Avaya Aura™ Call Center Feature Reference

Release 6.0

November 2010

Notice

While reasonable efforts have been made to ensure that the information in this document is complete and accurate at the time of printing, Avaya assumes no liability for any errors. Avaya reserves the right to make changes and corrections to the information in this document without the obligation to notify any person or organization of such changes.

Documentation disclaimer

Avaya shall not be responsible for any modifications, additions, or deletions to the original published version of this documentation unless such modifications, additions, or deletions were performed by Avaya. End User agree to indemnify and hold harmless Avaya, Avaya's agents, servants and employees against all claims, lawsuits, demands and judgments arising out of, or in connection with, subsequent modifications, additions or deletions to this documentation, to the extent made by End User.

Link disclaimer

Avaya is not responsible for the contents or reliability of any linked Web sites referenced within this site or documentation(s) provided by Avaya. Avaya is not responsible for the accuracy of any information, statement or content provided on these sites and does not necessarily endorse the products, services, or information described or offered within them. Avaya does not guarantee that these links will work all the time and has no control over the availability of the linked pages.

Warranty

Avaya provides a limited warranty on this product. Refer to your sales agreement to establish the terms of the limited warranty. In addition, Avaya's standard warranty language, as well as information regarding support for this product, while under warranty, is available to Avaya customers and other parties through the Avaya Support Web site: http://www.avaya.com/support. Please note that if you acquired the product from an authorized Avaya reseller outside of the United States and Canada, the warranty is provided to you by said Avaya reseller and not by Avaya.

Licenses

THE SOFTWARE LICENSE TERMS AVAILABLE ON THE AVAYA WEBSITE, HTTP://SUPPORT.AVAYA.COM/LICENSEINFO/ ARE APPLICABLE TO ANYONE WHO DOWNLOADS, USES AND/OR INSTALLS AVAYA SOFTWARE, PURCHASED FROM AVAYA INC., ANY AVAYA AFFILIATE, OR AN AUTHORIZED AVAYA RESELLER (AS APPLICABLE) UNDER A COMMERCIAL AGREEMENT WITH AVAYA OR AN AUTHORIZED AVAYA RESELLER. UNLESS OTHERWISE AGREED TO BY AVAYA IN WRITING, AVAYA DOES NOT EXTEND THIS LICENSE IF THE SOFTWARE WAS OBTAINED FROM ANYONE OTHER THAN AVAYA, AN AVAYA AFFILIATE OR AN AVAYA AUTHORIZED RESELLER, AND AVAYA RESERVES THE RIGHT TO TAKE LEGAL ACTION AGAINST YOU AND ANYONE ELSE USING OR SELLING THE SOFTWARE WITHOUT A LICENSE. BY INSTALLING, DOWNLOADING OR USING THE SOFTWARE, OR AUTHORIZING OTHERS TO DO SO, YOU, ON BEHALF OF YOURSELF AND THE ENTITY FOR WHOM YOU ARE INSTALLING, DOWNLOADING OR USING THE SOFTWARE (HEREINAFTER REFERRED TO INTERCHANGEABLY AS "YOU" AND "END USER"), AGREE TO THESE TERMS AND CONDITIONS AND CREATE A BINDING CONTRACT BETWEEN YOU AND AVAYA INC. OR THE APPLICABLE AVAYA AFFILIATE ("AVAYA").

Avaya grants End User a license within the scope of the license types described below. The applicable number of licenses and units of capacity for which the license is granted will be one (1), unless a different number of licenses or units of capacity is specified in the Documentation or other materials available to End User. "Designated Processor" means a single stand-alone computing device. "Server" means a Designated Processor that hosts a software application to be accessed by multiple users. "Software" means the computer programs in object code, originally licensed by Avaya and ultimately utilized by End User, whether as stand-alone products or pre-installed on Hardware. "Hardware" means the standard hardware originally sold by Avaya and ultimately utilized by End User.

Copyright

Except where expressly stated otherwise, no use should be made of materials on this site, the Documentation(s) and Product(s) provided by Avaya. All content on this site, the documentation(s) and the product(s) provided by Avaya including the selection, arrangement and design of the content is owned either by Avaya or its licensors and is protected by copyright and other intellectual property laws including the sui generis rights relating to the protection of databases. You may not modify, copy, reproduce, republish, upload, post, transmit or distribute in any way any content, in whole or in part, including any code and software. Unauthorized reproduction, transmission, dissemination, storage, and or use without the express written consent of Avaya can be a criminal, as well as a civil, offense under the applicable law.

Third-party components

Certain software programs or portions thereof included in the Product may contain software distributed under third party agreements ("Third Party Components"), which may contain terms that expand or limit rights to use certain portions of the Product ("Third Party Terms"). Information regarding distributed Linux OS source code (for those Products that have distributed the Linux OS source code), and identifying the copyright holders of the Third Party Components and the Third Party Terms that apply to them is available on the Avaya Support Web site: http://www.avaya.com/support/Copyright/.

Preventing toll fraud

"Toll fraud" is the unauthorized use of your telecommunications system by an unauthorized party (for example, a person who is not a corporate employee, agent, subcontractor, or is not working on your company's behalf). Be aware that there can be a risk of toll fraud associated with your system and that, if toll fraud occurs, it can result in substantial additional charges for your telecommunications services.

Avaya fraud intervention

If you suspect that you are being victimized by toll fraud and you need technical assistance or support, call Technical Service Center Toll Fraud Intervention Hotline at +1-800-643-2353 for the United States and Canada. For additional support telephone numbers, see the Avaya Support Web site: http://www.avaya.com/support/. Suspected security vulnerabilities with Avaya products should be reported to Avaya by sending mail to: securityalerts@avaya.com.

Trademarks

Avaya, the Avaya logo, Avaya one-X Portal, Communication Manager, Application Enablement Services, Modular Messaging, and Conferencing are either registered trademarks or trademarks of Avaya Inc. in the United States of America and/or other jurisdictions.

All non-Avaya trademarks are the property of their respective owners, and "Linux" is a registered trademark of Linus Torvalds.

Downloading documents

For the most current versions of documentation, see the Avaya Support Web site: http://www.avaya.com/support

Contact Avaya Support

Avaya provides a telephone number for you to use to report problems or to ask questions about your product. The support telephone number is 1-800-242-2121 in the United States. For additional support telephone numbers, see the Avaya Web site: http://www.avaya.com/support

Contents

Chapter 1: ACD Basics

Communication server features

What a communication server does

A communication server is an electronic device that processes incoming, outgoing, and internal calls and connects them to the proper destinations. The telephone company communication server in your local area is called a Central Office (CO). A communication server owned by a company or organization processes incoming, outgoing, and internal calls. Throughout this section, the term communication server is used to refer to a company or organization's communication server.

Trunks, trunk groups, and extensions

Incoming calls to a company first pass through the CO. The CO sends calls to the company communication server over trunks. Trunks are telephone lines that carry calls between two servers, between a CO and a server, or between a CO and a phone.

The CO receives dialed digits from the caller, processes the digits, and seizes a trunk that is assigned those digits. After the CO seizes a trunk, it sends a continuing transmission to the destination phone or communication server, and no other calls can be sent over that trunk until the current call disconnects.

Since a trunk can carry only one call at a time, trunk groups are usually created. A trunk group is a group of trunks that are assigned to the same digits. With a trunk group, the CO receives the digits of a dialed phone number and checks the trunk group assigned to that number to see if any of the trunks are available. The CO then seizes an available trunk. As many simultaneous calls can be made over a trunk group as there are trunks in that trunk group. A trunk group, therefore, can carry multiple calls for the same phone number. When a trunk group carries incoming calls (that is, calls made outside the company's communication server location) to the communication server, the communication server then connects the calls to their proper destinations within the company.

The communication servers previously listed, in addition to connecting incoming calls to the proper destinations, are also like private COs for company employees. Employee phones are connected to a communication server by telephone lines called extensions. Extensions are then assigned numbers, and these numbers become the employee phone numbers for internal (intra-company) calls.

Automatic-in processing

Automatic-in processing is one type of call processing. With automatic-in processing, the CO processes all of the digits of an incoming call. The CO then seizes a trunk from the trunk group, but since processing is complete, the call connects directly to a destination identified in the communication server software. That destination can be a phone, a queue (in which callers wait to be answered in the order in which their call was received), or special treatment like an announcement.

Communication server attendant

Incoming calls can also go to a communication server attendant. A communication server attendant is a person who manually routes calls to their proper destinations using an attendant console. Normally an attendant serves as an internal operator who transfers calls to the proper extensions. Often, a communication server will have more than one attendant, and all of the communication server’s attendants will answer calls directed to the attendant queue, which holds calls until an attendant is available. The attendant queue receives internal calls made from employee extensions, and also receives incoming calls through DID processing and automatic-in processing. Attendant call handling varies, depending on the company’s needs. However, if the attendant has an automatic-in number, it will normally be the number published in the phone book, and the DID number will most likely be used by off-site employees who know only the attendant’s extension number.

Centralized Attendant Service (CAS) is a communication server feature that enables attendants to be consolidated at one private-network location. The attended location is called the CAS main and each unattended location is called a CAS branch. At branch locations, calls requiring attendant services route by way of Release Link Trunks to the main location.

Direct inward dialing processing

With Direct Inward Dialing (DID) processing, incoming trunks do not connect the CO directly to an employee’s phone; instead, the incoming trunks are pooled by the communication server, and this pool of trunks is then shared by employee phones. Extension numbers may serve as the final digits of employee phone numbers for incoming calls. That is the CO may assign a 2- 3- or 4- digit prefix to a trunk group. Then, when a 7-digit employee phone number is dialed, the call is processed as follows:

1. The CO processes the prefix of the dialed number, and then seizes a trunk in the trunk group that is assigned that prefix.
2. The CO passes the remaining digits of the dialed number to the communication server.
3. The communication server recognizes the remaining digits as an employee extension number and sends the call to that extension.

DID processing example

As an example of DID processing, say that Employee A has the external phone number 538-1000 and the extension number 1000. Employee B has the phone number 538-9999 and the extension number 9999.

The steps in completing calls to Employees A and B might be as follows:

1. Employee A's client dials 538-1000.
2. The CO serving Employee A's company identifies the digits 538 (the common prefix for all phone numbers to that company) and seizes Trunk 1 in the trunk group assigned the digits 538.
3. The CO passes the digits 1000 to the communication server at Employee A's company.
4. The communication server identifies the digits 1000 as Employee A's extension number and sends the call to Employee A's extension.
5. Employee A's phone rings and Employee A answers.
6. Meanwhile, Employee B's client dials 538-9999.
7. The CO identifies the digits 538 and seizes Trunk 2 in the trunk group assigned the digits 538.
8. The CO passes the digits 9999 to the communication server.
9. The communication server identifies the digits 9999 as Employee B's extension number and sends the call to Employee B's extension.
10. Employee B's phone rings and Employee B answers.

While Employees A and B continue to talk, Trunks 1 and 2 in the 538 trunk group will not accept any more calls, so another call beginning with the digits 538 will seize yet another trunk in the trunk group.

What the ACD does

About ACD

Automatic Call Distribution (ACD) is a communication server software feature that processes high-volume incoming, outgoing, and internal calls and distributes them to groups of extensions called hunt groups or splits. The communication server also sends information about the operation of the ACD to the CMS, which stores and formats the data and produces real-time and historical reports on ACD activity.

ACD is used by a call center to route incoming calls to specifically assigned splits/skills and agents. ACD allows a system administrator to create an efficient call management environment. This administrator can add or remove splits/skills from the system, add or remove announcements, add or remove agents, add trunk groups and route calls to the appropriate splits/skills. The administrator can also specify ACD measurement criteria and use an optional CMS package to provide reports on ACD efficiency.

Things to know before you start

A voice response port or a person who answers ACD calls is called an agent. Companies that operate high-volume call-answering centers, for example, a catalogue sales center, a reservations center, or a customer service center, use the ACD feature to process incoming calls and distribute them to agents. In addition to agents, each ACD split can be assigned a split supervisor. The split supervisor uses various communication server and CMS features to monitor split and agent performance and to provide assistance if necessary. Maintaining trunks from the CO to the communication server and hiring agents to answer calls costs money. However, if customers who call to purchase goods or services have difficulty reaching an agent and, therefore, stop trying to get through, the call center loses revenue. Call center management needs, therefore, to determine how many trunks and agents are necessary to minimize costs and maximize the ability of customers to purchase goods or services. Management can then set up and maintain the ACD accordingly.

Automatic-in processing of ACD calls

Through communication server administration, each automatic-in trunk group is assigned to an ACD split. All calls that come in on an automatic-in trunk group are directed to the assigned split. Then the ACD software distributes the calls to the agent extensions assigned to the split according to the assigned call distribution method (described later).

DID processing of ACD calls

The communication server enables you to dial directly to various extensions such as a VDN, a hunt group, an agent, or a login ID. Each extension can be assigned to a split as a DID extension.

For DID processing, trunk groups are not assigned to the split. The creation of associated extensions is sufficient to send calls arriving over DID trunk groups to the appropriate split. Each split can receive incoming calls through DID processing, automatic-in processing, or both. Automatic-in trunk groups carry calls only to the split, whereas DID trunk groups carry calls to any extension identified in the communication server software, not just a split.

Split queues

Split queue call processing

A split queue is a holding area for calls waiting to be answered, usually in the order in which they were received. When a call is put into queue, the caller may hear one or more delay announcements, music, and/or silence, depending on the treatment assigned for the split. (Treatment of calls in queue is assigned through communication server administration.)

Things to know about split queues

Calls enter the queue at the bottom and move toward the top or head of the queue. After a call reaches the head of the queue, it connects to the next available agent.

For communication servers with the Call Vectoring feature, all call treatment including routing, queuing, announcements, and music is specified by call vectors. When a call arrives at a split, the ACD software checks to see if an agent is available to handle the call. If an agent is not available, or busy, the call enters the split’s queue.

Calls queue only if no agents are available, a queue is assigned to the split, and the queue is not full. If the queue is full, the caller hears a busy tone or the call goes to coverage. If the split is vector controlled, then this step will fail. Furthermore, if no agents are logged into the split or if all agents are in AUX work mode (described later), calls do not queue.

Priority and normal split queues

Each split can have two queues: a normal queue and a priority queue. A split always has a normal queue and can also be assigned a priority queue. The ACD distributes all calls in the priority queue before it distributes any calls in the normal queue. Therefore, the priority queue, if one exists, must be empty before the ACD distributes calls in the normal queue.

Priority queuing may be assigned in the Class of Restriction (COR) associated with the split extension number. A split may also be assigned Priority Queuing on Intraflow, which means that calls to that split, if rerouted to another local split, will enter the destination split's priority queue.

Dynamic queue slot allocation

Beginning with Communication Manager 2.1, Communication Manager dynamically allocates queue slots to hunt groups or skills. You no longer have to estimate and administer queue slots. The system dynamically allocates hunt group or skill queue slots on an as needed basis. When a queue slot resource is needed, it is extracted from a common pool. When the call is removed from queue, the queue slot resource is relinquished and returned to the common pool. There are enough queue slots to allow all possible calls to queue.

Dynamic queue slot allocation has the following advantages:

- Reduced administration
- Expanded capacities, such as increased skill availability in your call center
- Elimination of lost or blocked calls when all queue slots are full

Note:
You can limit the actual number of calls that can be queued for a specific hunt group by using the *calls-queued* conditional in the **`check split/skill if calls-queued`** or **`goto step/vector if calls-queued in split/skill`** vector commands.

Use the**Queue Limit** field to specify the maximum number of calls that can be queued to the hunt group. For more information about this field, see Hunt Group screen field descriptions in *Avaya Aura™ Call Center Feature Reference*.

Announcements for calls in a split queue

When a call enters a split queue, the caller hears ringing until the call is connected to an agent or an announcement. Depending on the treatment assigned to a split, the caller may hear one or two announcements, music, or silence. An announcement is a recorded message that provides information such as the destination the call has reached or a company's business hours, or it tries to persuade the caller to stay on the line.

Things to know about split queue call announcements

Announcements and delay time are assigned to splits through communication server administration. Delay time is the amount of time a call will wait in queue before receiving an announcement. If a call connects to an agent before the delay time expires, the caller does not hear the announcement. If a call connects to an agent while an announcement is playing, the announcement stops. After the first announcement plays, the caller hears music or silence until the second announcement plays or the call connects to an agent. The type of caller feedback (music or silence) is also assigned to a split through communication server administration.

For communication servers with the Call Vectoring feature, announcement capabilities are more flexible than those described in this section. See Call Vectoring.

Split queues: Related features

The communication server supports both internal and external announcement devices.

The announcement delay time can be from 0 to 99 seconds. A 0-second delay time causes a forced announcement, which means callers always hear the entire first announcement, whether an agent is available or not. A second announcement can be administered to recur each time the announcement delay time expires.

Rules for announcements

The announcement is played from beginning to end unless an agent becomes available. In such a case, the announcement is interrupted and (if manual answering operation is assigned to the agent, or if calls are delivered to the agent on a manual answering basis) ringback is provided. If the call is queued, the call remains as such while the announcement is played. Any feedback that is provided before an announcement (for example, a wait with music or ringback) continues until the announcement is played.

Without vectoring - If an announcement queue is full, the system continues to try to connect the call to the proper announcement until the call connects to an agent, connects to an announcement, or enters the announcement queue. The following rules apply to announcements without vectoring implemented:

- Calls directly entering a split queue always receive a forced first announcement if assigned. The caller also hears first and second delay announcements if administered and delay intervals are met.
- Calls that reach a split by way of Call Coverage from another split (Intraflow) or a station do not receive a forced or delay first announcement at the destination split. The caller hears a second delay announcement if administered and the delay interval is met.
- Calls that reach a split by way of Call Forwarding from another split (Interflow) or station do receive delay first and second announcements if administered and the delay intervals are met.

With vectoring - If the announcement's queue is full, the call retries the announcement step for an indefinite period before any new vector steps are processed. If an **`announcement`** command follows a failed **`adjunct routing`** command, the announcement is interrupted. If the **`adjunct routing`** command succeeds (that is, the communication server receives a destination from the ASAI adjunct), the announcement terminates immediately. The **`announcement`** command step is skipped, and vector processing continues at the next vector step, whenever any of the following conditions exist:

- Requested announcement is busied out, not available, or not administered.
- Integrated board is not installed.
- External aux trunk or analog equipment is not attached.

Announcement queuing

External and internal announcement units are available. The number of calls that can be queued to an announcement depends on the size of the communication server you have. The capacity tables in the System Description have details for each communication server model. Queuing for internal announcements is quite different. Internal announcements are delivered by a multi-port/channel announcement board, and a call receives an announcement only when it connects to one of the announcement ports or channels. Therefore, all calls wait in a single queue to access a channel on the announcement board regardless of the split announcement they are waiting to receive. The same announcement can be delivered over multiple channels. Announcements are delivered on demand, so a call that connects to a channel receives an announcement immediately and does not have to wait for the announcement to finish and start again.

Answer supervision and abandoned calls

Answer supervision is a signal sent by the communication server to the serving Central Office (CO). This signal tells the CO that an incoming call is answered and that the CO should begin tracking toll charges for the call (if they apply). Answer supervision is sent immediately before a call connects to an agent's telephone, to music, or to an announcement.

Abandoned calls

An abandoned call is a call that reaches a call center, but does not connect to an agent because the caller hangs up. A call can abandon while in queue or while ringing at an agent position. Abandoned calls represent lost sales or lost good will. Adequate split staffing and effective use of announcements can reduce the number of abandoned calls. Splits should be staffed so that calls do not have to wait in queue for an unreasonable amount of time, and announcements can be used to persuade the caller to wait until someone answers the call.

Abandoned call search

If answer supervision is sent before a caller abandons, ghost calls can occur. A ghost call is a call that is sent to an agent after the caller hangs up. Ghost calls occur because, after a caller hangs up, some COs wait 2 to 25 seconds before sending a disconnect signal to the communication server. Ghost calls are a problem because they waste agents' time, and they can delay or prevent other calls from connecting to an agent. To minimize this problem, Abandoned Call Search can be assigned to specific trunk groups for the communication server.

With Abandoned Call Search, the communication server checks the incoming trunk before delivering an ACD call to an agent. If the trunk is on-hook at the CO (the call has been abandoned), the communication server releases the trunk and does not deliver the call. If a call is still in progress on the trunk, the communication server delivers the call to an agent.

Intraflow and interflow

Intraflow and interflow allows you to redirect ACD calls to another split or other local or remote destinations. Redirecting calls to a local destination is called intraflow. Redirecting calls to a destination outside the communication server is called interflow.

Things to know about intraflow and interflow

Intraflow and interflow are set up differently on the Generic 3 and newer communication servers. If Call Vectoring is active on the communication server, redirection of calls differs significantly from the following intraflow/interflow descriptions.

Description of intraflow and interflow

As many as three intraflow destinations OR one interflow destination can be established for a split through communication server administration. Intraflow uses the Call Coverage feature to redirect ACD calls to a coverage path that contains one, two, or three of the following internal destinations:

- An extension
- An ACD split (including messaging system and message center splits) or Hunt Group. The term “Hunt Group” refers to groups of extensions that receive distributed calls. The term “split” refers to a hunt group that is measured by CMS.
- An attendant group
- An announcement followed by a forced disconnect

Call Forwarding and ACD splits can be set up to intraflow calls unconditionally.

Interflow destinations are the same as those listed above for intraflow (plus the CAS attendant), except interflow sends calls to destinations outside the communication server.

Setting up splits

If a split is assigned more than one intraflow destination, the communication server tries each destination in the order in which it was assigned. If no destination can accept the call, the communication server leaves the call in the original split’s queue. If an interflow destination is specified and activated, the communication server tries only that destination. If the interflow destination cannot accept the call, the caller hears a busy signal. ACD splits can be set up to intraflow calls unconditionally. Unconditional intraflow redirects all calls to the specified destination. Unconditional intraflow is normally used to redirect calls when a split is not staffed.

Splits can also be set up to intraflow calls when one or all of the following criteria are met:

- Don’t Answer - Calls redirect if not answered within the assigned Don’t Answer Interval (1 to 99 ringing cycles).
- No Agents Staffed or All Agents in AUX Mode - Call redirect if there are no agents staffed or if all agents are in the AUX work mode.

Queue status assignment

If an intraflow destination has a queue, that queue may be assigned an inflow threshold. The inflow threshold, which is established through communication server administration, is the length of time the oldest call in queue has waited. Once the inflow threshold is reached, that queue does not accept intraflowed calls and the communication server tries the next administered destination.

Through communication server administration, a split can be assigned Priority Queuing on Intraflow, which allows intraflowed calls to enter the priority queue at the destination split.

Types of calls for a split

The following types of intraflow/interflow can be used for a split:

- Don’t Answer Time Interval intraflow (using the Call Coverage feature)
- Busy intraflow (using the Call Coverage feature)
- Unconditional intraflow (using the Call Forwarding-All feature).

When calls are intraflowed using the Call Coverage feature, CMS only reports inflowed and outflowed calls if the call queues to the original split. For example, a call that covers using the busy criterion will not be recorded as in/outflowed since it could not queue to the original split. Calls that queue before covering using the Don’t Answer criteria are recorded as in/ outflowed calls.

Intraflow/interflow setup

A split can have either intraflow or interflow active, but not both. However, both conditional (Call Coverage) and unconditional (Call Forwarding) intraflow can be active for a split at the same time. In this case, unconditional intraflow is first invoked for the split’s incoming calls. Then, after the communication server forwards a call to the unconditional destination, the communication server uses the conditional intraflow criteria to determine whether to redirect the call to the next destination. Thus, when unconditional and conditional intraflow are used together, the conditional intraflow criteria are applied to the forwarded-to destination, not to the original split.

This combination of unconditional and conditional intraflow allows Dialed Number Identification Service (DNIS) numbers to appear on agent display telephones. In this case, the DNIS number is actually a dummy split extension (that is, the split extension has no assigned agent extensions). The intraflow destinations are the real splits (with staffed agents). With such a configuration, CMS will count incoming calls for the DNIS number (that is redirected using unconditional intraflow to real splits) as outflows. CMS will also count the calls to the destination splits as ACD calls and inflowed calls. And regardless of the split where calls actually connect to agents, the agents will see the DNIS (dummy split) number on their display telephone.

The intraflow criteria and destinations are assigned through communication server administration. Console permissions and the Call Forwarding dial access code are also assigned through communication server administration. Unconditional intraflow or interflow can be activated by entering the Call Forwarding dial access code from a station with console permission, the split's extension, and the interflow or intraflow destination number.

The split supervisor cannot establish conditional intraflow from a telephone. Furthermore, CMS cannot be used to set up or activate intraflow/interflow.

Night Service

Night Service redirects all calls to one of the following internal destinations:

- An ACD split
- An extension
- An attendant group
- An announcement with forced disconnect.

Night service is available for a hunt group, a trunk group, or a system.

Hunt Group Night Service

Hunt Group Night Service redirects all calls arriving at a split to an internal destination. The Night Service destination for the split and the telephone button used to activate the feature are assigned through communication server administration.

Trunk Group Night Service

Trunk Group Night Service redirects all calls arriving over a split's assigned trunk groups to an internal destination. The Night Service destination for the trunk group and the telephone button used to activate the feature are assigned through communication server administration.

Trunk Group Night Service by itself does not guarantee that all calls to a split will be redirected. Calls from local extensions and DID calls will still connect to the split.

Trunk Group Night Service and Hunt Group Night Service can both be active at the same time. If the Trunk Group Night Service is active, its destination will be used for calls that come in over the trunk group even if they go to a split that has a Hunt Group Night Service destination assigned.

System Night Service

System Night Service redirects all calls arriving over all trunk groups to the Night Service destination. System Night Service overrides any Hunt Group Night Service set up for an individual split. If Trunk Group Night Service is active for a particular trunk group, System Night Service does not affect that trunk group. When any type of Night Service becomes effective, calls already in a split's queue are not redirected. To avoid dissatisfied callers, agents should continue to staff the split until the queue is empty.

Distributing and handling calls

About call distribution

ACD calls are delivered to agents according to the type of call distribution (also known as hunting) that is assigned to the split or skill. When the ACD and Queue field on the Hunt Group screen are set to y, queues for agents and queues for calls are established in the communication server. These queues are used to handle incoming calls based on the type of call distribution that is implemented. The types of call distribution available for use depend on whether or not EAS is used by the call center.

Call distribution methods without EAS

Without EAS, the following call distribution methods are available:

- Direct Department Calling
- Uniform Call Distribution-Most Idle Agent (UCD-MIA)

Note:

The following descriptions of ACD call distribution assume that the Multiple Call Handling (MCH) feature is not assigned. Agent availability is different for splits assigned the MCH feature.

Direct

ACD software searches for an available agent in the order that extensions were assigned to the split (through communication server administration), starting with the first extension assigned to the split. This type of call distribution is most useful when management wants the most effective or most experienced agents to handle more calls. Agents are rank-ordered from most to least effective and then are assigned to the split in that order. Direct call distribution is called Direct Department Calling (DDC).

If you administer a split for DDC, an incoming call is routed to the first available agent extension in the administered sequence. If the agent is not available, the call routes to the next available agent, and so on. Incoming calls are always routed to the first agent in the sequence, so calls are not evenly distributed among agents.

UCD-MIA for call distribution without EAS

When the UCD-MIA call distribution method is used, the communication server searches for the agent extension that has been idle (waiting) the longest and delivers the call to that extension if the agent is available to handle an ACD call. This type of call distribution ensures a high degree of equity in agent workloads even when call-handling times vary.

The ACD software determines which agent extension has been idle the longest by maintaining an ordered list (queue) of agents who are eligible to receive the next ACD call. Eligible agents enter the queue at the bottom and move toward the top of the queue. The agent who has been in queue the longest receives the next ACD call unless the agent is not available at the time the call is to be distributed. If the agent at the top of the queue is not available, the ACD software checks the availability of the next agent in queue until an available agent is found.

When an agent completes an ACD call, the agent is added to the bottom of the eligible-agent queue for the split or skill associated with the call. The MIA across splits/skills options is used to put an agent at the bottom of all split or skill queues that the agent is logged in to when the agent completes any ACD call. Agents move toward the top of the eligible-agent queue as long as they remain staffed and available or on AUXIN or AUXOUT extension calls from the available state, or on an ACD call for another split (unless the MIA across splits/skills option is turned on). Agents in After Call Work (ACW) mode are in eligible agent queues on Generic 3 communication servers. You can choose whether these agents are or are not in the eligible-agent queues for the communication server.

An agent is marked as unavailable to take an ACD call if the agent is:

- In ACW
- On an AUXIN or AUXOUT extension call from the available state
- On an ACD call for another split or skill

The agent remains in queue moving toward the top of the queue. Agents in multiple splits enter multiple eligible-agent queues. The agents' progress in each queue is independent of any activity in other queues. Agents in the AUX state are not in the eligible-agent queue.

You can set the communication server to maintain a separate queue for available agents in each split or skill, or you can create one combined queue for agents in all splits/skills. If the **MIA Across Splits/Skills?** field on the Feature-Related System Parameters screen is set to `n`, the communication server maintains available agent queues for each split or skill. When agents answer a call, they are only removed from the available agent queue for the split or skill at which that call arrived. If the field is set to `y`, then the agent is removed from all split or skill queues that the agent is logged in to whenever they answer a call for any of their assigned splits/skills.

The agent is returned to the agent queues, based on how you administer the following:

- If forced Multiple Call Handling applies, the agent is placed in the queue when the call stops alerting.
- If the **ACW Agents Considered Idle?** on the Feature-Related System Parameters screen is set to`y`, the agent is queued when the call completes.
- If **ACW Agents Considered Idle?** is `n`, the agent is queued when ACW completes.

Note:
If you are using an Expert Agent Distribution method (EAD-MIA or EAD-LOA), then the agent is put back in queue(s) after completing an ACD call based on skill level. If you are not using an EAD call distribution method, then the agent is put at the bottom of the queue(s) after completing an ACD call.

Call distribution methods with EAS

With EAS, the following call distribution methods are available:

- Uniform Call Distribution-Most Idle Agent (UCD-MIA)
- Expert Agent Distribution-Most Idle Agent (EAD-MIA)
- Uniform Call Distribution-Least Occupied Agent (UCD-LOA)
- Expert Agent Distribution-Least Occupied Agent (EAD-LOA)

The following table summarizes the different call distribution methods, which are further defined in the following sections.

Agents available, call arrives, and agent selection method is:	THEN the communication server selects:
EAD-MIA	the highest skill level, most idle agent.
UCD-MIA	the most idle agent, without regard to skill level.
EAD-LOA	the highest skill level agent with the lowest occupancy.
UCD-LOA	the least occupied agent, without regard to skill level.

UCD-MIA for call distribution with EAS

UCD-MIA works the same in the EAS environment as it does without EAS, except that the communication server searches for the most idle agent with the required skill.

UCD-MIA does not select an agent based on skill level. Therefore, if an agent is the most idle agent with the required skill, even if the skill is assigned a secondary skill level for that agent, the call is delivered to that agent.

EAD-MIA

The EAD-MIA call distribution method selects the most idle agent with the required skill to handle the call and the highest skill level.

This method of call distribution adds a layer of processing on top of the Most Idle Agent distribution call processing. EAD-MIA sorts the agents in the eligible-agent queue into multiple queues based on skill level. Agents with the skill assigned at higher-priority levels appear in the eligible-agent queue ahead of agents with the skill assigned at lower-priority levels. The call is delivered to the most idle, most expert agent available.

When you are using EAS Preference Handling Distribution (EAS-PHD), the agent can enter the MIA queue at one of 16 levels. The lower the level, the higher the level of expertise; so an agent with skill level 1 is the most qualified to answer a call to that skill. Without EAS-PHD, agents enter the MIA queue as either level 1 or level 2 agents. When agents with a lower skill level become idle, they enter the MIA queue in front of agents with a higher skill level. See Expert Agent Selection for more information about EAS Call Distribution.

UCD-LOA

When the UCD-LOA call distribution method is in use, the communication server delivers the call to the least occupied agent, without regard to skill level.

The least occupied agent is the agent who has spent the lowest percentage of their time on ACD calls since logging in. The agent’s place in the queue of available agents is determined by this percentage. The agent occupancy (the percentage of time on calls) is always calculated separately for each skill an agent is logged into, so there is an available agent queue for each skill.

EAD-LOA

When the EAD-LOA call distribution method is in use, the communication server delivers the call to the least occupied agent with the highest skill level.

The agent occupancy is calculated as described in the UCD-LOA section.

How agents handle calls

An agent can receive split calls and, in most cases, personal calls that are not related to a split. Calls distributed to an agent's telephone by the ACD feature on the communication server are considered ACD calls. Calls dialed directly to an individual agent using the agent's extension number (such as internal calls and DID extension calls) are called extension-in (EXT-IN) calls. Outgoing calls the agent makes are called extension-out (EXT-OUT) calls. EXT-IN and EXT-OUT calls are considered non-ACD calls.

Things to know about handling calls

The capability of a telephone to receive EXT-IN calls or to make EXT-OUT calls can be restricted through communication server administration. The following descriptions of agent call handling assume that the Multiple Call Handling (MCH) feature is not assigned. Agent availability and call handling are different for splits assigned the MCH feature.

ACD calls are distributed only to available agent extensions. To be considered available, an agent must first staff an agent extension and then select a call-answering mode (automatic in or manual in).

Staffing agent extensions without EAS

To staff an agent extension on the communication server without the EAS feature, an agent must dial a login access code or press the **LOGIN** button on the agent's telephone. The agent must then dial a split number and a login ID. The login ID length, the login dial access code, and, if desired, the **LOGIN** button are assigned through communication server administration. The split number may also be assigned to the **LOGIN** button or to another telephone button.

Staffing multiple splits

An agent can log in from any extension assigned to a split. An agent can log into as many as four splits. To the communication server and CMS, each login counts toward the maximum number of agent members that can be measured. That is, if four agents are each logged into three splits, the agent member count is 12.

Agent login

Agent login lets ACD (and CMS) know an extension is active and logged into the system (AUX work mode). Pressing the login button and then following the appropriate system login procedure makes the extension staffed in AUXWORK. This procedure varies with the type system you have.

Agent logout

Agent logout lets ACD (and CMS) know an extension is no longer active.

Agent request for supervisor assistance

When supervisor assistance is needed, an agent can press the ASSIST button or dial the ASSIST feature access code to bring the designated person on line. Pressing ASSIST automatically places the current call on hold.

Agent states and call-answering modes

Agent State is the current status of an agent. Work modes are the work function(s) the agent is performing at a given time.

When the agent is engaged in an ACD call, the agent is in the ACD agent state.

After staffing an extension, the agent is in the auxiliary work (aux-work) mode, which is considered non-ACD work.

Auto-in versus manual-in

In aux-work mode, the agent is not yet available to receive ACD calls. To become available for ACD calls, the agent must press the manual-in or auto-in button to select a call-answering mode.

Work mode	Description
Auto-in	Like the manual-in button, the auto-in button tells the ACD that the agent is available for an ACD call. However, when the call ends, the agent is immediately available for another ACD call according to the established call distribution method. The agent does not have to press any buttons to receive another ACD call. This type of call answering increases the number of calls that agents can answer in a given period and is most effective if agents have little or no call-related work to do after finishing each ACD call. The communication server has a timed ACW feature for auto-in operation. This option automatically puts the agent into ACW for a preset length of time at the end of an auto-in call. When the time is up, the

Work mode	Description
	agent automatically becomes available to take an ACD call. manual-in and auto-in dial access codes and telephone buttons are assigned through communication server administration.
MANUAL-IN	The MANUAL-IN button tells the ACD that the agent extension is available for an ACD call. The ACD then distributes a call to the agent according to the established call distribution method. When the call ends, the agent automatically enters the ACW state. While in ACW, the agent is not available to receive ACD calls. When ACW ends, the agent presses MANUAL-IN to receive another ACD call. The manual-in mode is most effective if an agent must perform call-related tasks after finishing each ACD call. MANUAL-IN dial access codes and telephone buttons are assigned through communication server administration.

Ringing versus zip tone for incoming calls

When a call arrives at a telephone, the agent may hear ringing or zip tone (beeping), depending on how the telephone is administered. Ringing is recommended when an agent answers calls using the handset. When a call connects to the agent telephone, the telephone rings, and the agent picks up the handset to answer the call.

Zip tone is recommended when the agent uses a headset to answer calls. Zip tone can also be used with a handset, but the agent must hold the handset and listen for the zip tone. When a call connects to an agent telephone, the agent hears one burst of zip tone for calls dialed directly to the split (or agent extension on the communication server) and, without pushing any buttons, the agent greets the caller.

Ringing or zip tone is established on a per-telephone basis through communication server administration.

Ringing is also called manual answer and zip tone is also called automatic answer.

Auxiliary Work (AUXWORK) and ACW

To temporarily stop ACD calls from arriving at an agent's telephone, an agent can press the Auxiliary Work (AUXWORK) or After Call Work (ACW) button.

Mode	Description
AUXWORK	The agent is involved in non-ACD work, is on break, in a meeting or at lunch. CMS recognizes the extension as staffed but does not want ACD to route calls there for an extended time. AUX-IN implies that the extension received an extension-in call while in AUX. AUX-OUT implies that the agent placed an outgoing call while in AUX.

Mode	Description
	The **AUXWORK** button temporarily stops ACD calls from arriving at the agent's telephone. The agent normally presses this button before doing non-ACD-related work such as taking a break or doing personal business. Instead of unstaffing the extension or logging off, an agent can press this button, which places the agent in the auxiliary-work state. To receive ACD calls again, the agent presses the manual-in or **auto-in** button. The **AUXWORK** button (or the dial access code, if no button is available) is assigned through communication server administration. If an agent is normally logged into more than one split, an **AUXWORK** button for each split may be assigned. Then, when the agent presses the **AUXWORK** button for a particular split, the agent will not receive calls from that split. However, the agent will still be available for calls from the other splits the agent is logged into. Also, if an agent is logged into more than one split or skill and receives an ACD call for one split or skill, the agent is unavailable for calls for other splits/skills. When the service level threshold for an interruptible hunt group (skill) is exceeded, agents with that interruptible skill who are in AUX with an interruptible reason code are notified that they are needed. The notification consists of a display message ("You are needed"), flashing **auto-in** and/or **manual-in** buttons and an audible tone. Agents who move to an interruptible Aux mode after the threshold is exceeded are also notified. The duration of notification to "Auto-In-Interrupt" agents is administrable using the Interruptible Aux Notification Timer (sec) field on page 13 of feature-related system-parameters form. For more information on Interruptible Aux feature, see Interruptible Aux work.
ACW	The agent is engaged in work associated with a call, but not on a call. ACW-IN implies that the station received a call while in ACW. ACW-OUT implies that the agent made an outgoing call while in ACW. The ACW button temporarily stops ACD calls from arriving at the agent's telephone. An agent who is in auto-in mode presses this button during a call so that when the call is finished, the agent will not receive another ACD call and can, instead, do ACD call-related work such as filling out a form, completing data entry, or making an outgoing call. The lamp indicator next to the **ACW** button lights when the agent is in ACW. When in the manual-in mode, an agent automatically enters ACW when the call ends. However, if the agent needs to get out of auto-in mode or the auxiliary work state to do additional call-related work, the agent can press the **ACW** button (or dial the appropriate access code). An agent can press the **MANUAL-IN** button (or dial the appropriate access code) while on an ACD call to automatically enter ACW when the call ends. If an agent is logged into more than one split, pressing the **ACW** button makes the agent unavailable for calls in all splits. CMS considers the agent to be in the OTHER state for all splits other than the split in which the agent is currently in ACW.

The following table lists additional agent states/work modes that may display.

Agent state/ work mode	Description
UNSTAF	Unstaffed (Agent State). The agent is not logged in and being tracked by CMS.

Agent state/ work mode	Description
DACD	The agent is on a direct agent ACD call.
DACW	The agent is in the after call work state for a direct agent ACD call.
OTHER	The agent is doing other work. If an agent is working in three splits/skills and receives a call from one, the ACD puts the agent in OTHER for the other two.
UNKNOWN	CMS does not recognize the current state. Unknown remains until the condition is cleared, and/or the agent completes the current ACD call and any current ACW, or a current agent state message is sent to CMS from the communication server.
RING	The time a call rings at an agent's telephone after leaving the queue and before the agent answers.

Trunk states

Trunk State indicates the current status of a specific trunk, or the ability to change that state. Trunk states are described in the following table.

Trunk State	Description
Idle	The trunk is waiting for a call.
Seized	The trunk is seized by an incoming or outgoing call.
Queued	An ACD caller has the trunk and is waiting for the agent to answer.
Conn	The agent and caller are connected in an ACD call.
Abandoned	The queued caller has just abandoned the call.
Fwrd	A queued call has been intraflowed outside the ACD or has been interflowed to another PBX/communication server.
Mbusy	Maintenance Busy, or out of service for maintenance purposes.
Hold	The agent has put the call on hold.

Other telephone buttons

This section describes other buttons that can be assigned to an agent’s telephone.

Button	Description
CALL APPEARANCE	These buttons are used to place (originate) and answer calls. Two status lamps (red and green) are next to each call appearance button. The red lamp lights when an agent presses an appearance button to make or answer a call. The green lamp flashes to indicate an incoming call. Except with Multiple Call Handling, incoming ACD calls always arrive at the first call appearance. However, telephones may be assigned more appearances to provide additional call-handling capabilities. For example, an agent can use a second call appearance to transfer or place calls since the line will be free of ACD calls. On a two-appearance telephone, the second appearance can only be used to originate calls.
ADD SKILL	For communication servers with EAS, logged-in agents or telephone users with console permissions can press this button to add a skill. This is an Abbreviated Dialing (AD) button programmed with the add skill Feature Access Code (FAC).
ALERT CHANGE	The lamp associated with this telephone button flashes when another user changes an agent's assigned skills or moves an agent from his or her current split to a different split. The lamp does not flash when an agent changes his or her own skills from the telephone.
ASSIST	Press this button to request help from the split supervisor. The ASSIST button automatically dials the split supervisor's extension and connects the agent to the supervisor. Pressing the ASSIST button automatically puts the current call on hold.
AUDIO TROUBLE	Agents press this button to report a call with poor transmission quality to CMS. The message the communication server sends CMS includes the agent's extension, the trunk being used, and the time of day the trouble occurred. This information is reported in CMS exception reports and is useful for trouble-shooting trunk and extension problems. For more information, see Avaya CMS Administration. Stroke count button 0 is used for reporting audio difficulty.
CONFERENCE	Press this button to add another person to a two-person call. An agent with a multi-appearance telephone can add up to four additional people to a 2-person call. For single-appearance telephones, only one person can be added. Single appearance telephones do not have a CONFERENCE button. Agents must use the RECALL button to conference a call. If an agent adds another agent into a conference call, the resulting conference is not considered an ACD call for the added agent. The ACD considers the added agent to be on an extension-in call.
CALL WORK CODE	Agents press this button and enter up to 16 digits to record the occurrence of a customer-defined event. Call Work Codes are stored on CMS, not on the communication server.
CALLER-INFO	With the Call Prompting feature, agents press this button to display the digits collected by the last `collect digits`vector command.
EMERGENCY	Press this button to report a malicious call to the controller. The controller can then trace the call.

Button	Description
HOLD	Press this button to put a call on hold. The ACD will not send any more calls to an agent who has a call on hold. For communication server with Multiple Call Handling, an agent can put an ACD or non-ACD call on hold and receive an ACD call by pressing the auto-in or manual-in button. With Multiple Call Handling, multiple ACD calls can be delivered automatically to an agent in auto-in or Manual-In work mode, provided that an unrestricted line appearance is available on the telephone. Single appearance telephones do not have a HOLD button. Agents must use the RECALL button or the telephone's switchhook to put a call on hold. A single appearance telephone cannot be used to handle multiple ACD calls.
LOGIN	Press this button to staff the extension and start CMS collection of agent data. This is an Abbreviated Dialing (AD) button programmed with the login Feature Access Code (FAC).
LOGOUT	Press this button to unstaff the extension and end CMS collection of agent data. This is an Abbreviated Dialing (AD) button programmed with the logout Feature Access Code (FAC).
RECALL/ Flash switchhook	To put calls on hold, transfer calls, and create conference calls, agents using single-appearance telephones press the RECALL button or flash the switchhook if the phone is not equipped with a RECALL button.
RELEASE	Press this button to disconnect a call. Do not use the DROP button.
REMOVE SKILL	With EAS, logged-in agents or telephone users with console permissions can press this button to remove a skill. This is an Abbreviated Dialing (AD) button programmed with the remove skill Feature Access Code (FAC).
STROKE COUNT	As many as nine STROKE COUNT buttons can be assigned. Agents press these buttons to record call events of interest. CMS records and reports stroke-count information. Stroke count button 0 is reserved for audio difficulty.
TRANSFER	Agents normally press the TRANSFER button to transfer calls to other agents or the split supervisor. This button is only available on multi-appearance telephones. Single-appearance telephone users must use the telephone's switchhook. Agents can also use the TRANSFER button to transfer calls to external destinations. External transfer must be assigned to a telephone as a feature over and above the normal transfer feature. If an agent transfers a call to another agent, the call is not considered an ACD call for the agent receiving the call unless the transferring agent dialed a split extension, VDN, or agent login ID, an EAS capability known as Direct Agent Calling (DAC). The ACD considers the agent receiving the transfer to be on an extension-in call. For the agent transferring a call, the call is counted as an EXT-OUT call.
VUSTATS	Agents with display telephones press this button to display agent, split or skill, VDN, or trunk group data similar to that reported by CMS.

Queue status lamps

The lamps associated with the queue status buttons provide the following information.

Lamp	Description
NQC	The lamp associated with the Number of Queued Calls (NQC) button tells the agent that calls are in queue and when the number of calls in queue has met or exceeded the assigned queue threshold for the split. If no calls are in the split's queue, the status lamp associated with the button is dark. When one or more calls are in queue, the lamp lights steadily. When the number of calls in queue reaches the assigned queue threshold, the lamp flashes on and off.
OQT	The lamp associated with the Oldest Queued Time (OQT) button tells the agent that calls are in queue and when the oldest call in queue has been waiting longer than the assigned wait time threshold (0 to 999 seconds) for the split. If no calls are in the split's queue, the status lamp is dark. When calls are in queue, the lamp lights steadily. When the assigned wait time threshold has been met or exceeded by the oldest call in queue, the lamp flashes on and off. A flashing queue status lamp tells agents they need to handle calls more quickly. The thresholds that cause the lamps to flash and the telephone buttons are assigned through communication server administration.
Auxiliary queue status lamps	An auxiliary queue status lamp indicates that either the NQC threshold or the OQT threshold has been reached. The lamp lights when the assigned threshold is met or exceeded. Unlike the lamps on a telephone, the auxiliary queue status lamp does not indicate when calls queue to the split.

Display buttons

The following telephone buttons control the information that appears on the display.

Button	Description
NORMAL	Press this button to display information about the active call appearance. Press this button to display incoming call information (either an extension-in call or an intraflowed/interflowed call) for a different call appearance.telephone

Split supervisor telephone buttons

A split supervisor is normally assigned to each split. The capabilities that allow monitoring of agent performance, adding and removing agents, and performing other split-related activities must be assigned with separate communication server administration procedures.

Telephone button definitions

The following telephone buttons are available only to the split supervisor's extension.

Button	Description
NIGHT SERVICE	The split supervisor presses this button to send all calls to night service. The Night Service may be Trunk Group Night Service or Split Night Service. Also, a separate button for each type of night service may be available.
RECORD ANNCT	The supervisor presses this button to either listen to or to record an announcement for the split.
SERVICE OBSERVE	The supervisor presses this button and dials an agent extension number to listen to conversations on the telephone. The Service Observe feature permits the supervisor to check an agent's call-handling technique. An agent's telephone may also be assigned the SERVICE OBSERVE button so that the agent can listen to another agent's conversations. This capability is especially useful for agent training. Service Observing can be set up for listening only or for both listening and talking. For communication servers with EAS, a logical agent ID, which is associated with an agent, not the telephone the agent is currently using, can be service observed. For communication servers with Call Vectoring, VDNs can be service observed. Feature Access Codes which allow Service Observing from an external location or from a telephone that does not have feature buttons can be assigned through communication server administration.
VU STATS	Split supervisors and agents with display telephones press this button to display agent, split or skill, VDN, or trunk group data similar to that reported by CMS.

Chapter 2: ACD Call center features

This section describes the Avaya Call Center features that are administered on the Avaya communication server.

Related feature or screen:

See the *Avaya Aura™ Communication Manager Feature Description and Implementation*, for more information about the following related features or forms:

- Announcements/Audio Sources
- Calling Party/Billing Number
- CallVisor Adjunct-Switch Application Interface
- Class of Restriction
- Hunt Groups
- Malicious Call Trace
- Recorded Announcements
- Service Observing
- Callmaster™ phones
- 500, 2500, K2500, 7101A, 7102A, 7103A, 7104A, 8110, OPS, DS1FD, DS1SA, and VRU phones

Abandoned Call Search

About Abandoned Call Search

Abandoned Call Search allows the communication server to identify abandoned calls if the Central Office (CO) does not provide timely disconnect supervision. An abandoned call is one in which the calling party hangs up before the call is answered. Note that Abandoned Call Search is suitable only for older COs that do not provide timely disconnect supervision. Most COs provide timely disconnect supervision and do not require Abandoned Call Search.

Before an incoming Automatic Call Distribution (ACD) call rings a hunt group member or agent, the system checks to make sure that the calling party has not abandoned the call. If the calling party has abandoned the call, the call does not ring the hunt group member or agent.

If a call has been abandoned, the system determines if the calling party is still connected to the ground-start trunk at the CO. To do this, the system flashes (that is, opens the tip-ring loop for 150 to 200 ms) the CO end of the trunk. If the calling party is still connected, the CO does not respond. If the calling party has abandoned the call, the CO sends the system a disconnect signal within 800 ms. The system interprets this as an abandoned call, releases the trunk, and the call does not ring the hunt group member or agent.

Outside of the U.S., a flash of this duration may be handled differently. For more information about trunk flash, see *Avaya Aura™ Communication Manager Feature Description and Implementation*.

Abandoned Call Search considerations

- Abandoned Call Search works with ground-start analog trunks that do not provide disconnect supervision and that do react to a 500-ms break.
- Some older COs can take as long as two minutes to notify the communication server of a disconnect. Thus, the communication server must determine within one second whether the call has been abandoned, before extending the call. Even with Abandoned Call Search or disconnect supervision, there is a small probability that a call will be extended to the destination hunt group after the caller has hung up. Abandoned Call Search and disconnect supervision significantly reduce that probability.
- Abandoned Call Search allows agents and hunt group members to answer more calls because time is not wasted on abandoned calls. In addition, call-handling statistics that the Call Management System (CMS) generates are more accurate because it is clear when a call is abandoned.
- Abandoned Call Search adds an overhead of up to one second to each call delivered to an agent.

ACD options by agent

ACD options by agent description

You can now set the following ACD options for individual agents:

- ACW Agent Considered Idle
- AUX Work Reason Code Type
- Forced Agent Logout from ACW
- MIA Across Skills
- Logout Reason Code Type

In previous releases, you could set these options only system-wide.

Reasons to use ADC options by agent

Customers may want to set the ACD options differently for each agent.

ACD options by agent considerations

Use the Login ID screen to set the ACD options for each agent. The **ACD option** fields currently on the Feature-Related System Parameters screen remain the same. Consider the following:

- The ACD option settings on the Agent Login ID screen take precedence over the system-wide settings on the Feature-Related System Parameters screen.
- If any options are set to system on the Login ID screen, the system-wide setting is applied.
- The system setting is the default setting for both new and upgraded systems.
- The settings on the Feature-Related System Parameters screen or the Login ID screen override the system option and apply to both ACD calls and Direct Agent calls.
- For these changes to take effect, the agent must log out and log back in.

Add/Remove Skills

About Add/Remove Skills

Add/Remove Skills allows an agent using Expert Agent Selection (EAS) to add or remove some of their administered (assigned) skills. A skill is a numeric identifier in the communication server that refers to an agent's specific ability. For example, an agent who is able to speak English and Spanish could be assigned a Spanish-speaking skill with an identifier of 20. The agent then adds skill 20 to his or her set of working skills. If a customer needs a Spanish-speaking agent, the system routes the call to an agent with that skill.

Agents can dial feature access codes (FACs) to add or remove a skill. Or a supervisor with console permission can enter an agent's login ID and add or remove an agent's skill. If a supervisor adds or removes a skill for an agent, the agent receives a change notification.

To determine if they need to add or remove a skill, agents and supervisors can use:

- Queue-status indications
- Avaya Basic Call Management System Reporting Desktop VuStats
- Avaya Call Management System (CMS) or Basic Call Management System (BCMS) information

When adding a skill, the agent must specify the skill priority level (1 - 16).

On phones with displays, the system prompts the agent through the process of adding or removing a skill and displays the updated set of skills.

Add/Remove Skills feature considerations

Consider the following when using the Add/Remove Skills feature:

- A skill cannot be removed from an agent's skill set if the agent is on a call for that skill or in the After Call Work (ACW) state for that skill.
- With EAS, agents cannot remove their direct agent skill.

Interactions with other features and systems

The Add/Remove Skills feature has the following interactions with other features and systems:

Interaction	Description
Auto-Available Skills (AAS)	If an agent adds a skill that is administered as Auto-Available, you must set the AAS field to y for that agent's login ID on the Agent Login ID screen.
BCMS	BCMS begins tracking a new skill as soon as it is added. When an agent removes a skill, the real-time agent information specific to that skill is removed from the real-time reports, but it still appears on the historical reports.
EAS-PHD	When EAS-PHD is set as an option, agents cannot remove their direct agent skill. In an EAS environment, agents must have at least one skill assigned to them during a login session. With EAS-PHD, agents can specify up to 20 or 120 skills (depending on platform). **Note:** If EAS-PHD is not enabled, agents can specify only four skills.
VuStats	Because VuStats displays information gathered by BCMS whether BCMS is enabled or not, the BCMS interaction above applies to VuStats.

Agent/Caller Disconnect tones

About agent/caller disconnect tones

Call Center policy often expects that an agent or other call center resource remains on a call until the caller disconnects or is redirected to an automated entity such as an after-call survey. A feature related system yes/no option is available to play one of two distinct disconnect tones to help you identify whether a caller or the call center (e.g., agent) has disconnected first from an active incoming ACD or DAC call. These tones can be used to alert an agent that the caller has disconnected, and can be monitored by a service observer as part of agent training or by call recording features.

On an incoming, internal or external, ACD or DAC call, the remaining parties hear a:

- *Caller Disconnect Tone*: three (3) beeps heard when the caller terminates the call
- *Agent Disconnect Tone*: two (2) beeps (in most countries) heard when the agent or other call center party terminates the call

The remaining parties still on the call that hear the relevant disconnect tone include an agent, Auto Available Split/Skill (AAS) agents, service observers (human and systems such as call recorders), and any additional conferees other than the caller depending on who disconnects first. If the caller is the only remaining party on the call, no tone is played.

The caller disconnect tone is played if the caller disconnects first while at least one agent, station or other active party is still on the call. The remaining parties on the call hear the tone. Service Observers on the call will hear the caller disconnect tone along with the active party.

The agent disconnect tone is played if the caller is the last active party on the call and at least one observer (e.g., Service Observer or Call Recording System) is still on the call. The observers hear the tone but the caller does not.

Note:

If an incoming trunk is measured, CMS and IQ may offer reports that indicate which party hung up first. I but If the incoming trunk is unmeasured, CMS and IQ cannot determine which party hung up first. This is true regardless of whether these tones are used. However, you may compare your call recording with reports and possibly notice a discrepancy. To guarantee consistency, measure the trunks.

Reason to use Agent/Caller Disconnect tones

You may want an agent or other call center resource to remain on a call until the caller disconnects or is to be redirected to an automated entity, such as an after-call survey. This option helps you identify whether a caller or the call center (agent) disconnected first from an active ACD or DAC call.

The following table covers the operation of relevant scenarios for the disconnect tones:

Party Leaving Call	Remaining Parties [1]	Action[2]
Agent	caller	no tone played, connection is dropped
	caller + 1 or more observers	Agent Disconnect Tone is played to observers, connection is dropped
	caller + transferred to party	no tone played, call remains active
	caller + 1 or more conferees	no tone played, call remains active
Last call center party or trunk connection	caller is last party after a conference/transfer	no tone played, connection is dropped,

1 Up to two observers can be observing an agent or transferred to agent/station. Once the observed agent/station disconnects, those observers will also be removed from the call connection. One or more observers can be observing the call via VDN Observing; VDN observers will remain with the call until the call connection is dropped

2 The connection will not be dropped after the agent or Call Center party releases if VDN Return Destination applies to the remaining caller.

Party Leaving Call	Remaining Parties [1]	Action[2]
	caller is last party after a conference/transfer + 1 or more observers	Agent Disconnect Tone is played to observers, connection is dropped
Caller	agent	Caller Disconnect Tone is played to agent, connection is dropped
	agent + 1 or more observers	Caller Disconnect Tone is played to agent and observers, connection is dropped
	agent + 1 or more conferees	Caller Disconnect Tone is played to agent and conferees, call remains active
	transferred to single party connection + 0 or more observers	Caller Disconnect Tone is played to remaining party and any observers, connection is dropped
	conference connection + 0 or more observers	Caller Disconnect Tone is played to conferees and any observers, call remains active

Definition of the Tones

Tone	In the country	Defined as
The Agent Disconnect Tone	US	• 440hz at -17db for 200msec • Silence for 200 msec • Repeat once
	Australia	• 525hz at -11 db for 100msec • Silence for 100 msec • Repeat once
	Italy	• 425 hz at -4db for 100msec • Silence for 100 msec • Repeat once

Tone	In the country	Defined as
	Hong Kong	• 440hz at -13db for 200msec • Silence for 200 msec • Repeat once
	Belgium	425 hz at -11db for 200msec
	France	440 hz at -11db for 300msec
The Caller Disconnect Tone is defined as:	All countries/Regardless of the administered country	• 404hz at -16db for 100ms • Silence for 50ms • Repeat twice

Agent/Caller Disconnect Tones interactions

1. Agent/caller disconnect tones will apply after the VDN Return Destination occurs.
2. Agent/caller disconnect tones will apply to No Hold Conference or ASAI Single Step Conference as described in the table.
3. Any tones applied to the connection such as Service Observing Warning Tone, Conference Drop Tone, etc. will still apply and take precedence if there is a conflict in applying one tone versus a disconnect tone. The agent or caller disconnect tone will still be given when appropriate.
4. Agent/Caller Disconnect tone does not apply to an outgoing call (OCM-ACD or otherwise) placed by or for the agent.
5. The application of Agent/Caller Disconnect Tones does not apply to shuffled calls and call preserving failovers are not expected to hear the tones.
6. The playing of the Caller Disconnect tone as indicated in the relevant scenarios table also applies when the caller disconnect is received while the caller is on hold.
7. When the agent and caller drop simultaneously or almost simultaneously, generally the tone will be played for the first one to drop. However, the drop (or release) of the second party can immediately cancel or truncate the playing of that disconnect tone. This could result in making a three burst tone sound like a one or two burst tone resulting in indeterminate results.

Agent Call Handling

About Agent Call Handling

Agent Call Handling allows you to administer functions that Automatic Call Distribution (ACD) agents use to answer and process ACD calls.

You define the following agent capabilities in support of Agent Call Handling:

- Agent Login and Logout
- Agent Answering Options: Automatic Answer (zip tone) or Manual Answer
- ACD Work Modes: Auxiliary Work (AUX Work), Auto-in, Manual-in, or After Call Work (ACW)
- Timed ACW (After Call Work)
- Agent Request for Supervisor Assistance
- ACD Call Disconnect (Release button)
- Stroke counts
- Call Work Codes (CWCs)
- Forced Entry of Stroke (Event) Counts and/or Call Work Codes

Agent capacity and related limits on page 58 describes agent-capacity planning.

Note:

All of these agent capabilities are also supported through the CallVisor Adjunct/Switch Applications Interface (ASAI). For more information about the CallVisor Adjunct-Switch Application Interface, see *Avaya Aura™ Communication Manager Feature Description and Implementation*.

Communication server controls

This section describes how the communication server controls agent work.

Note:

This information applies generally to ACD; see Expert Agent Selection for more information on EAS.

This section includes the following topics:

Agent login and logout

To receive ACD calls, an agent must log into the system. An agent can be logged into multiple splits/skills. If a hunt group is measured by Call Management System (CMS) or Basic Call Management System (BCMS) or is a skill, an agent must enter a login ID.

Login to a split (non-Expert Agent Selection)

To log in, an agent goes off-hook and dials the login feature access code (FAC), followed by the split number and the login ID, if required. If login is successful, the agent automatically enters Auxiliary Work mode for that split. The Auxiliary Work button lamp for that split lights steadily and the agent hears the confirmation tone.

If the split is measured, the system sends messages to CMS or BCMS that the agent (identified by login ID) has logged in and has entered Auxiliary Work mode.

Login to a skill (Expert Agent Selection)

To log in, an agent goes off-hook and dials the system assigned login feature access code (FAC), followed by the agent's assigned login ID and password, if required. If login is successful, the agent automatically enters Auxiliary Work mode for all the skills assigned to the agent and the assigned skills are displayed on the station set. The Auxiliary Work button lamp(s) on the station set lights steadily and the agent hears the confirmation tone.

Cancelled logins

Login is cancelled and the agent receives an intercept tone if any of the following occur during login:

- The agent dials an invalid login FAC.
- With non-EAS the agent:
  - Dials an invalid split number.
  - Dials a split number the agent is not assigned for.
  - Dials a split number the agent is already logged in to.
  - Is logged in to the maximum number of splits (4).
  - Dials an invalid or unassigned BCMS/VuStats login ID.
- With EAS the agent dials an invalid agent login ID or password.

Login is cancelled and the agent receives a reorder tone if the system maximum number of agents are already logged in.

An EAS agent can be denied login to some of his assigned skills if the system maximum number of agent-skill pairs has been reached. The display of skills will show a "*" for each skill not logged in.

Logout

The agent should log out when he or she leaves for an extended period and is unavailable for ACD calls. If the split or skill is measured by CMS or BCMS and an agent logs out, a message is sent to the CMS or BCMS so that the agent's status is no longer measured. In a non-EAS environment, if an agent is logged into multiple splits, the agent should log out of each split.

When temporarily unavailable for calls, an agent should use Auxiliary work mode, rather than logging out. CMS or BCMS can continue tracking the agent's auxiliary work time.

To log out of a split, an agent goes off-hook and dials the logout FAC followed by the split number. To log out of a skill the agent dials the logout FAC and is automatically logged out of all the assigned skills. If logout is successful, the agent hears confirmation tone and work-mode button lamps darken. The logout is canceled and the agent receives an intercept if any of the following occur during logout:

- The agent dials an invalid logout FAC or split number.
- The agent dials a split number for a split that he or she is not logged into.

If an agent is using a handset in Automatic Answer mode, the agent can log out simply by hanging up or turning off the headset. (This does not mean pressing the release button on a Callmaster phone.) This does not apply to quick-disconnect. If the agent pulls the handset to log out, the agent is automatically logged out of all splits that he or she has logged into.

Agent answering options

An agent can answer ACD calls by using either a headset, handset, or speakerphone. You can assign an agent as either Automatic Answer or Manual Answer.

Note:

Use Automatic Answer with a headset. See Agents with Automatic Answer on page 63 for more information.

Automatic Answer

The information in this section applies to ACD and EAS environments.

An agent assigned to Automatic Answer hears zip tone and connects directly to incoming calls without ringing.

Note:

You can administer Automatic Answer to apply only to ACD calls or to apply to all calls terminating to the agent's set. If all calls are Automatic Answer and the agent receives direct-extension calls, he or she should always activate Call Forwarding, or Send All Calls when leaving temporarily or for an extended period, so that calls do not terminate to an unstaffed station.

Manual Answer

An agent assigned to Manual Answer hears ringing, and then goes off-hook to answer the incoming call.

ACD work modes

At any given time, an agent can be in one of four work modes:

- Auxiliary Work (AUX)
- auto-in
- manual-in
- After Call Work (ACW)

An agent can change work modes at any time.

To enter any work mode, an agent presses the button or dials the FAC for that mode, depending on what you have administered. If the agent has no active or held calls, the work-mode button lamp lights steadily and CMS or BCMS is informed of the agent's mode change. If the agent has active or held calls, the lamp flashes until all calls are dropped, then the new work mode's lamp lights steadily and CMS or BCMS is informed of the agent's mode change.

The attempt is cancelled and the agent receives an intercept if the agent:

- Tries to enter a work mode for an invalid split or skill
- Tries to enter the work mode for a split or skill of which he or she is not a member
- Dials an invalid FAC

Auxiliary Work mode

An agent should enter Auxiliary Work mode whenever taking a temporary break. This makes the agent unavailable for ACD calls and removes them from the most-idle-agent queue. CMS and BCMS can continue to track the agent.

In a non-EAS environment, when an agent is in AUX Work mode for a particular split, the agent may be available for ACD calls to other splits that the agent is logged into, depending on the agent's state in those splits. Even in AUX, the agent is still available for non-ACD calls. CMS/BCMS is notified whenever an agent in AUX Work mode receives an incoming non-ACD call or makes an outgoing call. When an agent logs into a split, he or she automatically enters AUX Work mode for that split.

Note:

Agents in vector-controlled splits/skills can go into AUX Work mode even if they are the last agent and calls are queued to that split or skill.

Although an agent in Aux work mode is unavailable to receive ACD calls, such an agent may become available to receive ACD calls if the agent:

- Has moved to Aux with an interruptible reason code
- Is logged into one or more skills that are administered as interruptible
- Is logged into one or more skills that have exceeded an administered service level threshold

For more information on this feature, see Interruptible Aux work. The agent is either automatically moved (“forced”) to an available state or requested to become available, depending on how the agent’s skill is administered. After changing state, agents are available to receive calls from any of their skills, not just the one that caused the interruption from Aux. Applicable agent selection criteria are used once the agent has become available.

Auto-in mode

In auto-in mode, the agent automatically becomes available for answering new ACD calls upon disconnecting from an ACD call.

Manual-In mode

In Manual-In mode, the agent automatically enters ACW mode for the split or skill upon disconnecting from an ACD call and is not available for any ACD calls. To become available for ACD calls, the agent must manually reenter either auto-in mode or manual-in mode.

After Call Work mode

An agent should enter ACW mode when he or she needs to perform ACD-related activities, such as filling out a screen related to the ACD call. The agent is unavailable for ACD calls to all splits/skills while in ACW mode. Communication server administration determines whether the agent remains in the Most Idle Agent queue while in ACW.

When an agent is in the Manual-In mode and disconnects from an ACD call, he or she automatically enters ACW mode. Although no longer available for ACD calls, the agent is still available for non-ACD calls. CMS, IQ and/or BCMS is notified whenever an agent in ACW mode receives an incoming non-ACD call or makes an outgoing call.

Timed After Call Work

With Timed ACW administered either on the active VDN for the call (takes precedence) or the hunt group (skill) that the call queues to, an auto-in agent is immediately placed in the ACW state for the specified length of time after completing the currently-active ACD or Direct Agent

Call (DAC) call. When the Timed ACW Interval expires, the agent automatically returns to the available auto-in work mode. If the agent manually activates ACW while not on a call, the agent is placed in ACW (not timed) state regardless of whether the agent is in auto-in or manual-in mode.

Timed ACW applies to an agent when handling an ACD/DAC call that has the feature enabled for the active VDN for the call. The active VDN for the call is based on the VDN Override rules For more information on VDN Override, see the *Programming Call Vectors in Avaya Aura™ Call Center*document. When the call is placed in queue for a split/skill which also has a Timed ACW interval assigned on the hunt group screen, the interval assigned to the active VDN is applied when the agent is put into ACW.

When the Timed ACW Interval is administered, if the caller drops while on hold or the agent transfers the call, an auto-in agent is immediately made available instead of being placed in timed ACW. For these cases, the Timed ACW After Xfer or Held Call Drops option can be used to place the agent in the Timed ACW mode (for the assigned interval) instead of available. For more information, see About Timed ACW after transfer or held call drops option.

Use Timed ACW to allow agents to rest between incoming ACD calls, or to pace agents when they have to complete work from the previous call within an allotted time.

Timed ACW after transfer or held call drops option

An auto-in agent handling an incoming ACD or DAC call can be placed into the Timed ACW mode instead of immediately making the agent available if the caller drops from a held call or the agent transfers the call. You can enable this option for the agents in a hunt group or for calls delivered from a VDN by setting the **After Xfer or Held Call Drops?** option to `y` on the same line as the Timed ACW Interval (sec) field when Timed ACW is set to y (the Timed ACW Interval field is set to a non-zero value) either on the hunt group screen or the VDN screen.

If the agent places an ACD/DAC call on hold or holds a conference that includes an ACD/DAC call and the caller disconnects the call, the agent is put in Timed ACW for the interval specified in the associated Timed ACW Interval (sec) field assigned on the VDN screen or Hunt Group screen if the **After Xfer or Held Call Drops?** option is enabled. As with the basic Timed After Call Work feature the setting on the active VDN for the call takes precedence over the setting on the hunt group. This feature also places an Auto-in agent into the Timed ACW mode after the agent transfers an incoming ACD/DAC call.

The following table explains the call treatment based on the Timed ACW after transfer or held call drops feature for an ACD or DAC call:

Agent Auto-in or Manual-In	Scenario	Resulting agent state if the Timed ACW after Xfer or Held Call Drops option is NOT enabled…	Resulting agent state if the Timed ACW after Xfer or Held Call Drops option IS enabled…
Auto-in	agent or caller drops call	Timed ACW	Timed ACW
	caller drops from hold	Available	Timed ACW
	agent transfers a call then drops	Timed ACW	Timed ACW
	agent initiates a consultative transfer, caller drops from hold	Available	Timed ACW
	caller drops while call is ringing or initial VOA (VDN of Origin Announcement) is playing	Available	Available
	call drops after the agent selects pending ACW mode	ACW	ACW
	In a Multiple Call Handling (MCH) condition, an ACD/ DAC call is on hold, a 2nd ACD call is active and a Timed ACW interval setting applies based on the 2nd call[3] – held ACD/ DAC call with MCH drops and then 2nd active ACD call drops and TACW applies	Available	Timed ACW (matches Manual-In operation)
	In an MCH condition, an ACD/DAC call is on hold, a station call is active and Timed ACW applies, the held ACD/DAC call with MCH drops and	Available	Timed ACW (matches Manual-In operation)

[3] Whether Timed ACW would apply in this scenario is to be based on the settings for the last active ACD/DAC call. MCH is Multiple Call Handling.

Agent Auto-in or Manual-In	Scenario	Resulting agent state if the Timed ACW after Xfer or Held Call Drops option is NOT enabled…	Resulting agent state if the Timed ACW after Xfer or Held Call Drops option IS enabled…
	then an active station call drops		

TACW After Held Call Drops/Call Transferred Option Interactions

1. If the agent places an ACD/DAC call on hold while initiating a consultative conference or transfer and the caller disconnects, TACW is pending while the agent is active making a call or talking to another party, then is activated after the active call drops.
2. When application of Timed ACW is based on the VDN for the call and VDN Override is set, the setting of the After Xfer or Held Call Drops option is taken from the active VDN for the call on call delivery along with the timed interval from the associated **Timed ACW Interval (sec)** field.
3. Timed ACW as well as the option can apply to VDN calls to Auto Available Split/ Skill (AAS) stations or AAS stations in a hunt group with a Timed ACW interval set on the hunt group screen. The timed interval provides a period when a call will not be delivered to the AAS station port and the station port will then be automatically made available after the timed period expires.

Cancelling Timed ACW

Timed ACW is cancelled under the following conditions:

Agent activates auto-in or manual-in mode

When an agent activates auto-in or manual-in mode during Timed ACW, the agent becomes available and timed ACW is cancelled. An agent can change to manual-in mode before or during a call. The system cancels Timed ACW and applies ACW (not timed) mode when the call is released. The agent remains in ACW until he or she requests another mode. When the agent releases an ACD call, the ACW lamp (if provided) lights. At the end of the administered Timed ACW interval, the ACW lamp goes dark and the auto-in lamp lights.

Agent manually activates ACW

Timed ACW is canceled when an agent presses the ACW button or dials the ACW FAC.

Agent activates Auxiliary Work mode

If an agent activates Auxiliary Work mode during Timed ACW, the agent is placed in that mode and Timed ACW is cancelled.

Ringing or held ACD call is dropped by the caller

If the Timed ACW after held call drops option is not enabled and a ringing or held ACD call to an auto-in agent is dropped by the caller, the agent is immediately made available in the auto-in mode. However if the Timed ACW after held call drops option is enabled and the caller disconnects while on hold, Timed ACW applies. The agent will still be made available if a ringing call is disconnected. For more information, see About Timed ACW after transfer or held call drops option.

Agent request for supervisor assistance with or without an active ACD call

To request assistance from the split or skill supervisor, an agent, with or without an active ACD call, presses the Assist button or puts the call on hold and dials the Assist FAC plus the split or skill number. The agent must be logged into the split or skill. Assist generates 3-burst ringing at the supervisor’s station. If a split or skill supervisor is not assigned, the agent receives intercept tone.

Attendants should press the Start button before pressing the Assist button. This allows them to later transfer the call. This rings like a priority call at the supervisor’s set.

When the agent presses the Assist button, the following happens:

1. If the agent is active on an ACD call, the ACD call is automatically placed on hold and a call is placed to the split or skill supervisor. If the agent is not active on an ACD call, a call is automatically placed to the supervisor.
2. CMS or BCMS is notified of the request and the supervisor’s display shows that the call is a request for assistance. This rings like a priority call at the supervisor’s set.
3. The caller hears silence or music on hold.
4. After the agent has talked to the supervisor, the agent can drop the assist call and return to the ACD call, set up a conference call with the supervisor and the calling party, or transfer the call to the supervisor.

When the agent puts the call on hold and dials the Assist FAC plus the split or skill number, the system handles the request as if the agent pressed the Assist button, except that the Assist call does not follow the supervisor’s coverage path.

Stroke counts

Stroke counts allow you to record in CMS the number of times that a particular customer-related event occurs. For example, agents could press a button each time a customer requests information on a certain item.

Stroke counts are reported to CMS in real time. The communication server does not store stroke counts. Use stroke counts only when CMS is connected and you have defined ACD splits/skills to be measured by CMS.

Stroke counts allow agents to record up to nine administrator-defined events on a per-call basis. You can assign 10 Stroke Count button types. Stroke Count 0 is reserved for tracking Audio Difficulty or poor transmission quality.

For troubleshooting purposes, CMS records the equipment location of the trunk that the agent was using when he or she pressed the Audio Difficulty button. Make sure that agents are aware that pressing this does not improve audio transmission quality.

To enter a stroke count, an ACD agent presses a Stroke Count button while off-hook. The system validates that the agent is either active on an ACD call or in the ACW mode for an ACD split or skill. If yes, the feature lamp lights steadily for two seconds to indicate activation and the stroke count is sent to CMS. If not, the feature lamp flutters and no message is sent.

Call work codes

Call work codes are up to 16-digit sequences that ACD agents enter to record customer-related information. You define the codes for your site. Codes that agents enter are sent to CMS for storage for splits/skills measured by CMS and only when the link to the CMS is up. Agents must have multiappearance phones (for example, Callmaster) to enter call work codes.

To enter call work codes, the agent must be off-hook and either:

- On an ACD call
- In ACW mode after disconnecting from a call while in Manual-In mode remaining off-hook
- In Timed ACW after disconnecting from a call while in auto-in mode
- In auto-in mode and pending for ACW mode

The sequence of event is as follows:

1. The agent selects **Call Work Code (CWC)** button.
2. The CWC lamp lights steadily and a C: prompt appears on the agent's display. The agent must wait for the ready indication before entering the call work code or the caller hears the touch-tone digits being dialed.
3. Agent enters up to 16 digits on the dial pad. The agent can press `*` to erase digits.
4. The agent presses `#` to send the code entry to CMS.
5. The Call Work Code lamp goes dark and the display returns to normal.
6. If the agent presses any feature button or hangs up during digit collection, the code entry is cancelled and data is not sent to CMS. The CWC lamp goes dark and the display is cleared.

Call work codes may be used by as many as 100 agents simultaneously. If 100 agents are simultaneously using this function, and another agent attempts to enter a call work code, the agent receives a display message to try again later.

Forced entry of stroke counts and call work codes

You can administer a split or skill so that agents must enter a stroke count and/or a call work code before becoming available for another call using Manual-In mode.

Note:
Multi-appearance phones or an attendant console are required for agents to enter stroke counts or call work codes.

To enter a stroke count and/or call work code, the agent must be on a call, or in ACW mode after releasing a call in Manual-In mode.

After releasing a call, the agent automatically enters ACW mode and cannot return to Manual-In mode until entering a stroke count or call work code. If the agent presses the Manual-In button or FAC before entering a stroke count or a call work code, the Manual-In lamp flutters or intercept tone is given.

Once the agent enters a stroke count or call work code and presses the Manual-In button or FAC, he or she returns to Manual-In mode and the Manual-In lamp lights.

Any of the agent's splits/skills can have Forced Entry assigned. If the agent goes into Auxiliary Work mode in any split or skill, the Forced Entry requirement for all other splits/skills is removed.

Agent capacity and related limits

Agent Sizing adds an overriding capacity limit to the number of logged-in ACD agents. It can be used to limit the number of logged-in ACD agents to a number less than or equal to the maximum supported by the system configuration.

The logged-in ACD agents limit applies to ACD agents in traditional or non-EAS ACD splits or in Expert Agent Selection (EAS) skills. Auto-Available split or skill (AAS) agent ports are logged in and counted when they are first assigned, while the non-AAS agents are counted when they actually log in. Each logged-in agent is counted as a single agent independent of the number of splits/skills logged in to for the Logged-in ACD agents limit. AAS and non-AAS agents are counted towards this limit whether they are BCMS/CMS measured or not.

The Logged-in Advocate Agent Count feature

The Logged-in Advocate Agent Count feature counts the number of Avaya Business Advocate agents who are logged in at the call center. The feature bases the count on whether or not a logged-in agent has any Avaya Business Advocate features, except Predicted Wait Time, assigned or associated with the agent. With this feature, Advocate-counted agents are still counted as ACD agents.

Avaya Business Advocate licensing

When an agent logs in, the**Logged-In Advocate Agents** license setting is counted only if any of the following fields are set as described in the table.

screen	Field	Set to
Login ID for the agent	**Service Objective**	`y`
	Call Handling Preference	`percent-allocation`
	Reserve Level	`1` or `2`
hunt group for the skill the agent logs into	**Service Level Supervisor**	`y`
	Group Type	`pad`
	Dynamic Queue Position	`y`

The**Service Objective** field setting on the hunt group screen is not used for Avaya Business Advocate agent counting. Only agents whose Login ID screen have the**Service Objective** field set to`y` are counted. Skills with Least Occupied Agent assignments of type ucd-loa or ead-loa are not counted as Avaya Business Advocate agent types starting with Communication Manager Release 9.

The agent sizing license limit is administered by authorized Avaya personnel. The `Logged-in ACD Agents` field (and `Logged-in Advocate Agent Count`) on the System-Parameters Customer-Options screen are set by the loaded license file. The maximum number of allowed logged-in ACD and Avaya Business Advocate agents is set to correspond to the configuration you purchase.

Agent sizing when agents work in shifts

For agent sizing, if you have agents working in shifts, you should purchase enough agent capacity to allow for a smooth shift change. If agents on a subsequent shift are logging in before agents in the previous shift have logged out, agents could be denied login because too many agents are currently logged in. Additionally, the non-ACD and/or non-agent (AAS/VRU) use of Hunt Group resources must be considered. Call center managers need to be aware of their logged-in ACD agent and other related limits when adding agents to handle a traffic peak or when planning a special campaign. Some of the resource utilization is displayed dynamically on the Display Capacities screen.

Limit considerations

In addition to the logged-in ACD agents limit, the number of agents supported is dependent on the upper limits that the system platform supports. The following limits must also be considered.

- Maximum Hunt Group members
  - Non-ACD members include hunting groups with or without queues, message center service groups, messaging-system groups, and remote messaging-system groups. Each line or port in a group is counted once when assigned.
  - ACD members (also called agent-split pairs or agent-skill pairs with EAS). For agents in multiple splits/skills, each combination (pair) is counted as a member (e.g., an EAS agent logged into 4 skills or a non-EAS agent assigned to 4 splits counts as 4 members). Non-EAS ACD members are counted when assigned (note that many more splits can be assigned to an agent than can be logged into but each agent-split pair is still counted towards the limit). EAS ACD members are counted when they log in.
  - Avaya Business Advocate Agents - Each logged-in Avaya Business Advocate agent is counted as both an ACD member and as an Avaya Business Advocate agent.
- Hunt Group members per group - Count of non-ACD or ACD members within a split or skill. Counting is done as above for maximum Hunt Group members.
- Additional traditional ACD (non-EAS) agents limits:
  - Maximum logged-in agents system limit
  - Maximum splits an agent can log into
- Additional EAS limits:
  - ACD members (skill pairs) administered - Limits skill assignments to agents (each AAS port is counted as one skill pair)
  - Agent login IDs administered - Limits number of AAS ports and EAS agents that can be pre-assigned
  - Agent login IDs logged-in (staffed) system limit - Upper limit on the number of EAS agents (and AAS ports) that can be logged-in simultaneously
  - Skills per agent - The maximum number of skills a particular agent can be assigned
- Call Management System (CMS) logged in ACD members (agent-split or skill pairs) limits assigned. Both an Avaya setup and a customer-administered limit are assigned in CMS. These limits are related to the CMS memory/hardware configuration equipped and are passed over the link to the communication server to reduce/set the externally measured

logged-in ACD member component of the Hunt Group member limit to that supported by CMS.

- BCMS internally measured ACD agents system limit. Non-EAS ACD agents counted when assigned while EAS agents are counted when logged in.

When the maximum number of ACD agents are logged in or any of the other above limits are reached, an agent who attempts to log in hears reorder tone or is otherwise denied log in. Also with EAS, an agent logging in may not have all the assigned skills logged in if the ACD member limit is reached.

The administrator of a non-EAS system can be blocked from adding agents to splits using the Hunt Group screen.

The administrator of an EAS system can be blocked from assigning additional login IDs or skills to an agent using the Login ID screen if the relevant system limits are reached.

Callr-info display options

This feature allows administrators to decide when an agent's station display is cleared of caller information (Callr-info). Options include:

- Clearing the existing call information when the next call is received
- Clearing the existing call when the call is released - whether the agent enters After Call Work (ACW) or not
- Clearing the existing call when the agent leaves ACW mode or if the agent does not enter ACW, when the call is released

Reason to use

This feature is designed to meet U.S. government privacy requirements as specified in the Health Insurance Portability and Accountability Act (HIPAA). HIPAA has a specification that medical records cannot be left where they can be viewed by others.

Call centers also have a requirement that agents can see the data on the station display when the agent goes into ACW mode. Agents must be able to see the data in order to use it for other purposes without having to write it down.

Administering Callr-info

This feature applies only to those stations supporting a two-line display, such as Callmaster IV, Callmaster VI, or 8434D stations. It also applies to an IP Softphone or IP Agent that is emulating a two-line display telephone.

To administer the Callr-info display options:

1. Go to thefeature related system parameters screen.
2. Set one of the following options on the**Clear callr-info** field.

Option	Clears the display of the existing call information
next-call	When the next call is received. This is the default setting.
on-call-release	As soon as the call is released. A call is released when the agent presses the release button or when the caller disconnects from the call.
leave-ACW	As soon as the call is released except when the agent is put into the ACW state. The station display is cleared when the agent leaves ACW to go into any other state including when the agent is forced out of ACW.

Ringer-off control of auto-answer for non-ACD calls

Starting in Release 3.1, administrators can allow agents to use the ringer-off button to prevent non-ACD auto-answer calls from ringing.

Previously, besides agents hearing an answer tone in their headsets, agents also heard an audible single ring for non-ACD auto answer calls on their station set when any of the following settings applied:

- The **Auto-Answer** field on the Agent LoginID screen was set to `all`, or
- The **Auto-Answer** field on the Agent LoginID screen was set to `station` and the **Auto-Answer** field on the Station screen was set to `all`

Agents could not use the ringer-off button to prevent this single ring from occurring.

Reason to use: Some call centers prefer quiet environments where they do not want to hear the audible ring for every non-ACD call that is received.

Administration: Set the **Allow Ringer-off with Auto-Answer** field on the feature-related system parameters screen to `y`.

Agent Call Handling considerations

Release button

Agents using Automatic Answer are logged out of all splits/skills when they disconnect from an ACD call by hanging up or by using the Drop button. Therefore, agents should always use the Release button to force the release of a connection.

Timed ACW

To prevent agents from canceling Timed ACW by pressing the Manual-In or ACW buttons, do not assign these buttons to the agents' phones. Timed ACW cannot be assigned to AAS, adjunct-controlled, messaging system, Remote AUDIX, or Message Center splits/skills. In addition, VDN-Timed ACW does not apply to calls routed to a converse split or skill by way of the **`converse-on`** vector command. Timed ACW assigned to a converse hunt group applies.

BCMS and CMS track Timed ACW as standard ACW work states. Time spent in Timed ACW is not specifically identified.

Non vector-controlled splits/skills

For non vector-controlled splits/skills, the last available agent in a split or skill cannot enter Auxiliary Work mode if any calls remain in the queue. (However, the agent can log out.)

When the last available agent tries to enter Auxiliary Work mode, the following occurs:

- The Auxiliary Work button flashes indicating the change is pending.
- New calls on the ACD split or skill either receive busy tone or redirect to coverage. Calls in the queue continue to route to the last available agent until the queue is empty.
- At the last available phone or console, the Auxiliary Work button lamp flashes until the queue is empty. The telephone then enters Auxiliary Work mode and the associated lamp lights steadily.

Agents logged into multiple splits/skills

If an agent is logged into multiple splits/skills, the agent may become unavailable for calls to one split or skill because of activity at another split or skill. For example, if an agent enters After Call Work mode for one split or skill, the agent becomes unavailable for calls to other splits/skills.

An agent should not log into a split or skill while a call is on hold at the extension.

Agents with Automatic Answer

Agents who use Automatic Answer should use a headset. The agent hears zip tone through the headset and automatically connects to a call.

If either the incoming trunk group or the agent's extension is data-restricted, the agent does not hear zip tone. Therefore, do not assign data-restriction to a headset user's extension.

It is not recommended that you use Automatic Answer with a handset or speakerphone. The handset or speakerphone must be off-hook (handset lifted or speakerphone turned on) all the time for the agent to hear zip tone.

If automatic answer is assigned for all calls, when a non-ACD call arrives, non-ACD Auto-Answer agents hear Incoming Call ID tone, not ringing.

Callmaster telephones

Calls for Callmaster digital phones and attendant stations are announced by double tones. The tones that are doubled are zip (Auto-Answer ACD agent calls) and Incoming Call ID (for End of VDN of Origin announcements and all other Auto-Answer calls). The user hears part of the first tone and all of the second tone.

Agents assigned to hunt-group and ACD calls

Do not use agents for hunt-group calls and ACD split or skill calls simultaneously. Otherwise, all of the calls from one split or skill (either ACD or hunt-group) are answered first.

The oldest call-waiting termination is supported only for agents who are servicing ACD calls only.

Agent Call Handling interactions

Interaction	Description
Abbreviated Dialing	Assign Abbreviated Dialing buttons to make agent login easier. You can program an Abbreviated Dialing button to dial an access code, split number, and/or agent login ID. You can use Autodial feature buttons to assign login and logout feature buttons.
Auto Answer with Interruptible Aux	Even if an agent's skill is administered to be forced interruptible (RL set to 'a' or 'm' on the Agent LoginID form), the agent administered as Auto Answer is treated as requested interruptible (RL set to 'n' on the Agent LoginID form), not forced. Otherwise, a forced interruptible agent administered as Auto Answer would be made available using the interruptible aux feature even if the agent is not physically present or ignores the call. In such a situation, a call may be delivered and automatically answered by the endpoint but no one will answer the caller. For more information on Interruptible Aux, see Interruptible Aux work.
Bridging	ACD split or skill calls are not bridged. Station calls are bridged and agents are able to bridge onto them. If an agent bridges onto a call, the call is considered a non-ACD extension-in call. The agent is not available for an ACD call unless

Interaction	Description
	the agent is a member of a many-forced, one-forced, or one-per-skill MCH split or skill. The agent can put the call on hold and become available to receive ACD calls even in non-MCH splits/ skills if only bridged appearances are active.
Call Coverage	If an ACD call routes to an agent as a result of covering to a VDN (where the VDN is the last coverage point in the coverage path), Timed ACW applies as administered for the VDN or split or skill.
Call Forwarding	If an ACD call routes to an agent after being call-forwarded to a VDN, Timed ACW applies as administered for the VDN or split or skill.
Call Pickup	When an ACD agent answers a call with Call Pickup, the call is treated as an incoming non-ACD call. The agent can put the call on hold and become available for additional calls.
Call Work Codes	The CWC 100-agent limit is shared with reason codes. Therefore, no more than 100 agents can simultaneously enter either a call work code or reason code.
CallVisor ASAI Adjunct	If a split or skill hunt group has CallVisor ASAI as the controlling adjunct, you cannot administer Timed ACW for the split or skill. Additionally, if an ACD call is routed to an agent in an adjunct-controlled split or skill, the agent is not placed in Timed ACW when the call ends.
Avaya CMS	Timed ACW is reported on CMS reports in the same way as any other ACW. CMS gives exception notification only on ACW intervals that are longer than the defined threshold.
Conference	If an agent receives an ACD call through a VDN and then conferences in other agents, the agents added to the call use the Timed ACW interval associated with the number dialed to conference them. An ACD agent on conference with more than three parties may cause inaccurate CMS measurements.
Expert Agent Selection	When EAS is active, all ACD hunt groups are assigned as vector-controlled skills. Agents log in using Logical Agent IDs. Skills can be preassigned to login IDs, however, assignment on the Login ID screen does not actually assign a non-AAS login ID to the skills until the ID is logged in. When the login ID is logged in, each skill is counted as a hunt-group member towards the system hunt-group member limit, the per-group member limit, and each agent is counted as a logged-in ACD agent.
Interruptible Aux notification indicators	Interruptible Aux notification indicators on an agent's endpoint (display message, flashing lamp and ring-ping alert tone) persist across call-preserving system resets (levels 1 and 2). This keeps agents aware of critical service level conditions even when certain serious system events occur. For more information on Interruptible Aux, see Interruptible Aux work.

Interaction	Description
Manual Answer with Interruptible Aux	If a forced interruptible agent is administered as Manual Answer and is moved to an available mode, when a call is delivered to the agent's endpoint the agent must explicitly answer the call. RONA applies to the calls delivered to the endpoint that are not answered within the set number of rings. For more information on Interruptible Aux, see Interruptible Aux work.
Multiple Call Handling	If MCH calls are on hold at an agent's telephone and the agent completes a call that normally is followed by Timed ACW, the agent is not placed in ACW. If no MCH calls are on hold, but one is alerting at the station when the Timed ACW call completes, the agent is placed in ACW. MCH affects when agents can enter different work modes and when calls are delivered to agents in Manual-In or auto-in work modes. See Multiple Call Handling for detailed information.
Transfer	If an agent receives an ACD call through a VDN and then transfers the call to another agent, the second agent uses the Timed ACW interval assigned to the number that was dialed to transfer the call. For an EAS agent, this is the Timed ACW interval associated with his or her direct agent skill. For an agent receiving a call transferred to a second VDN, this is the VDN Timed ACW interval of the second VDN. The agent who originally transferred the call uses the ACW associated with the VDN or split or skill that first received the call.
VDN Override	If a VDN has VDN Override set to no and the vector routes a call to a second VDN, the first VDN's Timed ACW interval is used for Timed ACW. If VDN Override is set to yes, the second VDN's Timed ACW interval is used. If no interval is set for the second VDN, no Timed ACW is associated with the call.
Voice Response Integration	If an ACD call routes on a **`converse`** vector command, any VDN-Timed ACW associated with the call is ignored for agents in the converse split or skill. However, if the converse split or skill has an administered Timed ACW interval, the answering agent associated with the split or skill is placed in Timed ACW when **`converse`** vector command processing completes.

Auto-Available Split/Skill

About AAS

Auto-Available Split/Skill (AAS) allows members of an ACD split or skill to be in auto-in work mode continuously. An agent in auto-in work mode becomes available for another ACD call immediately after disconnecting from an ACD call.

Use AAS to bring ACD agents back into auto-in work mode after a system restart. Although not restricted to such, this feature is intended to be used for splits/skills containing only nonhuman members - for example, recorders or voice response units (VRUs). To allow the IVR hardware to take back to back calls, you can assign a Timed-ACW delay of 1 to 2 seconds for the Auto Available Splits/Skills.

AAS detailed description

This section includes the following topics:

- Agent login with AAS on page 67
- Agent logout with AAS on page 68

Agent login with AAS

With AAS, ACD splits or skills generally operate as usual. The major difference is in how work modes are handled.

For splits/skills with AAS, agents are automatically logged in under the following circumstances:

- Call Management System (CMS) completes an Agent Move request into an Auto-Available split or skill.
- A maintenance-busied-out port, which is defined as an agent in an Auto-Available split or skill, is released.
- The system reinitializes and requires agents to log in again.
- You administer a split or skill on the Hunt Group screen as AAS = y.
- You administer an agent into an existing AAS split or skill.

Once an agent is logged into an Auto-Available split or skill, it is immediately moved to the auto-in work mode and subsequent requests to change out of that mode are denied.

Agent logout with AAS

For splits/skills with AAS, agents are automatically logged out under the following circumstances:

- CMS completes an Agent Move request out of an Auto-Available split or skill.
- The Auto-Available agent's port is unavailable because maintenance is being performed.
- You administer a split or skill as AAS = n.
- You remove an agent from an existing AAS split or skill.
- Redirection on No Answer (RONA) redirects a call that the agent has not answered after an administered number of rings.

AAS considerations

- AAS is intended primarily for non-BX.25 and non-ASAI PBX adjuncts such as an IVR system VIS, that require extra help in getting PBX ports back online after a restart. AUDIX is incompatible with AAS because it uses BX.25 messages to automatically activate its ACD agent ports after a PBX restart.
- Because AAS is intended for nonhuman agents, do not administer an Auto-Answer telephone as a member of an AAS.
- AAS is not intended for any agent port hardware that can change its work mode state since a request to move to any state other than auto-in is denied; however, administration of such telephones is not blocked.

AAS interactions

Auto-Answer

Do not administer an Auto-Answer telephone as a member of an AAS.

Auto-Answer was originally implemented for human agents. If a non analog telephone is administered as Auto-Answer and that telephone is logged into a split or skill, when the telephone goes on-hook, it is logged out.

Agents at analog telephones defined as Auto-Answer who are logged into a split or skill must dial a log-out FAC to log out. If a telephone is a member of an AAS, a log-out FAC is denied.

To log the agent out, you must either remove the agent from the split or skill when not active on a call or busy-out the physical extension.

If an agent in an AAS with an Auto-Answer telephone goes off-hook, the telephone is logged into any Auto-Available splits of which it is a member. To log out of the AAS splits/skills, the agent goes on-hook, is placed in AUX work mode, and then presses the RELEASE button on non analog sets or disconnects on analog sets. Because agents are not placed immediately in auto-in work mode, they may place personal or emergency calls rather than answering ACD calls that may be in queue.

CMS

For each agent, AAS notifies CMS of any login, logout, or change into the auto-in work mode. In a non-EAS environment, an AAS agent is identified to CMS with a login ID equivalent to the agent's administered extension. With EAS, the AAS login ID and port are assigned on the Login ID screen.

With CMS Move Agent, you can move a member from one AAS split or skill to another while that member is logged in.

Automatic Call Distribution

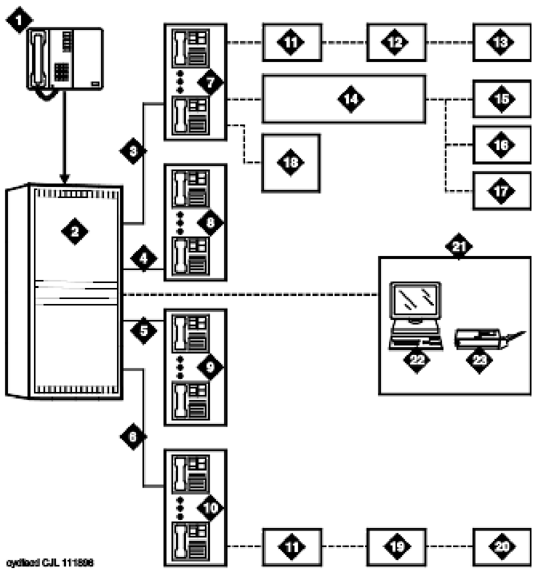
About ACD

Automatic Call Distribution (ACD) allows incoming calls to connect automatically to specific splits/skills. An ACD split or skill is simply a hunt group that is designed to receive a high volume of similar calls. ACD hunt groups with basic ACD, but non-EAS, are called splits. ACD hunt groups with basic ACD with EAS are called skills. Calls to a specific split or skill are automatically distributed among the agents, or hunt group members, assigned to that split or skill. Calls queue to the split or skill until an agent is available.

An ACD agent can be a physical telephone extension, an individual attendant extension, or, in an Expert Agent Selection (EAS) environment, an agent login ID. An agent can be logged into multiple splits/skills. However, in a non-EAS environment, agents can be logged into only one split if that split is administered for Multiple Call Handling (MCH).

You can assign a supervisor to each split or skill. The split or skill supervisor can listen in on agent calls, monitor the split or skill queue status, and assist agents on ACD calls. Although supervisors can assist agents on ACD calls, the supervisors do not normally receive ACD calls unless they are also members of the split or skill.

If you have Call Management System (CMS) or Basic Call Management System (BCMS), you can measure and create reports on the status of ACD agents, splits/skills, and trunks. See Agent Call Handling and Reporting adjuncts or Basic Call Management System before setting up your ACD splits. See Agent Call Handling for detail on administering agent functions and operations.



1. Incoming calls
2. ACD switch
3. Trunk group 1
4. Trunk group 2
5. Trunk group 3
6. Trunk group 4
7. Split 1 Business Travel (10 agents)
8. Split 2 Personal Travel (8 agents)
9. Split 3 Group Travel (5 agents)
10. Split 4 General Information (15 agents)
11. Queues
12. Announcement 1
13. Announcement 2
14. Intraflow (Call Coverage)
15. Split 2 Personal Travel (3rd choice)
16. Split 3 Group Travel (2nd choice)
17. Split 4 General Information (1st choice)

18. Supervisor (with Service Observing)
19. Announcement
20. Disconnect
21. Call Management System (CMS)
22. Terminal
23. Printer

ACD detailed description

Table of call distribution methods

The following table summarizes the different call distribution methods.

WHEN agents are available, a call arrives, and the agent selection method is:	THEN the communication server selects:	This distribution method is available with:
Direct Department Calling	the first available agent found in the hunt sequence.	Non-EAS
UCD-MIA	the most idle agent, without regard to skill level.	Non-EAS, EAS, Avaya Business Advocate
EAD-MIA	the highest skill level, most idle agent.	EAS, Avaya Business Advocate
UCD-LOA	the least occupied agent, without regard to skill level.	EAS, Avaya Business Advocate
EAD-LOA	the highest skill level agent with the lowest occupancy.	EAS, Avaya Business Advocate

Basic ACD (non-EAS without vectoring) queuing and announcements

Starting with Communication Manager 2.1, the system automatically allocates queue slots. The queue slot pool allows callers to wait for an agent to become available. The next available agent is automatically connected to the call in the queue.

For non-vector-controlled splits, calls do not queue in the following cases:

- No agents are logged in
- All logged-in agents are in Auxiliary Work mode

The caller gets a busy signal (or busy coverage, if administered) unless a call comes in using an automatic-in Central Office (CO) facility. In this case, the caller hears ringback from the CO and the system continues trying to place the call in the queue.

You can assign two announcements to each split and administer a second announcement to repeat. When an incoming call is directed to an ACD split, the call is either directed to an agent or is automatically connected to the first announcement. For information on how announcements are affected by call forwarding and call coverage, see *Avaya Aura™ Communication Manager Feature Description and Implementation*.

Related topic

For more information, see Dynamic queue slot allocation on page 18.

First announcement

After a call enters a queue, the caller hears ringing and the first announcement delay interval begins. If an agent becomes available during the first announcement delay interval, the call is connected to the agent. Otherwise, the interval expires and the system tries to connect the incoming call to the first announcement, with one of the following results:

- If the first announcement is available, the caller hears ringing, then the first announcement.
- If the announcement is busy and has no queue, the caller hears ringing and the first announcement delay interval is reset. The system tries to access the announcement again when the interval expires.
- If the announcement is busy and has a queue, then:
    - If the queue is full, the caller hears ringing and the first announcement delay interval is reset. The system tries to access the announcement again when the interval expires.
    - If the queue is not full, the call enters the announcement queue and the caller hears ringing, then the first announcement. The system then tries to connect the call to an agent.
- If the announcement is not busy, but is still unavailable, the second-announcement delay interval begins and the system attempts to connect the call to the second announcement.

If there is no first or second announcement, the call remains in queue until answered or removed from the queue.

Forced first announcement

The first-announcement delay interval defines how long a call remains in queue before the call is connected to the first announcement. If this interval is 0 seconds, the incoming call is automatically connected to the first announcement. This is a forced first announcement - the call is not routed to an agent until after the caller hears the first announcement.

With a forced first announcement, the following occurs:

- If a first announcement is available, the caller hears ringing and then the first announcement. The system then tries to connect the call to an agent.
- If the announcement is busy and has no queue, the system waits 10 seconds and then tries to access the announcement.
- If the announcement is busy and has a queue, then:
    - If the queue is full, the system waits 10 seconds, then tries to access the announcement.
    - If the queue is not full, the call enters the announcement queue and the caller hears ringing, then the first announcement. The system then tries to connect the call to an agent.
- If the announcement is not busy but is still unavailable (for example, it may have been deleted), then the system tries to connect the call to an agent.

After a forced first announcement, the caller always hears ringback (or music-on-hold, if administered) until the call is answered or is connected to a second delay announcement. After a first or second delay announcement, the caller hears music-on-hold, if administered.

Second announcement

After the first announcement, the second-announcement delay interval begins and the caller hears ringing (if there is no forced first announcement), or music, if provided. If an agent becomes available during the interval, the call is connected. Otherwise, the interval expires and the system tries to connect the incoming call to the second announcement, resulting in one of the following:

- If the second announcement is available, the caller hears ringing or music, then the second announcement.
- If the announcement is busy and has no queue, the caller hears ringing and the second-announcement delay interval is reset. The system tries to access the announcement again when the interval expires.
- If the announcement is busy and has a queue, then:
    - If the queue is full, the caller hears ringing (only if the first announcement has not been heard) and the second-announcement delay interval is reset. The system tries to access the announcement again when the interval expires.
    - If the queue is not full, the call enters the announcement queue and the caller hears ringing (only if the first announcement has not been heard), then the second announcement. The system then connects the call to an agent.
- If the announcement is not busy but is still unavailable, the call remains in queue until answered or removed from the queue.

After the second announcement, the caller hears music, if provided, or silence and then:

- If you administered the split or skill to repeat the second announcement, the system tries to connect the call to the second announcement after the delay expires.
- If you administered the split or skill not to repeat the second announcement, the call remains in the queue until answered or removed from the queue.

Forced disconnect

You can connect an incoming call directly to an announcement and then disconnect the call after the announcement has completed in one of two ways:

- Administer an announcement extension as the incoming destination. The caller is directed to the announcement and is disconnected, without being queued for a split.
- Administer an announcement extension as a point in a split coverage path. Calls that have been in the queue for a long time are forced to go directly to the announcement and are disconnected.

Announcement rules

The following rules govern announcements a caller hears:

- Calls that reach a split directly always hear a forced first announcement, if assigned, regardless of subsequent call coverage, call forwarding, night service, or busy signal processing. If these calls queue long enough, they hear first and second announcements.
- Calls that reach a split using call coverage receive a second announcement only, if administered. The assumption is that a caller has likely heard a first announcement at the original split or station before being redirected.
- Calls that reach a split using call forwarding receive first and second announcements at the destination split, if administered. These calls can receive a forced first announcement at the original split, if administered, but not at the split they are forwarded to.

Entering the queue

When a forced first announcement is not assigned, the system tries to connect an incoming call to an available agent. If an agent is available, the call is connected to the agent. If all agents are active (either on an ACD call or in ACW mode), the call enters the split or skill queue.

When you have administered Intraflow and Interflow with Call Coverage and Call Forwarding All Calls, the caller hears a busy tone or the call is redirected in any of these cases:

- No agents are logged in
- All logged-in agents are in AUX work mode, and the incoming facility is a digit-oriented facility (digits are sent to the communication server as in DID, incoming wink, or immediate tie trunks)

Note:
Central office trunk (non-DID) calls receive ringback from the CO, so the PBX cannot give these callers a busy signal. The system tries to put such calls into queue until successful or until the call is abandoned.

Priority queuing

Priority queuing allows priority calls to be queued ahead of calls with normal priority. You can implement priority queuing in two ways:

- Assign Priority Queuing to a calling party's Class of Restriction (COR).
- Assign Priority on Intraflow to an ACD split. This allows calls from the split, when intraflowed into another split, to be queued ahead of non priority calls. For more information, see Information Forwarding.

Queue status indications

You can assign queue status indications on agent or supervisor telephones or consoles for ACD calls in queue. For more information, see Queue Status Indications.

Configuring a call center or call center with EAS checklist

This section includes the following topics:

- Configuring the basics on page 76
- Defining the applications on page 76
- Defining trunks on page 76
- Defining hunt groups (skills) on page 77
- Defining agents on page 77
- Defining caller treatments and backup treatments on page 77
- Configuring and recording announcements on page 78
- Defining vectors on page 78

Configuring the basics

1. Confirm the configured options for the license file.
2. Set the system date and time.
3. Configure the dial plan.
4. Define the Feature Access Codes.
5. Define the Class of Service (COS).
6. Define the Class of Restrictions (CORs).
7. Configure Abbreviated Dialing.

Next steps

For detailed information about how to perform these steps, see *Avaya Aura™ Communication Manager Feature Description and Implementation*.

Defining the applications

1. Define caller types.
2. Define incoming called numbers.
3. Define the corresponding VDNs.
4. Define the skills needed to support call types.

Next steps

For detailed information about how to perform these steps, see *Planning an Avaya Aura™ Call Center Implementation*.

Defining trunks

For detailed information about how to perform this step, see *Avaya Aura™ Communication Manager Feature Description and Implementation*.

Defining hunt groups (skills)

Defining hunt groups (skills) includes all of the attributes. For detailed information about how to perform this step, see *Avaya Aura™ Communication Manager Feature Description and Implementation*.

Defining agents

To define agents:

1. Determine station types.
2. Define stations using the station screen.
   Also select auto-answer or manual-answer.
3. Define work modes.
   For example, auto-in, manual-in, or mixed environment.
4. Assign feature or work buttons.
5. Administer the Login ID forms.
   a. Define the skills.
   b. Define the attributes.
6. Determine what options (system, VDN, hunt group, or agent, and so on) are required for the call center, and what applications you want supported.

Result

For detailed information about how to perform these steps, see *Avaya Aura™ Communication Manager Feature Description and Implementation* and *Planning an Avaya Aura™ Call Center Implementation*.

Defining caller treatments and backup treatments

For detailed information about how to perform these steps, see *Planning an Avaya Aura™ Call Center Implementation* and *Administering Avaya Aura™ Call Center Features*.

Configuring and recording announcements

For detailed information, see Administering recorded announcements in *Administering Avaya Aura™ Call Center Features*.

Defining vectors

For detailed information about how to perform these steps, see *Programming Call Vectors in Avaya Aura™ Call Center*.

Call and agent selection

Call selection

Call selection methods are used when calls are in queue and an agent becomes available. This is known as a call surplus condition. During such conditions, the communication server considers the call selection method that is administered for the agent on the Agent LoginID Screen to determine which skill to serve. Once a skill is identified, the call at the head of that queue is selected and delivered to the agent. Call selection is based on such things as call handling preference, call selection measurement, and the use of service objectives.

Agent selection

Agent selection methods are used when there are one or more available agents for an incoming call. This is known as an agent surplus condition. Agent selection methods are administered as a hunt group type for the skill. Avaya Business Advocate allows you to select agents according to occupancy, idleness, individual skill level, and the percentage of time that you want the agent to spend serving each skill.

Automated agent staffing adjustments

Avaya Business Advocate provides you with options that automate staffing during call center operation. These methods simplify call center management and eliminate the need for moving agents from skill to skill to ensure coverage as call conditions change.

Avaya Business Advocate offers you the ability to assign reserve agents and set overload thresholds to determine when those reserve agents will be engaged. The Dynamic Advocate feature, known as Dynamic Threshold Adjustment, takes this a step further by automatically adjusting the thresholds as needed to help maintain the service levels you defined.

The Dynamic Percentage Adjustment feature, gives you the ability to automate adjustments to predefined allocations for your agents' time to maintain defined service levels. Auto Reserve Agents, another feature that is new with R9, allows you to intentionally leave an agent idle in a skill when the agent's adjusted work time has exceeded the percentage that you administered for that skill.

Call selection at a glance

The following table shows what happens during call surplus conditions, according to the call selection methods that have been administered on the communication server.

IF calls are waiting when an agent becomes available and the agent's selection method is:	THEN the communication server takes the highest priority call:
Skill Level without Service Objective	With the highest skill level and the longest CWT or PWT.
Skill Level with Service Objective	With the highest skill level and the highest ratio of CWT/SO or PWT/SO.
Greatest Need without Service Objective	With the longest CWT or PWT.
Greatest Need with Service Objective	With the highest ratio of CST/SO or PWT/SO.
Percent Allocation	That is the oldest call waiting that best maintains the administered target allocations for all skills.

Agent selection at a glance

The following table shows what happens during agent surplus conditions, according to the agent selection method that has been administered.

WHEN agents are available, a call arrives, and the agent selection method is:	THEN the communication server selects:
EAD-MIA	The highest skill level, most idle agent.
UCD-MIA	The most idle agent, without regard to skill level.
EAD-LOA	The highest skill level agent with the lowest occupancy.
UCD-LOA	The least occupied agent, without regard to skill level.

WHEN agents are available, a call arrives, and the agent selection method is:	THEN the communication server selects:
PAD	The agent with the lowest ratio of adjusted work time and target allocation for the skill.

Combining agent and call selection methods

Avaya Business Advocate provides a variety of features to help meet your business goals and to help you manage your agent resources. The table below shows some of the ways you can combine call and agent selection methods to meet your company's specific needs.

IF your goal is to:	THEN consider:
Maintain service levels while controlling the time agents spend serving each of their skills	• Percent Allocation • Dynamic Percentage Adjustment • PAD
Maintain service levels using more or less time from reserve resources to supplement staffing as needed	• Greatest Need • Service Level Supervisor • Dynamic Threshold Adjustment • UCD-LOA
Add customer segmentation with differentiated levels of service while routing all segments to the same skill to simplify staffing	• Greatest Need • Dynamic Queue Position • UCD-LOA
Increase revenue by assigning agents their best skills as primary skills and limiting the use of reserve skills to eliminate long call wait times	• Greatest Need • Service Objective • Service Level Supervisor • UCD-LOA
Ensure that critical skills are covered, regardless of caller wait time in other skills	• Greatest Need • Service Level Supervisor • Call Selection Override • Oldest Call Waiting • UCD-LOA
Control the time your agents spend serving their assigned skills while maintaining the	• Percent Allocation • Dynamic Percentage Adjustment • Call Selection Override

IF your goal is to:	THEN consider:
ability to change to meet service level requirements for the center	• Service Level Supervisor • PAD
Automate agent staffing to activate back up agents a little sooner or a little later to meet service level goals	• Greatest Need or Skill Level • Service Level Supervisor • Dynamic Threshold Adjustment • UCD-LOA or EAD-LOA
Minimize the complexity of differentiating service levels for different types of calls that require similar agent abilities	• Greatest Need or Skill Level • Dynamic Queue Position • UCD-LOA or EAD-LOA
Maximize the amount of time that agents spend in high contribution roles while limiting their use of lesser skills to address wait time problems	• Greatest Need • Service Objective • UCD-LOA
Spread calls more evenly among agents while delivering the right level of service to each skill	• Greatest Need • Service Objective • UCD-LOA
Use agents in their most proficient skills while minimizing the hot seat problem to some extent	• Skill Level • EAD-LOA

Different needs within a call center

You may find that one Avaya Business Advocate solution does not fit for your entire organization. Your call center may have different needs within particular areas or departments, and Avaya Business Advocate can help to meet these varying needs. A sales department, for example, may choose to use Dynamic Queue Position to create differentiation among various types of customer without creating a different skill for each type of sales call. A service department, on the other hand, may be more interested in working toward similar goals for each technical support skill, while eliminating the hot seats often experienced by the well trained, multi-skilled agents.

Avaya offers a subscription service for Avaya Business Advocate customers that provides access to skilled consultants with expertise in understanding how Avaya Business Advocate helps to solve business problems. For more information, please contact the Advisory Team at 877-977-0078 or by e-mail at advisoryhelp@avaya.com.

Feature compatibility

It is important to choose the right combination of features to meet your organization's needs and ensure that Avaya Business Advocate is set up to work most effectively. This section summarizes the features that provide the best results when used together and also lists those that are not designed to work together.

Call selection methods (call handling preferences)

The following table shows the features that work effectively with the various Avaya Business Advocate call selection methods.

Call selection method	Recommended to work with
Greatest Need	• Predicted Wait Time • Service Objective • Service Level Supervisor • UCD-MIA • UCD-LOA
Skill Level	• Predicted Wait Time • Service Objective • Service Level Supervisor • EAD-MIA • EAD-LOA
Percent Allocation	• Dynamic Percentage Adjustment • Auto Reserve Agents • Service Level Supervisor • PAD

Agent selection methods (hunt group types)

The following table shows which features work with the various agent selection methods.

Agent Selection Method	Recommended to work with
UCD-MIA	• Greatest Need • Predicted Wait Time

Agent Selection Method	Recommended to work with
	• Service Objective • Service Level Supervisor
EAD-MIA	• Skill Level • Predicted Wait Time • Service Objective • Service Level Supervisor
UCD-LOA	• Greatest Need • Predicted Wait Time • Service Objective • Service Level Supervisor
EAD-LOA	• Skill Level • Predicted Wait Time • Service Objective • Service Level Supervisor
PAD	• Percent Allocation • Dynamic Percentage Adjustment • Auto Reserve Agents • Service Level Supervisor

Feature combinations to avoid

The PAD agent selection method should not be used with Greatest Need or Skill Level call selection methods.

Avaya Business Advocate

This section provides an overview of Avaya Business Advocate. For extensive information on implementing and using Avaya Business Advocate, refer to the *Avaya Business Advocate User Guide*.

This section includes the following topics:

About Avaya Business Advocate

Avaya Business Advocate is a collection of features that provide flexibility in the way a call is selected for an agent in a call surplus situation and in the way that an agent is selected for a call in an agent surplus situation.

Note:

Avaya Business Advocate requires the Expert Agent Selection (EAS) feature to be enabled.

Avaya Business Advocate provides predictive and adaptive methods for call centers that address three fundamental questions in terms of how the most expensive resource of the center, its agents, are used every time a call is handled.

What should this agent do next?

Avaya Business Advocate decides what the agent should do after he or she becomes available and calls are waiting in queue. With Avaya Business Advocate, this decision does not come from executing a set of pre-programmed directives such as, the highest priority or oldest-waiting call. Such a fixed plan of attack considers nothing in terms of consequences. Instead, Avaya Business Advocate understands the consequences of its choices and the business objectives for each type of call.

Which agent should take this call?

Avaya Business Advocate decides which agent should take a call when there is more than one agent waiting for the call. Avaya Business Advocate can make this choice so that workloads are distributed fairly across the agents to eliminate hot seats. Avaya Business Advocate can also promote fairer opportunities for compensation by delivering a predetermined mix of calls to agents.

Does the call center need to adjust its operations?

Avaya Business Advocate continuously evaluates the performance of the call center and makes adjustments accordingly. Avaya Business Advocate can prevent callers from waiting too long, and makes sure that the call center consistently meets service-level goals.

Basic Call Management System

About Basic Call Management System

Basic Call Management System (BCMS) provides real-time and historical reports to assist you in managing agents, ACD splits/skills (hunt groups), VDNs, and trunk groups. You can display BCMS reports on a terminal or print a paper copy.

BCMS provides the following reports:

- Real Time Reports
  - Split/Skill Status
  - System Status
  - VDN Status
- Historical Reports
  - Agent
  - Agent Summary
  - Split/Skill
  - Split/Skill Summary
  - Trunk Group
  - Trunk Group Summary
  - VDN
  - VDN Summary

For a detailed description of BCMS and the reports it provides, see Communication Manager Call Center Software - Basic Call Management System (BCMS) Operations.

BCMS interactions

Call redirection and conference calls

For information about how BCMS records redirects and conferences calls, see *Communication Manager Call Center Software - Basic Call Management System (BCMS) Operations*.

Move Agents From CMS:

If agents are moved from one split or skill to another split or skill using CMS/Supervisor, measurements are stopped for the agent's from split or skill and started for the agent's to split or skill.

If an attempt is made to move an agent from a non-BCMS-measured split or skill to a measured BCMS split or skill using CMS/Supervisor, and the move would exceed the maximum number of measured agents, the communication server rejects the move. Otherwise, internal BCMS measurements are started for the agent. If the an agent is moved from a split or skill that is measured by BCMS to a split or skill that is not measured by BCMS using CMS/Supervisor, then internal measurements for the agent stop.

Night Service

When night service is activated for a split or skill, new calls go to the alternate destination. BCMS does not record these calls as OUTFLOW. If the destination is a measured split or skill, BCMS treats the calls as new incoming calls (that is, BCMS does not record them as INFLOW).

System Measurements

The system can simultaneously produce BCMS reports, adjunct CMS reports, and communication server traffic measurements.

Although some of the CMS and BCMS report information is similar, BCMS measurements are not determined in the same way as trunk group and hunt group measurements are reported in CMS. Therefore, representation of data in the two report types is not identical.

Best Service Routing (BSR)

About BSR

The Best Service Routing (BSR) feature compares specified splits/skills and selects the one that provides the best service to a call. To respond to changing conditions and operate more efficiently, BSR monitors the status of the specified resources and adjusts call processing appropriately.

BSR can be configured for either single-site or multi-site operation. Single-site BSR compares splits/skills on the Communication Manager where the BSR resides to find the best resource to service a call. Multi-site BSR extends this capability across a network of Communication Managers, comparing local splits/skills, remote splits/skills, or both, and routing calls to the resource that provides the best service.

Benefits of BSR

Both single-site and multi-site BSR intelligently compare specific resources to find the one that can best service a call. In addition, multi-site BSR makes it possible for you to integrate a network of call centers for better load balancing and optimal agent utilization. Depending on your specific application, BSR can yield a variety of other benefits as shown in the following table.

Note:

If a call center network is heavily overloaded and a significant number of calls are being blocked or abandoned, shorter wait times may not result when BSR is used. Rather than reducing wait times, any productivity gains will allow more calls to gain access to the network.

Table 1: Best Service Routing benefits

You can benefit from…	As a result of…
Increased revenue	• Better agent utilization, thus allowing more calls to be handled with a given staff level. • Lower abandonment rates - By balancing the load between resources, BSR reduces extremes in wait times across local resources or across an entire network. • In call centers with Expert Agent Selection, the ability to deliver calls to the best qualified or highest revenue generating agents.
Lower costs	• Better agent utilization. • Shorter trunk holding times. • Reductions of ineffective interflows. • Operation over ISDN-BRI trunks and public networks.
Improved customer satisfaction	• Interflowing calls from centers with a surplus of calls to centers with a surplus of agents. You can achieve uniform service levels across your network. This means that all callers for a given application experience approximately equivalent waiting times. • Shorter wait times. • In call centers with Expert Agent Selection, the ability to deliver calls to the best qualified or highest revenue generating agents. • Robust information forwarding capabilities. Multi-site BSR can forward original service requirements and any caller-entered digits with each call, and can use both QSIG and non-QSIG information transport methods over private or public networks.

You can benefit from…	As a result of…
Increased performance and more efficient trunk usage	• Less messaging and processing required per call than in traditional LAI scenarios. • Eliminates phantom calls to remote agents. • Intelligent interflows that only route calls to centers with available agents.
BSR's easy configuration	Simple vector commands. You do not need to learn complex programming languages or design comparison steps. All that you have to do is list the local and remote resources to be considered for calls and instruct the Communication Manager to queue or deliver the call to the best resource on the list.
Improved agent productivity	• Increased efficiency. Improve your service without adding staff, or reduce staff while maintaining your current level of service. Network-wide load balancing means that agents at one location are less likely to sit idle while calls wait in queue at another location. • No call delivery delays. In contrast to approaches that queue calls at all remote centers simultaneously, with BSR there is no delay in delivering a call when an agent becomes available.
Increased operating flexibility, easier staffing and scheduling	• Larger pool of agents available to take calls in a split/skill. Through its network-wide call distribution and information forwarding, BSR effectively converts distributed locations into a virtual call center. Thus, staffing problems do not need to be solved on a center-by-center basis. BSR can automatically react to staff shortages at one center by routing more calls to other locations. • Automatic management of sudden and unexpected increases in call volume. Large increases in call volume for a single split/skill can be distributed across other splits/skills. Spikes in call volume at a single call center can be distributed across all call centers, provided that sufficient trunk capacity is available between servers.
Improved service levels	Lower average speed of answer (ASA).

Server and network requirements for BSR

For single-site BSR applications, your Avaya Communication Manager must meet the requirements that are shown below. The requirements for ISDN trunks and LAI do not apply to single-site BSR applications.

To use multi-site BSR applications, all servers involved and the network connecting them must meet all of the requirements that are described in this section.

Caution:

To ensure that your network meets the requirements for BSR support, contact your Account Executive about BSR network certification.

This section includes the following topics:

- Server requirements on page 89
- Network requirements

Server requirements

Your Avaya Communication Manager must meet the requirements shown in the following table to support BSR.

Table 2: Requirements to use Best Service Routing

Screen	Page	Field	Must be set to...
System-Parameters Customer-Options	2	**ISDN-BRI Trunks**[4]	`Y`
		ISDN-PRI Trunks[45]	`Y`
	3	**Vectoring (G3V4 Advanced Routing)**	`Y`
		Vectoring (Best Service Routing)	`Y`
		Lookahead Interflow (LAI)[6]	`Y`
Feature-Related System Parameters	8	**Adjunct CMS Release**	`R3V6` or higher, or left blank

Tip:

If you begin using BSR and then turn it off, you can not set **Vectoring (Best Service Routing)** to `n` until you remove all BSR commands from vectors. If you are using multi-site BSR with Look-Ahead Interflow and want to turn LAI off, you can not set **Lookahead Interflow (LAI)** to `n` until you remove all **`consider location`**, **`reply-best`**, and **`interflow-qpos`** commands from vectors.

4 Multi-site BSR operates over both BRI and PRI trunks. ISDN connectivity is only necessary if you want to use multi-site BSR, in which case one or both of these fields must be set to `Y`.

5 ATM trunking and IP trunking can be set up to emulate ISDN PRI. For information on setting this up, see the Administering Network Connectivity on Avaya Aura™ Communication Manager, and ATM Installation, Upgrades and Administration using Communication Manager.

6 Look-Ahead Interflow is only necessary if you want to use multi-site BSR.

Network requirements

To support multi-site BSR, networks must meet both the criteria for LAI call control operation over switched networks (see Look-Ahead Interflow (LAI)) and the following criteria:

- The network must support end-to-end transport of codeset 0 user data, either as a User-to-User Information Element (UUI IE) or by QSIG Manufacturer Specific Information (MSI IE), in the ISDN SETUP and DISCONNECT messages. For more information, see Determining user information needs.
- With BSR poll calls, the information is forwarded back in the DISCONNECT message. In this case, the network must support forwarding of UUI in the first call clearing message, while the call is still in the call proceeding state, prior to the active state.
- Private networks can be configured for either QSIG (using MSI packaged in codeset 0 Facility IEs) or non-QSIG (using a codeset 0 UUI IE) transport. Currently, public networks do not support QSIG and user data can only be transported by the UUI IE when supported by the network. Future public network offerings may support QSIG, possibly by Virtual Private Network.
- The switch must support the ISDN country protocol.
- The network byte limit for the user data portion of the user information contents must be large enough to carry the data needed for the customer application.

Note:
Some public network providers may require service activation, fees for user information transport, or both.

BSR, LAI, enhanced information forwarding, and UCID have been tested with several major carriers. To find out if these capabilities work with your carrier, check with your account team for the most current information.

If testing has not been done to verify operation over the public networks that are involved with the preferred specific configuration, use of private ISDN trunking between the nodes should be assumed until successful testing is complete.

Special BSR terminology

Understanding the BSR terms listed below will be helpful as you read through the material in this section. The following table contains terms pertaining to both single-site BSR and multi-site BSR.

Term	Description
adjusted EWT	Expected Wait Time plus a user adjustment set by a **`consider`** command.

Term	Description
agent selection method	The method that the Communication Manager uses to select an agent in a hunt group when more than one agent is available to receive the next call. Possible methods are: • UCD-MIA • UCD-LOA • EAD-MIA • EAD-LOA The agent selection method is a property of hunt groups and is set in the **Group-Type** field on the Hunt Group screen. **Note:** To use an EAD available agent strategy, Expert Agent Selection (EAS) must be enabled.
Alternate Selection on BSR Ties	The BSR feature compares splits or skills using a series of consider steps and selects the one that provides the best service to a call. When that comparison results in a tie (the results are equal in value), the Alternate Selection on BSR Ties determines how BSR chooses which agent, skill, or location to select. For more information about this feature, see Alternate Selection on BSR Ties under Best Service Routing.
application	A general term for a system in any call center that handles calls of a particular type. In relation to BSR, any specific implementation of multi-site BSR.
application plan	Used only in multi-site applications, the application plan identifies the remote switches that may be compared in consider series. The plan also specifies the information that is used to contact each switch and to interflow calls to it.
best	Includes the following conditions • No agents available - When no agents are available in any of the specified splits/skills, the best resource is the one with the lowest adjusted EWT. • Agent available in one resource - When an agent is available in one and only one of the splits/skills that are specified in a consider series, that agent is the best and the call is delivered to that agent. If the **BSR Available Agent Strategy** field is set to `1st-found`, BSR ignores all subsequent steps in the consider series. If any other available agent strategy is used, all remaining resources are still considered before the call is delivered. • Agents available in two or more resources - When agents are available in two or more splits/skills, the best agent is the one that best meets the criteria that are specified in the **BSR Available Agent Strategy** field. For example, if the available agent strategy is UCD-MIA, the best agent out of those available will be the agent with the longest idle time.

Term	Description
Best Service Routing (BSR)	A feature that is based on call vectoring and routes ACD calls to the resource that is best able to service each call. BSR can be used on a single switch, or it can be used to integrate resources across a network of switches.
BSR available agent strategy	A field that appears on the VDN screen when either version of BSR is enabled. The entry in this field is a property of the VDN and its assigned vector. Possible entries are: • `1st-found` • `UCD-MIA` • `UCD-LOA` • `EAD-MIA` • `EAD-LOA` When the VDN is the active VDN for a call, as determined by VDN Override, this field determines how BSR commands in the vector identify the best split/skill when several have available agents.
consider series	consider commands are typically written in a set of two or more. This set of consider commands is called a consider series. A consider series in a status poll vector might have just one consider step.
consider sequence	A consider sequence is a consider series plus a queue-to best, check-best, or reply-best step.
Expected Wait Time (EWT)	Expected Wait Time is an estimate of how long a call in the queue will have to wait before it is connected to an agent.
Intelligent polling	An automatic feature of BSR that significantly reduces the number of status polls that are executed. When a remote location cannot be the best resource at a given moment in time, the intelligent polling feature temporarily suppresses polls to that location.
interflow	The process of routing an incoming call to an external switch without answering it at the origin switch.
poll suppression	A component of BSR intelligent polling that eliminates wasteful polling of remote locations which have returned poor adjusted EWTs.
resources	An agent, split, skill, or location
status poll	A call that is placed by a **`consider location`** vector command to obtain status data from a remote location in a multi-site BSR application.

Single-site BSR

About single-site BSR

Single-site BSR is a simple, logical extension of call vectoring. Like any other vector, vectors with BSR commands are assigned to one or more VDNs. Using new vector commands and command elements, you tell the Communication Manager to compare, or consider, specific splits/skills for each call that is processed in that particular vector. Throughout the comparison, the server can remember which resource is the best based on how you define best. BSR vectors can deliver a call to the first available agent found, or they can consider all of the specified resources and deliver the call to the best split/skill. If no agents are available in any split/skill, the call is queued to the split/skill with the shortest adjusted EWT.

Command set - single site BSR

The following table shows the screens, the vectors, and the vector commands and command elements that are used in single-site BSR. The following table shows the vector commands and command elements used in single-site BSR applications.

Table 3: Vector commands and usage for single-site BSR

Commands and command elements		Use this…
Screens	Vector Directory Number screen	To link a VDN to a BSR vector. To set the agent selection strategy that will be used for all calls to that VDN.
	Call Vector screen	To confirm that BSR is administered. To write vectors that use BSR commands.
Commands	**`consider split/ skill`**	To obtain the Expected Wait Time or agent data that is needed to identify the best local resource. One consider step must be written for each split/skill that you want to check. Since the **`consider`** command is designed to compare two or more resources, consider commands are typically written in a series of two or more with the sequence terminating in a queue-to best vector step. This set of consider commands and a queue-to best step is called a consider sequence.
	`queue-to`	With the best keyword to queue calls to the best resource that is identified by the consider sequence.

Commands and command elements		Use this…
	check	With the best keyword to queue calls to the best resource that is identified by the consider sequence if the resource meets certain conditions.
Key word	**best**	Use the best keyword in queue-to, check, and goto commands that refer to the resource that is identified as best by a series of consider steps
Conditional	**wait-improved**	To prevent calls from being queued to an additional split/skill when the reduction in Expected Wait Time is not enough to be useful. Wait improved means that a call's EWT must be improved by a specific amount, specified in seconds, over its current EWT or the Communication Manager does not queue the call to the additional split/skill.
User adjustment	**adjust-by**	To specify your preferences for the splits/skills that might handle the calls for a particular application, reflecting factors such as agent expertise or reducing calls to a backup split/skill. When a vector considers a local resource you can make the selection of that split/skill less desirable. The higher the setting, the less chance that resource will be selected over another with a lower setting (for example, set to 30 makes that choice 30% less desirable). With EWT returned, the setting increases the returned expected wait time for comparison with other returned EWTs. As a result, this split/skill is less likely to service the call unless its EWT is significantly less than that of any other available split/skill. Optionally, the adjust-by setting applies in the available agent case. If you are using the UCD-MIA or EAD-MIA available agent strategy, the setting decreases the returned agent idle time, making the agent appear less idle (busier). If you are using the UCD-LOA or EAD-LOA available agent strategy, the setting increases the returned agent occupancy, making the agent appear busier. In either case with EAD, the MIA or the LOA is used as a tie breaker if more than one site has an agent available with the same highest skill level.

How BSR determines the best resource

BSR determines the best resource to service a call by examining one or all of the following variables:

- The EWT of the resource
- Any user adjustments

- The availability of agents
- The selection strategy for the active VDN

Note:
The BSR available agent strategy that applies to a given call is the strategy that is assigned to the active VDN for that call, as determined by VDN override.

This section includes the following topics:

- Call surplus situations on page 95
- Agent Surplus situations on page 95
- Agent selection adjustments on page 96

Call surplus situations

Every BSR application compares a set of predetermined resources (splits/skills) and selects the best resource to service the arriving ACD call.

In a call surplus situation when no agents are available, the best resource is always the skill with the lowest Expected Wait Time (EWT). For purposes of calculating the best resource in a call surplus situation, BSR allows you to adjust the EWT figure for any split/skill. The actual EWT for calls in queue is not changed. Only the figure used in the calculations performed by the BSR feature is changed. You do not have to enter adjustments, but the ability to adjust the EWT for splits/skills allows you to program preferences in vectors. Because of agent expertise, for example, or the availability or cost of tie trunks, you might prefer that some resources do not service a call unless doing so significantly decreases the time in queue for the call.

It is possible for you to make adjustments to agent availability using the `consider` step. For more information, see Agent selection adjustments on page 96.

Agent Surplus situations

In an agent surplus situation when one or more agents are available to take incoming calls, BSR delivers a new call according to the**BSR Available Agent Strategy** field that is specified on the VDN screen. The best resource is the split/skill that meets one of the five (5) criteria that you have defined by the strategy as the VDN BSR Available Agent Strategy. BSR can use any of the five strategies shown in the following table to select an agent when agents are available.

Table 4: BSR available agent strategies

If BSR Available Agent Strategy is set to…	The call will be delivered to…
1st-found	The first available agent. BSR will not consider any other resources as soon as it finds an available (idle) agent.
ucd-mia	The resource with a staffed agent who has been idle (available) for the longest amount of time. BSR compares all of the splits/skills

If BSR Available Agent Strategy is set to…	The call will be delivered to…
	that are specified in the vector before delivering the call.
ead-mia	The resource with a staffed agent who has the highest (most primary) skill level that is relevant to the call and who has been idle the longest. BSR compares all of the splits/skills that are specified in the vector before delivering the call.
ucd-loa	The resource with the least-occupied agent. BSR compares all of the splits/skills that are specified in the vector before delivering the call.
ead-loa	The resource with an agent who has the highest (most primary) skill level that is relevant to the call and who is the least occupied. BSR compares all of the splits/skills that are specified in the vector before delivering the call.

For more information on LOA, see UCD-LOA on page 28, or *Avaya Business Advocate User Guide*. LOA is available with the Contact Center Elite package.

When agents are available in one or more of the specified resources, BSR does not consider resources (local or remote) that return an Expected Wait Time value (as would be the case in a call queue/call surplus situation) in selecting the best place to send the call.

Note:

With the exception of "first-found" the **BSR Available Agent Strategy** assigned to a VDN should match the agent selection method that is used in the splits/skills considered by a BSR application.

Agent selection adjustments

An option has been provided to have the BSR adjust-by value apply in the agent surplus (agents available) situation. This adjustment provides the ability to use the **`consider`** step adjustment value to prioritize (handicap) agent resources when agents are available.

When the adjustment is used, the **`consider`** step uses the following syntax:

```
consider split/location adjust-by x
```

The server applies the agent adjustment in the same manner as the calls in queue/call surplus (lowest EWT) situation.

To select an adjustment, think in terms of reducing the importance of a resource/site and in relative percentage — the higher the adjustment, the less desirable it is to pick that agent/site. So, if **`x`** = 30, then the agent/site is 30% less desirable.

The available agent adjustment applies to the UCD-MIA, UCD-LOA, EAD-MIA, and EAD-LOA call distribution methods. For the most idle agent distribution methods, the adjust-by lowers the idle time value returned by the agent/site. For the least occupied agent distribution methods, the adjust-by raises the returned occupancy level of the agent/site. In either case, with EAD, the MIA or LOA is used as a tie breaker if more than one site has an agent available with the same highest skill level.

The same adjust-by value in the **`consider`** step applies to both agent surplus and call surplus situations.

Example of basic single-site BSR

This example shows the simplest use of BSR. The central element of all single-site and multi-site BSR is a VDN/vector pair. The vector contains the commands that actually process the call, but the active VDN for the call contains information that is used by some vector steps. For single-site BSR, the active VDN for a call sets the available agent strategy that is used by the vector.

Single-site BSR example VDN screen

```
change vdn xxxxx                                                  page 1 of 3
        VECTOR DIRECTORY NUMBER

                            Extension: 5000
                                 Name: Single-site BSR
                        Vector Number: 234
                  Attendant Vectoring? n
                   Meet-me Conference? n
                   Allow VDN Override? n
                                  COR: 59
                                   TN: 1
                             Measured: internal
       Acceptable Service Level (sec): 20
              Service Objective (sec):
        VDN of Origin Annc. Extension:
                            1st Skill:
                            2nd Skill:
                            3rd Skill:
```

```
change vdn xxxxx                                                 page 2 of 3
        VECTOR DIRECTORY NUMBER

                           Audix Name:

                   Return Destination:
               VDN Timed ACW Interval:
                      BSR Application:31
         BSR Available Agent Strategy: 1st-found
```

In the example Vector Directory Number screen shown above, the **BSR Available Agent Strategy** field is set to `1st-found`. If vector 234 uses BSR commands, as soon as a consider step locates a resource with an available agent any subsequent consider steps are skipped and the call is delivered to that resource. Resources that are specified in any subsequent consider commands are not checked. If no split has an available agent, the call is queued to the split with the lowest adjusted EWT.

If the **Allow VDN Override?** is set to `n` and a second VDN and vector are used to process this call, the 1st-found strategy specified in VDN 5000 will still be used.

In the preceding example, Vector Directory Number 5000 is associated with vector 234, which is shown below. In this example, vector 234 compares two splits. No adjustment is assigned to either resource, indicating that both splits are equally suited to service calls since neither is preferred to the other. In reality, such a vector would probably have additional steps after step 4, such as **`announcement`** or **`wait-time`** commands. These steps are omitted in this example for purposes of clarity.

Single-site BSR example vector

```
1. wait time 0 secs hearing ringback
2. consider split 1 pri l adjust-by 0
3. consider split 2 pri l adjust-by 0
4. queue-to best
```

Notice that the **`consider`** commands follow each other in unbroken sequence and that the **`queue-to best`** command immediately follows the last **`consider`** command. This structure is called a *consider series*, and it is recommended that you typically write such series in uninterrupted order. A few commands, such as the **`goto`** command, which cause little if any delay in the execution of the **`consider`** steps, may be used. In general, however, do not put other commands between **`consider`** steps, or between a **`consider`** step and a **`queue-to best`** step. Even if BSR still works in that situation, you might seriously impair the performance of the vector.

Consider commands collect and compare information. When a call is processed in the vector above, the first consider step collects and temporarily saves the following information about split 1:

- The fact that split 1 is a local split
- The queue priority that is specified in the consider step
- The user adjustment that is specified in the consider step
- The split's
    - Split number
    - Expected Wait Time

If EWT=0, which indicates that one or more agents are available, the step also collects all of the agent information that might be needed by the BSR available agent strategy. This includes:

- Agent Idle Time (AIT)
- Agent Occupancy (AOC)
- The skill level of the agent in the split/skill who will receive the next call

In the example shown above, neither split has an available agent when the consider series executes. If one did, the call would be delivered to that split by the queue-to best step. Since there are no available agents in either split, the complete set of saved data now defines the best resource—for the moment. The second consider step collects the same data and compares it to the current best data. For this example, assume that the EWT for split 1 is 40 seconds and the EWT for split 2 is 20 seconds. When the second consider step executes, its data will replace the best data from step 1 because its adjusted EWT is lower. The best data is essentially a placeholder. When a queue-to best step executes, it reads the data that is saved as the best at that moment and queues the call to that split. In this case, the best data was collected from split 2, so the call is queued to split 2 at the specified priority.

What if there are available agents in both splits?

Since the "BSR Available Agent Strategy" in this example is 1st-found, the consider series will skip any consider steps after step 2 and the queue-to best step will deliver the call to split 1, which is the first split/skill with an available agent that is found by the vector.

In any BSR vector, the order of the consider steps should reflect your preferences for the resources to be considered. Put the step that considers the most preferred split/skill first, the step for your second preference second, and so forth in the consider series.

What if there are several available agents in split 1? Which agent receives the call?

When more than one agent is available in a split, the BSR `consider` command collects agent data only for the agent who will receive the next call to that split. This agent is identified according to the agent selection method that is specified in the **Group-Type** field on the Hunt Group screen.

Note:

For greatest efficiency, the agent selection method used in the splits/skills considered by a BSR vector should match the BSR Available Agent Strategy that is assigned to the active VDN.

User adjustments in single-site BSR

You may have preferences as to which splits/skills should answer certain types of calls. In both single-site BSR and multi-site BSR, the "adjust-by" portion of the `consider` command makes it possible for you to program these preferences into your vectors.

You can assign a value of 0 to 100 in user adjustments. The units of this value are supplied by the server depending on the conditions whenever that `consider` step executes. For example, in the command `consider split 1 pri h adjust-by 20`, the server interprets "adjust-by 20" to mean *add 20% to the EWT, but add at least 20 seconds*.

Note:

If the user adjustment were defined as a number of seconds, BSR would not be efficient when EWT was high. If the user adjustment were defined as a percentage, BSR would not be efficient when EWT was low. Such efficiencies, while always important, become critical in multi-site BSR applications where issues of trunk cost and capacity are involved.

For Expected Wait Times of 1 to 100 seconds, an adjustment of 20 will therefore add 20 seconds. Above 100 seconds, the same adjustment will add 20% to the EWT for the split/

skill that is specified in the **consider** step. The following table shows the results of applying a constant adjustment to a range of Expected Wait Times.

Table 5: User adjustments in BSR

EWT of resource (seconds)	User adjustment	Adjustment applied by the server (seconds)	Adjusted EWT used to select resource
10	20	20	30
60		20	80
120		24	144
300		60	360

Example of single-site BSR with adjustments

The following example shows a more complex implementation of single-site BSR. Four skills in an Expert Agent Selection environment are compared. The Expected Wait Time (EWT) for some skills is adjusted to reflect the administrator's preferences

Single-site BSR example VDN screen

```
change vdn xxxxx                                                  page 1 of 3
        VECTOR DIRECTORY NUMBER

                            Extension: 5001
                                 Name: Single-site BSR
                        Vector Number: 11
                  Attendant Vectoring? n
                   Meet-me Conference? n
                   Allow VDN Override? n
                                  COR: 59
                                   TN: 1
                             Measured: internal
       Acceptable Service Level (sec): 20
              Service Objective (sec):
        VDN of Origin Annc. Extension: 501
                            1st Skill:
                            2nd Skill:
                            3rd Skill:
```

```
change vdn xxxxx                                           page 2 of 3       VECTOR
DIRECTORY NUMBER
                           Audix Name:
                   Return Destination:
               VDN Timed ACW Interval:
                      BSR Application:19
         BSR Available Agent Strategy: EAD-MIA
```

In the example shown above, the **BSR Available Agent Strategy** field is set to `EAD-MIA`. If vector 11 uses BSR commands, calls are not automatically delivered to the first resource with an available agent that is found. All consider steps in vector 11 are executed and one of the following things happens:

If …	Then…
No skill has an available agent	The call queues to the skill with the lowest adjusted EWT.
Only one skill has an available agent	The call is delivered to that skill.
Two or more skills have available agents	The call is delivered to the skill with the most expert agent.
Two or more skills have available agents with the same skill level	The call is delivered to whichever of these agents has been idle the longest.

Also note that **Allow VDN Override?** is set to `n`. If a second VDN and vector are used to process this call, the EAD-MIA strategy that is specified in VDN 5001 is used. If **Allow VDN Override?** is set to `y` and vector 11 routes some calls to another VDN, the subsequent VDN's available agent strategy governs the operation of consider steps in its vector.

The following example vector 11, which compares four skills.

Single-site BSR example vector

```
1. wait-time  0 secs hearing ringback
2. consider skill  1 pri l adjust-by 0
3. consider skill  2 pri l adjust-by 30
4. consider skill 11 pri l adjust-by 30
5. consider skill 12 pri l adjust-by 30
6. queue-to best
7. wait-time 10 secs hearing ringback
8. announcement 1001
9. wait-time 30 secs hearing music
10. goto step 8 unconditionally
```

For this example, assume that the Expected Wait Times of the four skills are 95, 60, 180, and 50 seconds, respectively. Notice that all **`consider`** steps except the first adjust the EWT returned by the specified skill. Skill 1 is the preferred skill to handle calls to VDN 5001, so its EWT is not adjusted. Skills 2, 11, and 12 can handle this call type, but they are not preferred. The adjustment of 30 means that, in call surplus situations, these skills will not handle calls to VDN 5001 unless their EWT is at least 30 seconds better than the EWT in skill 1.

The following table shows the adjustments that would be applied to each skill given its EWT and the user adjustment specified in the **`consider`** step. The last column shows the adjusted EWT the server will use to select a skill for the call.

Table 6: User Adjustments

Skill number	User adjustment in the consider step	Actual EWT (seconds)	Adjustment applied by the server (seconds)	Adjusted EWT used in BSR calculations (seconds)
1	0	95	0	95

Skill number	User adjustment in the consider step	Actual EWT (seconds)	Adjustment applied by the server (seconds)	Adjusted EWT used in BSR calculations (seconds)
2	30	60	30	90
11	30	180	54	234
12	30	50	30	80

Since the available agent strategy is not 1st-found, all four `consider` steps are executed each time that the vector processes a call. In this example, there are no available agents in any of the skills. In fact, EWT is high enough in the first three skills for the server to queue the call to skill 12.

When the queue-to-best step executes, the data in the best data placeholder is the data from skill 12 and so the call is queued to that skill. From this point on, if the call is not answered during the execution of step 7, a common vector loop regularly repeats an announcement for the caller while he or she waits in the queue.

User adjustments also apply to available agent situations (with a strategy other than first found) in a manner that is similar to EWT.

What if there is an available agent in one skill? Will user adjustments be applied?

Since the “BSR Available Agent Strategy” in this example is EAD-MIA, the entire consider series will always be executed to check all of the skills for available agents. If only one skill has available agents, the call is delivered to that skill and user adjustments are not applied.

What if there are available agents in two skills? Which skill gets the call? Will user adjustments be applied?

Since the BSR Available Agent Strategy for VDN 5001 (the active VDN) is EAD-MIA, the call is delivered to the skill with the most expert agent. If there are available agents in both skills with the same skill level, their user adjusted idle times are compared and the call goes to the skill with the agent who has the longest adjusted idle time.

If a split/skill has more than one available agent, remember that it is the split/skill’s agent selection method that determines which agent’s data is used in BSR selection of the best resource.

What if no agents are staffed in a skill? Will the server recognize this?

Yes. Under any of the following conditions, the EWT returned from a split/skill is infinite:

- No agents logged in
- No queue slots available
- All agents in AUX work mode

The server logs a vector event and goes to the next vector step without changing the data in the best placeholder. A resource with an infinite EWT is never selected as the best resource.

Can VDN skills be used in consider steps?

Yes. For example, consider skill 1st [2nd, 3rd] pri m adjust-by 0 will collect data on the 1st [2nd, 3rd] skill, as defined for the active VDN.

Troubleshooting for single-site BSR

You should regularly execute a **`display events`** command for the appropriate vectors, especially if you have just implemented a new BSR application. Vector events will identify and indicate the source of common malfunctions and administration errors.

For a list of BSR vector events and definitions, see Troubleshooting vectors.

Note:

Only the most recent events are displayed when a **`display events`** command is executed. For this reason, you should periodically display vector events to help quickly identify problems.

To verify that your BSR vectors are operating as intended, use a **`list trace vdn`** or **`list trace vec`** command to observe processing of an individual call. For more information, see Clearing events in the *Programming Call Vectors in Avaya Aura™ Call Center* document.

Note:

The **`list trace vdn`** and **`list trace vec`** commands are blocked if the Tenant Partitioning feature is enabled.

Multi-site BSR

Multi-site BSR extends all of the capabilities of single-site BSR across a network of Communication Managers. Multi-site BSR compares local splits/skills and remote splits/skills, and route calls to the resource that provides the best service. Multi-site BSR has special features that work to ensure efficient use of processor power and network resources in your BSR applications.

This section includes the following topics:

- Example for handling excessive wait times on page 121
- Selecting or administering application plans on page 122

- Administering the BSR Application Plan

Multi-site BSR command set

The following table shows the screens, the vectors, and special vector commands and command elements that you use to administer multi-site BSR applications. The table also briefly describes the purpose of each component.

Table 7: Vector commands and usage for multi-site BSR

Screens	
Best Service Routing Application Plan screen	• To define the group of remote sites that will be polled by a specific application. • To assign a unique name and number to each application. • To assign routing numbers for the status poll and interflow VDNs.
Vector Directory Number screen	• To link a VDN to a BSR application by its application number. • To link the VDN to a BSR vector. • To set the agent selection strategy that will be used for all calls to that VDN.
Call Vector screen	To confirm that BSR is administered and to program the vector steps for BSR.
ISDN Trunk screens	To tell the Communication Manager whether to forward user information by Shared UUI or QSIG MSI.
List Best Service Routing Applications screen	To display a list of all the BSR applications by name and number.
System Capacity	To monitor the number of BSR application-location pairs that are assigned in your system.
VDNs and Vectors	
Primary VDN (the active VDN for the call at the origin, as determined by VDN override)	To define the application plan and available agent strategy that are used by the vector that is assigned to this VDN.
Primary vector	To control call processing at the original server and compare local and remote resources.

Status poll VDN/ vector	To respond to status poll calls from another server. The status poll vector considers a set of local splits/skills and returns data on the best resource to the original server.
Interflow VDN/ vector	To accept BSR calls from another server and queue them to the best of the local resources considered.
Commands	
`consider split/skill`	To obtain the Expected Wait Time or agent data that is needed to identify the best local resource. One consider step must be written for each split/skill that you want to check. Since the **`consider`** command is designed to compare two or more resources, consider commands are typically written in a series of two or more with the sequence terminating in a queue-to best vector step. This set of consider commands and a queue-to best step is called a consider sequence.
`consider location`	To obtain the Expected Wait Time or agent data that is needed to identify the best resource at a remote server. One consider step must be written for each location that you want to check. Routing information is obtained from the BSR Application plan for the active VDN.
`reply-best`	To return data to another server in response to a status poll
`queue-to`	With the best keyword to queue calls to the best resource that is identified by the consider sequence.
`check`	With the best keyword to queue calls to the best resource that is identified by the consider sequence if the resource meets certain conditions.
Key word	
best	In **`queue-to`**, **`check`**, and **`goto`** commands that refer to the resource identified as best by a series of consider steps
Conditional	
wait-improved	To prevent calls from being queued to an additional split/skill—local or remote—when the reduction in Expected Wait Time is not enough to be useful. Wait improved means that a call's EWT must be improved by a specific amount, which is a figure that you specify in seconds, over its current EWT or the server will not queue it to the additional split/skill.
User adjustment	
adjust-by	To control long-distance costs and limit trunk usage, reflecting factors such as availability of the trunks or agent expertise at remote locations. When a vector polls a local or remote resource, you can make the selection of that site less desirable. The higher the setting, the less chance that resource will be selected over another with a lower setting.

	With EWT returned, the setting increases the returned expected wait time for comparison with other returned EWTs. Optionally, the adjust-by setting applies in the available agent case. If you are using the UCD-MIA or EAD-MIA available agent strategy, the setting decreases the returned agent idle time, making the agent appear less idle (busier). If you are using the UCD-LOA or EAD-LOA available agent strategy, the setting increases the returned agent occupancy, making the agent appear more occupied (busier). In either case with EAD, the MIA or the LOA is used as a tie breaker if more than one site has an agent available with the same highest skill level.

Multi-site BSR applications

You can implement BSR at a single location solely by using the BSR commands in vectors. Using BSR across a network is more complex and requires additional administration.

A series of consider location steps in a multi-site BSR vector contacts one or more remote locations. You need to define these locations, tell the server how to contact each one, and set up VDNs and vectors to handle communications between the origin server and the remote (or receiving) servers. The BSR application should support some larger application in your call center that handles calls of a particular type.

Note:

Any combination of split/skill numbers, VDN numbers, and vector numbers can be used to support a single customer application or call type across a network. For clarity and simplicity, Avaya recommends that the BSR Application Plan number and the location numbers for a given application be the same on all servers.

You also need to set up ISDN trunk groups, set the parameters for information forwarding (UUI Transport), and administer numbering plans and AAR/ARS tables.

Multi-site BSR starts with the active VDN for a call, as determined by VDN override. If you want any specific VDN/vector pair to interflow calls using multi-site BSR, you must create a specific application for it. A multi-site application must contain the elements shown in the following table.

Table 8: Required elements of a multi-site BSR application

A BSR application consists of…	Which serves this purpose…
The Primary VDN	The Primary VDN is the active VDN for a call at the origin server, as defined by VDN override. Therefore, the Primary VDN in a BSR application does not have to be the VDN that originally received the incoming call. The primary VDN links its assigned vector to a BSR application plan and sets the “BSR Available Agent Strategy”.
The Primary vector that handles the incoming call on the origin server	The Primary vector contacts the specified remote servers, collects information, compares the information, and delivers or queues the call to the resource that is likely to provide the best service.

A BSR application consists of…	Which serves this purpose…
An application plan	The application plan identifies the remote servers that you can compare and specifies the information that will be used to contact each server and to route calls to it.
Two VDN/vector pairs on each remote server: • Status poll VDN/ vector • Interflow VDN/ vector	Status poll VDN/vector The status poll vector compares splits at its location and replies to the origin server with information on the best of these splits. Each remote server in a given application must have a dedicated status poll VDN/ vector.
	Interflow VDN/vector When a given remote server is the best available, the origin server interflows the call to this VDN/vector on the remote server. Each remote server in a given application has to have a dedicated interflow VDN/ vector. The steps in this vector deliver or queue the call, as appropriate, to the best resource that is found by the status poll vector.

To create a multi-site BSR application, you start by creating an application plan on the origin server.

Note:

Remember that the terms local, origin, and remote are relative terms. In most networks that use multi-site BSR, every server can interflow calls to other servers and receive interflowed calls from other servers. Therefore, every server in the network may have all the elements described above. For clarity in the following discussions, local or origin means a server that is considering or might consider whether to interflow a call. Remote means any server that is polled or might be polled by this first server.

Application plans

The application plan identifies the remote servers that you can compare and specifies the information that is used to contact each server and to route calls to it.

The plan for each application is identified by the application number and a name. It specifies the remote servers that might be polled by the application and identifies each with a number called the location number. The plan also specifies the numbers for the status poll and interflow VDNs for each remote server. Whatever you would dial to reach these VDNs is what should be entered in these fields: full length numbers as well as AAR, ARS, UDP, or public network numbers will work.

You create application plans on the Best Service Routing Application screen. A plan for an application with three remote servers might look like the following example.

Sample multi-site BSR Application Plan

```
BEST SERVICE ROUTING APPLICATION PLAN

 Number: 15   Name: Customer Service  Maximum Suppression Time: 60   Lock? y

Num Location Name    Switch Node            Status Poll    VDN   Interflow VDN  Net
Redir?
```

```
1  New Jersey        320   84015           84115               n
2 Denver       18   913031234015       913031234115        n
4 New York       12345   912121234015    912121234115        n
___ ______________      ____  ____   __________  __      ____________        n
___ ______________      ____  ____   __________  __      ____________        n
___ ______________      ____  ____   __________  __      ____________        n
___ ______________      ____  ____   __________  __      ____________        n
___ ______________      ____  ____   __________  __      ____________        n
___ ______________      ____  ____   __________  __      ____________        n
___ ______________      ____  ____   __________  __      ____________        n
___ ______________      ____  ____   __________  __      ____________        n
___ ______________      ____  ____   __________  __      ____________        n
___ ______________      ____  ____   __________  __      ____________        n
___ ______________      ____  ____   __________  __      ____________        n
```

The maximum number of application plans may vary depending on your Communication Manager software release and platform. For more information, see *System Capacities Table for Communication Manager on Avaya Media Servers*. You can find the latest capacity tables from the Avaya support Website at:

http://www.avayadocs.com

By entering the application number from this plan on a VDN screen, you can link a given VDN on your local server to this list of locations. This VDN becomes the primary VDN for the application. For example, if the primary vector contains instructions to consider locations 1 and 2, the server places a status poll call to the status poll VDN at the New Jersey and Denver servers and compares the results. If location 2 is better than either location 1 or any splits that are considered on the originating server, the call will be interflowed to the interflow VDN that is specified in the plan for location 2.

Example of multi-site BSR

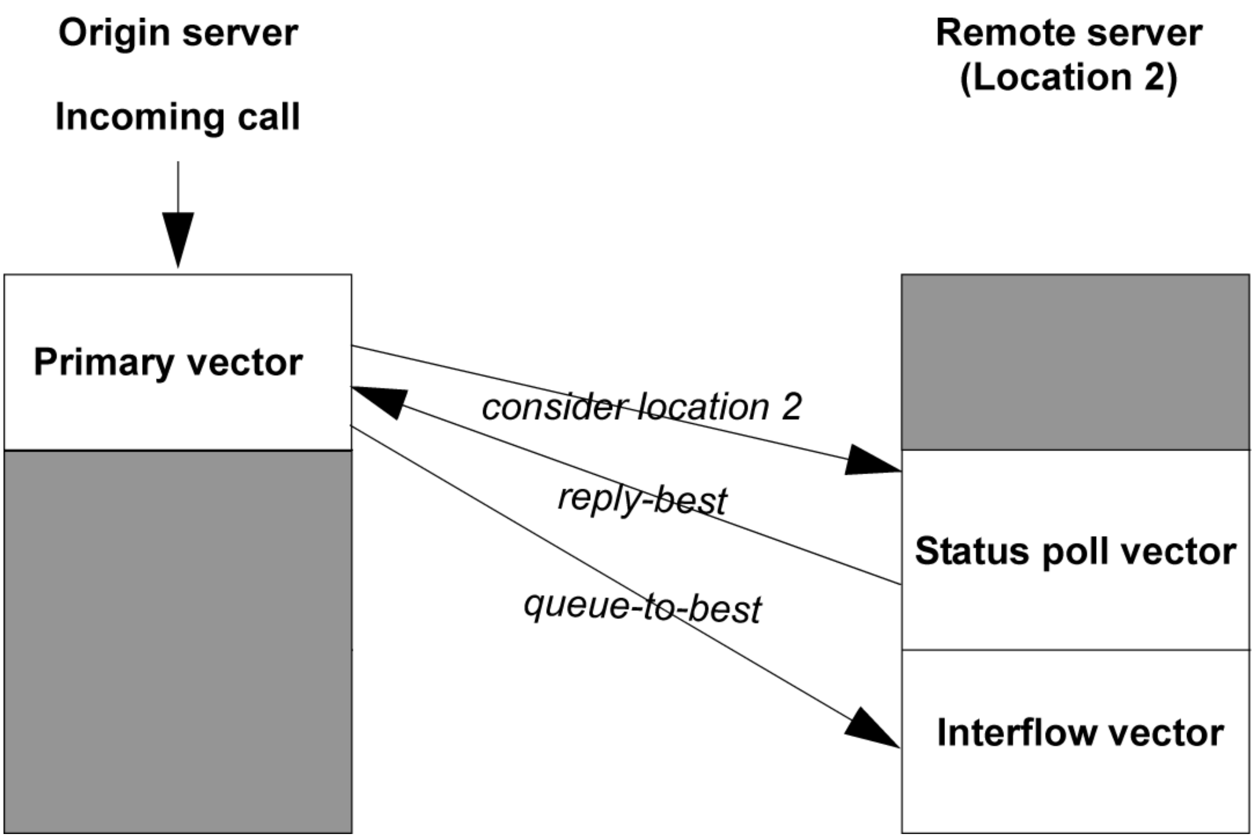
This section includes the following topics:

Simple multi-site BSR

To see how the basic elements of multi-site BSR work, consider a simple application in a two-server network. Multi-site BSR compares local and remote splits/skills and queues calls to the resource that provides the best service. Remember that each BSR application has two main parts:

- An application plan. This plan identifies the remote servers that you want to compare.
- A set of three VDN/vector pairs:
  - The primary VDN/vector. This vector on the origin server contacts the specified remote servers, collects information, compares the information, and routes the call to the server that is likely to provide the best service.
  - The status poll VDN/vector. The status poll vector on the remote server compares resources on that server and replies to the origin server with information on the best of these. Each remote server in a given application must have a dedicated status poll vector.
  - The interflow VDN/vector. When a given remote server is the best available, the origin server interflows the call to this vector on the remote server. Each remote server in a given application has to have a dedicated interflow vector.

The general operational scheme for multi-site BSR is shown in the following figure.



The following example shows the primary VDN using a multi-site BSR application.

BSR example primary VDN

```
change vdn xxxxx                                                    page 1 of 3
        VECTOR DIRECTORY NUMBER
```

```
                        Extension: 52222
                             Name: Multi-site BSR
                    Vector Number: 222
              Attendant Vectoring? n
               Meet-me Conference? n
               Allow VDN Override? n
                              COR: 59
                               TN: 1
                         Measured: internal
   Acceptable Service Level (sec): 20
          Service Objective (sec):
    VDN of Origin Annc. Extension:
                        1st Skill:
                        2nd Skill:
                        3rd Skill:
```

```
change vdn xxxxx                                           page 2 of 3       VECTOR
DIRECTORY NUMBER
                          Audix Name:
                  Return Destination:
              VDN Timed ACW Interval:
                     BSR Application:15
         BSR Available Agent Strategy: UCD-MIA
```

In the example shown above for VDN 52222, the entry in the **BSR Application** field links this VDN to BSR Application Plan 15. Also note the UCD-MIA entry in the **BSR Available Agent Strategy** field. If vector 222 uses BSR commands, calls are not automatically delivered to the first resource found with an available agent. All **`consider`** steps in vector 222 are executed, and one of the following things happens:

If:	Then:
There is no available agent in the local or the remote splits	The call will be queued to the split with the lowest adjusted EWT.
Only one split has an available agent	The call will be delivered to that split.
Two or more splits have available agents	The call will be delivered to the split with the most idle agent.

Also note that **Allow VDN Override?** is set to `n`. If a second VDN and vector are used to process this call, the UCD-MIA strategy and the application plan that are specified in VDN 52222 are used.

Application plan 15 (which is shown in Sample multi-site BSR Application Plan on page 107) identifies the remote server and provides the digit strings to dial into the VDNs for both the status poll vector and the interflow vector.

BSR primary vector

When a call arrives at the origin server, it is processed by the primary vector. This vector begins the BSR process by considering the resources that are specified. The following example shows a primary vector used for that purpose.

BSR example of primary vector on origin Communication Manager

```
1. wait time 0 secs hearing ringback
```

```
2. consider split   1 pri m   adjust-by 0
3. consider location 2 adjust-by 30
4. queue-to-best
```

In this example, the consider commands in steps 2 and 3 collect information to compare local split 1 with one or more splits at location 2. (Location 2 is the Denver server identified on the BSR Application Plan screen.) Step 4 queues the call to the best split that is found. As in single-site BSR, the adjust-by portion of the **consider** command allows you to set preferences for each resource, whether the resource is a remote location or a split/skill on the origin server. In multi-site BSR, this user adjustment enables you to control the frequency of interflows by adjusting the EWT that is returned by a particular resource on a remote server. In this example, the Communication Manager administrator has chosen to adjust the EWT value for location 2 by 30.

BSR status poll vector

To collect information from the remote server, the command **consider location 2 adjust-by 30** in the primary vector places an ISDN call, known as a status poll, to the status poll vector on the server at location 2. The following example shows a status poll vector on the remote server used for that purpose.

BSR example of status poll vector on remote Communication Manager

```
1. consider split   2  pri m   adjust-by 0
2. consider split 11    pri m   adjust-by 0
3. reply-best
```

The status poll only obtains information and returns it to the origin server; the call is not connected to the status poll VDN.

This vector compares splits 2 and 11, identifies the better of the two, and sends this information back to server 1 with the **reply-best** command. Notice that the **adjust-by** command could be used on the remote server to adjust the EWT that is returned by either of the splits. When EWT adjustments are applied at both the origin and remote servers, the two adjustments are added at the origin server. For more detail on user adjustments in multi-site applications, see User adjustments in multi-site BSR on page 114.

The **consider** command is ISDN neutral and does not return answer supervision. The status poll call is dropped when the reply-best step executes, but the ISDN DISCONNECT message that is returned to server 1 contains the information from the best split considered at location 2. Once the remote server returns the necessary information, the consider series in the primary vector on server 1 can continue at the next vector step.

Caution:

It is recommended that status poll vectors not be used to poll other servers. Status poll vectors should only consider resources on the server where the vector resides. Status poll vectors must always end with a reply-best step. A busy or disconnect should never be used.

Note:

Multi-site BSR includes mechanisms that automatically limit the number of status poll calls that are placed over the network when such calls are unlikely to yield better service for the caller. For a detailed explanation of these mechanisms, see Advanced multi-site routing on page 533.

BSR interflow vector

In this example, assume that no agents are available and that split 11 (location 2) has the lowest adjusted EWT. The **`queue-to best`** command in the primary vector will interflow the call to the interflow vector at location 2. The following example shows what the interflow vector looks like.

BSR example of interflow vector on remote Communication Manager

```
1. consider split   2   pri m   adjust-by 0
2. consider split 11     pri m   adjust-by 0
3. queue-to best
```

The interflow vector reconsiders the status of both splits to get the most current information and queues or delivers the call to the best split. Notice that the consider sequences in the interflow vector and the status poll vector are identical aside from their last step. When a call is interflowed, it is removed from any queues at the origin server and any audible feedback at the origin server is terminated.

Caution:

BSR will not operate correctly unless the consider series in the status poll vector and the interflow vector use the same splits/skills with the same queue priorities.

BSR call interflow with SIP

You can use SIP to interflow BSR calls, but BSR polling is not supported over SIP trunks. The polling must be done using either H.323 or ISDN trunks. Once the polling has determined the best site, the incoming call can be routed to another location by the queue-to-best step. The redirected call can be routed over a SIP trunk.

What happens to the call if the interflow attempt fails?

If the interflow attempt fails, for example, because there are no available trunks, the call is queued to the best local split. The call is not disconnected. The call is not dropped from vector processing on the origin server. For the call to be queued to a local split, however, that split must have been considered at some previous point in the consider series. In writing primary vectors, always consider local splits/skills before considering remote resources.

Adjusting the AIT or AOC returned by an available resource

1. Go to the feature related system parameters screen.
2. Enable the**Available Agent Adjustments for BSR?** option field.
3. Go to the Vector screen and program a **`consider split/skill`** or **`consider location`** vector command specifying both the split/skill or location and the **`adjust-by`** parameter.

   The **`adjust-by`** parameter can be used to provide a percentage value during vector processing and can be:

   - A percentage (0 through 100)
   - A vector variable (A-Z, AA-ZZ)

- A VDN variable (V1-V9)

Result

Once the vector command is executed, the adjustment factor has the following result when the remote site has an available agent:

- For the MIA strategies, the adjustment reduces the agent idle time (AIT) received.
- For the LOA strategies, the adjustment increases the agent occupancy percentage (AOC) received.

Depending on the available agent strategy assigned to the VDN for the call, the adjusted AIT or the adjusted AOC is used for that local split/skill or remote location when choosing between available agents over multiple locations.

Example: You have an agent whose current AIT is 40%. You want to increase this agent's idle time to 60% to handicap sending the call to that remote location. If the strategy is ucd-loa, you can program the following vector command:

```
consider location 4 adjust-by 50
```

The occupancy used for location 4 is increased by 50% of the actual occupancy. The occupancy originally sent was 40%. A 50% adjust-by results in multiplying 40 by 50% resulting in 20. Therefore, 40 + 20 = 60%.

BSR available agent strategies

In multi-site BSR applications, the 1st-found available agent strategy results in fewer interflows and thus minimizes the load on trunking between Communication Managers. The Communication Manager also has less processing to perform for each call in BSR vectors, since it may not need to compare as many resources to identify the best. If processing power and tie trunk capacity are issues in your multi-site applications, you may want to use the 1st-found strategy.

The other strategies typically result in a much greater percentage of calls being interflowed, thus optimizing load balancing across locations. For a strategy that greatly increases agent fairness across the network while limiting the number of trunks used, see Example of multi-site BSR with limited trunking on page 115.

More on status poll and interflow vectors

The following points are important to consider when you write status poll and interflow vectors.

- Since status poll vectors do not return answer supervision, call charges are not normally incurred for the status poll portion of the call flow.
- When a “consider location” step performs a status poll, it also checks for the availability of a B-channel. If no B-channel is available, the remote resource is never considered the best since the call cannot be redirected to it.

- If only one split/skill on a remote server can service the call type that is handled in a BSR application, you do not need to write a consider series in the interflow vector. You can just queue the call to the appropriate resource.
- If status poll and interflow vectors consider more than one split/skill, the VDNs for these vectors must be administered with the appropriate BSR available agent strategy.

User adjustments in multi-site BSR

User adjustments are especially important in multi-site applications, where unnecessary interflows may be costly and use trunk capacity inefficiently.

User adjustments in multi-site applications function in the same way they do in single-site BSR with one important difference: user adjustments may be applied at the remote servers in an application as well as at the origin server. Since a status poll vector uses consider steps to evaluate resources on the server where it resides, the adjust-by portion of each **`consider`** command allows the administrator at each server to set preferences for the splits/skills at that server. In BSR applications, any such adjustment for a split/skill is considered by the status poll vector in selecting the best resource on its server. The adjustment is then returned to the origin server along with the other data for that resource. When the server receives this adjustment from the remote server, it adds it to any adjustment that was assigned to that location in the consider location step. The following example assumes, of course, that no agents become available during the time these vectors are processing the call.

The following example shows a primary vector that considers one remote location, to which it assigns an adjustment of 30.

Vector with consider step for one location

```
1. wait time 0 secs hearing ringback
2. consider split     pri m  adjust-by  0
3. consider location 2 adjust-by 30
4. queue-to-best
```

The following example shows the status poll vector at location 2.

Status poll vector

```
1. consider split   2  pri m   adjust-by 0
2. consider split 11    pri m   adjust-by 20
3. reply-best
```

Consider split/skill commands in status poll vectors work just like they do in single-site BSR vectors. The user adjustments are applied to a single split/skill and not to the entire location. In this case, the two splits are assigned different adjustments. Say that split 11, despite having the larger adjustment, returns the lower adjusted EWT for a call. The **`reply-best`** command in step 3 returns the user adjustment of 20 to the primary vector on the origin server, along with the rest of the data for split 11.

In saving the data that is returned by location 2, the origin server adds the remote adjustment of 20 to the adjustment of 30 that is specified in step 3 of the primary vector. As a result, the call will not interflow to location 2 in this example unless the EWT for location 2 is more than 50 seconds better than the EWT in split 1 on the origin server.

Example of multi-site BSR with limited trunking

Multi-site BSR applications must balance improvements in wait times and agent utilization with the cost of interflows and the availability of inter-server trunking for status polls and interflows. The following example shows an application that is recommended for balancing agent workload across the network while still limiting tie trunk usage.

BSR example of Application Plan

```
BEST SERVICE ROUTING APPLICATION PLAN

 Number: 10   Name: International       Maximum Suppression Time: 60   Lock? y

Num Location Name    Switch Node           Status Poll    VDN   Interflow VDN  Net
Redir?

1 Kansas City     1111   919131234015        919131234115          n
2 New York      1112   912121234015        912121234115          n
3 Montreal      1113   915141234015        915141234115          n
3 London        1114   90114411234015      90114411234115        n
___ ______________      ____  ____   __________  __      ____________         n
___ ______________      ____  ____   __________  __      ____________         n
___ ______________      ____  ____   __________  __      ____________         n
___ ______________      ____  ____   __________  __      ____________         n
___ ______________      ____  ____   __________  __      ____________         n
___ ______________      ____  ____   __________  __      ____________         n
___ ______________      ____  ____   __________  __      ____________         n
___ ______________      ____  ____   __________  __      ____________         n
___ ______________      ____  ____   __________  __      ____________         n
___ ______________      ____  ____   __________  __      ____________         n
___ ______________      ____  ____   __________  __      ____________         n
```

The following Vector Directory Number example shows the VDN screen for VDN 51110, the VDN that is used in this BSR Application Plan example. In the example, the entry in the **BSR Application** field links this VDN to BSR Application Plan 10. Also note the `EAD-MIA` entry in the **BSR Available Agent Strategy** field. If vector 100 uses BSR commands, calls are not automatically delivered to the first resource found with an available agent. In each consider sequence, when the **`queue-to best`** or **`check best`** step executes, one of the following things happens:

If …	Then…
No skill has an available agent	The call is queued to the skill with the lowest adjusted EWT.
Only one skill has an available agent	The call is delivered to that skill.
Two or more skills have available agents	The call is delivered to the skill with the most expert agent, which is the agent with the lowest skill level.
Two or more skills have available agents with the same skill level	The call is delivered to the skills that has the most idle agent.

Also note that **Allow VDN Override?** is set to `n`. If a second VDN and vector are used to process this call, the, the EAD-MIA strategy and the application plan that is specified for VDN 51110 is still used.

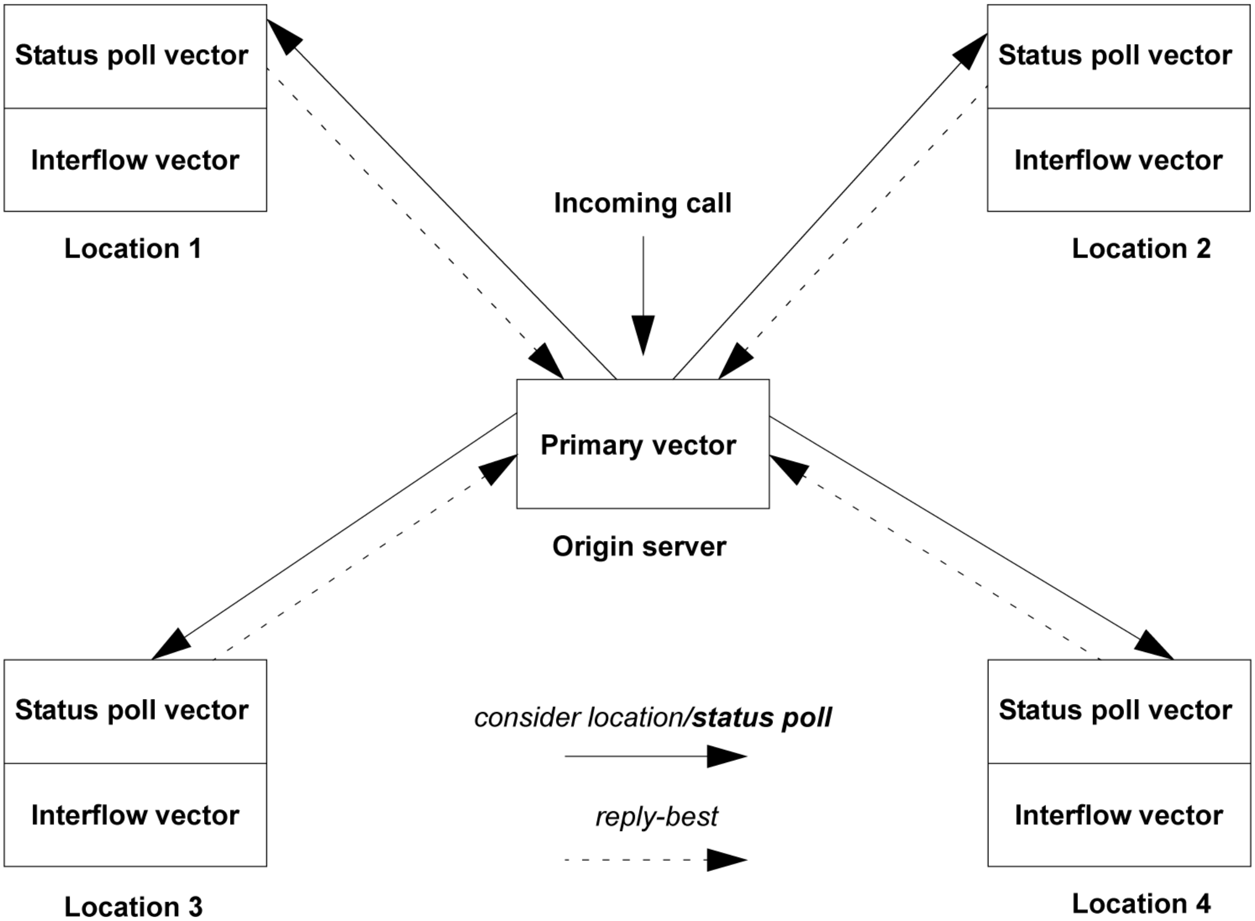
BSR example of primary VDN

```
change vdn xxxxx                                                  page 1 of 3
        VECTOR DIRECTORY NUMBER

                            Extension: 51110
                                 Name: Multi-site BSR
                        Vector Number: 100
                  Attendant Vectoring? n
                   Meet-me Conference? n
                   Allow VDN Override? n
                                  COR: 59
                                   TN: 1
                             Measured: none
       Acceptable Service Level (sec): 20
              Service Objective (sec):
        VDN of Origin Annc. Extension: 1001
                            1st Skill:
                            2nd Skill:
                            3rd Skill:
```

```
change vdn xxxxx                                                 page 2 of 3
        VECTOR DIRECTORY NUMBER

                           Audix Name:
                 Messaging Server Name:
                    Return Destination:
                VDN Timed ACW Interval:
                       BSR Application:15
         BSR Available Agent Strategy: UCD-MIA
             Observe on Agent Answer?:n
```

With four remote servers to be considered, the overall application is represented in the following figure. Application plan 10 on the origin server identifies the remote servers and provides the digit strings to dial into the VDNs for both the status poll vector and the interflow vector on each server.

Each **`consider location`** command in the primary vector places a status poll call to its specified location. The status poll vector at that location executes a series of **`consider skill`** commands and returns data on the best resource to the origin server through a **`reply-best`** command.



The following example shows the primary vector for this application. The first consider series in the primary vector tests two local skills. If either skill has an available agent, step 4 jumps to step 9 and the call is queued locally. No remote locations are polled. If no agents are available in either local skill, though, steps 5 to 8 test 4 remote locations. In general, you should not put other commands between **`consider`** steps. This use of the **`goto`** step is one of the few exceptions to that rule.

If the best remote location's adjusted EWT can reduce the call's current adjusted EWT, step 9 interflows the call to that location. In this vector, a local available agent is always favored over a remote available agent. Whichever location services a call, it will always be directed to the most idle, best skilled agent available.

Multi-site BSR example

```
1. wait time 0 secs hearing ringback
2. consider skill 1 pri m    adjust-by  0
3. consider skill 2 pri m   adjust-by 20
4. goto step 9 if expected-wait for best = 0
5. consider location 1      adjust-by 30
6. consider location 2      adjust-by 30
7. consider location 3      adjust-by 50
8. consider location 4      adjust-by 50
9. queue-to best
10. announcement 1001
```

```
11. wait time 60 secs hearing music
12. goto step 10 if unconditionally
```

In the primary vector, note that user adjustments are entered for local skill 2 as well as for all the remote locations. These indicate the administrator's preferences regarding both local and remote resources. For this example, let's say that neither local resource has an available agent and therefore an EWT greater than 0.

Status poll vector in a multi-site BSR application

Each receiving server in a multi-site application must have a status poll vector. To collect information from these locations, each **consider location** command in the primary vector places a status poll to the status poll vector for the appropriate server. The following example shows the status poll vector on the server at location 3.

BSR example of status poll vector at location 3

```
1. consider skill    2   pri m  adjust-by 0
2. consider skill 11     pri m  adjust-by 20
3. consider skill 21   pri m  adjust-by 30
4. reply-best
```

This vector compares skills 2, 11, and 21, identifies the best one, and sends this information back to the origin server through the **reply-best** command. Notice that user adjustments are applied to skills 11 and 21 to adjust the skill's EWT. When EWT adjustments are applied at both the origin and remote servers, the two adjustments are added at the origin server. For more detail on user adjustments in multi-site applications, see User adjustments in multi-site BSR on page 114.

In this example, suppose that skill 11 has the best adjusted EWT at location 3. Its data, including a user adjustment of 20, is returned to the origin server by the **reply-best** command.

Finding the best resource

Once the remote servers have returned the best data for each location, the second consider series in the primary vector can be completed. In this example, let's suppose that no agents are available at any remote location.

The following table shows how user adjustments at the origin and remote servers yield the adjusted EWT for each location.

Table 9: BSR best resource user adjustments

Location	Actual EWT of remote best (sec.)	User adjustment on origin server	User adjustment on remote server	Adjustment applied by origin server (sec.)	Adjusted EWT used in BSR calculations (sec.)
1	60	30	0	30	90
2	45	30	10	40	85
3	40	50	20	70	110
4	70	50	0	50	120

The second consider series identifies location 2 as the best remote location, with an adjusted EWT of 85, and the queue-to best step interflows this call to location 2.

Interflow vector in a multi-site BSR application

The interflow vector on a remote server in a multi-site application accepts the interflowed call from the origin server. It also executes the same consider series as the status poll vector to identify the current best resource, in case conditions have changed since the status poll.

The following example shows the interflow vector on a remote server.

BSR example of interflow vector at location 2

```
1. consider skill  2     pri m   adjust-by 0
2. consider skill 11     pri m   adjust-by 20
3. consider skill 21  pri m   adjust-by 30
4. queue-to best
```

As happens today when a call is interflowed, it is removed from any queues at the origin server and any audible feedback at the origin server is terminated.

Caution:

BSR will not operate correctly unless the consider series in the status poll vector and the interflow vector use the same splits/skills with the same queue priorities.

Example of multi-site BSR with slow networks

Network response times are not an issue for most users. This example is intended for those users, if any, who experience such a problem. This example uses the same VDN, application plan, and four-server network that is described in the Example of multi-site BSR with limited trunking on page 115. The vector in that example minimized interflows by using a goto step that skips the remote consider series if a local resource has an available agent. This design is especially useful if network response times are slow. Calls are always queued once locally before remote locations are considered.

Furthermore, both status polls and interflows are conditional. The call can wait in the queue for a local resource while BSR looks for a better split/skill at remote locations.

This example also shows the function of the **`check best`** command and the wait-improved conditional.

The following example shows the primary vector for this application, vector 100. The first consider series in the primary vector tests two local splits and queues the call to the best one. If the EWT for the best split is 30 seconds or less, step 5 jumps to the loop in step 11 and the second consider series is not executed. If the EWT for the best split is over 30 seconds, though, steps 6 through 9 test 4 remote locations. If the best remote location can reduce the call's EWT by more than 30 seconds as compared to its EWT in the best local queue, step 10 interflows the call to that location.

Caution:

Be certain to queue calls at least once before using the wait-improved conditional in a vector step. If calls are not already queued when the step with the wait-improved conditional executes, The server reads the call's EWT as infinite. This could result in a vector that interflows all calls, even if that is not its intended function.

Multi-site BSR with EWT

```
1. wait time 0 secs hearing ringback
2. consider skill  1    pri m   adjust-by  0
3. consider skill   2 pri m     adjust-by 20
4. queue-to-best
5. goto step 11 if expected-wait for call <= 30
6. consider location 1      adjust-by 30
7. consider location 2      adjust-by 30
8. consider location 3      adjust-by 50
9. consider location 4      adjust-by 50
10. check best if wait-improved > 30
11. announcement 1001
12. wait time 60 secs hearing music
13. goto step 11 if unconditionally
```

A consider series can end with either a queue-to best or a check best step. The **`check best`** command lets you set conditions that must be met before a call is queued to the best resource. In this example, step 10 in the primary vector is **`check best if wait-improved > 30`**. In other words, step 10 interflows the call to the best location found by the consider series only if the EWT for that location is more than 30 seconds better than the call's EWT in the local queue.

You can use up to 3 consider series in one vector. It is possible to write more than 3 consider series in a vector, but there's no benefit in doing so. The server only allows you to queue a call simultaneously to 3 different local resources. Since each consider series ends by queuing a call (assuming no agent is available), using more than 3 series in a vector will not place the calls in additional local queues. If the call interflows to another Communication Manager, it's removed from vector processing and any queues it was in on the origin server.

It is also possible to combine single-site and multi-site consider series, as this example shows. Note that user adjustments are entered for local skill 2 as well as for locations 3 and 4. These indicate the administrator's preferences regarding both local and remote resources. In this example, say that step 2 queues the call to skill 1, which has an EWT of 65 seconds, before the second consider series is executed.

Status poll vector in a multisite BSR with slow networks

Each receiving server in a multi-site application must have a status poll vector. To collect information from these locations, each **`consider location`** command in the primary vector places a status poll to the status poll vector for the appropriate server. The following example shows the status poll vector on the server at location 3.

BSR example of status poll vector at location 3

```
1. consider skill     2  pri m  adjust-by 0
2. consider skill 11     pri m  adjust-by 20
3. consider skill 21   pri m  adjust-by 30
4. reply-best
```

This vector compares skills 2, 11, and 21, identifies the best one, and sends this information back to the origin server through the **`reply-best`** command. Notice that user adjustments are applied to skills 11 and 21 to adjust the skill's EWT. When EWT adjustments are applied at both the origin and remote servers, the two adjustments are added at the origin server. For more details on user adjustments in multi-site applications, see User adjustments in multi-site BSR on page 114.

Suppose that skill 11 has the best adjusted EWT at location 3. Its data, including a user adjustment of 20, is returned to the origin server by the **reply-best** command.

Remember that the first consider series queued the call to local skill 1. Say that the second consider series identifies location 2 as the best remote resource. The **check** command in step 10 recalculates the call's current, unadjusted EWT in skill 1 and compares it to location 2's unadjusted EWT. If the call's actual (unadjusted) EWT can be improved by more than 30 seconds, the call is interflowed.

Note:

BSR uses adjusted EWT to determine which of the resources in a consider series is the best. Once the best resource is identified, subsequent **expected-wait** and **wait-improved** conditionals use the actual EWT values.

Interflow vector in a multisite BSR with slow networks

When a call is interflowed to any of the remote locations, the interflow vector on that server accepts the interflowed call from the origin server. It also executes the same consider series as the status poll vector to identify the current best resource, in case conditions have changed since the status poll. The following example shows such an interflow vector.

BSR example of interflow vector at location 2

```
1. consider skill   2  pri m  adjust-by 0
2. consider skill 11     pri m  adjust-by 20
3. consider skill 21   pri m  adjust-by 30
4. reply-best
```

Caution:

BSR will not operate correctly unless the consider series in the status poll vector and the interflow vector use the same splits/skills with the same queue priorities.

If the call is queued to a remote resource by step 10 in the primary vector, is the call removed from the local queue that it entered in step 4?: When a call is interflowed, the call is removed from any queues at the origin server and any audible feedback at the origin server is terminated.

The second consider series can compare local and remote resources. If it does, and if step 10 queues the call to another local skill, will the call be removed from the local queue that it entered in step 4?

No. In general, the server can queue a call to as many as 3 local splits/skills simultaneously. BSR does not change this limit.

Example for handling excessive wait times

This short example shows a simple primary vector in a multi-site BSR application. If wait times are sometimes excessive because of high call volumes, step 4 of this vector directs calls to a disconnect after announcement step when wait time in the network exceeds 5 minutes. The following example shows a simple primary vector.

Multi-site BSR using disconnect for excessive wait times

```
1. wait 0
2. consider skill 1     pri m   adjust-by 0
3. consider location 2 pri m   adjust-by 30
4. goto step 6 if expected-wait for best ≤ 300
5. disconnect after announcement 3001
6. queue-to best
```

Announcement 3001 might say something like, *We're sorry. We are currently experiencing heavy call volume and cannot service your call at this time. Please try again later. We are normally least busy between 8 a.m. and 11 a.m. each morning.*

Planning and administering multi-site BSR

Selecting or administering application plans

To select or administer a BSR application plan:

1. Select the VDNs on each server that serve the group of callers you have identified.

   On each server these are the Primary VDNs for your application. You may, of course, want or need to create new VDNs. In either case, record the extensions of each VDN that will point to a vector with a BSR application.
2. Select the locations that you want to include in each application plan.

   To uniquely identify each location, assign a number between 1 and 255 and a short name of 15 characters or less.
3. Record the node number of the server at each location.
4. Create Status Poll VDNs on each of the servers in the application plan.

   Record the full numbers you will need to route calls to these VDNs. These numbers will be entered on the Best Service Routing Application Plan screen when you create the plan.

   If you are creating new VDNs on the Communication Managers that will receive interflowed calls, record these numbers too. You will need them to complete the BSR Application Plan screen. Remember: you cannot use the same number for a Status Poll VDN and an Interflow VDN.

Administering the BSR Application Plan

Defining the application plan

To create an application plan on each Communication Manager:

1. At the command line prompt, type `add best-service-routing xxx` and press `Enter` (where `xxx` is a number between 1 and 255 that you want to assign to this BSR application.)

   The system displays the Best Service Routing Application Plan screen. The number that you typed in the command appears in the**Application Number** field.
2. Assign a name to the plan.

   The best names are short and descriptive. This name cannot be longer than 15 characters.
3. Type in the information for the first remote location.

   Fill in the information for each field as shown below.

   Note:
   Each row on the screen contains all of the information the BSR application needs to identify and communicate with one of the resources in the plan.

   Table 10: Fields on application plan screen

Field	Type	Description
Num	Required	Type the number that you assigned to this location in 2.
Location Name	Optional	Type the name that you assigned to this location in 2.
Switch Node	Optional	This field is for user reference only. Leave it blank. If you are using the Universal Call ID feature, you may want to type each Communication Manager node identity in this field. The server node identity is the number that is entered in the UCID Network Node ID field on page 4 of the Feature-Related System Parameters screen.
Status Poll VDN	Required	This is the complete digit string that your Communication Manager will dial for the status poll call. The string can be up to 16 digits long.

Field	Type	Description
Interflow VDN	Require d	This is the complete digit string that your Communication Manager dials to interflow a call to this location. The string can be up to 16 digits long.

4. Repeat 3 on page 0 for each of the locations that you want to include in the application plan.
5. Press `Enter` to save your changes.

Result

Note:
You must set up trunk groups to other sites. For information on setting up trunk groups, see Look-Ahead Interflow (LAI) and Information Forwarding on page 225.

Linking the application plan to a primary VDN and enter an agent selection strategy

To link the application plan to a primary VDN and enter an agent selection strategy:

1. Go to the Vector Directory Number screen for the first VDN that you identified in 1.
   If this is a new application, create the VDN.
2. In the **Allow VDN Override?** field, type `y` or `n`.
   If the call is directed to another VDN during vector processing:
   - **`y`** allows the settings on the subsequent VDN, including its "BSR Available Agent Strategy", to replace the settings on this VDN.
   - **`n`** allows the settings on this VDN, including its "BSR Available Agent Strategy", to replace, or override, the settings on the subsequent VDN.
3. In the **BSR Application** field, type the application number you assigned to the plan.
4. In the **BSR Available Agent Strategy** field, type the identifier for the agent selection method you want this application to use:

If you enter…	The application will select the resource with…
`1st-found`	The lowest Expected Wait Time. If the application finds an available agent before it has compared all the locations in the plan, the application routes the call to that agent without contacting any other locations.
`ucd-mia`	The agent who has been idle the longest. The application compares all the locations in the plan.

If you enter…	The application will select the resource with…
ead-mia	The agent with the highest skill level, which is the lowest skill number, who has been idle the longest.
ucd-loa	The least-occupied agent.
ead-loa	The agent with the highest skill level, which is the lowest skill number, who is the least occupied.

5. Press `Enter` to save your changes.

Result

Repeat steps 1 on page 124 through 5 on page 125 on each server that needs an application plan and a Primary VDN/vector pair.

This process covers the administration that is needed for BSR vector commands to function. Now, of course, you need to write or modify the vectors that will control call processing.

Local treatment for remotely queued IP and ISDN calls

About BSR local treatment for calls queued remotely

Note:
Voice Portal Call Back Assist is not supported with BSR Local Treatment, as the use of the converse-on vector step does not have knowledge of the queue position and wait time for the call at the remote end.

In a multi-site BSR configuration, a call that arrives at a local Communication Manager can be rerouted to a remote server located in a different part of the world. To better meet the needs of such multi-site call centers, Communication Manager 2.0 or later includes a newBSR Local Treatment for Calls Queued Remotely Over IP or ISDN Trunks feature that allows you to provide local audio feedback for IP and ISDN calls while a call waits in queue on a remote server.

This feature provides the following potential benefits for call center operations:

- For multi-site BSR operations that include sites located in different countries, the new local treatment feature can result in significant bandwidth savings for IP calls.
- Audio quality concerns that occur when music is sent over wide area networks that use low bit-rate codecs are eliminated.
- Announcements and other treatments can be maintained and managed in a central location.

Overview of local treatment operations

This section describes local treatment feature operations that occur when BSR redirects IP or ISDN calls to a remote queue.

Important:
The local treatment operations described in this section assume that the required feature and vector administration steps are implemented on both the local *and* remote Communication Managers.

For information about feature administrations, see Screens and fields used to administer local treatment in *Administering Avaya Aura™ Call Center Features*.

For information about required vector design, see Example vectors for the local treatment feature on page 127.

The following steps describe the basic process for local treatment operations in a multi-site BSR environment:

1. A call arrives at the local Communication Manager and is processed by a VDN that is enabled for BSR local treatment.
2. The local vector includes the **`consider`**, **`queue-to best`**, and **`wait`** hearing announcement steps that are required for BSR local treatment operations.
3. A skill on a remote server is identified as best location and the local server attempts an interflow to the remove server. Vector processing is temporarily suspended on the local server while the interflow attempt is in progress.
4. If the interflow attempt succeeds, the remote server returns an ISDN_PROGRESS message with progress indicator of in-band information (8) to indicate that the call is in queue and local treatment operations can proceed.

   The remote server must meet the following requirements for the appropriate ISDN_PROGRESS message to be sent back to the local server:

   - The remote server is administered for BSR local treatment.
   - The call is directed to a VDN that is also enabled for local treatment.
   - The vector associated with the VDN includes only those steps and commands that are required for successful local treatment operations.
5. The local server receives the ISDN_PROGRESS message with progress indicator of in-band information (8), vector processing resumes with an appropriate treatment step and the caller receives feedback provided by the local server while they wait in the remote queue.

   Important:
   To ensure that the local treatment feature operates as designed, use only the vector commands that are recommended for local treatment implementation.

Although local treatment operations do not impose restrictions on the types of vector steps that are administered on the local server after call processing resumes, use of inappropriate vector steps can interfere with local treatment operations. For more information, see Example vectors for the local treatment feature on page 127.

6. When an ACD agent on the remote server accepts the call, an ISDN_ALERTING message is sent to the local server. Vector processing is discontinued on both the local and remote servers.

Local treatment system requirements

The BSR Local Treatment for Calls Queued Remotely Over IP or ISDN Trunks feature works on all platforms and operating systems that are supported by the Avaya Communication Manager. You must meet the following licensing and system requirements to use the local treatment feature:

- The Communication Manager release must be 2.0 or later.
- The system license file must be configured to enable the following features:
  - Call Center Release 12.0 and later
  - LAI
  - BSR
  - BSR Local Treatment for IP and ISDN

Example vectors for the local treatment feature

This section provides vector guidelines and examples that describe how to implement the local treatment feature. Vector administration typically requires polling vectors on both the local and remote Communication Manager and an interflow vector on the remote server. The polling vector on the local server should also be administered to provide an appropriate local call treatment.

This section includes the following topics:

- Implementation guidelines for local treatment vectors on page 127
- Data return on page 552
- Example polling vector for the remote Communication Manager on page 130
- Example interflow local treatment vector for the remote Communication Manager on page 130

Implementation guidelines for local treatment vectors

This section describes the best practices for successful implementation of the local treatment feature.

Important:

Read these guidelines before you implement the local treatment feature.

Implementation of the local treatment feature requires use of specific vector steps to generate the correct ISDN messages between the local and remote Communication Managers. If the treatment, polling and interflow vectors that are administered to implement this feature include vector steps other than those recommended in this section, the feature may not work as intended and the associated bandwidth savings may not be realized.

For polling vectors: You must be careful to administer your local treatment polling vectors so that calls are not unintentionally dropped or phantom calls are generated. If the **`queue-to best`** step is followed by vector steps that include any commands other than announcement, **`wait`**, or **`goto`**, the trunk to the remote queue may be dropped. For example, the addition of **`consider`** steps after a **`queue-to best`** command can cause intermittent call behavior. The addition of a **`queue-to`** step after a **`queue-to best`** step may cause phantom calls to be queued to the remote server.

Tip:

You can also exploit this functionality to allow the local server to *take back* calls that remain in queue on a remote server after a specified time limit is exceeded. For more information, see Take back example on page 129.

Interflow local treatment vectors on the remote Communication Manager: When the BSR Local Treatment feature is enabled, specific ISDN messages must be exchanged between the remote and local Communication Managers. If additional vector steps are included either before or after the **`consider`** steps (if used) and **`queue-to best`** in the interflow vector on the remote server, the following results occur:

- Either an ALERTING or PROGRESS message (with in-band information) is returned from the remote server to the local server.
- In response to the message, trunk bandwidth is immediately allocated and the call is removed from the local queue.
- Local treatment operations cease, trunk bearer resources are allocated for the call sooner than required and cost savings associated with the local treatment feature are not realized.

Example vectors for the local Communication Manager

The following examples shows two different vector strategies that you can use to implement the local treatment feature on the local server. Vectors created for this purpose are the same as those used in all BSR polling operations, which include a consider series followed by a queue-to best step.

Important:

You must be careful to administer your local treatment polling vectors so that calls are not unintentionally dropped. For more information, see Implementation guidelines for local treatment vectors on page 127.

After the various skills and locations are polled and the call is placed in queue at the identified best location, the local server continues to maintain control of the call until it is answered by an agent. While the call is in queue, the local server continues to provide additional vector steps to implement the local call treatment.

At a minimum, the local treatment vector should include announcement and wait-time steps to provide appropriate feedback to the caller. However, the local treatment vector can be designed to use either a continuous loop or take back strategy. These alternate local call treatment strategies are described in the following sections.

Continuous loop example

the following example shows a vector that provides a sequence of call treatment steps on the local server that proceed in a continuous loop until an agent answers the call at the remote location.

In the following vector example, step 6 places the call in queue at the identified best location. Step 7 provides an appropriate announcement and step 8 provides 10 seconds of music. Step 9 uses an unconditional goto step to loop call processing back to step 6, where the treatment process continues.

```
change vector 40                                                        Page 1 of 3
                                    CALL VECTOR

Number: 40                    Name: Local BSR vector
                 Attendant Vectoring? n         Meet-me Conf? n           Lock? n
    Basic? y    EAS? y   G3V4 Enhanced? y      ANI/II-Digits? y   ASAI Routing? y
Prompting? y    LAI? y  G3V4 Adv Route? y       CINFO? n BSR? y       Holidays? y

01 announcement 3000
02 consider skill 4 pri m adjust-by 0
03 consider skill 6 pri m adjust-by 0
04 consider location 1 adjust-by 10
05 consider location 2 adjust-by 10
06 queue-to best
07 announcement 3001
08 wait-time 10 secs hearing music
09 goto step 7 if unconditionally
```

Take back example

The previous example set up the local treatment process as a continuous loop that repeats indefinitely while the call remains in queue at the identified best location. However, you can also design vectors that allow the local server to take back a call after it remains in queue for a specified amount of time.

In the following vector example, the **`queue-to best`** in step 6 is followed by a series of **`announcement`** and **`wait-time`** commands provided in steps 7 through 12. If the treatment steps complete and the call still remains in the remote queue, vector processing proceeds to step 13, which uses a **`route-to`** command that causes the call to the remote server to be dropped. The **`route-to`** step can be used to provide alternate services for the call.

Note:
When the call to the remote server is dropped, a type 305 vector event is logged.

```
change vector 40                                                   Page 1 of 3
                                    CALL VECTOR

Number: 40                     Name: Local BSR vector
                 Attendant Vectoring? n          Meet-me Conf? n           Lock? n
    Basic? y    EAS? y   G3V4 Enhanced? y      ANI/II-Digits? y   ASAI Routing? y
Prompting? y    LAI? y  G3V4 Adv Route? y       CINFO? n BSR? y       Holidays? y

01 announcement 3000
02 consider skill 4 pri m adjust-by 0
03 consider skill 6 pri m adjust-by 0
04 consider location 1 adjust-by 10
05 consider location 2 adjust-by 10
06 queue-to best
07 announcement 3001
08 wait-time 10 secs hearing music
09 announcement 3001
10 wait-time 10 secs hearing music
11 announcement 3001
12 wait-time 10 secs hearing music
13 route-to number 54010 if unconditionally
```

For another method to take back the call based on the amount of time the call has been in the system, see vdn type variable in the *Programming Call Vectors in Avaya Aura™ Call Center* document.

Example polling vector for the remote Communication Manager

The following example shows a call vector that polls skills on the remote server. This vector does not differ from other typical BSR polling vectors.

```
change vector 31                                                   Page 1 of 3
                                    CALL VECTOR

Number: 31                     Name: Remote BSR poll vector
                 Attendant Vectoring? n          Meet-me Conf? n           Lock? n
    Basic? y    EAS? y   G3V4 Enhanced? y      ANI/II-Digits? y   ASAI Routing? y
Prompting? y    LAI? y  G3V4 Adv Route? y       CINFO? n BSR? y       Holidays? y

01 consider     skill    3   pri m adjust-by 0
02 consider     skill    4   pri m adjust-by 0
03 reply-best
```

Example interflow local treatment vector for the remote Communication Manager

The following example shows a call vector that is used to interflow the call to the remote server while local treatment is provided for the call.

Important:
When the BSR Local Treatment feature is enabled, specific ISDN messages must be exchanged between the remote and local Communication Managers. If additional vector steps are included either before or after the consider steps (if used) and queue-to best in the interflow vector on the remote server, the following results occur:

- Either an ALERTING or PROGRESS message (with in-band information) is returned from the remote server to the local server.
- In response to the message, trunk bandwidth is immediately allocated and the call is removed from the local queue.
- Local treatment operations cease, trunk bearer resources are allocated for the call sooner than required and cost savings associated with the local treatment feature are not realized.

```
change vector 32                                                        Page 1 of 3
                                    CALL VECTOR

Number: 32                    Name: Remote BSR interflow vector
                 Attendant Vectoring? n          Meet-me Conf? n            Lock? n
    Basic? y    EAS? y   G3V4 Enhanced? y      ANI/II-Digits? y   ASAI Routing? y
Prompting? y    LAI? y  G3V4 Adv Route? y        CINFO? n BSR? y      Holidays? y

01 consider     skill    3   pri m adjust-by 0
02 consider     skill    4   pri m adjust-by 0
03 queue-to     best
```

Special BSR local treatment considerations

You should also understand the following items that pertain to the BSR local treatment feature:

Trunk group status

Calls that are queued remotely but are receiving local treatment are displayed as 'active' trunk members if the 'status trunk-group' command is performed on the interflowed trunk group. Even though the H.323 (IP) trunk member is 'active', no bandwidth is used because no voice packets are transmitted while local treatment is performed.

Path replacement

\ Path replacement is not supported for BSR Local Treatment calls. Both ends of the connection must be answered for path replacement to work. When BSR local treatment is enabled, the local VDN has answered, but the remote VDN where the call is queued has not answered. Therefore, path replacement can not occur when a call is queued remotely by local treatment VDNs. For more information about BSR path replacement, see BSR-initiated path replacement for calls in vector processing on page 133.

Troubleshooting for multi-site BSR

You should regularly execute a **display events** command for the appropriate vectors, especially if you have just implemented a new BSR application. Vector events will identify and indicate the source of common malfunctions and administration errors.

When tie-trunks or queue slots become exhausted, BSR cannot effectively balance calls across the network. If such problems are revealed frequently by vector events, review the design of the BSR application involved. If tie-trunks are frequently exhausted, the user adjustments on consider location steps may be set too low.

For a list of BSR vector events and definitions, see Tracking unexpected events in the *Programming Call Vectors in Avaya Aura™ Call Center* document.

Note:

Only the most recent events are displayed when a **`display events`** command is executed. For this reason, you should periodically display vector events to help quickly identify problems.

To verify that your BSR vectors are operating as intended, use a **`list trace vdn`** or **`list trace vec`** command to observe processing of an individual call. For more information, see Clearing events in the *Programming Call Vectors in Avaya Aura™ Call Center* document.

Note:

The **`list trace vdn`** and **`list trace vec`** commands are blocked if the Tenant Partitioning feature is enabled.

BSR status poll vectors must always end with a **`reply-best`** step. A **`busy`** or **`disconnect`** command should never be used.

Tips for writing BSR vectors

Before you write your first vector using BSR, you should study the sample vectors that are provided and familiarize yourself with the new commands and command elements. Sample vectors are provided in Single-site BSR and Multi-site BSR on page 103. The new commands and command elements are explained in Call Vectoring commands.

As you write BSR vectors, it is strongly recommended that you follow the guidelines below.

- Arrange your consider steps in order of preference.

  The consider step that tests the main, or preferred, resource should be the first in the series. The second consider step should test the resource that is your second preference for handling the given call type, and so on. To avoid unnecessary interflows, put consider steps for local resources before steps that consider remote resources. This arrangement also provides a local best as a backup in case the interflow fails.

  Arranging consider steps in order of preference is recommended for all BSR vectors. It is especially important when the active VDN for the call is using the 1st-found agent strategy since the server delivers the call to the first available agent found, arranging consider steps in order of preference ensures that calls are delivered to the best of the available resources and that unnecessary interflows are avoided.
- Do not put any commands between the steps of a consider series that would cause a delay. Goto commands are OK.
- Do not put a consider series in vector loops.
- Confirm that calls queue successfully.

  This check is recommended for all vectors. Since EWT is infinite for a call that has not queued, a step that checks EWT after a queue attempt is a good confirmation method.

After a queue-to best step, for example, a command such as **`goto step X if expected-wait for call < 9999`** should be included.

- Do not use the wait-improved conditional in a vector before you have queued the call at least once.

  The wait-improved conditional compares the call's EWT in its current queue to the best resource that is found by a consider series. If a call has not been queued and a vector step such as check best if wait-improved > 30 is executed, the server interprets the call's current EWT as infinite and the check best step always routes the call to the best resource. In other words, in this situation the check best step functions like an unconditional **`goto`** or **`route-to`** command.

BSR-initiated path replacement for calls in vector processing

Path replacement for calls in queue and vector processing can be accomplished using QSIG or DCS with Reroute using ISDN SSE. For calls that are waiting in queue or in vector processing, even if the call is not connected to an answering user, path replacement can be attempted to find a more optimal path for this call. This results in more efficient use of the trunk facilities.

The **`queue-to best`** command is used in BSR to initiate a QSIG path replacement for a call. The following scenario can take place:

At the terminating Communication Manager, if a Path Replacement Propose operation is received for a call that is in queue or vector processing, the server can immediately initiate path replacement using the Path Replacement Extension if the **Path Replace While in Queue/ Vectoring** field is set to`y` and the**Path Replacement Extension** field has a valid entry. These fields are located on the ISDN parameters page of the Feature-Related System Parameters screen.

The ability to track a measured ACD call after a path replacement has taken place is available for CMS versions r3v9ai.o or later. Starting with the r3v12ba.x release, CMS reports a path replacement as arename operation rather than a path replacement. There name operation properly reports scenarios where a path replacement takes place from a measured to an unmeasured trunk facility. Avaya recommends that you upgrade CMS to r3v12a.x or later and administer all trunks associated with path replacement as measured by CMS to ensure better CMS tracking of path-replaced calls.

Example BSR vector written to trigger path-replacement

The following example shows how a BSR vector can be written to trigger path-replacement at the terminating Communication Manager.

Note:

In order for a path-replacement to be attempted, the incoming and outgoing trunks that are used for the call must be administered with the **Supplementary Service Protocol** field set to `b`.

BSR-initiated path-replacement vector

```
1. wait 0
2. consider skill 1 3. consider skill 5
4. consider location 10       adjust-by 10
5. consider location 24       adjust-by 20
6. queue-to best
```

At the terminating (receiving) server, the vector that is executed by the incoming call must be programmed with an **announcement**, or **wait hearing music** vector command. The use of one of these commands is what makes it possible for path-replacement to take place while the call is in vector processing.

Alternate Selection on BSR Ties

Understanding Alternate Selection on BSR Ties

The Best Service Routing (BSR) feature compares splits or skills using a series of consider steps and selects the one that provides the best service to a call. When that comparison results in a tie (the results are equal in value), the Alternate Selection on BSR Ties determines how BSR chooses which agent, skill, or location to select. The Alternate Selection on BSR Ties chooses between:

- Skills or locations with the same expected wait time (EWT)
- Available agents that are weighted with the same criteria (most idle agent or least occupied agent) in a consider series that is designed to locate the best skill or location

You can set the Alternate Selection on BSR Ties for the system or on a per Vector Directory Number (VDN) basis. Each consider skill or location step applies the assigned strategy by comparing the current best choice from a previous consider step to the value obtained from the current consider step.

If the Alternate Selection on BSR Ties on the active VDN for the call is 1st-found, when the algorithm finds an available agent at a consider step, the processing of the remaining consider steps stops. The Alternate Selection on BSR Ties does not apply in this case.

Vector commands for single-site BSR

The following table shows the vector commands and command elements used in single-site BSR applications.

Commands and command elements		Use this …
Commands	consider split/ skill	7 to obtain the Expected Wait Time or agent data needed to identify the best local

Commands and command elements		Use this …
		resource. One **`consider`** step must be written for each skill you want to check.[7]
	`queue-to`	with the **`best`** keyword to queue calls to the best resource identified by the consider sequence.
	`check`	with the **`best`** keyword to queue calls to the best resource identified by the consider sequence if the resource meets certain conditions.
Key word	`best`	in **`queue-to, check,`** and **`goto`** commands that refer to the resource identified as best by a series of consider steps
Conditional	`wait-improved`	Prevents calls from being queued to an additional skill when the reduction in Expected Wait Time is not enough to be useful. *Wait-improved* means that a call's EWT must be improved by a specific amount (a figure you specify in seconds) over its current EWT or the communication server will not queue it to the additional skill.
User adjustment	`adjust-by`	To specify your preferences for the skills that might handle the calls for a particular application, reflecting factors such as agent expertise or reducing calls to a backup skill. When a vector considers a local resource you can make the selection of that skill less desirable. The higher the setting, the less chance that resource will be selected over another with a lower setting (for example, set to 30 makes that choice 30% less desirable). With EWT returned, the setting increases the returned expected wait time for comparison with other returned EWTs. As a result, this skill is less likely to service the call unless its EWT is significantly less than that of any other available skill. Optionally, the adjust-by setting applies in the available agent case. If you are using the UCD-MIA or EAD-MIA available agent strategy, the setting decreases the returned agent idle time, making the agent appear less idle (busier). If you are using the UCD-LOA or

[7] Since the **`consider`** command is designed to compare two or more resources, **`consider`** commands are typically written in sequences of two or more with the sequence terminating in a **`queue-to best`** step. This set of **`consider`** commands and a **`queue-to best`** step is called a consider series.

Commands and command elements		Use this …
		EAD-LOA available agent strategy, the setting increases the returned agent occupancy, making the agent appear more occupied (busier). In either case with EAD, the MIA or the LOA is used as a tie breaker if more than one site has an agent available with the same highest skill level.

Vector commands for multi-site BSR

The following table summarizes the vector commands and command elements that support multi-site BSR applications.

Commands and command elements		Use this…
Commands	`consider split/skill`	[8] To obtain the Expected Wait Time or agent data needed to identify the best local resource. One **`consider`** step must be written for each skill you want to check.[8]
	`consider location`	To obtain the Expected Wait Time or agent data needed to identify the best resource at a remote communication server. One **`consider`** step must be written for each location you want to check. Routing information is obtained from the BSR Application plan for the active VDN.
	`reply-best`	To return data to another communication server in response to a status poll.
	`queue-to`	With the **`best`** keyword to queue calls to the best resource identified by the consider sequence.
	`check`	With the **`best`** keyword to queue calls to the best resource identified by the consider sequence if the resource meets certain conditions.
Key word	`best`	In **`queue-to, check,`** and **`goto`** commands that refer to the resource identified as best by a series of consider steps
Conditional	`wait-improved`	To prevent calls from being queued to an additional skill - local or remote - when the reduction in Expected Wait Time is not enough to be useful. *Wait-improved* means that a call's EWT must be improved by a specific amount (a figure

[8] Since the **`consider`** command is designed to compare two or more resources, **`consider`** commands are typically written in sequences of two or more with the sequence terminating in a **`queue-to best`** step. This set of **`consider`** commands and a **`queue-to best`** step is called a consider series.

Commands and command elements		Use this…
		you specify in seconds) over its current EWT or the communication server will not queue it to the additional skill.
User adjustment	`adjust-by`	To control long-distance costs and limit trunk usage, reflecting factors such as availability of the trunks or agent expertise at remote locations. When a vector polls a local or remote resource, you can make the selection of that site less desirable. The higher the setting, the less chance that resource will be selected over another with a lower setting. With EWT returned, the setting increases the returned expected wait time for comparison with other returned EWTs. Optionally, the adjust-by setting applies in the available agent case. If you are using the UCD-MIA or EAD-MIA available agent strategy, the setting decreases the returned agent idle time, making the agent appear less idle (busier). If you are using the UCD-LOA or EAD-LOA available agent strategy, the setting increases the returned agent occupancy, making the agent appear more occupied (busier). In either case with EAD, the MIA or the LOA is used as a tie breaker if more than one site has an agent available with the same highest skill level.

BSR considerations

- If one or more of the resources considered have an available agent, the resources with EWT are ignored. This means that there is an agent surplus.
- If the available agent strategy (assigned to the active VDN) is 1st-found, the adjust-by is ignored and the first consider with an available agent is used for the queue-to best.
- If the available agent strategy is UCD-MIA, EAD-MIA, UCD-LOA, or EAD-LOA and there is more than one consider step with an available agent, then adjust-by is applied as part of the algorithm to select the best of the possible choices.

BSR interactions

Agent Telephone Display

If collected digits are forwarded with an interflowed call, the forwarded digits are displayed on the answering agent's telephone display (unless they're overridden with newly collected digits).

Best Service Routing (BSR)/LAI

Restrictions and interactions that apply to LAI also apply to BSR status poll and interflow calls. See Interflow and Intraflow interactions on page 250 for more information.

BCMS

BCMS does not report accumulated in-VDN time. BCMS does not log LAI attempts and therefore will not log BSR status polls, which are treated as LAI attempts.

Call Vectoring:

The following considerations apply to ALL vectors when BSR is enabled on your communication server.

Call Vectoring considerations when BSR is enabled	
route-to VDN	If a call is routed to a new VDN, any best resource data defined by a series of consider steps in the previous VDN will be initialized (cleared)
goto vector	If a **goto vector** command is executed, any best resource data produced by a series of consider steps in the original VDN will remain with the call and can be used in the subsequent vector.
consider	• Do not use other commands within a series of **consider** steps, since these may delay the execution of the series. • Skills used in **consider** commands must be vector controlled.
converse	Collected digits forwarded with the call will be passed to VRU using the digits data passing type.
best (keyword)	The **best** keyword can be used in the following commands, but only with the conditionals listed: • **goto** step or **goto vector** commands using the **expected-wait** or **wait-improved** conditionals • **check** commands using the **unconditional**, **expected-wait**, or **wait-improved** conditionals The **best** keyword can not be used as a replacement for split or skill in the following vector commands: • **converse-on split/skill** • **messaging split/skill**

Direct Department Calling

BSR will function when the considered splits use DDC call distribution. Once the best resource is determined, the actual call distribution will follow the split's DDC setting regardless of the BSR Available Agent Strategy. DDC may not be used as a BSR Available Agent Strategy.

Distributed Networking using QSIG Manufacturers Specific Information (MSI):

BSR will not function with systems from other vendors (unless that vendor develops a corresponding capability that works with the Avaya communication server).

Expert Agent Selection

EAS is required to use the EAD-MIA or EAD-LOA Available Agent Strategy. EAS VDN skills (1st, 2nd, 3rd) can be used in consider skill commands.

Facility Restriction Levels

The FRL applies to status poll and interflow calls in the same way it works with the route-to number command.

ISDN

Best Service Routing and globally supported information transport are fully functional over ISDN PRI or ISDN BRI trunking facilities.

Note:

Asynchronous Transfer Mode (ATM) trunking and IP trunking can be set up to emulate ISDN PRI. For information on setting this up, see Administration for Network Connectivity for Communication Manager, and ATM Installation, Upgrades and Administration using Communication Manager.

Location Preference Distribution:

Local Preference Distribution is used to select an available agent within the call center during **`consider`** and **`queue-to best`** step operations. Local Preference Distribution is not used across system sites. In this case, there is no notion of a multi-site network region.

Look Ahead Routing (LAR) - BSR incompatibility:

Look Ahead Routing (LAR) and BSR are incompatible. If a trunk is not available at the site being polled, an alternative route (as a secondary route using an ARS pattern) can be used to poll, assuming there is a secondary route available that supports transporting shared UUI in the DISconnect message. This does not use LAR. If no route is available for polling when a consider location step is executed, then BSR processing handles the situation and after a period of 30 seconds, subsequent calls will try to poll that location again.

The use of alternative routes for polling only works if there are alterative routes for the interflow path, regardless of whether LAR or BSR is in use.

Multi-Split/Skill Queuing:

A call may be queued up to three times by **`queue-to`** or **`check`** commands in the same vector. One vector may therefore contain up to three series of **`consider`** steps. Each series must be followed by a **`queue-to best`** step. Each consider series will select the best remote resource from the options you specify and queue the call to that resource.

BSR can only queue simultaneously on the origin communication server. BSR gives up control of a call once it queues the call at a remote resource.

A call may be queued up to three times by **`queue-to`** or **`check`** commands in the same vector. One vector may therefore contain up to three series of **`consider`** steps. Each series must be followed by a **`queue-to best`** step. Each consider series will select the best remote resource from the options you specify and queue the call to that resource.

Network Access:

BSR interflow operates over public, private, or virtual private (for example, SDN) ISDN-BRI and ISDN-PRI networks that meet the criteria explained in Network requirements on page 90. BSR interflow can also operate over SIP trunks. Best Service Routing requires that the network support transport of user-to-user data using MSI or UUI as a codeset 0 Information Element, or as shared user-to-user information over SIP trunks. The numbers administered on the BSR Application Plan screen are expected to access VDNs using ISDN, H.323, or SIP trunks, or using the Network Call Redirection (NCR) feature.

Administration or call processing will not prevent access to other types of trunks or to destinations that are not VDNs. However, BSR is only intended to support the types of applications described in this section. Attempts to use the BSR feature for any other purposes may not work.

Operating Support System Interface (OSSI):

The new administration commands, conditionals, keywords and forms are available using OSSI.

Path replacement for QSIG/DCS ISDN calls

For calls that are waiting in queue or in vector processing, even if the call is not connected to an answering user, path replacement can be attempted to find a more optimal path for this call. This results in more efficient use of the trunk facilities.

The QSIG ISDN or DCS ISDN trunk path-replacement operation can be triggered for ACD calls by the Look-Ahead Interflow **`route-to number`** vector step, BSR **`queue-to best`** vector step, and the Adjunct Routing vector steps.

The ability to track a measured ACD call after a path replacement has taken place is available for CMS versions r3v9ai.o or later. Starting with the r3v12ba.x release, CMS reports a path replacement as a *rename* operation rather than a path replacement. The *rename* operation properly reports scenarios where a path replacement takes place from a measured to an unmeasured trunk facility. Avaya recommends that you upgrade CMS to r3v12a.x or later and administer all trunks associated with path replacement as *measured* by CMS to ensure better CMS tracking of path-replaced calls.

Note:

Path replacement is not supported for BSR Local Treatment calls. Both ends of the connection must be answered for path replacement to work. When BSR local treatment is enabled, the local VDN has answered, but the remote VDN where the call is queued has not answered. Therefore, path replacement can not occur when a call is queued remotely by local treatment VDNs.

For more information on path replacement, see BSR-initiated path replacement for calls in vector processing on page 133, About path replacement for calls in vector processing on page 279, and Adjunct routing-initiated path replacement on page 441.

QSIG

LAI, BSR, and information forwarding function over QSIG trunk facilities if the remote locations are Avaya communication servers.

Redirection on No Answer (RONA):

Calls redirected to a VDN by RONA can be subsequently processed by BSR or LAI applications. When the RONA feature redirects a call to a VDN, any best resource data defined in a previous vector will be initialized (cleared).

SLM:

The following interactions occur between BSR and SLM:

- The SLM algorithm applies only within a particular call center location, not across locations in a multi-site configuration
- Assignment of reserve agents applies only to skills within a local site.
- SLM always selects the agent for an SLM skill at the remote site. BSR uses SLM to determine the best available agent and when to route the call to that skill.
- The best skill selected at a particular site or across sites when due to multiple consider steps is based on an existing BSR operation. In other words, the shortest adjusted EWT or skill as defined by the available agent strategy.
- The selection of the agent, and delivery of the call in the best-chosen skill, is based on what is assigned to the skill.
- BSR does not override the skill distribution algorithms and pick a reserve agent unless the skill distribution algorithm selects that agent due to the current conditions at that site.

Service Observing:

You can observe a call in BSR or LAI processing as long as the call is still connected through the local communication server. All current restrictions on Service Observing still apply.

Transfer

If a call is transferred to a VDN, any best resource data defined in previous vector processing will be initialized (cleared). Transferred calls do not forward any of the information that is forwarded with interflows (previously collected digits, In-VDN time, etc.).

Trunk Access Code (TAC)

Use of routing numbers (status poll or interflow) that utilize TACs is not recommended since the required in-band outpulsing slows the setup operation significantly.

VDN Override:

VDN Override applies to the BSR Application Number and the Available Agent Strategy option assigned on the VDN screen. It also applies to the VDN name forwarded using Information Forwarding. When a consider step is executed, the application number and available agent strategy assigned to the active VDN for the call will be used.

VDN Return Destination

The best resource data for a call is initialized when the call first leaves vector processing and therefore will not be available should the call return to vector processing.

VuStats

No enhancements have been added for BSR.

Call Prompting

About Call Prompting

Call Prompting provides flexible call handling that is based on information that is collected from a calling party. This information is in the form of dialed digits that originate from an internal or external touch-tone telephone or from an internal rotary telephone that is on the same switch as the vector. Call Prompting allows for the temporary transfer of call management control to the caller.

With Call Prompting and Vectoring enabled, the switch can collect caller entered digits (ced) and customer database provided digits (cdpd) that are supplied by the network. The system can receive Call Information Forwarding (CINFO) digits in an incoming call's ISDN message when the AT&T Network Intelligent Call Processing (ICP) service is in use. A switch can collect digits and forward those digits to other switches by way of interflow commands. For more information, see Caller Information Forwarding in the *Programming Call Vectors in Avaya Aura™ Call Center* document.

With Voice Response Integration (VRI), digits can be returned to the switch by a Voice Response Unit (VRU) script that is accessed by a **`converse-on split`** command. Such digits can also be used for call management.

Call Prompting can be used in various applications so that calls can be handled with more flexibility.Call Prompting uses specialized vector commands to process incoming calls based on information collected from the caller or from an ISDN-PRI message. It can be used in various applications to better handle incoming calls.

The following list gives a brief description of some Call Prompting applications.

- Automated Attendant - Allows the caller to enter the extension of the party that he or she would like to reach. The call is routed to that extension.
- Data In/Voice Answer (DIVA) Capability - Allows the caller to hear an announcement based on the digits that he or she enters, or to be directed to a hunt group or another system extension.
- Data Collection - Allows the caller to enter data that can be used by a host/adjunct to assist in call handling. This data, for example, may be the caller's account number.
- CINFO (Caller Information Forwarding) Routing - Allows a call to be routed based on digits supplied by the network in an ISDN-PRI message.
- Message Collection - Gives the caller the option of leaving a message or waiting in queue for an agent.

Call Prompting considerations

Call prompting, with the exception of CINFO, competes with several features for ports on the call classifier - detector circuit pack or equivalent.

Call Prompting interactions

The following interactions apply specifically to Call Prompting. For general Call Vectoring interactions that may affect Call Prompting applications see Call Vectoring.

Authorization Codes

If authorization codes are enabled, and a **`route-to`** command in a prompting vector accesses AAR or ARS, if the VDN's FRL does not have the permission to use the chosen routing preference, then the system does not prompt for an authorization code and the **`route-to`** command fails.

CallVisor ASAI

ASAI-provided digits can be collected by the Call Vectoring feature using the **`collect`** vector command as dial-ahead digits. CINFO is passed to CallVisor ASAI.

Hold

With the exception of CINFO, if a call is put on hold during the processing of a **`collect`** command, the command restarts, beginning with the announcement prompt, when the call is taken off hold. All dialed-ahead digits are lost. Similarly, if a call to a vector is put on hold, vector processing is suspended when a **`collect`** command is encountered. When the call becomes active, the **`collect`** command resumes.

Inbound Call Management (ICM)

You can use Call Prompting to collect information that may later be used by an adjunct to handle a call.

Transfer

If a call to a VDN is transferred during a **`collect`** command, the **`collect`** command restarts when the transfer is complete, and all dialed-ahead digits are lost. Similarly, if a call to a vector is transferred, vector processing is suspended when a **`collect`** command is encountered. When the transfer is complete, the **`collect`** command resumes. This is not true when a **`collect`** command collects CINFO digits. In this case vector processing is not suspended. Attendant extended calls do suspend vector processing in the same way as transferred calls.

Call Prompting command set

The following table show the commands that are used for Call Prompting.

Table 11: Call Prompting command set

Command category	Action taken	Command
Information collection	Collect information from the calling party, from the public network in an ISDN SETUP message, from a Voice Response Unit (VRU), or from CallVisor Adjunct Switch Application Interface (ASAI).	**`collect digits`**
Treatment	Play an announcement. Delay with audible feedback of silence, ringback, system music, or an alternate audio/music source.	**`announcement`** **`wait-time`**
Routing	Leave a message. Route the call to a number that is programmed in the vector. Route the call to digits that are supplied by the calling party.	**`messaging split`** **`route-to number`** **`route-to digits`**
Branching/ programming	Go to a vector step. Go to another vector. Stop vector processing.	**`goto step`** **`goto vector`** **`stop`**

Touch-tone collection requirements

Before the switch can accept the touch-tone digits that are entered by a caller, the switch must be equipped with a collection resource. The resource used for collecting and interpreting touch-tone digits is a unit of hardware called a Touch-Tone Receiver (TTR). These TTRs are provided on the call classifier and tone detector circuit packs, one of which is required for Call Prompting.

The number of TTRs that are required is configured according to two sources:

- Customer input to the Avaya Account Team
- Account team input to the configurator tool

For existing systems that are adding a Call Prompting application, the Account Team recommends the appropriate number of TTRs based on two factors:

- Account team input to the configurator tool
- Application review by the Avaya Design Center

The process of collecting CINFO digits does not require TTRs.

Outside callers must have a touch-tone telephone to enter the digits that are requested by the **`collect digits`** command. For callers who are using rotary dialing, the Call Prompting timeout takes effect, the **`collect digits`** command times out, and vector processing continues at the next step. As a precaution, always provide a default treatment, such as a

route-to attendant command or a **queue-to split** command, in the vector script unless the script is created exclusively for users of touch-tone telephones.

Note:
The **Call Prompting interdigit timeout** can be administered for any number of seconds from 4 to 10. This value is administered on the Feature-Related System Parameters screen.

Provisions for users of rotary telephones are illustrated in the vector scripts in this section.

Call Prompting digit entry - collect digits command

About the collect digits command

The touch-tone digits that are entered by a Call Prompting user are collected by the **collect digits** command. This command allows the system to collect up to 24 digits from a touch-tone telephone. Sixteen of these digits may be collected immediately, while any remaining digits are stored as dial-ahead digits, which are explained later.

Call Prompting allows some flexibility in entering digits. Specifically, the caller can:

- Remove incorrect digits strings
- Enter variable-length digit strings
- Enter dial-ahead digits.

The following sections explain these processes.

Removing incorrect digit strings

An announcement that requests the caller to enter digits can be included in call treatment. As an option, the announcement can instruct the caller to enter an asterisk (*) if he or she enters incorrect data.

When the caller enters a *, the following happens:

1. Digits that were collected for the current **collect digits** command are deleted.

Note:

Also deleted are any dial-ahead digits that are entered and that do not exceed the maximum digit count of 24. (Dial-ahead digits are explained later.)

2. Digit collection is restarted.
3. The announcement is not replayed.

Result

Once the caller enters an asterisk, the caller can reenter digits for processing.

Variable-length digit strings

The maximum number of digits that are requested from the caller must be specified in the administration of the **`collect digits`** command. In some cases, the caller might be permitted to enter fewer digits than the maximum specified. In fact, the number of digits that the caller enters can vary for several variations of one **`collect digits`** command. Each such grouping of digits is called a variable-length digit string.

Call Prompting allows for variable-length digit strings by providing an end-of-dialing indicator in the form of the pound sign (#). The pound sign is used to end any digit string that is entered by the caller, and it does the following:

- Tells the system that the caller has finished entering digits
- Causes the next vector step to be processed immediately.

Whenever the caller is permitted to enter a variable-length digit string, the announcement portion of the **`collect digits`** command should specify the largest possible number of digits that can be entered. Accordingly, each **`collect digits`** command should be administered to collect no more than the intended maximum number of digits. The caller can enter a pound sign part of a variable digit string entry either:

- At the end of each variable digit string that is entered. In this case, the pound sign should be included in the count of the number of maximum digits that can be entered.
- At the end of each such string that, not counting the pound sign, contains fewer characters than the maximum number of allowable digits. In this case, the pound sign should not be included in the count of the number of maximum digits that can be entered.

If the caller enters more digits than the maximum number specified, the additional digits are saved as dial-ahead digits for subsequent **`collect digits`** commands. If the vector or vectors chained to it do not contain another **`collect digits`** command, the extra digits are discarded.

If the caller enters fewer digits than the maximum number specified and does not complete the entry with the pound sign, a Call Prompting timeout occurs. The timeout terminates the command, and any digits collected prior to the timeout are available for subsequent vector processing. The Call Prompting timeout period is set to 10 seconds by default but can be changed to a value between 4 and 10 seconds using the Prompting Timeout field on the Feature-Related Customer-Options screen. See collect digits command in the *Programming Call Vectors in Avaya Aura™ Call Center* document for detailed information.

A common application involving the entering of variable-length digit strings allows the user to dial either the number for the attendant or an extension to reach the desired destination. If the maximum number of digits that can be entered is administered to be 3 and the user wishes to reach the attendant, the user should dial 0#. However, if the user chooses to dial a 3-digit extension, the user should dial, for example, 748 and not 748#. Since the maximum number of digits that can be dialed in this case is three, dialing 748# would cause # to be saved as a dial-ahead digit. On the other hand, if the caller dials 748#, and if the maximum number of digits that can be entered is 4, # is not saved as a dial-ahead digit since it is the fourth of four digits that can be entered in this case.

Dial-ahead digits

When digit collection for the current **`collect digits`** command is completed, vector processing continues at the next vector step. However, the switch continues to collect any digits that the caller subsequently dials until the TTR disconnects. For more information, see Collecting Digits on the switch in the *Programming Call Vectors in Avaya Aura™ Call Center* document. These dialed-ahead digits are saved for processing by subsequent collect digits commands. Dial-Ahead Digits are explained fully in Dial-ahead digits - collect digits command.

Functions and examples

Call Prompting uses some of the functions found in Basic Call Vectoring. Call Prompting also provides some additional functions that involve digit processing. These functions are included in the following sections:

- Treating digits as a destination on page 147
- Using digits to collect branching information on page 148
- Using digits to select options on page 150
- Displaying digits on an agent set on page 151
- Passing digits to an adjunct on page 152
- Creating Service Observing vectors on page 153

Treating digits as a destination

Call Prompting allows you to route calls according to the digits that are collected from the caller. Once the digits are collected by the **`collect digits`** command, the **`route-to digits`** command attempts to route the call to the destination that the digits represent. The command always routes the call to the destination that is indicated by the digits that are processed by the most recent collect digits command.

The digits can represent any of the following destinations:

- Internal (local) extension, for example, split/hunt group, station, and announcement
- VDN extension
- Attendant
- Remote access extension
- External number, such as a trunk access code (TAC) or an Automatic Alternate Route/ Automatic Route Selection (AAR/ARS) feature access code (FAC) followed by a public network number, for example, 7-digit Electronic Tandem Network (ETN), 10-digit DDD.

The following example shows how a call is routed by digits that are collected from a caller.

Using Call Prompting to route by collected digits

```
1. wait-time 0 seconds hearing ringback
2. collect 5 digits after announcement 300 [
You have reached Redux Electric      in Glenrock. Please dial a 5-digit extension
or wait for the attendant.
]
3. route-to digits with coverage y
4. route-to number 0 with cov n if unconditionally
5. stop
```

In this vector, the caller is prompted to enter the destination extension of the party that he or she would like to reach (step 2). The extension in this vector may contain up to 5 digits. The vector collects the digits and then routes to the destination by the **`route-to digits`** command in step 3.

If the **`route-to digits`** command fails because the caller fails to enter any digits, or because the extension number entered is invalid, the **`route-to number`** command in step 4 routes the call to the attendant, which is the default routing option. However, as long as the destination is a valid extension, the **`route-to digits`** command succeeds, coverage applies, and vector processing terminates. If the destination is busy, vector processing terminates because coverage call processing takes effect.

Note:

Occasionally, all of the system's TTRs might be in use. As a result, when you are collecting digits from a caller, you should avoid starting your main vector with a **`collect digits`** command, since the caller in this case receives no audible feedback if he or she has to wait for a TTR to become available. Accordingly, it is a good practice to include some treatment, for example, wait-time 0 seconds hearing ringback, before the initial collect digits step.

In addition, if calls are likely to be transferred to this vector, a wait-time step of sufficient length is recommended before the collect step to allow the transferring party enough time to complete the transfer.

Using digits to collect branching information

Call Prompting allows you to direct a call to another step or vector based on the digits that are entered by the caller. This branching is accomplished with a goto step. For example, in the following vector example, digits are used to route calls to different vectors based on an assigned customer number.

Using Call Prompting to branch by collected digits

```
1. wait-time 0 seconds hearing ringback
2. collect 5 digits after announcement 200 [
Please enter your customer number.
]
3. goto vector 8 if digits = 10+
4. goto vector 9 if digits = 11+
5. goto vector 10 if digits = 12+
6. route-to number 0 with cov n if unconditionally
7. stop
```

The wildcard + indicates that the two digits can be followed by zero or any number of additional digits. Callers with a number that begins with the digits 10 are routed to vector 8, callers with a number that begins with the digits 11 are routed to vector 9, and callers with a number that begins with the digits 12 are routed to vector 10.

Vector Routing Tables

You also can test digits against entries in a Vector Routing Table.

Vector Routing Tables contain lists of numbers that can be used to test a **`goto...if digits`** command. Digits that are collected with the collect digits step can be tested to see if they are either in or not-in the specified table. Entries in the tables can include either the + or ? wildcard.

- The + represents a group of digits and can only be used as the first or last character of the string.
- The ? represents a single digit. Any number of them can be used at any position in the digit string.

Tables are entered on the Vector Routing Table screen. For complete instructions for creating Vector Routing Tables, see *Administering Avaya Aura™ Call Center Features*.

The following Call Vector example could be used to test against the numbers provided in the Vector Routing Table.

Testing for digits in Vector Routing Table

```
1. wait-time 0 seconds hearing ringback
2. collect 7 digits after announcement 200 [
Please enter your account      number.
]
3. goto vector 8 if digits in table 10
4. queue-to split 5 pri l
5. wait-time 10 seconds hearing ringback
6. announcement 2771
7. wait-time 10 seconds hearing music
8. goto step 6 if unconditionally
```

If the caller enters an account number that is listed in the Vector Routing Table, the call is routed to vector 8. If the caller enters an account number that matches the wildcard entry (for example 1345987), the call is routed to vector 8.

If the caller enters an account number that is not listed in the Vector Routing Table, or if the caller does not enter an account number, the call is queued to split 5.

Suppose that, instead of containing a list of premier accounts, the Vector Routing Table contains a list of accounts with a poor payment record. The following example shows a vector that only queues calls with account numbers that are not in the table. Calls in the table route to the collection department.

Testing for digits not in Vector Routing Table

```
 1. wait-time 0 seconds hearing ringback
 2. collect 7 digits after announcement 200 [
Please enter your account      number.
]
 3. goto step 11 if digits = none
 4. goto step 6 if digits not-in table 10
 5. route-to number 83456 with cov y if unconditionally      [collections]
 6. queue-to split 5 pri l
 7. wait-time 10 seconds hearing ringback
 8. announcement 2771
 9. wait-time 10 seconds hearing music
10. goto step 8 if unconditionally
11. route-to number 0 with cov n if unconditionally
12. stop
```

If no digits are collected, the call is routed to the operator.

Note:

Entries in Vector Routing Tables also can be tested against the telephone number of the caller Automatic Number Identification (ANI). For more information, see ANI /II-digits routing and Caller Information Forwarding (CINFO) in the *Programming Call Vectors in Avaya Aura™ Call Center* document.

Using digits to select options

Call Prompting makes it possible to provide a menu of options that the caller can use to satisfy his or her information needs. The caller selects the desired option by entering the appropriate requested digit. Once the digit is entered, a conditional branch to the appropriate treatment is made. The treatment is usually provided by the **`route-to number`** command.

The following example shows how digits are used to select options.

Using Call Prompting to select options

```
1. wait-time 0 seconds hearing ringback
2. collect 1 digits after announcement 3531 [
Thank you for calling Bug Out
      Exterminators. If you wish to learn about the services we provide,      please
      dial 1. If you would like to set up an appointment for one of our
representatives
 to visit your home or place of business, please dial 2.
]
3. route-to number 4101 with cov y if digit = 1
4. route-to number 4102 with cov y if digit = 2
5. route-to number 0 with cov n if unconditionally
6. disconnect after announcement none
```

In step 2 of this vector, the user is asked to enter either 1 or 2, depending on the service he or she uses. If one of these digits is entered, the appropriate one of the next two steps (3 through 4) routes the call to the relevant extension, that is, either 4101 or 4102. If one of the digits is not entered, the call is routed to the attendant (step 5).

Displaying digits on an agent set

A CALLR-INFO button can be included at the agents' display stations to help process calls that are serviced by the Call Prompting feature. However, if the agent has a two-line display set, and the display is in normal or inspect mode, the collected digits are automatically displayed on the second line. As a default, these digits remain on this line until they are overwritten, even after the call is released by the agent. As an option, administrators can decide when an agent's station display is cleared of caller information. For more information, see Clearing caller information from the station display on page 152. For other display sets, the agent must press the CALLR-INFO button to display the collected digits.

It may be beneficial to install the CALLR-INFO button if you want to expedite calls by reducing the amount of time agents spend on the telephone. For example, the button could be set up to collect specific information such as a customer account number before the call is answered by the agent, thus eliminating the need for the agent to ask for this information.

The CALLR-INFO button displays information in the following format:

x = Info: 1234567890

where:

- x is a call appearance letter, for example, a, b, c, and so forth
- 1234567890 represents the digits that are collected from the caller

The digits that are entered by the caller are collected by the most recent collect digits command. Any digits that were dialed ahead and not explicitly requested by the most recently executed **`collect digits`** command are not displayed.

Assume that digits have been collected by Call Prompting. If the agent presses the CALLR-INFO button when the call rings at the agent station or when the station is active on a call appearance, the following events occur:

- The 10-second timer for display interval is set.
- The status lamp (if available) that is associated with the button is lit.
- The display is updated. Specifically, the incoming call identification (calling party ICI) is replaced with the collected digits in the format that was presented earlier in this section. Only those digits that were collected for the last **`collect digits`** command are displayed.

If all the conditions to use the button (except for the collection of digits) are set, and the agent presses the button, the status lamp (if available) that is associated with the button flashes denial.

One or more events may occur during a successful execution after the button is pushed. These events include the following:

- The 10-second timer times out.
- The incoming call arrives at any call appearance.
- An active call changes status, for example, another caller is added to the conference.

If any of these events occur, the following takes place:

- The status lamp (if available) that is associated with the button is turned off.
- The display is updated as previously described.

Note:
If the agent needs to display the collected digits again, the CALLR-INFO button can be pressed again to repeat the operation that is described in this section, provided that the agent is active on the call or the call is still ringing. Also, the agent can flip between the collected digits and the ICI by alternately pressing the CALLR-INFO and NORMAL buttons.

Clearing caller information from the station display

Administrators can decide when an agent's station display is cleared of caller information. Options include:

- Clearing the existing call information when the next call is received
- Clearing the existing call information when the call is released - whether the agent enters After Call Work (ACW) or not
- Clearing the existing call information when the agent leaves ACW mode or if the agent does not enter ACW, when the call is released

For more information, see Callr-info display options on page 61.

Passing digits to an adjunct

Call Prompting allows for the passing of information in the form of collected digits to an adjunct for further processing. Digits are passed to the adjunct by the ASAI Adjunct Routing capability.

An adjunct is any processor that is connected to a switch by the ASAI link. The adjunct makes a routing decision using the **`adjunct routing link`** command according to caller information and/or agent availability, and it returns the routing response to the switch. For example, the adjunct can indicate that the call be routed to a specific ACD agent. This is known as Direct Agent Calling (DAC).

A maximum of 16 Call Prompting digits from the last **`collect digits`** command can be passed to the adjunct by the **`adjunct routing link`** command.

The following example, shows how Call Prompting digits are passed to an adjunct.

Using Call Prompting to pass digits to an adjunct

```
1. wait-time 0 seconds hearing ringback
2. collect 10 digits after announcement 300 [
Please enter your 10-digit      account
```

```
   number.
]
3. adjunct routing link 15
4. wait-time 10 seconds hearing music
5. route-to number 52000 with cov y if unconditionally
6. stop
```

In step 2 of this vector, the caller is asked to enter a 10-digit account number. Once the account number is entered, the adjunct receives this information from the **adjunct routing link** command in step 3. This command then makes the appropriate routing decision if it is able to do so. If the command succeeds within the specified wait time, the command routes the call to the appropriate destination, and the call leaves vector processing. If the command fails, vector processing continues at the next step.

In addition to the Adjunct Routing capability, collected digits also can be passed by way of ASAI to an adjunct by prompting for the digits in one vector and then routing the call to a VDN that is monitored by an Event Notification (VDN) association. The collected digits (up to 16) are sent to the adjunct in a Call Offered to Domain Event Report. For detailed information, see Communication Manager CallVisor ASAI Technical Reference.

Note:

Adjunct Routing is fully discussed in Adjunct (ASAI) Routing.

Creating Service Observing vectors

Service Observing vectors can be constructed to allow users to observe calls from a remote location or local station. When combined with Call Prompting, Service Observing vectors can route calls to a:

- Remote access Service Observing vector example on page 153
- User-entered FAC and extension example on page 153
- Pre-programmed FAC and extension example on page 154

Remote access Service Observing vector example

The following vector example connects a user to Remote Access. Once connected, the user can dial either a listen-only or listen/talk Service Observing FAC followed by the extension number to be observed. Although it is not required, Call Prompting increases security by providing passcode protection with remote service observing.

Remote access Service Observing vector

```
1. wait-time 0 secs hearing ringback
2. collect 5 digits after announcement 2300 [
Please enter your 5-digit      security
  code.
]
3. goto step 5 if digits = 12345 (security code)
4. disconnect after announcement 2000
5. route-to number 5000 with cov n if unconditionally
6. stop
```

User-entered FAC and extension example

The following vector example connects a user directly to the Service Observing FAC and extension based on the digits that are collected by Call Prompting.

Service Observing vector with user-entered FAC and extension

```
1. wait-time 0 secs hearing ringback
2. collect 5 digits after announcement 2300 [
Please enter your 5-digit      security
  code.
]
3. goto step 5 if digits = 12345 [security code]
4. disconnect after announcement 2000
5. wait-time 0 seconds hearing ringback
6. collect 6 digits after announcement 3245 [
Please enter the number 11 for
     listen-only observing or the number 12 for listen/talk observing     followed
by
 the number of the extension you would like to observe.
]
7. route-to digits with coverage n
8. stop
```

Pre-programmed FAC and extension example

The following example shows a vector that connects a user to a pre-programmed FAC and extension using Call Prompting to allow the observer to select the extension that he or she wants to observe. In this example, the observer will be Service Observing a VDN.

Service Observing vector with programmed FAC and extension

```
1. wait-time 0 secs hearing ringback
2. collect 5 digits after announcement 2300 [
Please enter your 5-digit      security
   code.
]
3. goto step 5 if digits = 12345 [security code]
4. disconnect after announcement 2000
5. wait-time 0 seconds hearing ringback
6. collect 1 digits after announcement 2310 [
Enter 1 to observe sales, 2 to
      observe billing.
]
7. route-to number 113001 with cov n if digit = 1 [11 = listen-only observe,     3001
 = Sales VDN]
8. route-to number 113002 with cov n if digit = 2 [11 = listen-only observe,     3002
 = Billing VDN]
9. goto step 6 if unconditionally
```

Dial-ahead digits - collect digits command

About dial-ahead digits

Dial-ahead digits provide the caller with a means of bypassing unwanted announcement prompts on the way to acquiring the information or servicing he or she wants. These digits are available for use only by subsequent **collect digits** commands. The digits are never used by other vector commands that operate on digits, for example, **route-to digits**, and **goto...if digits**, until they are collected. These digits are not forwarded with interflowed calls. In addition, these digits are not displayed as part of the CALLR-INFO button operation until they are collected by a **collect digits** command.

Collection of dial-ahead digits continues until one of the following occurs:

- Vector processing stops or is terminated.
- The sum of the digits collected for the current **collect digits** command plus the dial-ahead digits exceeds the switch storage limit of 24. Any additional digits are discarded until additional storage is made available by a subsequent **collect digits** command.

  Note:
  Any asterisk (*) and pound sign (#) digits that are dialed ahead count toward the 24 digit limit, as do any dial-ahead digits that are entered after the asterisk or pound sign digit.
- The TTR required by the user to collect digits is disconnected. This happens whenever one of the following conditions is true:
  - A successful or unsuccessful **route-to number** step is encountered during vector processing, except where the number routed to is a VDN extension.
  - A successful or unsuccessful **route-to digits** step is encountered during vector processing, except where the number routed to is a VDN extension.
  - A successful or unsuccessful **adjunct routing link** step is encountered during vector processing.
  - A successful or unsuccessful **converse-on** step is encountered during vector processing.
  - A Call Prompting timeout occurs, during which time the caller has not dialed any additional digits, asterisks (*) or pound signs (#).
  - Vector processing stops or is terminated.
  - A successful or unsuccessful collect ced/cdpd step is encountered.

Note:
When the TTR is disconnected due to a **route-to number**, **route-to digits**, **converse-on**, **adjunct routing link** , or **collect ced/cdpd** step, all dial-ahead digits are discarded. This means that following a failed **route-to**, **converse**, or **adjunct routing link** step, a subsequent **collect digits** step always requires the user to enter digits.

Dial-ahead digit vector examples

The vectors shown in the following examples illustrate a situation where a caller can enter dial-ahead digits. In this case, the caller is required to have a touch-tone telephone. An alternative handling sequence should be programmed in case the caller has a rotary telephone or the caller does not dial a touch tone digit before the timeout period.

Step 2 of Vector 30 gives the caller two options, each of which provides different information. The caller is prompted to enter either 1 or 2, depending on what information he or she wants to hear. Once the caller enters a digit, the digit is collected by the **`collect digits`** command. Thereafter, an attempt is made by the **`route-to number`** command to route the call to the appropriate vector (step 3 or 4). If the caller enters a digit other than 1 or 2, the appropriate announcement is provided (step 5), and the digit entry cycle is repeated (step 6).

If the caller enters 1, Vector 31 is accessed.

Using dial-ahead digits to bypass announcements, example 1

```
VDN (extension=1030   name=‘‘Coastal’’   vector=30)
Vector 30:
1. wait-time 0 seconds hearing ringback
2. collect 1 digits after announcement 3000 [
Thank you for calling Coastal    League
Baseball Hotline. You must have a touch-tone telephone to use this    service. If you
wish to hear the scores of yesterday’s games, please press 1.  If you wish to hear
today’s schedule of games, please press 2.
]
3. route-to number 1031 with cov y if digit = 1
4. route to number 1032 with cov y if digit = 2
5. announcement 301 [
Entry not understood. Please try again.
]
6. goto step 2 if unconditionally
```

In step 1 of Vector 31 (below), the caller is given three options that supplement the original option that was provided in Vector 30. The caller is prompted to enter either 3, 4, or 5, depending on what information he or she wants to hear. If the caller enters an incorrect digit, the customary digit correction routine is implemented (steps 5 and 6). Once an appropriate digit is entered, the call is routed, in this example by a **`goto step`** command (step 2, 3, or 4), to the appropriate announcement (step 7 or step 9).

In step 10 of Vector 31, the caller is prompted with the choice of returning to the main menu provided in Vector 30 or of terminating the call. If the caller selects the former option (by entering 9), the call is routed to Vector 30, and the entire process is repeated.

Using dial-ahead digits to bypass announcements, example 2

```
VDN (extension=1031  name=‘‘Scores’’   vector=31)
Vector 31:
 1. collect 1 digits after announcement 4000 [
If you wish to hear scores of      games
in both divisions, please press 3.  If you wish to hear scores for      Northern
Division games only, please press 4. If you wish to hear scores for      Southern
 Division games only, please press 5.
]
 2. goto step 7 if digits = 3
 3. goto step 7 if digits = 4
 4. goto step 9 if digits = 5
 5. announcement 301 [
Entry not understood. Please try again.
]
 6. goto step 1 if unconditionally
 7. announcement 4002 [Northern Division scores]
 8. goto step 10 if digits = 4
 9. announcement 4003 [Southern Division scores]
```

```
10. collect 1 digits after announcement 4004 [
If you wish to return to the main
menu, please press 9. Otherwise, press 0.
]
11. route-to number 1030 with cov n if digit = 9
12. goto step 15 if digit = 0
13. announcement 301 [
Entry not understood. Please try again.
]
14. goto step 10 if unconditionally
15. disconnect after announcement none
```

Vector 32 (below) is similar in design to Vector 31. The major difference is the information provided and the requested digit entries.

In this example, the caller has to go through at least two sets of options to get the information that he or she wants. Each option set is introduced by an announcement. However, because of the dial-ahead digit capability, the caller can bypass the announcements if he or she chooses. Thus, the caller could enter 1 and 5 within a matter of seconds to hear yesterday's Southern Division scores.

The caller may enter digits while he or she is being queued for an announcement or while the announcement is playing. If digits are entered during an announcement, the announcement is disconnected. If digits are entered while a call is queued for an announcement, the call is removed from the announcement queue.

Dial-ahead digits, example 2

```
VDN (extension=1032   name=Schedule   vector=32)
Vector 32
 1. collect 1 digits after announcement 5000 [
If you wish to hear today's      schedule
of games in both divisions, please press 6.  If you wish to hear     today's schedule
of games in the Northern Division only, please press 7.      If you wish to hear
 today's schedule of games in the Southern Division      only, please press 8.
]
 2. goto step 7 if digits = 6
 3. goto step 7 if digits = 7
 4. goto step 9 if digits = 8
 5. announcement 301 [
Entry not understood. Please try again.
]
 6. goto step 1 if unconditionally
 7. announcement 5002 [Northern Division schedule]
 8. goto step 10 if digits = 7
 9. announcement 5003 [Southern Division schedule]
10. collect 1 digits after announcement 4004 [
If you wish to return to the      main
menu, please press 9.  Otherwise, press 0.
]
11. route-to number 1030 with cov n if digit = 9
12. goto step 15 if digits = 0
13. announcement 301 [
Entry not understood. Please try again.
]
14.goto step 10 if unconditionally
15.disconnect after announcement none
```

ASAI-requested digit collection

The ASAI-requested digit collection feature gives an adjunct the ability to request that a DTMF tone detector be connected for the purpose of detecting user-entered digits. The digits that are collected as a result of this feature are passed to ASAI monitoring and/or controlling adjuncts for action. The switch handles these digits as if they were dial-ahead digits. This feature allows the caller to request Sequence Dialing after the call has been routed to the final destination and has resulted in an unanswered call, that is busy, no answer, and so forth.

These digits are not necessarily collected while the call is in vector processing. They are sent to an ASAI adjunct, or they may be used by Call Prompting features, or both.

ASAI Adjunct Routing and Call Prompting features must be enabled on the switch for this feature to work.

ASAI-provided dial-ahead digits - collect digits command

The ASAI-provided digits feature allows an adjunct to include digits in a Route Select capability. These digits are treated as dial-ahead digits for the call. Dial-ahead digits are stored in a dial-ahead digit buffer and can be collected (one at a time or in groups) using the **`collect digits`** command(s). Although the adjunct may send more than 24 digits in a Route Select, only the first 24 digits (or 24-`x`, where`x` is the number of digits that are collected by vector processing prior to executing the **`adjunct routing link`** vector command) are retained as dial-ahead digits. An application can use this capability to specify the digits that the switch should pass to the VRU as part of the **`converse-on`** vector step.

Note:
The maximum number of dial-ahead digits that can be stored in the buffer is dependent on the number of digits that were already collected for the call by a previous **`collect digits`** vector command. If`x` digits were collected by vector processing prior to executing an **`adjunct routing link`** vector command, the`x` digits collected reduces the maximum number of digits that can be stored as dial-ahead digits as a result of a Route Select. The rest are discarded.

ASAI-provided dial-ahead digits - collect digits command considerations

You should keep the following considerations in mind when working with Call Prompting:

- To enter the digits requested using a **`collect digits`** command, outside callers must have a touch-tone telephone. For such callers using rotary dialing, a 10 second inter-digit timeout takes effect, and the **`collect digits`** command is omitted. As a precaution, a default treatment (for example, **`route-to attendant`** command, **`queue-`**

to split command) should always be provided in the vector script unless the script is created exclusively for users of touch-tone telephones.

- If a caller does not enter the full number of digits specified in a **collect digits** step, an administered timeout occurs. Thereafter, vector processing continues with subsequent vector steps, and an attempt is made to process the call using the digits that have been collected. If the digits entered do not represent a valid destination, and if Automated Attendant is being implemented using a **route-to digits** command, the **route-to digits** command fails, and vector processing continues at the next step, which should be a default treatment.
- It may be prudent to take steps in case a **route-to attendant** command fails, such as providing a disconnect announcement.
- From time to time, all of the system's touch-tone receivers might be in use. As a result, you should avoid starting your main vector with a **collect digits** command, since the caller on a DID or tie trunk in this case receives no audible feedback if he or she has to wait for a receiver to become available. Accordingly, it is a good practice to include some treatment (for example, a **wait-time 0 seconds hearing ringback** step) before the initial **collect digits** step. The **wait-time** step is not necessary if the collect step is collecting ced or cdpd digits.

Direct Agent Calling

What is DAC?

Note:
Direct Agent Calling (DAC) requires CallVisor Adjunct-Switch Application Interface (ASAI) or EAS. Both originating and called party Class of Restrictions (CORs) must be set to allow Direct Agent Dialing. See Expert Agent Selection for information on Direct Agent Announcements (DAA).

DAC is an EAS feature that lets a caller:

- Contact a specific agent instead of a skill hunt group
- Queue for the agent if the agent is on a call
- Use Agent LoginID for callbacks and transfers
- Hear system-wide DAC delay announcement while holding
- Follow the agent's coverage path, if the call is not answered immediately.

Advantages of DAC

DAC calls have two important advantages:

- They reduce the need to transfer callers who want or need to speak with a certain agent, such as the agent spoken to on a previous call.
- They provide more accurate reporting of calls, because CMS counts direct agent calls as ACD calls. In this way, agents get proper credit for taking them. By comparison, calls transferred to an agent are not counted as ACD calls.

How DAC works

DAC works as described below:

- Callers can dial the agent's login ID as part of a DID or from auto attendant as an extension number.
- Direct agent calls have a special ringing sound, regardless of the agent's work state, and the current work mode button on the agent's telephone flashes.
- If the agent is on a call, he or she can use multiple call handling to decide whether to put the call on hold in order to take the direct agent call.
- If the agent is available, the call is delivered according to the answering and alerting options.
- If the agent is not available, or if multiple call handling is not used, call coverage or RONA routes the call to backup.
- While on direct agent calls, agents are unavailable for subsequent ACD calls. If the agent logs off by unplugging the headset, he or she can still answer a direct agent call in the queue by logging back in and becoming available. Agents who have direct agent calls waiting are not allowed to log off using a FAC. If the agent is in Manual In mode or pushes the After Call Work (ACW) button while on a direct-agent call, the agent goes to ACW mode.

Generally, direct agent calls are queued and served in first-in, first-out order before other calls, including priority calls. However, if you administer skill level for the Call Handling Preference, direct agent calls must be assigned the highest priority for them to be delivered before other ACD calls. Otherwise, calls with a higher skill level are distributed before direct-agent calls.

Note that you can use Multiple Call Handling (MCH) to allow agents to answer a DAC with another ACD call active.

Direct agent calls follow the receiving agent’s coverage and call forwarding paths, if these features are administered. Once a call goes to coverage or is forwarded, the call is no longer treated as a direct-agent call, and CMS is informed that the call has been forwarded.

Administering DAC

1. On the Agent LoginID screen, enter the agent's direct agent skill.

   It is suggested that you use the Hunt Group screen to set up a skill for all direct agent calls. This skill will:

   - Tell the communication server how to handle calls to the skill
   - Show report users how much time each agent has spent on direct agent calls

   Note:
   Any agent who will receive DACs should have at least one non-reserve skill assigned to the agent loginID.
2. Add the skill to the agent's administered skills on this screen.

   Whenever an outside caller dials the agent's extension, the communication server looks at the entry in that field to determine the skill for tracking call data.
3. On page 8 of this Feature-Related System Parameters screen, you may specify:
   - A Direct Agent Announcement Extension that plays an announcement to DACs waiting in queue.
   - Amount of delay, in seconds, before the announcement.

Result

You also need to administer a Class of Restriction (COR) for direct agent calls.

Direct Inward Dialing (DID) is administered on the Trunk Group screen.

On the second page of the Hunt Group screen, consider administering Multiple Call Handling On-Request for this hunt group. This feature will enable agents to see that the incoming call is a direct agent call and put their current call on hold to answer the direct agent call.

If there is no answer after a certain number of rings, you may use RONA to redirect the caller to a VDN that points to a vector. You can set up the vector to provide appropriate routing and treatment for the call.

On page 3 of the Hunt Group screen, administer messaging for the direct agent hunt group.

Next, you need to assign this hunt group to agents who need to receive direct agent calls.

DAC considerations

This section includes the following topics:

Maximum number of agents

If an agent is assigned to more than one split or skill, each assignment applies to the maximum number of agents. When computing the number of agents measured by BCMS, count one agent as one agent regardless of the number of splits/skills that the agent will be logged into. For CMS sizing, count one agent for each agent in each split or skill measured by CMS; one agent logged into three splits/skills counts as three agents.

Using the Number of Agents System Capacity screen, you can view the Used, Available, and System Limit counts.

MIA across splits/skills

MIA Across Splits/Skills distributes calls more equally to agents with multiple splits/skills. When agents handle a call for one split or skill, they go to the back of all their idle agent lists.

With MIA Across Splits/Skills, agents may not receive calls from all of their splits/skills. If, for example, split 20 has a very short average agent idle time and split 22 has a very long average agent idle time, agents with both of these skills may never become the most-idle for skill 22 because they continuously take calls for split 20.

Announcements

Announcements can be analog, aux trunk, DS1, or integrated. Integrated announcements use the TN750, TN2501AP, or co-resident announcement board, and queuing is based on whether one of the playback channels is available. When a channel becomes available, any announcements on the board can be accessed, including the announcement already being played. A caller may be in queue for an announcement because a channel is not available, even though that announcement is not being used.

Queues for analog and aux trunk announcements are on a per-announcement basis. You can also install multiple Integrated Announcement boards to allow for more announcements.

If a delay announcement is used, answer supervision is sent to the distant office when the caller is connected to the announcement. Charging for the call, if applicable, begins when answer supervision is returned.

Storing and retrieving messages

Leave Word Calling messages can be stored for an ACD split or skill and retrieved by a split or skill member, a covering user of the split or skill, or a system-wide message retriever. The message retriever must have a telephone display and proper authorization. You can also assign a remote Automatic Message Waiting lamp to an agent telephone to indicate when a message has been stored for the split or skill.

Class of Restriction

Each ACD split or skill and each individual agent is assigned a Class of Restriction (COR). You can use Miscellaneous Restrictions to prohibit selected users from accessing certain splits/skills. You can use Miscellaneous Restrictions or restrictions assigned through the COR to prevent agents from being accessed individually. Unless you administer such restrictions, each agent can be accessed individually as well as through the split or skill.

An agent with origination and termination restriction can receive ACD calls and use the assist function. A telephone in a COR with termination restriction can receive ACD calls.

If you are using Service Observing, administer a COR for observers and agents being observed.

Trunk groups and ACD splits

- If you assign an ACD split extension as the incoming destination of a trunk group and the split extension is later changed, you must also change the incoming destination of the trunk group to a valid extension.
- Calls incoming on a non-DID trunk group can route to an ACD split instead of to an attendant. Calls incoming on any non-DID trunk group can have only one primary destination; therefore, the trunk group must be dedicated to the ACD split or a VDN.
- For MEGACOM 800 Service with DNIS over a wink/wink-tie trunk, if all agents are logged out or in AUX work mode, incoming MEGACOM calls receive a busy signal if no coverage path is provided (unlike other automatic-in trunk groups, which receive ringback from the central office).
- CO communication servers usually drop calls that remain unanswered after two to three minutes. Therefore, if an incoming CO call queues to a split without hearing an

announcement or music, and the caller hears CO ringback for two to three minutes, the CO drops the call.

Agent considerations

- Agents should not be used for hunt group calls and ACD split or skill calls simultaneously. Otherwise, all calls from one split or skill (either ACD or hunt group) are answered first. For example, if ACD calls are answered first, none of the hunt-group calls are answered until all of the ACD calls are answered.
- Agents with multiappearance phones can receive only one ACD call at a time unless Multiple Call Handling is active. Without MCH, a phone is available for an ACD call only if all call appearances are idle. The agent may, however, receive non-ACD calls while active on an ACD call.

Vector-controlled splits/skills

- You can enhance ACD by using Call Prompting, Call Vectoring and Expert Agent Selection. For detailed information on vector-controlled splits/skills, see *Programming Call Vectors in Avaya Aura™ Call Center*. Vector-controlled splits/skills should not be called directly using the split or skill extension (instead of using a VDN mapped to a vector that terminates the call to a vector controlled split or skill). However, if split or skill extensions are called, the calls do not receive any announcements, are not forwarded or redirected to coverage, and do no intraflow/interflow to another hunt group.
- The oldest-call-waiting termination, which is available with Call Vectoring, is supported for agents who are servicing ACD calls only.

Changing hunt groups from ACD to non-ACD

Before you change a hunt group from ACD to non-ACD, all agents in that hunt group must be logged out. When you change a hunt group from ACD to non-ACD, the system places all agents in that hunt group in busy state. If any phones in the hunt group have an Auxiliary Work button, the button lamp lights. To become available for calls, the agent presses the Auxiliary Work button or dials the Hunt Group Busy Deactivation FAC followed by the hunt-group number.

Direct Agent Call (DAC) interactions

Interaction	Description
Attendant Call Waiting	An attendant can originate or extend a call to an ACD split. Attendant Call Waiting cannot be used on such calls. However, such calls can enter the split queue.
Attendant Intrusion	Attendant Intrusion does not work with ACD split extensions because an ACD extension has many agent extensions. It is not possible to determine which agent extension to intrude upon.
Automatic Callback	Automatic Callback calls cannot be activated toward an ACD split or skill.
Call Coverage	Calls can redirect to or from an ACD split or skill. A vector-controlled split or skill cannot be assigned a coverage path.

Interaction	Description
	If the queue is not full, a call enters the queue when at least one agent is on an ACD call or in ACW mode. Queued calls remain in queue until the Coverage Don't Answer Interval expires before redirecting to coverage. If any split or skill agent becomes available, the call is directed to the agent. Calls that redirect on the Don't Answer coverage criterion are reported to BCMS/CMS as intraflowed calls. If a call is queued for an ACD split or skill and redirects using Call Coverage directly to an announcement, the call is dropped after the announcement. Calls to a split or skill that are directed to an agent do not follow the agent's call coverage path. If an agent activates Send All Calls it does not affect the distribution of ACD calls. An ACD split or skill call directed to an agent station follows the split or skill call coverage path, once the agent's Don't Answer interval is met. For a call to an ACD split or skill to be redirected to call coverage on the Busy coverage criterion, one of the following conditions must exist: • All agents in the split or skill are active on at least one call appearance and the queue, if there is one, is full. • No agents are logged in. • All agents are in Auxiliary Work mode.
Call Forwarding All Calls	Call Forwarding All Calls activated for an individual extension does not affect the extension's ACD functions. When activated for the split or skill extension, calls directed to the split or skill are forwarded from the split or skill. Calls receive no announcements associated with that split or skill (other than a forced first announcement, if administered). The system reports to BCMS/CMS that calls are queued on the split or skill. The system reports to CMS when the call is removed from the queue and forwarded. Calls can be forwarded to an off-premises destination to activate Intraflow and Interflow. For more information, see Intraflow and Interflow in the chapter ACD Call Center features.
Data Call Setup	Telephone or data terminal dialing can be used on calls to or from a member of an ACD split or skill.
Data Restriction	If the trunk group used for an ACD call has data restriction activated, agents with Automatic Answer activated do not hear the usual zip tone.
DCS	CMS cannot measure ACD splits/skills on a Distributed Communications System (DCS) network as if they were one communication server. Agents for a split or skill must be all on the same communication server. If a call to an ACD split or skill is forwarded to a split or skill at another DCS node, the caller does not hear the forced first announcement at the second split or skill. If an ACD split or skill is in night service, with a split or skill at second DCS node as the night service destination, a call to the first split or

Interaction	Description
	skill is connected to the second split or skill's first forced announcement.
Dial Intercom	An agent with origination and termination restriction can receive ACD calls and can make and receive dial intercom calls.
Forced Agent Logout from ACW mode	After an agent handles a Direct Agent Call (DAC), the Forced Agent Logout from ACW feature applies when the agent enters the ACW state after the DAC is released.
Hold	If an agent puts an ACD call on hold, information is reported to the CMS using Personal Call Tracking. CMS records the amount of time the agent actually talks on the call.
Individual Attendant Access	Individual attendant extensions can be assigned to ACD splits. Unlike telephone users, individual attendants can answer ACD calls as long as there is an idle call appearance and no other ACD call is on the console.
Internal Automatic Answer (IAA)	Internal calls directed to an ACD split or skill are eligible for IAA. You cannot administer IAA and ACD Automatic Answer simultaneously on the same station.
Intraflow and Interflow	Intraflow and Interflow, when used with Call Forwarding All Calls or Call Coverage, allows splits/skills to be redirected to other destinations on and outside the system.
Multiappearance Preselection and Preference	All assigned call appearances must be idle before an ACD call is directed to a phone.
Location Preference Distribution	Direct Agent calls take precedence over Location Preference Distribution.
Night Service - Hunt Group	When Hunt Group Night Service is activated for a split or skill and the night-service destination is a hunt group, a caller hears the first forced announcement at the original split or skill. The call is redirected to the night-service destination hunt group. If all agents in the hunt group are busy, the caller hears whatever you have assigned.
Terminating Extension Group	A TEG cannot be a member of an ACD split or skill. • Transfer - Calls cannot be transferred to a busy split or skill. The transfer fails and the agent transferring the call is re-connected to the call. If an agent presses the Transfer button, dials the hunt-group extension number, and then disconnects while the split or skill is busy, the call is disconnected. • Phone Display - For calls dialed directly to an ACD split or skill extension, the identity of both the calling party and ACD split or skill are shown on the phone display.

Expert Agent Selection

What is EAS?

Expert Agent Selection (EAS) helps call center managers provide the best possible telephone service to callers by matching the needs of the callers with the skills or talents of the agents. Caller needs and agent skills are matched using Call Vectoring. All the Call Vectoring features described in this guide can be used with EAS.

Matching the call to an agent with the appropriate skills reduces transfers and call-holding time. Accordingly, customer satisfaction is increased. Also, since an entire agent group need not be trained at the same time for the same skills, employee satisfaction is increased.

In addition to matching the skills that are required for a call to an agent with one of those skills, EAS provides other capabilities:

- Logical Agent associates hardware (the telephone) with an agent only when the agent is logged in. While the agent is logged in, calls to the agent login ID are directed to the agent. For more details, see Logical Agent capability on page 189.
- Direct Agent Calling (DAC) allows a user to call a particular agent and have the call treated as an ACD call. For more details, see Direct Agent Calling.

Most EAS administration can be completed before you activate it, thus minimizing the down time for upgrading to EAS.

EAS requires ACD and Call Vectoring. All of the existing ACD features and Call Vectoring capabilities can be used within EAS applications.

As with Call Vectoring calls, EAS calls are directed to VDNs, which in turn point to vectors. However, unlike Basic Call Vectoring, skills can be assigned in EAS to VDNs, or they can be associated with vector steps to represent caller needs. As for Call Vectoring calls, EAS calls are queued to ACD hunt groups. However, with EAS enabled, ACD hunt groups are called *skill hunt groups* instead of splits.

Skill hunt groups deliver calls to EAS agents. Agent skills are administered on the Agent Login ID screen.

Note:

These are the same login IDs that are used by Avaya Call Management System (CMS) and Basic Call Management System (BCMS).

Logical Agent implies that telephones are no longer preassigned to hunt groups. When the agent logs, the telephone becomes associated with all of the skill hunt groups that are assigned to that agent login ID.

With EAS optioned and enabled, ACD calls can also be directed to a particular agent, instead of to the skill hunt group, by using the DAC feature. The direct agent call is treated like an ACD call, but it waits in queue for a specific agent to become available. direct agent calls have a higher priority than skill hunt group calls.

About EAS

Use Expert Agent Selection (EAS) to route incoming Automatic Call Distribution (ACD) calls to the agent who is best qualified to handle the call. That is, the agent with the specialized skills or experience required to best meet the caller's needs.

In addition, EAS provides the following capabilities:

- You assign all agent functions to the agent login ID and not to a physical phone. Therefore, EAS agents can login to and work at any phone in the system.
- Using the agent login ID, a caller places a call directly to a specific agent. These calls can be treated and reported as ACD calls.

EAS ensures the best possible service to the caller.

Detailed administration for EAS

In general, EAS uses vectors to route calls to agents with the required skills. To administer EAS you must:

- Assign skills to VDNs on the Vector Directory Number screen.
- Create vectors that will route a call to the correct skill.
- Assign skills with priority levels to agents on the Agent Login ID screen.

EAS administration method	Description
VDN administration	You can administer up to three VDN skill preferences on the Vector Directory Number screen in the 1st Skill, 2nd Skill and 3rd Skill fields. These fields indicate the skills that are required to handle calls to this VDN. All of the VDN skills on the VDN screen are optional. For example, only the first and third, or only the second and third VDN skills might be assigned. Vector steps can then refer back to these fields to route calls. For example,

EAS administration method	Description
	`queue-to skill 1st` routes calls the skill administered as 1st on the VDN screen.
Vector administration	When a call routes to a VDN, the VDN directs the call to the vector that is specified on the Vector Directory number screen. The vector then queues the call to the skill specified in a vector step. You can write vectors that route calls either to specific skill numbers or to the skill preferences administered on the Vector Directory Number screen.
Agent administration	Assign skills to each agent. In addition, assign a skill level to each skill for the agent. When a vector routes incoming calls to a skill, the call is delivered to an available agent with the skill assigned. If no agents are available, the call is queued until it can be answered by an agent who has the skill required to handle the call.
Agent selection	The administered agent selection method and Call Handling Preference determine which agent will receive an incoming call.
Agent selection method	EAS can use either Uniform Call Distribution (UCD) or Expert Agent Distribution (EAD) to select agents for calls. Both methods can use the Most-Idle Agent (MIA) or the Least Occupied Agent (LOA) algorithm to select agents.

For more information on agent selection methods, see Automatic Call Distribution.

Call handling preference administration

This section includes the following topics:

The call handling preference selected on the Agent Login ID screen can route calls based on either greatest need or agent skill level. The following table summarizes how a call is routed based on greatest need or agent skill level administration with either UCD or EAD distribution.

If:	EAD/UCD with Skill Level	EAD/UCD with Greatest Need
Agents are available. When a new call arrives it is delivered to:	• EAD - Most-idle agent with the highest skill level for the calls' skill. • UCD - Most-idle agent with the call's skill.	• EAD - Most-idle agent with the highest skill level for the call's skill. • UCD - Most-idle agent with the call's skill.
Agents are not available, calls are in queue. When an agent becomes available, he or she receives:	• EAD - Highest priority oldest call waiting for agent's highest level skill with calls in queue. • UCD - Highest priority oldest call waiting for the agent's highest level skill with calls in queue.	• EAD - Highest priority oldest call waiting for any of the agent's skills. • UCD - Highest priority oldest call waiting for any of the agent's skills.

MIA Across Splits/Skills for EAS

In addition, both UCD and EAD can be used in conjunction the MIA Across Splits/Skills option. With MIA Across Splits/Skills, the system has the option to either:

- Retain the agent's position in other splits/skills MIA lists while handling an ACD/DA call (default), or
- Remove the agent from all MIA lists when handling a call from any of the splits/skills.

The distribution is based on total call activity rather than activity in a single skill.

See Automatic Call Distribution for more information about UCD, EAD, and MIA Across Splits/ Skills.

Additional agent login ID capabilities

The following capabilities are also associated with agents' login IDs.

Capability	Description
Auto-Answer	When EAS is optioned, auto answer settings can be assigned to agents on the Agent LoginID screen. An agent's auto answer setting will apply to the station where the agent logs in. If the auto answer setting for that station is different, the agent's setting overrides the station's.
Calls	To call an EAS agent, the caller dials the login ID extension. The call is extended to the physical extension where the agent with that login ID is logged in. Calls to the login ID reach the agent independent of the phone the

Capability	Description
	agent is currently using. For example, when agents use multiple phones because they have multiple offices or rotate desks, login IDs allow these agents to be reached independent of their current location.
Name	Calls to the login ID display the name associated with the login ID and not the name associated with the phone. This is also true for calls made from a phone with an agent logged in.
Coverage	When the agent is logged out, or when calls go to coverage because the agent is busy, or does not answer, calls to the login ID go to the coverage path associated with the agent and not the phone. When an agent is logged out, calls go to the agent's busy coverage destination.
Restrictions	Calls to the login ID or from the agent use the restrictions associated with the agent and not the phone. Phones are fully functional when an agent is not logged in. The restrictions, coverage, and name revert to the phone administration when the agent logs out.

DAC

Calls to an agent's login ID are treated as direct agent calls if the caller and the agent have the Direct Agent Calling Class of Restriction (COR). Direct agent calls can be originated by stations or trunks. If the caller or agent does not have the proper COR, the call is treated as a normal non-ACD (personal) call.

See Avaya Business Advocate on page 83 for additional information on how DAC works, is used in the call center, and is administered.

Direct agent calls are treated as ACD calls and receive zip tone answer, queue as other ACD calls do, allow the agent to enter after call work following the call, and are measured by BCMS and CMS.

Any of the agent's skills can be the direct agent skill. When greatest need is optioned as the Call Handling Preference, the agent always gets direct agent calls before any skill calls. This is because direct agent calls have a higher priority than skill calls. However, when skill level is optioned as the Call Handling Preference, the agent will get direct agent calls first only if the direct agent skill has the agent's highest skill level. Otherwise calls from a skill with a higher level will be distributed before direct agent calls. If the direct agent skill and another skill are the same skill level, the agent will always receive direct agent calls before the other skill calls because direct agent calls have a higher priority.

A **`route-to`** vector command with an EAS login ID as the destination is treated as a IC Email call if the VDN and agent have the COR and the **Direct Agent** field is set to `y`.

EAS considerations

Station User records cannot be shared between TTI ports and EAS LoginID extensions. This causes a reduction in the number of possible EAS LoginID extensions allowed by the System depending on the number of administered TTI ports. For example, if 2,000 TTI ports are administered, the maximum number of allowable EAS LoginIDs is reduced by 2,000.

EAS agent login IDs are also tracked for personal calls. CMS uses the first skill an EAS Agent is logged into to track personal calls. If the first logged-into skill is unmeasured, CMS credits the agent login ID with the personal call, but no skill hunt group is credited with the personal call.

The system can have either splits/skill hunt groups but not both simultaneously. Non-ACD hunt groups can exist with either splits or skills. Skill hunt groups are required when using EAS.

EAS interactions

Unless otherwise specified, the feature interactions for skill hunt groups are the same as for vector-controlled splits.

Interaction	Description
Abbreviated Dialing	Abbreviated Dialing is used to log in or log out EAS agents. Abbreviated Dialing lists/buttons can only be administered for stations.
Add/Remove Skills	In the EAS environment, agents have the ability to add and remove skills during a login session by dialing a FAC. Other phone users with console permissions can add or remove an agent's skill on behalf of the agent. (Note that the ability to add and remove skills depends on whether a user has a class of restriction (COR) that allows adding and removing skills.)
Administration Without Hardware	EAS login ID extensions are extensions without hardware. Login ID extensions require space in the dial plan.
Agent Work Mode States	With EAS, agents can only be in a single work mode at any one time for all their skills.
Assist	The Assist feature can be used with a skill hunt group (for example, where there is one supervisor per skill hunt group). When assist is selected, a call is placed to the supervisor associated with the skill for the active call.
AUDIX	Calls to the EAS agent login ID can cover to AUDIX.
Auto-Available Splits/Skills	If a skill hunt group is administered as an Auto-Available Skill (AAS) the EAS login IDs assigned to this skill must also be administered as Auto-Available. When the communication server reinitializes, these login IDs are automatically logged in with the auto-in work-

Interaction	Description
	mode. If any communication server features attempt to change the work-mode to anything except to auto-in, this attempt is denied. This feature is not intended for human agents.
Automatic Answering with Zip Tone	The Automatic Answer option can only be administered for a physical extension.
Automatic Callback	Users cannot activate Automatic Callback to an EAS agent's login ID. They can activate Automatic Callback to the phone where the agent is logged in.
Call Forwarding	Skill hunt groups (since they are vector-controlled) cannot be call forwarded. EAS agent login IDs cannot be forwarded, but the physical extension where the EAS agent is logged in can be forwarded.
Call Park	Calls cannot be parked on the skill hunt group extension.
Call Pickup	Skill hunt group extensions and EAS login ID extensions cannot be members of a call pickup group.
Class of Restriction (COR)	Skill hunt groups do have a class of restriction. This is used if the skill hunt group extension is called directly. The COR for an EAS agent login ID overrides the physical extension's COR of the phone an EAS agent logs into.
Class of Service (COS)	EAS agents do not have a COS associated with their login ID. Therefore, the COS of the telephone is not affected when an EAS agent logs into it.
Directed Call Pickup	An EAS agent can use the Directed Call Pickup feature to pick up a call and/or have his or her calls picked up by another agent. The Class of Restriction of the agent will override the Class of Restriction of the station where the agent is logged in. If both the station's COR and the logged-in agent's COR allow the call to be picked up using Directed Call Pickup, the user picking up the call can use either the station's extension or the agent's loginID.
Displays - Phone	When an EAS agent logs in, the display for originators who call the login ID shows the login ID and agent name (as administered using the Agent Login ID screen). Calls that the agent originates show the agent login ID and agent name at the receiving telephone display. However, the user can display the name of the physical extension where the EAS agent is logged in. To do this, the user must be active on a call with the agent, and must have a telephone with an alphanumeric display and an inspect button. When the inspect button is pressed during a call to or from the EAS agent, the physical extension name of the agent is displayed.
Leave Word Calling	Calls to the physical extension show the physical extension's number and name on the originator's display. When an EAS agent is logged into a station, the agent can only retrieve LWC messages left for that agent's login ID. To retrieve LWC messages left for that station, the agent must log out.

Interaction	Description
	When an EAS agent is logged into a station, its Message lamp defaults to tracking the status of LWC messages waiting for the station. However, you can assign the Message lamp to track the status of LWC messages waiting for the agent's login ID.
Look Ahead Interflow	VDN skills are not sent to another ACD/PBX when a call interflows using Look Ahead Interflow. If skills have the same meaning on both ACDs, then a Look Ahead Interflow command to a VDN with the same skills assigned can provide a mapping of the skills.
Message Waiting Lamp	The Message Waiting Lamp by default tracks the status of messages waiting for the logged in EAS agent LoginID rather than messages for the physical extension. The operation of the Message Waiting Lamp can be changed so that it tracks the status of messages waiting for the physical extension where the agent is logged in. For more information about Feature-Related System Parameters, see Administrator Guide for Communication Manager.
Queue Status Indications	Physical extensions can be administered with Queue Status Indicator buttons and lamps for skill hunt groups. Queue Status Indicators can be administered for all skills needed by agents using that physical extension, given that enough buttons are available.
Service Observing	The Service Observing feature is activated in the EAS environment by dialing either the physical extension of the telephone where an EAS agent is logged in or the login ID of the agent.
Tenant Partitioning and agent skills	The Tenant Partitioning feature was designed to support multiple customers using the same Communication Manager server. Tenant Partitioning separates entities, thereby avoiding or reducing interactions between entities in different partitions. Assign the same partition number to agents, groups, and entities to avoid blocking calls and to avoid any unexpected interactions that result from mixing tenant partitions. When Tenant Partitioning is active and used for restriction of service, assign the same partition number to: • ACD agents • Hunt groups (splits or skills) • Other entities that are involved with ACD agents and hunt groups, such as VDNs and announcements **Note:** An agent's skill set should contain only skills belonging to the *same* tenant partition; not doing so can result in unintended behavior.
VuStats	VuStats displays can show an agent's skill assignments and can show some measurements by skill.

EAS benefits

About EAS benefits

Because you can match caller needs to an agent who has the appropriate skills to handle the call, your call center can achieve the following:

- Maximum profitability.
- Greater customer satisfaction because the caller reaches, on the first call, an agent with the necessary skills to handle the call.
- Greater responsiveness to customer needs because you can base call distribution on either skill level or greatest need.
- Improved agent performance and satisfaction because agents handle calls they are most familiar and most comfortable with.
- Improved agent performance because supervisors have the option to have agents handle calls based on either skill level or greatest need. For agents, it offers an opportunity to learn new skills.
- Ability to track the number of calls that are handled by particular skills from the VDN perspective. You can see whether vectors are performing as expected.

Skill-based call distribution

With EAS, call distribution is based on agent skills. Caller needs are determined by the VDN called or by voice prompting.

An agent who has at least one of the skills that a caller requires is selected to handle the call. You assign skills and skill levels to agents to determine which types of calls go to which agents and to determine the order in which agents serve waiting calls.

Interruptible Aux

If a skill's designated service level is not met, unavailable EAS agents who are in Auxiliary (AUX) work mode and have an interruptible reason code can be made available. Using this feature, for example, during the call volume spikes, you can use agents in Auxiliary (AUX) work mode to maintain your desired service level.

For more information on Interruptible Aux, see About Interruptible Aux on page 244.

Greatest need call distribution

With EAS, you have the option of basing call distribution on greatest need instead of skill level. You can distribute the highest-priority, oldest call waiting to an agent with an appropriate skill, even if that skill is not the agent's highest-priority skill.

Percent allocation call distribution

Percent allocation enables you to assign a percentage of an agent's time to each of the agent's assigned skills, to comprise a total of 100% of the agent's staffed tome. Percent allocation then selects the call that is the best match for an agent's administered skill percentages.

Percent allocation is available with Avaya Business Advocate. For more information, see *Avaya Business Advocate User Guide*.

Percentage allocation routing for EAS

Based on specified percent allocation, percentage allocation routing helps you to distribute calls among a set of call centers or local/remote destinations using VDNs. Percentage allocation routing enables desired distribution of work by routing the incoming calls according to the percentage specified for each of the destinations.

For more information on Percentage allocation routing, see Percentage allocation routing.

ACD queuing and vector commands

ACD queuing and the vector commands **`queue to skill`** and **`check skill`** are used to route a call to an agent with the appropriate skill to handle the call.

Route calls by skill level

Under agent surplus conditions, this feature allows you to request selection of an agent with a particular skill level. Using vector processing, you can apply skill level as well as skill (and other factors) in agent selection. The **`check skill`** vector command is used to route calls to an agent with a particular skill. For more information on check skill vector command, see Check skill for available agents with level preference in *Programming Call Vectors in Avaya Aura™ Call Center*. For more information on how to route calls by skill level, see *Administering Avaya Aura™ Call Center Features*.

EAS considerations

When you implement the EAS feature, be aware of the following considerations:

- With EAS, skill hunt groups replace splits. You cannot administer both skills and splits on the same switch. All ACD hunt groups must be administered as either splits or skills. If EAS is optioned, all ACD hunt groups are skill hunt groups.
- With EAS, all skill hunt groups except for messaging-system hunt groups must be vector controlled.
- With EAS, non-ACD hunt groups are allowed, but they cannot be vector controlled.
- Agent login IDs are extensions in the dial plan, and they decrease the total number of stations that can be administered.
- With EAS, agents have a different login procedure and a single set of work mode buttons, regardless of the number of skills that are assigned to the agents.
- Skill hunt groups can distribute a call to the most-idle agent (UCD) or to the most-idle agent with the highest skill level for that skill (EAD). In either of these cases, the call can route to the most-idle agent for the specified skill, or to the most-idle agent in all of the skills. Direct Department Call (DDC) distribution is not allowed for skill hunt groups.
- With either UCD or EAD distribution, the system can be administered to deliver calls based either on greatest need or agent skill level. This is the Call Handling Preference that is administered on the Agent LoginID screen. When calls are in the queue, greatest need delivers the highest priority oldest call waiting for any of the agent's skills. With skill level administration, the system delivers the highest priority oldest call waiting for the agent's highest level skill with calls in the queue.
- The EAS-PHD customer option adds additional capabilities to the basic EAS capabilities.
    - It increases the number of skills an agent can log in to from 4 to 20
    - It increases the number of agent skill priority levels from 2 to 16

For information on converting a call center to EAS, see Converting a call center to EAS.

Expert Agent Selection (EAS) terminology

The following terms have special significance in the EAS environment.

Agent skill	The type of call that a particular agent can handle. With EAS, an agent can be assigned up to four skills each, with a primary (level 1) or secondary (level 2) skill level. With the following releases of Communication Manager for EAS-PHD: • Prior to 2.0, an agent can be assigned as many as 20 skills • Later than 2.0, an agent can be assigned up to 60 skills

Caller needs	The reasons why customers call your call center. Caller needs are determined by the VDN number that the caller dialed, by Call Prompting, or by Automatic Number Identification (ANI) database lookup. You define caller requirements in the vector in order to route calls to an ACD agent with particular skills to match the needs of the caller. These caller needs, which translate to skills, become active for an ACD call whenever a queue to the main skill or check backup skill vector command is executed and the threshold condition is met.
Skill	A specific caller or business need of your call center. You define your skills based on the needs of your customers and your call center. You specify skills by skill numbers, which are assigned to agents and are referenced in vectors to match caller needs with an agent who is skilled to handle those needs. When configuring your call center for skills, a particular skill number always has the same meaning, whether it is an agent skill, VDN skill, or skill hunt group.
Skill hunt group	Calls are routed to specific skill hunt groups that are usually based on caller needs. Agents are not assigned to a skill group; instead, they are assigned specific skills that become active when they log in.
Skill level	For each agent skill, a skill level may be assigned. With EAS-PHD, skill levels can range from 1 to 16, with 1 being the highest skill level (also known as the highest-priority skill). Without EAS-PHD, skill levels may be defined as primary (level 1) or secondary (level 2), with the primary being the highest-priority skill. When calls are queued for more than one of the agent's skills and the agent's call-handling preference is by skill level, the agent receives the oldest call waiting for the agent's highest level skill. If an agent's call-handling preference is by greatest need, then the agent receives the highest-priority, oldest call waiting for any of that agent's skills, regardless of skill level.
Top agent	An agent in a given skill who has the skill assigned as top skill.
Top skill	For EAS-PHD, an agent's first-administered, highest-priority skill. For EAS, an agent's first-administered primary skill (or first-administered secondary skill if the agent has no primary skill assigned). With call-handling preference by skill level, this is the skill for which the agent is most likely to receive a call.
VDN skill preference	Up to three skills can be assigned to a VDN. Calls use VDN skills for routing based on the preferences that you specify in the vector. VDN skill preferences are referred to in the vector as 1st, 2nd, and 3rd.

EAS-PHD - 120 skills/16 skill levels

EAS-PHD is a feature that allows an agent to be assigned to as many as 120 skills. For each skill, one of the 16 skill levels can be assigned, with 1 being the highest skill level and 16 being the lowest skill level.

If calls are waiting for some of the agent's skills and the agent's call-handling preference is by skill level, the agent receives the call that requires the agent's highest-priority skill. For an agent, the first-administered, highest-priority skill is known as the agent's top skill. The top skill represents the skill for which the agent is most likely to receive a call.

If an agent's call-handling preference is by greatest need, the top skill is not useful, because the agent receives the highest-priority, oldest call waiting that requires any of the agent's skills, regardless of skill level.

Identifying caller needs

About identifying caller needs

Caller needs for a particular call can be identified by any of the following methods:

- Interpreting information that is passed from the network in the screen of DNIS digits or ISDN messages.
- Processing Call Prompting digits, digits entered at a Voice Response Unit (VRU), or CINFO digits that are forwarded by the network.
- Using Adjunct Switch Application Interface (ASAI) or a VRU such as Avaya Interactive Response in a host database lookup.

To show how a call center manager might match caller needs and agent skills (which can be viewed as capabilities needed from the caller's perspective), assume that a call center receives inbound calls from automobile club members who speak Spanish or English. The callers in this case either need to plan a vacation route or have trouble with their car and are calling for assistance. The following table provides example associations between caller needs and agent capabilities.

Table 12: Example of caller need-to-agent skill matching

Caller need	Capability needed
Tourist information	Knowledge of the region
To speak Spanish	Bilingual
Emergency assistance	Handle stressful callers
Tow truck	Access to dispatch systems

The following list looks at the call center manager's strategy in matching the caller needs to the capabilities of the agent:

- Tourist information/knowledge of the region

  Travelers may need information while traveling or regarding a future trip. All assigned agents can provide this information.
- To speak Spanish/bilingual

  Separate numbers are published and used as part of Spanish membership information, or Call Prompting is used after a general number is dialed.
- Emergency assistance/handle stressful callers

  Separate emergency road service numbers are published and used, or Call Prompting is used after a general number is dialed. For example, a number is provided for towing.

Note that the call center chose to implement Call Prompting to identify Spanish-speaking callers and callers who require emergency assistance. This allows for quicker and more specialized treatment and therefore better satisfies the caller's needs.

In addition, some customers might prefer to speak to the agent that he or she spoke to on a previous call. To accommodate this request, a call center manager can implement Direct Inward Dialing (DID) at the call center. Also, Direct Agent Calling (DAC) can be used to direct a call to a specific agent.

The following sections explain further how caller needs are identified.

DNIS/ISDN called party

A set of DNIS digits can be interpreted as a VDN. The following table presents four services and their corresponding telephone number including DNIS digits that might be provided to the caller.

Table 13: Examples of services and corresponding DNIS digits

Service	Telephone number	Corresponding DNIS
Emergency road service (English)	800-765-1111	6001
Emergency road service (Spanish)	800-765-2222	6002
Route planning (English)	800-765-3333	6003
Route planning (Spanish)	800-765-4444	6004
General (Call Prompting)	800-765-5555	6005

Note:
DNIS digits must be extensions that are reflected in the dial plan.

Call Prompting/VRU Digits/CINFO digits

The Call Prompting/VRU/CINFO digits are entered by the caller in response to any recorded question about a caller's needs, or in the case of CINFO ced or cdpd digits, are provided by the call center host computer. For example, a hotline for a product may request that a product code be entered, or a travel service may request a 2-digit state code to indicate the state to which the caller would like to travel. The following table provides a prompt that encourages the caller to enter the appropriate Call Prompting digit for the needed service from the automobile club.

Table 14: Example of a prompt for entering Call Prompting digits

For emergency road service, dial 1.
Para asistencia con su automovil, marque el dos.
For travel route directions, dial 3.
Para informacion sobre rutas, marque el cuatro.

Host database lookup

A host database lookup uses DNIS and ANI (calling party's number) to determine what skills are required or even the agent desired. For example, the database may show that the caller speaks Spanish and has been working with Agent 1367. To access host information, either Adjunct Switch Application Interface (ASAI) or a VRU in conjunction with a converse-on skill step is used.

Direct Agent Calling

About DAC

Direct Agent Calling (DAC) is an EAS feature that lets a caller:

- Contact a specific agent instead of a skill hunt group
- Queue for the agent if the agent is on a call
- Use Agent LoginID for callbacks and transfers
- Hear system wide direct agent delay announcement while holding
- Follow the agent's coverage path, if the call is not answered immediately

DAC allows a call to a specific ACD agent to be treated as an ACD call. Zip-tone answer, ACW, and other ACD features can be used with direct agent calls.

If an agent is logged in but is not available, the call queues for that agent. If the agent is not logged in, the call follows the agent's coverage path.

EAS Direct Agent Calling is accomplished by dialing the login with the proper class of restriction (COR) settings. Both the caller (that is, trunk, VND, or station) and the agent must have the direct agent COR settings.

Customers might call an agent directly using Direct Inward Dialing (DID) if the agent's login ID is a published number, or customers might dial a toll-free number and be prompted for the agent's login ID extension. Vectors can be designed to handle the Call Prompting function.

Note:
DAC requires CallVisor Adjunct-Switch Application Interface (ASAI) or EAS. Both originating and called party Class of Restrictions (CORs) must be set to allow Direct Agent Dialing.

Advantages of DAC

Direct agent calls have two important advantages:

- They reduce the need to transfer callers who want or need to speak with a certain agent, such as the agent spoken to on a previous call.
- They provide more accurate reporting of calls, because CMS counts direct agent calls as ACD calls. In this way, agents get proper credit for taking them. By comparison, calls transferred to an agent are not counted as ACD calls.

How DAC works

DAC works as described below:

- Callers can dial the agent's login ID as part of a DID or from auto attendant as an extension number.
- Direct agent calls have a special ringing sound, regardless of the agent's work state, and the current work mode button on the agent's telephone flashes.
- If the agent is on a call, he or she can use multiple call handling to decide whether to put the call on hold in order to take the direct agent call.
- If the agent is available, the call is delivered according to the answering and ringing options.
- If the agent is not available, or if multiple call handling is not used, call coverage or RONA routes the call to backup.
- While on direct agent calls, agents are unavailable for subsequent ACD calls. If the agent logs off by unplugging the headset, he or she can still answer a direct agent call in the queue by logging back in and becoming available. Agents who have direct agent calls waiting are allowed to log off. Direct Agent calls left in-queue for any logged-out agent will remain in-queue until coverage takes over (unless the Direct Agent call was made with

the Priority Calling feature), or until the calling party abandons. If the agent is in Manual In mode or pushes the After Call Work (ACW) button while on a direct agent call, the agent goes to ACW mode.

Generally, direct agent calls are queued and served in first-in, first-out order before other calls, including priority calls. However, if you administer a skill level for Call Handling Preference, direct agent calls must be assigned the highest priority for them to be delivered before other ACD calls. Otherwise, calls with a higher skill level are distributed before direct agent calls.

Note that you can use Multiple Call Handling (MCH) to allow agents to answer a direct agent call with another ACD call active.

Direct agent calls follow the receiving agent's coverage and call forwarding paths, if these features are administered. Once a call goes to coverage or is forwarded, the call is no longer treated as a direct agent call, and CMS is informed that the call has been forwarded.

Functions and examples

Skills administration

A skill is an attribute that is:

- Administered as a skill hunt group
- Administered to VDNs (VDN skill preference)
- Assigned to agents (agent skill)

A skill hunt group is administered for each skill. A skill hunt group is a set of agents trained to meet particular customer needs.

Generally, if the ability *Spanish speaking* is assigned to skill 127, for example, it follows that Agent skill 127 and VDN skill 127 both signify *Spanish speaking*. However, note that the agent skill might be assigned a skill term that is broader than that for the corresponding VDN skill. For example, Agent skill 127 might be labeled, *bilingual,* for agents that can handle calls in English as well as Spanish.

Skills for an application are shown in the following table, which presents a very abbreviated example of such a skill distribution for an automobile club.

Table 15: Example of a skill table for an automobile club

Supergroup-99	
Emergency road service-bilingual-22	Route planning-bilingual-44
Emergency road service-English-11	Route planning-English-33

In the table shown above, five skills are defined. Each skill indicates knowledge or an ability on the part of the agent or a need for knowledge on the part of the caller. One or more of these skills can be attributed to the agent according to the agent's expertise with the corresponding highway services and his or her language-speaking ability. Similarly, one or more of these skills can be considered needs on the part of the caller.

The table shown above, is arranged in such a manner that the agents at the top level have the broadest knowledge, that is, these agents can handle emergency road service and route planning calls and can speak Spanish. The top level (skill group) here is called Supergroup, and it contains agents who, as a group, can take any type of call regarding the automobile club. Accordingly, this skill group serves as a backup skill group. As you descend through the table, each sublevel corresponds to a group of agents who have more specific skills and can therefore take more specialized calls.

Calls can be distributed to the most-idle agent by using either the Uniform Call Distribution (UCD) option or the Expert Agent Distribution (EAD) option. UCD distributes calls from the skill hunt group to the most-idle agent who has this skill assigned at any priority level. This scenario provides a more even distribution to calls and therefore keeps agents equally busy. EAD distributes calls from the skill hunt group to agents to an available agent who has the highest skill level. Skills that are assigned to an agent at higher skill levels indicate a higher level of expertise or preference by the agent than any lower skill level skills that are assigned to that agent. EAD distribution provides the caller with the best or most expert agent match.

Agents are usually given a preference for higher skill level calls. However, the system can be administered to give agents a preference for the greatest need call. The greatest need call is the highest priority oldest call waiting for any of the agent's skills.

Multiple Call Handling on Request and Forced Multiple Call Handling make it possible for an agent to receive additional ACD calls either after putting a call on hold, or when active on another ACD call. Forced Multiple Call Handling can be used to give priority to an ACD call over an in-progress non-ACD call, or to give priority to a call from one skill over an in-progress call from a different skill. For more information, see Avaya Aura™ Communication Manager Feature Description and Implementation.

To administer skills, set the Skill, ACD, and Vector fields to y. Instructions for completing the Hunt Group screen are included in Administering Avaya Aura™ Communication Manager.

VDN skills

EAS enhances the Call Vectoring and Automatic Call Distribution features of the switch by distributing incoming calls based on:

- Specific skills that are assigned to a VDN or used in a vector, and
- Skills that are assigned to an agent

For example, a caller dials a particular number (VDN). The VDN uses a vector to queue the call to an agent with a skill that matches the VDN skill.

You can assign up to three different skills to a VDN in an order that meets your callers' needs. The first skill assigned to a VDN might be the skill that is required to best meet the needs of the customer who called the VDN. The second and third skills assigned to the VDN might represent backup skills that can also meet the callers' needs.

Skills that are administered to a VDN are commonly called VDN skill preferences. VDN skill preferences are labeled 1st, 2nd, and 3rd.

Note:

While skills can be optionally assigned to VDNs, the vector controls when and to what VDN skill the call queues. The application of VDN skills is described later.

The following table shows how skill preferences can be assigned to the five VDNs that are used for the automobile club that we discussed earlier. For each VDN, the corresponding call type and the number of the vector to which the VDN points are indicated. For a description of each skill, see Example of a skill table for an automobile club.

Table 16: Example of VDN skill preferences assignments

	Skill Preferences				
Call type	**VDN**	**1st**	**2nd**	**3rd**	**Vector**
General number	6005				1
Emergency Road Service (English)	6001	11	22	99	3
Emergency Road Service (Spanish)	6002	22		99	2
Route Planning (English)	6003	33	44	99	3
Route Planning (Spanish)	6004	44	99		2

In the table shown above, note that two VDNs point to Vector 3, two VDNs point to Vector 2, and one VDN points to Vector 1. Note also that a 1st and 3rd VDN skill Preference, but no 2nd VDN skill Preference, are assigned to VDN 2222. This implies that the call to this VDN (if not already answered) will wait longer before queuing to the backup skill (Supergroup-99, in our example), provided that the vector is designed to execute accordingly.

The following table shows the skill preferences that are assigned for one specific VDN (6003) that is used for the automobile club:

Table 17: Skill preferences assignments for VDN 6003

Preference	Number	Description
1st:	33	Directed to an agent who is knowledgeable about Route Planning and speaks English
2nd:	44	Directed to an agent who is knowledgeable about Route Planning and is bilingual
3rd:	99	Directed to an agent who can field all calls

In the table shown above, the first VDN skill preference corresponds to a knowledge area that could be considered a subset of the knowledge area that is represented by the second and the third preference. Similarly, the second VDN skill Preference corresponds to a knowledge area that could be considered to be a subset of the knowledge area that is represented by the third preference. Such an approach is commonly used to assign VDN skill preferences. The result of this approach is that the longer a call waits, the larger the pool of agents that the ACD considers for handling the call.

Recall that the vector numbers for each VDN associated with the automobile club are listed in Example of VDN skill preferences assignments. VDN 6003 points to Vector 3. As such, the skill requirements that are associated with the VDN are forwarded to the vector. This process is shown in the following figure.

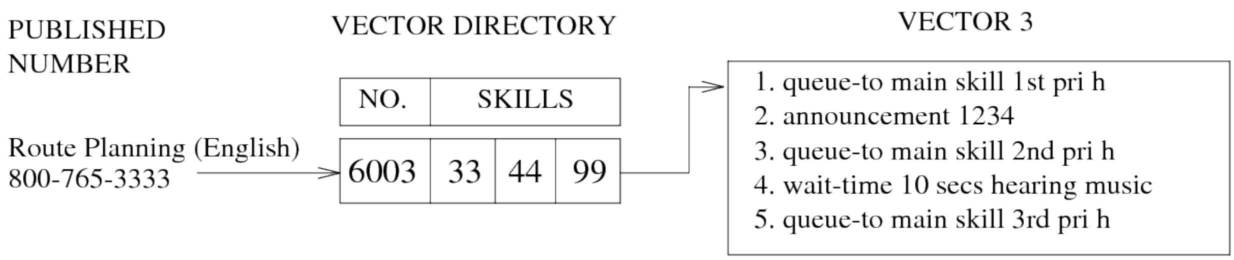


Figure 1: Example of VDN skill implementation

Assume that the English-speaking caller needs information on route planning and dials the appropriate number (800-765-3333). Network 800 features direct the call to 6003 (a VDN), the call enters the switch and is directed to VDN 6003, which points to the appropriate vector. As shown in Table 17: Skill preferences assignments for VDN 6003 on page 185, VDN skill Preferences 33, 44, and 99 are administered as the 1st, 2nd, and 3rd skill preferences, respectively, for VDN 6003.

Vector processing of this application is described in Delivering the call to the skill queue example on page 190.

Vector Directory Number (VDN) screen

The Vector Directory Number (VDN) screen shown in the following example is used to administer VDN skills.

Vector Directory Number (VDN) screen, page 1

```
change vdn xxxxx                                                     page 1 of 2
VECTOR DIRECTORY NUMBER

                            Extension: 2001
                                 Name: vdn 2001
                        Vector Number: 1

                  Attendant Vectoring? n
                   Allow VDN Override? n
                                  COR: 1
                                   TN: 1
                             Measured: internal
       Acceptable Service Level (sec): 20
              Service Objective (sec):

        VDN of Origin Annc. Extension:
                            1st Skill:
```

```
                            2nd Skill:
                            3rd Skill:
```

Vector Directory Number (VDN) screen, page 2

```
change vdn xxxxx                                                           page 2 of 2
VECTOR DIRECTORY NUMBER

                           Audix Name:
                Messaging Server Name:
                   Return Destination:
               VDN Timed ACW Interval:
                      BSR Application:
         BSR Available Agent Strategy: 1st-found
             Observe on Agent Answer?: n
```

Note:

Skills can be optionally assigned to VDNs, however, the vector controls when and to what VDN skill the call queues.

Complete instructions for completing the screen are included in Administering Avaya Aura™ Communication Manager.

Call Vector screen

Completion of the Call Vector screen is required for using vectors with EAS. The screen contains three pages. However, if the vector contains 11 or fewer instructions, you need to complete only the first page of the screen, as shown in the following example.

Call Vector screen (Page 1 of 3)

```
change vector 20                                                    Page 1 of 3

                           CALL VECTOR
 Number: 20                Name:_______________________
Multimedia? n     Attendant Vectoring? n          Lock? y
     Basic? y  EAS? y   G3V4 Enhanced? n  ANI/II-Digits? n ASAI Routing? n
 Prompting? n  LAI? n  G3V4 Adv Route? n   CINFO? n    BSR? y   Holidays? y

 01 _______________
 02 _______________
 03 _______________
 04 _______________
 05 _______________
 06 _______________
 07 _______________
 08 _______________
 09 _______________
 10 _______________
 11 _______________
```

Note:

Skills can be optionally assigned to VDNs, however, the vector controls when and to what VDN skill the call queues.

Instructions for completing the Call Vector screen are provided in Administering Avaya Aura™ Communication Manager, and in Creating and editing call vectors.

Agent skills

Agents are trained or hired to accommodate specific caller needs. Agent skills represent and define the ability of the agent to handle calls that require these skills. Agents are assigned skill numbers that are based on such characteristics as training or knowledge, access to systems or information, language ability, and interpersonal traits. Examples of agent skills include the following: speaks Spanish, knows about widget X, can handle complaint calls, has access to a database, and so forth.

You can assign up to 120 skills (with EAS-PHD) or 4 skills (without EAS-PHD). Each of these skills can be designated a skill level between 1 and 6 (EAS-PHD) or 1 and 2 (EAS), with 1 being the highest skill level, which is the highest-priority skill.

If an agent has multiple skills, a single skill group can be created for each set of skills. Agent skills are assigned to agents by completing the Agent Login ID screen. For more information, see the ACD login ID dialing on page 197.

It is highly recommended that you create a separate skill hunt group for direct agent calls. Direct agent calls are queued to the skill that is administered as the direct agent skill on the Agent LoginID screen. If an agent is not able to log in to his or her direct agent skill, direct agent calls are queued to the first-administered highest-level skill.

The following table shows the assignment of agent skills. For a description of the skills, see Example of VDN skill preferences assignments.

Table 18: Example of agent skill assignments

Agent	Skills assigned			
Jan O'Hara	22 (L1)	44 (L2)		
Sam Lopez	99 (L1)			
Sue Carlson	22 (L1)	11 (L1)	44 (L2)	33 (L2)
Mark Davis	44 (L1)			
Amy Brown	44 (L1)	22 (L2)		

Without EAS-PHD a maximum of four agent skills may be assigned to any one agent with one of two preference levels. With EAS-PHD up to 120 skills can be assigned to each agent with one of sixteen preference levels. The skill assignments table shows that four agent skills (22, 11, 44, 33) are assigned to Sue Carlson. These assignments indicate that Sue is bilingual and can service callers who need emergency road service or information on route planning. Only one agent skill (99-Supergroup) is assigned to Sam Lopez. This means that Sam is serving only as a backup.

A L1 or L2 next to the skill number indicates whether the agent skill is assigned as a level 1 or level 2 skill. For example, Jan O'Hara has**Emergency Road Service-Bilingual** as a level one skill and**Route Planning-Bilingual** as a level two skill. This means that whenever Jan O'Hara becomes available for an ACD call, provided that the Call Handling Preference is skill-level, the ACD software first looks for English-speaking callers who are requesting information on emergency road service from the agent. Only if there are no callers requesting emergency road service does the ACD software look for English-speaking callers who are requesting information on route planning. If the Call Handling Preference is greatest-need, Jan O'Hara receives the highest priority, oldest call waiting for either**emergency road service** or**route-planning bilingual** each time that she becomes available.

For any given application, EAS puts no restrictions on which agent skills can be assigned to an agent.

Note:

Agent skills are administered by completing the Agent Login ID screen. This screen is shown in ACD login ID dialing on page 197. Complete instructions for completing the screen are provided in *Administering Avaya Aura™ Call Center Features*.

Preference Handling Distribution

Preference Handling Distribution enables an agent to take calls based on either skill level or greatest need.

If an agent's call-handling preference is by skill level, the agent receives the call that requires the skill for which the agent's skill level is highest.

If an agent's call-handling preference is by greatest need, the agent receives the highest-priority, oldest call waiting that requires any of the agent's skills.

It is recommended that in any skill, all agents have the same call handling preference. This ensures the most consistent distribution of calls by either greatest need or skill level.

Preference Handling Distribution Examples

The following table is an example of how calls queue with Preference Handling Distribution.

Table 19: Preference Handling Distribution

Agent is assigned skills and skill levels...	These calls are in queue...
Skill 11; skill level 1	Waiting 15 seconds; priority medium
Skill 21; skill level 8	Waiting 30 seconds; priority low
Skill 31; skill level 16	Waiting 45 seconds; priority medium

Logical Agent capability

With Logical Agent and EAS, calls are routed to agents based on the login ID instead of the extension number that is assigned to the telephone. The agent's login ID must be consistent with the dial plan of the switch. When an agent logs in to an extension, the login ID overrides the extension as far as ACD tracking and characteristics, such as name and class of restriction (COR) are concerned.

When a specific login ID is called, the switch routes the call to the telephone that the agent is currently logged in to. Logical Agent allows agents to be called regardless of the telephone the agent is using. Calls to agent login IDs can be delivered as direct agent calls with the proper COR set for both the originating and the receiving login ID/facility.

Agents are not assigned to skill hunt groups with Logical Agent. Instead, an agent has specific skills that are assigned to his or her login ID. When an agent logs in, the agent is associated with the assigned skill hunt groups and tracking begins for the assigned skills.

Note:

Avaya CMS automatically measures a logical agent who is administered with at least one measured skill when the agent logs in.

Logical Agent uses a single set of work-mode buttons for all skills. This means that an agent is available or in AUX work for all skills at the same time. An agent cannot be available in some skills and in AUX work in others.

The telephone's button assignments and automatic answer options do not follow the agent because they are associated with the physical extension and not the agent login ID.

Note:

Converting to EAS may require a change to the CMS login ID if the current ID is not a valid extension number or cannot be made available in the switch dial plan. Agent login IDs are assigned names from the Dictionary-Login Identification window by way of Avaya Supervisor. Login IDs must be different from the telephone extensions.

Delivering the call to the skill queue example

This example shows how a call is delivered to a skill hunt group queue by vector processing.

The skills that are assigned to a VDN define the requirements in the vector for routing calls to an ACD agent with a particular set of skills. These skills become active for an ACD call whenever a **`queue-to skill`** command is executed. The skills also become active whenever a **`check skill`** command is executed and the threshold condition is met. Once a skill is active for an ACD caller, the call cannot be delivered to an available ACD agent unless the agent also has one of the active VDN skills.

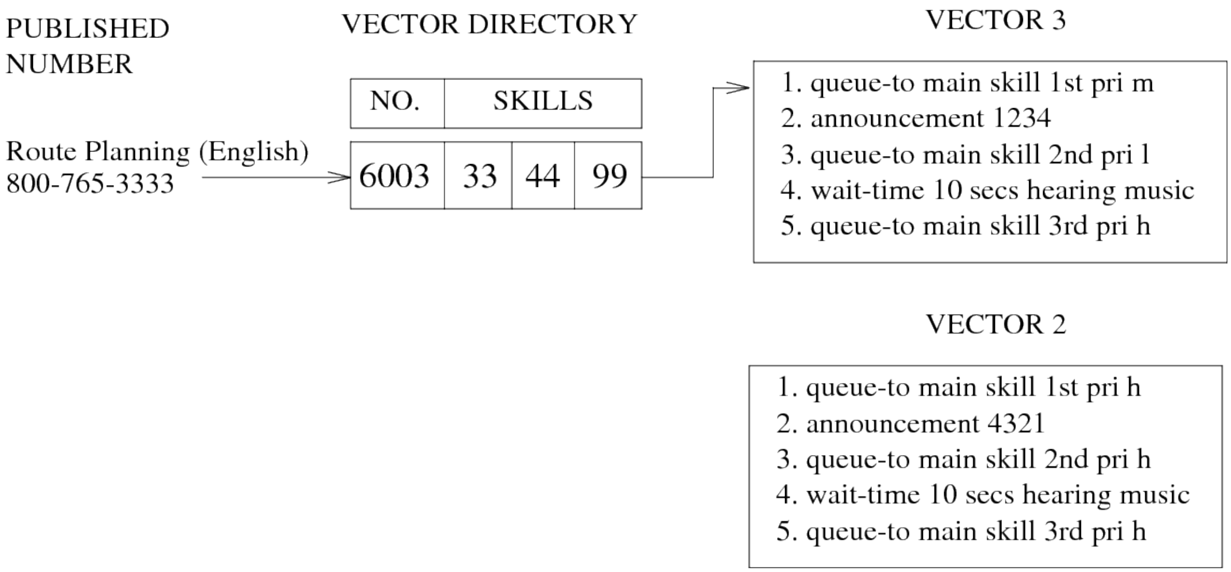


Figure 2: Process for delivery of a call to a skill queue

The process shown above assumes that an English-speaking caller needs information on route planning and dials the appropriate number (800-765-3333). In this case, the call enters the switch and is directed to VDN 6003, which points to Vector 3. Once vector processing starts, the **`queue-to skill`** command in step 1 queues the call to the skill hunt group that corresponds to the 1st VDN skill (33-Route Planning-English). If an agent with skill 33 is available, this agent answers the call. If such an agent is not available, the call is eventually queued to the skill hunt group that corresponds to the 2nd VDN skill (44-Route Planning-Bilingual) by the **`queue-to skill`** command in step 3. This time, if an agent with skill 44 is available, this agent answers the call. If the call is still not answered, the call is eventually queued to the skill hunt group that corresponds to the 3rd VDN skill (99-Supergroup) by the **`queue-to skill`** command in step 5.

In the process shown above, Vector 2 would be executed if a Spanish-speaking caller had called into the switch. Accordingly, the announcement that is provided in Vector 2 is in Spanish, whereas the announcement in Vector 3, which is executed in our example, is in English.

Note also that each of the queue-to skil commands in Vector 2 queues the call at a high priority, whereas only one of the queue-to skill commands in Vector 3 queues the call at this high a priority level. The strategy presented here is valuable when there is a limited number of bilingual agents because the bilingual such agents will be available more quickly to service callers who speak only Spanish.

VDN skills can also be used in **`check skill, messaging skill,`** and **`converse-on skill`** commands. Within any of these commands, a specific skill number can be used instead of a VDN skill Preference, provided that the relevant skill hunt group is correctly administered. For example, step 5 might have read queue-to skill 99 pri h. This concept is discussed further in Super agent pool on page 193.

Example using Call Prompting

The procedure that is described in the previous section can be enhanced by using Call Prompting. For example, the user can dial a general telephone number whose VDN points to a Call Prompting vector.

Staying with our automobile club example, recall that in Examples of services and corresponding DNIS digits on page 180, we define 800-765-5555 as the general telephone number for the service. Recall also that in Example of VDN skill preferences assignments we identify 6005 as the VDN for this 800 number. Also, we indicate that VDN 6005 points to Vector 1.

The following vector shows how Vector 1 might appear.

Call Prompting vector for the automobile club

```
1.  wait-time 0 seconds hearing ringback
2.  collect 1 digits after announcement 5678
    [
For emergency road service, dial 1.


Para asistencia con su automovil, marque el dos.

For travel route directions, dial 3.


Para informacion sobre rutas, marque el cuatro.
]
3. route-to number 6001 with cov n if digit = 1
    [English Emergency Road Service VDN]
4. route-to number 6002 with cov n if digit = 2
    [Bilingual Emergency Road Service VDN])
5. route-to number 6003 with cov n if digit = 3
    (English Route Planning VDN)
6. route-to number 6004 with cov n if digit = 4
    (Bilingual Route Planning VDN)
7. route-to number 6002 with cov n if unconditionally
    [Bilingual Emergency Road Service VDN]
```

Once the caller dials 800-765-5555, the call enters the switch and is directed to VDN 6005, which points to our Call Prompting vector. At this point, vector processing begins. Step 1 provides ringback if the caller has to queue for the announcement in step 2. The **`collect digits`** command in step 2 first provides an announcement that requests the caller to dial 1, 2, 3, or 4, depending upon the caller need and the caller's language speaking ability. If the caller dials a digit that is other than one of the four specified, each of the **`route-to...if digits`** commands in steps 3 through 6 fails, and control is passed to the **`route-to...if unconditionally`** command in step 7, which unconditionally routes the call to VDN 6002. This VDN is assigned the *bilingual emergency road service* skill and points to Vector 2, which is provided in the previous section.

Now we return to the collect digits step and assume that the caller dials 4. In this case, steps 3 through 5 fail because the required digit (1, 2, or 3, respectively) was not dialed. Thereafter, control is passed to step 6, where the **`route to...if digit`** command finds a digit match and consequently routes the call to VDN 6004. This VDN is assigned the *bilingual route planning* skill and also points to Vector 2, which is provided in the previous section.

Note:
VDN Override applies to the skills that are assigned to the VDN. For more information, see VDN Override.

Super agent pool

EAS allows a skill hunt group to function as a super agent pool. A super agent pool is a backup group of one or more agents that is able to handle many if not all types of calls coming into the application. In our automobile club examples, Skill Hunt Group 99 (Supergroup) serves as a super agent pool. Also, you might recall that 99 appears as both a VDN skill and an Agent skill. However, a super agent pool can be assigned a skill hunt group number that is not assigned to a VDN skill. This can and should be done whenever the application requires four levels within the skill table distribution, as shown in the following table.

Table 20: Modified skill table for the automobile club

Supergroup-99			
Emergency road service- bilingual-88		Route planning-bilingual-77	
English-66	Spanish-55	English-44	Spanish-33
Bostonian-11	Castilian-13	Bostonian-15	Castilian-17
New Yorker-12	South American-14	New Yorker-16	South American-18

Besides a new skill numbering scheme, our modified skill table has four levels instead of the three levels that are provided in Example of a skill table for an automobile club on page 183. Except for the skill numbering scheme, the top two levels (Supergroup-99 and Emergency Road Service-Bilingual-88/Route Planning-Bilingual-77) remain unchanged. However, note that the next level is reorganized into segments to indicate the ability to speak English or Spanish. Finally, note that a new level is added to denote particular types of accents or pronunciation in English and Spanish.

The following table shows how some of the skills in Table 20: Modified skill table for the automobile club on page 193 are administered to one relevant VDN (VDN 1616).

VDN 1616 - Skill preferences		
1st:	16	Knows about Route Planning, speaks English, has New York accent
2nd:	44	Knows about Route Planning, speaks English
3rd:	77	Knows about Route Planning, is bilingual

Now we are ready to consider the following vector to accommodate a super agent pool.

Modified vector to accommodate a super agent pool

```
1. queue-to skill 1st pri m
2. announcement 4555
3. queue-to skill 2nd pri l
4. wait-time 10 seconds hearing music
5. check skill 3rd pri l if calls-queued < 3
```

```
6. announcement 4666
7. check skill 99 pri l if available-agents > 0
```

Assume an English-speaking caller needs information on route planning and want to speak to an agent with a New York accent. In this case, the caller dials the appropriate number (800-765-1616, for example). Accordingly, the call enters the switch and is directed to VDN 1616, which points to the vector in the previous screen. Once vector processing starts, the **queue-to skill** command in step 1 queues the call to the skill group that corresponds to the 1st VDN skill (New Yorker-16). If an agent with skill 16 is available, this agent answers the call. If such an agent is not available, the call is eventually queued to the skill group that corresponds to the 2nd VDN skill (English-44) by the **queue to main skill** command in step 3. This time, if an agent with skill 44 is available, this agent answers the call. If the call is still not answered, the **check skill** command in step 5 attempts to queue the call according to the parameter indicated (if calls-queued < 3) to the skill group that corresponds to the 3rd VDN skill (Route Planning-Bilingual-77). If the call is queued, and if an agent with skill 77 is available, this agent answers the call. If the call is not queued, or if it is queued and an agent with skill 77 is not available, the **check skill** command in step 7 is executed.

Before we discuss the execution of step 7, note that a specific skill hunt group number (99) and not a VDN skill Preference designation (1st, 2nd, or 3rd) is included within the **check skill** command. Since the skill table for the application involves four levels of skills, and since there can be no more than three VDN skills, the specific skill group number (99) for the super agent pool must be included within the queuing command to allow caller access to the pool. Whereas a VDN skill is always represented in a vector by the term 1st, 2nd, or 3rd, a super agent pool is always represented by a whole number according to the parameters of the relevant switch. For the queueing commands, see Call Vectoring commands.

Returning to the vector execution, the **check skill** command in step 7 attempts to queue the call according to the parameter that is indicated (if available-agents > 0) to the super agent pool (Supergroup-99). If the call is queued, and if an agent in the super agent pool is available, this agent answers the call.

Note:

If the call has already queued to all three VDN skill hunt group preferences, it does not queue to the specific skill hunt group. This reflects the restriction that a call can only queue to a maximum of three splits or skills. The best approach is to test the splits/skills first to determine where to queue the call. Also see Expected Wait Time (EWT) on page 448.

Routing of the call to an agent

With EAS optioned, an agent becomes associated at login with one or more skill hunt groups. A single set of work mode buttons applies to all the skills that are assigned to a logged-in agent. For example, if the agent selects Aux Work, the agent is in Aux Work for all the skills associated with the agent. Therefore, logged-in agents need only a single set of work-mode buttons for all relevant skill hunt groups.

Calls can be routed to the agent from a skill hunt group by dialing an agent login ID or by dialing an agent telephone extension directly.

Delivery from a skill hunt group

An incoming call is matched to an agent who has at least one of the three VDN skills that are required to handle the call. This matching is done by ACD queuing and the **`queue-to skill`**, **`check skill`**, **`messaging skill`**, or **`converse-on skill`** commands in the vector. If more than one agent is available for a call, the call is delivered according to whether EAD or UCD is administered for the skill hunt group.

For any one login session, an agent can have a maximum of four skills, or a maximum of twenty skills with EAS-PHD. Each agent skill is administered with a skill level.

Remember that when the Call Handling Preference is administered as greatest need, the agent receives the highest priority oldest call waiting for any of the agent's skills. If the Call Handling Preference is skill-level, the ACD software distributes the call that is waiting for the agent's highest skill-level skills whenever the agent becomes available. If no calls are waiting for the highest skills, the queued calls for the next highest skills are distributed to the agent, and so on. The following scenario describes call distribution when the Call Handling Preference is skill level.

Once an agent becomes available, he or she receives a waiting call in the following order:

1. Oldest direct agent call waiting for the agent if the direct agent skill is administered at the agent's highest skill level
2. Oldest call waiting at the highest priority for the highest skill-level skill
3. Oldest call waiting at the next highest skill-level skill, and so on.

For example, assume that Jill is the only agent with skills 22 (L1), 13 (L1), 23 (L1) and 47 (L2). Also assume that, while Jill is in AUX work mode, five calls are queued, as shown in The following table, which also shows the skill level and priority level that are associated with each call:

Table 21: Example of skill call queue sequence

Call	Time in queue	Skill number	Priority level
A	8:00	13	Medium
B	8:01	47	Top
C	8:02	23	Direct Agent
D	8:03	22	Top
E	8:04	22	Medium

Given this scenario, the next table indicates and explains the order in which Jill handles the five calls.

Table 22: Example of skill call distribution for a single agent

Call handled	Reason
C	Only direct agent call queued at highest level skill.

Call handled	Reason
D	Oldest call waiting at the highest priority for highest skill-level skills (Call B has the same priority level (Top), but it is assigned a lower skill level (47). Also, Call E has the same skill (22), but it has a lower priority level (Medium) and has not been waiting as long as Call D).
A	Oldest call waiting at the highest priority level for highest skill-level skills (Call E also has a primary skill (22) and the same priority level as Call A, but Call A has been waiting four minutes longer than Call E).
E	Only remaining call with the highest skill level (22) (Call B has a lower skill level (47)).
B	Last remaining call, and the only one that has the lower skill level (47).

If no calls are waiting when an agent becomes available, the agent is placed into the agent queue according to the call distribution method that is in effect. For UCD, the agent is placed at the bottom of the most-idle agent queue. For EAD, the agent is placed at the bottom of the agents with the same skill level.

The following table shows a call scenario that is valid for either UCD or EAD.

Table 23: Example of UCD/EAD call scenario

Time	Event	Skills
9:00	Jill logs in	22(L1), 13(L1), 47(L2)
9:01	Jill available	22(L1), 13(L1), 47(L2)
9:02	Jack logs in	22(L1), 47(L1)
9:03	Jack available	22(L1), 47(L1)
9:04	Call A arrives	47
9:05	Call A drops	47
9:06	Call B arrives	13
9:07	Call B drops	13
9:08	Call C arrives	22

Given the scenario presented above, the following table shows how Calls A, B, and C are distributed by UCD and EAD:

Table 24: Example of call distribution by UCD and EAD

Time	UCD or EAD?	Result	Reason
9:04	UCD	Jill receives Call A.	Jill is the most idle agent for skill 47.

Time	UCD or EAD?	Result	Reason
	EAD	Jack receives Call A.	Jack is the more expert agent because he has skill 47 as a level 1 skill whereas Jill has skill 47 as a level 2 skill.
9:06	UCD	Jill receives Call B.	Jill is the only agent who is logged in to skill 13.
	EAD	Jill receives Call B.	Jill is the only agent with skill 13.
9:08	UCD	Jill receives Call C.	Jill is the most idle agent for skill 22. She receives Call C even if she handled Call A.
	EAD	Jill receives Call C.	Both Jill and Jack have skill 22 as a level 1 skill, but Jill has been logged in 2 minutes longer than Jack; that is, she is the most idle agent.

ACD login ID dialing

The ACD login IDs used in EAS are extension numbers that are included in a station numbering plan but not administered as stations. These IDs are administered by using the Agent Login ID screen, as shown in the following example. If EAS-PHD is not optioned, you can only administer four skills.

Agent Login ID screen

```
add agent-loginID 9011                                          Page  1 of  1
                             AGENT LOGINID

                Login ID: 9011_                                   AAS? _
                    Name: _________________________             AUDIX? _
                      TN: 1__                           LWC Reception: spe
                     COR: 1                  AUDIX Name for Messaging: _______
           Coverage Path: ____   Messaging Server Name for Messaging: _______
           Security Code: ____               LoginID for ISDN Display? n
      Direct Agent Skill: ___                                 Password: _________
Call Handling Preference: skill-level        Password (enter again): _________
       Service Objective? _                               Auto Answer: ______

   SN  RL SL  PA      SN  RL SL  PA       SN  RL SL  PA       SN  RL SL  PA
1: __  _  __  ___  6: __  _  __  ___  11: __  _  __  ___  16: __  _  __  ___
2: __  _  __  ___  7: __  _  __  ___  12: __  _  __  ___  17: __  _  __  ___
3: __  _  __  ___  8: __  _  __  ___  13: __  _  __  ___  18: __  _  __  ___
4: __  _  __  ___  9: __  _  __  ___  14: __  _  __  ___  19: __  _  __  ___
5. __  _  __  ___ 10: __  _  __  ___  15: __  _  __  ___  20: __  _  __  ___

      WARNING:  Agent must log in again before skill changes take effect
```

With EAS, an agent's ACD login ID is associated with a specific telephone only when the agent actually logs in at that telephone. When the agent logs off, the association of the agent's ACD login ID with a specific telephone is removed. If an agent does not answer a call, or if the agent is logged out, the call goes to the busy points on the coverage path.

When the agent logs in, the telephone display indicates the agent's skill assignments.

The agent logs in by doing the following:

- Going off-hook or selecting a line appearance
- Upon hearing the dial tone, entering the login Feature Access Code (FAC) or selecting the Login Abbreviated Dialing button
- Upon hearing the dial tone, entering the 1-digit to 5-digit login ID

  Note:

  If someone is already logged in at that telephone, the agent hears an intercept tone.
- Upon hearing the dial tone, entering (optionally) the 0-digit to 9-digit password.

Note:

If the agent is using a DCP telephone (such as a Callmaster), then the password digits are not shown unless an abbreviated dial button is used. BRI telephones show the password digits.

Once the login is accepted, confirmation tone is given. Also, the skills that are assigned are displayed for 5 seconds on the telephone display. If more skills are assigned than can be displayed, a plus sign (+) appears at the end of the display. If a skill is administered but the agent was not logged in to the skill, the skill number is displayed with a star (*). The previous login sequence allows an ACD call to be directed to a specific agent and to have that call tracked and treated as an ACD call.

When an EAS agent logs in to a station with the station administered for audible message waiting, the agent receives an Audible Message Waiting tone only when calls are waiting for the agent login ID extension. When the agent logs out, Audible Message Waiting tone then applies again to messages that are waiting for the physical extension. This field has no impact on whether an agent hears the EAS Login-ID Message Waiting tone during the login process.

The message waiting lamp by default tracks the status of messages that are waiting for the logged-in EAS agent LoginID rather than messages for the physical telephone. The operation of the Message Waiting Lamp can be changed so that it tracks the status of messages that are waiting for the physical telephone where the agent is logged in. For more information, see the Feature-Related System-Parameters screen in Administering Avaya Aura™ Communication Manager.

Other agent login capabilities

In addition to skill assignments, the following capabilities are associated with agents' login IDs.

Call routing

A call to the login ID reaches the agent independent of the telephone that the agent is currently using. In other words, such a call is sent to the telephone at which the agent is currently logged in.

If the proper Class of Restrictions (COR) is set, callers can initiate a direct agent call either by dialing the login ID extension directly or by calling a VDN that points to a vector that contains first a prompt for the login ID and then a **`route-to digits`** command. This allows external callbacks by way of Direct Inward Dialing (DID) or an 800 number. Both the receiving agent's login ID COR and the originator's (caller's) COR must have Direct Agent Calling (DAC) set to y. The caller's COR is for the following:

- Telephone extension (for internal calls or transfers)
- Trunk group (for DID calls)
- VDN (for prompted calls)

If the call covers or is forwarded, the COR of the originator (or VDN) and the final agent is used. All feature functionality for ACD calls, except Queue Status indications, is available for direct agent calls.

Internal and external users can originate direct agent calls by dialing the agent's login ID. Also, DAC can be used to transfer ACD calls from one agent to another agent.

If an agent who is receiving the direct agent call is staffed but unavailable, the call waits in front of the skill calls in the skill that is administered as the agent's direct agent skill until either the call is answered or a coverage timeout occurs. Also, the caller hears an optional direct-agent announcement that is followed by music or silence. There is one direct agent announcement per system. The agent, on the other hand, receives a ring-ping, and the current work mode button flashes. If the agent is available, the call is delivered to the agent according to the answering and ringing options. Calls are answered and handled in the same manner as ACD calls. For more information, see the Feature-Related System-Parameters screen in Administering Avaya Aura™ Communication Manager.

Login ID name on the telephone display

A call to a logged-in EAS login ID by default displays the name associated with the login ID and not the name that is associated with the telephone. This is also true on the receiving party's display for a call that is made from a telephone with an agent logged in. However, the user can display the name of the physical telephone where the EAS agent is logged in. The user must be active on a call with the agent, and must have a telephone with an alphanumeric display and an inspect button. When the inspect button is pressed during a call to or from the EAS agent, the physical telephone name of the agent is displayed.

Coverage path

Call coverage can occur whether or not the agent is logged in. If the agent is not logged in, the busy criteria is met and the call follows the points on the coverage path. If the agent is logged in but fails to answer, the don't answer criteria is met and the call follows the points on the coverage path. A call to the login ID goes to the coverage path that is assigned to the login ID rather than to the coverage path that is assigned to the telephone extension.

Agent restrictions

A call to the login ID or from the agent uses the restrictions that are associated with the agent and not the telephone.

Telephones are fully functional if an agent is not logged in. The restrictions, coverage, and name revert to the telephone administration when the agent logs out.

If a number of users are sharing one telephone (due to job sharing or shifts, for example), a unique login ID extension is assigned to each user. Therefore, whenever a user is logged out, any calls to that user (login ID) are sent to his or her coverage path. As a result, login IDs can be used to reach people independent of where they happen to be. Such people include those who use more than one phone because they have more than one office or (in the case of security guards, for example) sit at more than one desk.

Because AAS/messaging-system ports are not mobile, these ports are administered to agent login IDs. Whenever the **AAS** or **AUDIX** field is set to `y`, a field that requests the port number is brought up, and the **password** field disappears.

EAS feature interactions

This section discusses the feature interactions that involve EAS. Unless otherwise specified, the feature interactions for skill hunt groups are the same as for vector-controlled splits.

Interaction	Description
Abbreviated Dialing	Abbreviated Dialing is used to log in or log out EAS agents. Abbreviated Dialing lists or buttons can be administered only for stations.
Administration Without Hardware	Although EAS login IDs are extensions without hardware, they are not a part of the Administration Without Hardware (AWOH) feature.
Agents in multiple splits feature	With EAS, the Agents in Multiple Splits feature is called Agents in Multiple Skills. This feature allows an EAS agent to be logged in to multiple skills.
Agent work modes	With EAS optioned, an agent can be in only a single work mode for all skills at any one time. For example, an agent cannot be in AUX work mode in one skill hunt group and also available in another skill hunt group. Also, if the After Call Work (ACW) mode button is selected, the agent is placed into ACW for the first skill that is administered and logged in to.
Assist	This feature is used for skill hunt groups (that is, there is one supervisor per skill hunt group). A telephone can be administered with one or more Assist buttons for each skill that agents who are using the telephone might have. An Assist button can also be administered with no associated skill. In this case, the supervisor for the skill that the agent is currently active on is called. If the agent is not active on any skill, the supervisor for the agent's first skill is called. Any assist button that is selected is tracked as an assist for the current call, regardless of any skill that is assigned to the button. The administered association of an Assist button with a particular skill and assigned supervisor is not affected when an EAS agent logs in to that station.
Audible message waiting	If messages are waiting for an EAS agent login-ID extension, an agent hears a special 5-burst EAS Login-ID Message Waiting tone (instead of confirmation tone) after successfully logging in. This does not require Audible Message Waiting to be assigned to the telephone or the system. If Audible Message Waiting is optioned for the system and assigned to an agent's telephone, and messages are waiting for the agent login

Interaction	Description
	ID extension, the agent hears the Audible Message Waiting tone whenever the agent goes off-hook, or selects a line appearance and hears dial tone. Messages that are waiting for the physical extension do not cause an Audible Message Waiting tone when an EAS agent is logged in.
Auto-Available Skills	If a skill hunt group is administered as an Auto-Available Skill (AAS), the EAS login IDs that are assigned to this skill must also be administered as Auto-Available. When the switch reinitializes, these login IDs are automatically logged in with the auto-in work mode. If any switch features attempt to change the work mode to anything except auto-in, this attempt is denied. Agents cannot have both Auto-Available and Non-Auto-Available Skills. This feature is not intended for human agents.
Automatic answering with zip tone	This feature can be administered only for a physical extension. The feature is not associated with a LoginID.
BCMS	The BCMS user interface remains the same when EAS is optioned. The only change is that the labeling of the headings is changed from split to skill. When EAS is enabled, BCMS agent reports are based on the agent login IDs. BCMS tracks direct agent calls as skill calls. direct agent calls affect ACD talk time, ACW time, and Average Speed of Answer. Whenever direct agent calls are waiting, BCMS displays an asterisk (*) immediately after the CALLS WAITING column.
Best Service Routing (BSR)	EAS VDN skills (1st, 2nd, 3rd) can be used in consider split/skill commands. EAS skills levels are used for the EAD-MIA and EAD-LOA BSR Available Agent Strategies.
Bridging	ACD calls do not alert on bridged appearances. However, bridged users can activate features on behalf of agents. Features that can be activated include log in, log out, change work modes, and assist.
Call coverage	Call coverage can occur whether or not the agent is logged in. If the agent is not logged in, the busy criteria is met and the call follows the points on the coverage path. If the agent is logged in but fails to answer, the don't answer criteria is met and the call follows the points on the coverage path. A call to the login ID goes to the coverage path that is assigned to the login ID rather than to the coverage path that is assigned to the telephone extension.
Call Detail Recording (CDR)	For skill calls, the **called party** field can optionally be the agent login ID.
Call forwarding	Since they are vector-controlled, skill hunt groups cannot be call forwarded. EAS agent login IDs cannot be forwarded, but the physical extension where the EAS agent is logged in can be forwarded. If another station with console permissions tries to forward an EAS login ID, an intercept tone is given.

Interaction	Description
Call park	To retrieve a parked call by a Feature Access Code (FAC), the agent dials the Answer-Back FAC and the extension where the call is parked. If the person who is unparking the call dials the Answer-Back FAC and the physical extension of the station where the call is parked, he or she is connected to the parked call. In some cases, the person who is unparking the call may also be able to dial the Answer-Back FAC and the logical agent extension of the agent who parked the call. This operation is possible if the Class of Restriction (COR) of both the agent parking the call and the telephone or agent who is unparking the call have a COR with the DAC flag set to y. If the telephone that is unparking the call is not a logged-in agent, the telephone must have a COR with DAC set to y. If the station that is unparking the call is a logged in agent, then the COR of the logical agent extension must have DAC set to y.
Call pickup	Skill hunt group extensions and EAS login ID extensions cannot be members of a call pickup group.
Class of Restriction	Skill hunt groups do have a Class of Restriction (COR). The COR is used if the skill hunt group extension is called directly. The COR for an EAS agent login ID overrides the physical extension's COR of the telephone that an agent logged in to.
Class of Service	EAS agents do not have a COS associated with their login ID. Instead, the COS is associated with the physical extension. Therefore, the COS of the telephone is not affected when an EAS agent logs in to that telephone.
Dial plan	Agent login IDs are part of the dial plan, and they reduce the total number of stations.
Direct Agent Calling (DAC)	If a called EAS Agent login ID and the call originator (extension, trunk, or VDN) both have a COR that allows direct agent calls, the call to the login ID is treated as a direct agent call. A call to the telephone extension where an EAS agent is logged in, or a call to an EAS agent login ID where either the originator's or the login ID's COR does not allow direct agent calls, is treated as a personal (non-ACD) call.
Leave Word Calling	When an EAS agent is logged into a station, the agent can only retrieve LWC messages left for that agent's login ID. To retrieve LWC messages left for that station, the agent must log out. When an EAS agent is logged into a station, its Message lamp defaults to tracking the status of LWC messages waiting for the station. However, you can assign the Message lamp to track the status of LWC messages waiting for the agent's login ID.
Look-Ahead Interflow	Skills are not sent to another system when a call interflows using Look-Ahead Interflow (LAI). If skills have the same meaning on both ACDs, a LAI command to a VDN with the same skills assigned can provide a mapping of the skills.

Interaction	Description
Multiple Split Queuing	When EAS is enabled, the Multiple Split Queuing feature is called Multiple Skill Queuing, which has the same functionality. With Multiple Split/Skill Queuing, a call can queue to a maximum of 3 splits/skills.
OCM/EAS	If EAS is enabled on the switch, the Outbound Call Management (OCM)/Expert Agent Selection (EAS) feature is required for a CallVisor ASAI adjunct application to launch predictive Outbound Call Management (OCM) calls. Predictive Calling is an OCM feature that is often used in applications, such as sales or cold calling, where it does not matter which agent is accessed by a caller and for which it is important to keep the agents utilized fully. While OCM predictive calling is an outbound call management application, the EAS environment provides a number of desirable features for inbound call handling. The OCM/EAS feature allows the customer to enable both types of call handling on the switch. From a technical standpoint, if EAS is enabled, the feature is needed for the following reasons: • All skill hunt groups are vector controlled. However, to launch a predictive OCM call in a traditional ACD environment, the ACD split cannot be vector-controlled. • The traditional ACD environment and EAS cannot be enabled on the switch at the same time. The OCM/EAS feature extends the ASAI features to include launching predictive OCM calls from a VDN extension. Previously, ASAI hosts could launch predictive calls only from ACD split extensions. A limited number of Call Vectoring commands are supported in the VDNs that are used to launch or process OCM predictive calls. These commands are listed in the following section.

Commands for OCM predictive calls

EAS adjunct interactions

This sections describes adjunct interactions with the EAS feature.

ASAI interactions with EAS

ASAI support for EAS may be organized into the following categories: call control, feature requests, value queries, event notification, and adjunct-controlled skills. This section provides a high-level overview of the behavior of ASAI in the EAS environment.

Call control

Call-control capabilities work exactly the same in the EAS environment as in the traditional ACD environment except for the following:

- User-classified third-party make calls (calls classified by the originator) may originate from an EAS login ID and terminate to a login ID. User-classified calls that terminate to a login ID are given the same direct agent treatment that is provided for such calls that are dialed from a station extension.
- Switch-classified third-party make calls, which are classified by a call classifier board and delivered (when answered) to the originating hunt group, may originate from or terminate to EAS login IDs.
- Direct agent third-party make calls, which are ACD calls that are terminated to a selected member of an ACD skill group, may be requested by including a direct agent option, an agent's physical extension and a skill group extension (compatibility mode), or by requesting a user-classified third-party make call with a login ID destination. The primary differences between the two methods of requesting direct agent calls are that the compatibility mode allows the adjunct to specify the skill hunt group to which a given direct agent call is queued and that the non compatibility mode allows the adjunct to direct the call to a login ID, regardless of which station an agent is logged in to. Direct agent third-party make calls may not originate from an EAS login ID.
- Supervisor assist third-party make calls, which are supervisor assist calls that are originated by a selected member of an ACD split, may originate from an EAS login ID, and they may terminate to an EAS login ID. Unlike dialed direct agent calls, supervisor assist calls that are terminated to a login ID behave as though they have been previously directed to the requested login ID's physical extension. For example, they do not cover if the requested agent is not logged in and if the originator's display shows the agent's physical extension and not the agent's login ID.
- Extension (Domain) control may not be requested for an EAS login ID, but it may be requested on behalf of a Logical Agent's physical extension. Auto-dial calls, which are calls that are initiated by an extension-controlled station, may be terminated to an EAS login ID, in which case the call is given direct agent treatment.
- Adjunct-routing calls, which are vector calls that are routed by an ASAI adjunct by the **`adjunct routing link`** Call Vectoring command, are similar to third party make calls. Such calls may include a direct agent option, an ACD agent's physical extension, and a skill extension. If this is true, these calls are given compatibility mode direct agent treatment and may be terminated to an EAS login ID (in which case they behave like dialed direct agent calls.
- If EAS is optioned, ASAI launches OCM switch-classified or predictive calls from a VDN extension by the OCM/EAS feature. To launch a predictive call in a traditional ACD environment, an adjunct OCM application sends an ASAI request to the switch with an ACD split number as the originating number. The application also sends flags that identify the call as a switch-classified call. In the traditional ACD environment, the ACD split cannot be vector-controlled.

Feature requests

In the EAS environment, agent login, logout and change work-mode requests are fully supported. Agent login requests must contain an EAS agent login ID and optional password (delimited by ‘#’) in the login request’s user code IE. Agent logout requests and change work-mode requests may contain the desired agent’s physical extension or login ID. Call Forwarding and Send all Calls feature requests are denied for EAS login IDs but may be requested for EAS physical extensions where an EAS agent is logged in.

Multiple monitors

Multiple Monitors provides the ability for up to three ASAI applications to monitor the same ACD Split or VDN domain.

This is helpful in environments were OCM is primary and it can also be used to add an OCM application to launch calls at off-peak times without disrupting the primary application in any way. Multiple Monitors can also be used to monitor an ACD split over 2 links in call environments where ASAI link failure recovery is important.

Value queries

Value queries function identically in the EAS and traditional environments, except that the Extension Type/Class Information Query returns a new indication that a requested extension is an EAS login ID along with an indication of whether the login ID is currently logged in and where, in other words, at which physical extension.

Event notification

Because all skill hunt groups are vector controlled, event notification may not be requested on the basis of a skill hunt group extension. Event notification may, however, be requested on the basis of a controlling VDN extension. Generally, all event reports that involve EAS agents contain the agent’s physical extension rather than the agent’s login ID.

Adjunct-controlled skills

Agents with adjunct-controlled skills are considered to be adjunct-controlled agents. Adjunct-controlled agents exhibit the same behavior as agents within adjunct-controlled splits in the traditional ACD environment. The following list provides more details:

- Stations are locked for all logged-in adjunct-controlled agents. The only action an agent can take from the station is to go on hook (or unplug the headset) from an auto-answer station, which causes the agent to be logged out.
- Stations are unlocked whenever the controlling adjunct’s ASAI link stops functioning. Stations are locked again when the adjunct’s link is reestablished.
- The adjunct controls all skill and agent activities such as login, logout, and change work-mode (with the exception of agent logout using the telephone hook).
- Only adjunct-controlled calls can terminate to the extension of an adjunct-controlled agent.
- Only adjunct-controlled calls can terminate to an adjunct-controlled skill hunt group extension.
- Adjunct-controlled EAS Agents can be administered with only one skill. Accordingly, EAS agents may not mix adjunct-controlled and non-adjunct-controlled skills.

Messaging system

Calls to the EAS agent login ID can cover to the messaging system. Each agent must enter his or her agent login ID when calling the messaging system to obtain messages.

Messaging-system agents are assigned to EAS agent extensions. These login IDs are used for CMS and BCMS tracking if the associated messaging-system skill hunt group is externally measured. The aut-msg-wt button or message waiting light can be used to indicate that the login ID has a message.

An agent cannot have both messaging-system and non messaging-system skills.

CMS

Note:

CMS reports show only the first 15 skills that an agent is logged into.

The following items apply to Avaya CMS Agent Tables:

- Separate direct agent database items starting with DA_ are tracked.
- Standard reports combine statistics for direct agent calls and skill calls. However, reports can be customized to separate these statistical groupings.

The following is true for the CMS Skill Tables:

- Skill queues can be monitored for direct agent calls on the Queue/Agent Summary report.
- Direct agent calls are not tracked.
- Agent time while on a direct agent call is tracked as other time.
- Non-ACD calls while in direct agent ACW are tracked.

The following item is true for the CMS VDN/Vector Tables:

Direct agent calls and skill calls are combined as ACD calls.

Speech-processing adjuncts

Speech-processing adjuncts that have a line interface to the Communication Manager are able to initiate direct agent calls by dialing the login ID for an agent.

Listing Agents Logged into a Split or Skill

When administering a split or skill, you can use the `list members hunt-group` command to verify that all agents are logged out and to identify any agents who are logged in. You can list all logged in agents for a split or skill, or limit the list to a range of login IDs or physical extensions:

- list members hunt-group <hunt group nnn>
- list members hunt-group <loginid nnnn to-loginid nnnn>
- list members hunt-group <ext nnnn to-ext nnnn>

For example, to list the agents logged in to skill 37:

Type `list members hunt-group 37` and press `Return`.

The List Hunt Group Members report screen appears.

```
list members hunt-group 37

                           HUNT GROUP MEMBERS

Group Number: 37     Group Name: Platinum Card    Group Extension: 3002
  Group Type: ucd-mia       ACD? y     Skill? y           Members: 4

    Phys  Phys            Login Login          Agt     Per       Wrk
    Ext   Name            Ext   Name           Prf Lvl All SO DF Tim Occ
 1: 1002  1002-Al MacInni 2902  Agent 2902     grt 04      y      10  33
 2: 1022  1022-Kelly Chas 2901  Agent 2901     lvl 14      n      15  55
 3: 1001  1001-Chris Pron 2904  Agent 2904     pal R2         n    0   0
 4: 1021  1021-Maria Esta 2903  Agent 2903     pal 08   30    y   18  45
```

Note:

This screen shows a system using EAS and Avaya Business Advocate. For systems without either of these features, the related columns will be blank.

You can also use this command to list the agents *administered* in non-ACD hunt groups. However, since non-ACD hunt groups don't use agent logins the report will not identify agents who are currently active.

Upgrading to the EAS environment

For information on converting a call center to EAS, refer to Appendix P: Converting a call center to EAS.

Forced Agent Logout from ACW mode

About Forced Agent Logout from ACW mode

The Forced Agent Logout from After Call Work (ACW) feature automatically logs out an Expert Agent Selection (EAS) agent who spends too much time in ACW mode. The timeout period is specified on a per system basis and on a per agent basis. The timeout is reported with a customer-assignable reason code set on a system basis.

The per agent timeout setting takes precedence over the per system setting. For Auto-In agents, the Timed ACW feature takes precedence over the Forced Agent Logout from ACW feature.

Reasons to use Forced Agent Logout from ACW mode

This feature is typically used when customers want to:

- Require that agents not remain in ACW longer than a set time limit in order to monitor agents who exceed the time limit
- Logout agents who walk away from their position while in ACW mode

Prerequisites for using Forced Agent Logout from ACW mode

You can set Forced Agent Logout from ACW only if all of the following conditions are true:

- Expert Agent Selection (EAS) is enabled and active.
- The Reason Codes feature is active. If the Reason Codes feature is not active, you can still set the maximum time the agent can be in ACW on a system-wide and on an agent basis, but you cannot administer a reason for the logout.
- The Call Center release is 3.0 or later.

If any of these values are not true, you will not be able to change the default values on the Forced Agent Logout from ACW fields that enable the feature.

Administering Forced Agent Logout from ACW mode

The following forms are used to administer Forced Agent Logout from ACW mode.

Screen	Field
Page 13 of Feature-Related System Parameters	**Maximum time agent in ACW before logout (sec)** - Sets the maximum time the agent can be in ACW on a per system basis.
Agent Login ID	**Maximum time agent in ACW before logout (sec)**- Sets the maximum time the agent can be in ACW on a per agent basis and sets the reason code that explains why the system logged out the agent.

Note:
Changes do not apply until the agent logs out and logs back in again.

Tips for administering Forced Agent Logout from ACW mode

Consider the following tips when administering Forced Agent Logout from ACW mode.

If	Then
You want this feature to apply to all agents, and you want the timeout period to be the same for all agents	1. Set the**Maximum time agent in ACW before logout (sec)** field on the Feature-Related System Parameters screen to the desired timeout value in seconds. This value can be from 30 to 9999 seconds. 2. Leave the default setting of system on the Agent Login ID forms.
You want different timeout periods assigned to specific agents	Set the**Maximum time agent in ACW before logout (sec)** field on theAgent Login ID screen to the desired timeout value for each agent.
You do not want the timeout feature to apply to certain agents	Set the **Maximum time agent in ACW before logout (sec)** field on the Agent Login ID screen to `none` for those agents.

Forced Agent Logout from ACW interactions

Only the features that are impacted by the Forced Agent Logout from ACW feature are described in this section.

Interaction	Description
Call Work Codes and Stroke Count	If the agent is in the process of entering a Call Work Code (CWC) or Stroke Count and the Forced Agent Logout from ACW timer expires before the Digits message is sent to CMS, the following actions occur: • The software aborts sending the message. • The CWC session is closed as the agent is being logged out. Even if the setting for CWC is forced, logging out of the agent is allowed and takes precedence over the CWC entry.
Direct Agent Calls	After an agent handles a Direct Agent Call (DAC), the Forced Agent Logout from ACW feature applies when the agent enters the ACW state after the DAC is released.

Interaction	Description
Multiple Call Handling	An agent in ACW is logged out because the Forced Agent Logout from ACW timer has expired, even if the agent has ACD calls on hold.
Timed ACW	The Timed ACW feature immediately switches an Auto-In agent into ACW mode for a specific length of time after the agent disconnects from a call. If both Timed ACW and Forced Agent Logout from ACW are administered, consider the following: • If the agent disconnects from a call while in Auto-In mode, the Timed ACW settings apply and the agent is not logged out based on the Forced Agent Logout from ACW settings. • If the agent, after disconnecting from a call, uses the ACW button to enter ACW, or enters ACW while in Manual-In mode, the Forced Agent Logout from ACW feature settings apply.

Forced Agent Logout by Clock Time

About Forced Agent Logout by Clock Time

The Forced Agent Logout from Clock Time feature allows administrators to:

- Set a specific time when the system automatically logs out Expert Agent Selection (EAS) agents.
- Set a logout reason code for agents on a system-wide basis.
- Administer the system so that agents can override this feature when they press a forced logout override button.

  For more information, see Forced Logout cancellation on page 211.

If the agent is still on an ACD call when the forced agent logout time is reached, the agent is put into pending logout mode. In pending logout mode, the forced logout override button flashes, and the agent hears a repeating tone. The forced logout occurs when the call is disconnected.

Reasons to use Forced Agent Logout by Clock Time

This feature allows you to set a pre-determined time to automatically log out agents when the agents forget to log out at the end of their shifts. You will have a more accurate view of staffing if off-shift agents are logged off the system and calls are not delivered to an autoanswer agent position after the agent leaves.

Prerequisites for using Forced Agent Logout by Clock Time

You can set Forced Agent Logout by Clock Time only if all of the following conditions are true:

- The **Expert Agent Selection (EAS)** field is set to `y` on the System-Parameters Customer-Options screen.
- The **AAS?** field on the Agent LoginID screen is set to `n`.
- The **Call Center Release** field is set to `4.0` or later on the System Parameter Customer-Options screen.

Forced Logout cancellation

An agent can cancel a forced logout by:

- Pressing the flashing forced logout override button during a pending logout. The lamp turns off to indicate that the forced logout will not occur.
- Pressing the forced logout override button (lamp is dark) anytime during the agent's shift. If the override applies, the lamp lights steady. The lamp turns off when the forced logout time is reached, and resets for the next day. Agents can toggle the button to remove the override.
- Logging back in after the forced logout occurs. The forced logout resets for the next day.

If the forced logout override button flutters, one of the following has occurred:

- The agent has pressed the forced logout override button, but a logout time has not been administered.
- The agent pressed the forced logout override button to override the forced logout prior to the logout time, the forced logout time has passed, and the agent has tried to press the forced logout override button again.

Forced Agent Logout from Clock Time interactions

Only the features that are impacted by the Forced Agent Logout from Clock Time feature are described in this section.

Interaction	Description
Call hold	If the agent has ACD, ACDO, or DAC calls on hold when the forced logout time is reached, the agent is put into a pending logout mode. If

Interaction	Description
	the agent has non-ACD calls on hold, the forced logout occurs at the assigned time.
Call Work Codes or stroke counts	If a forced logout time is reached when the agent is in the process of entering a Call Work Code (CWC) or stroke count and the Digits message has not yet been sent to reporting adjunct, the message is aborted and the CWC session is closed. Even if the setting of the CWC is forced, the agent is allowed to be logged out during ACW and this log out takes precedence over the CWC entry. An agent on an ACD call remains connected while in pending logout mode and can complete entering and sending the CWC or stroke count.
Caller Info Display	When a forced logout occurs, the Callr-Info from collected digits that is displayed on the station is cleared.
Conference	A forced logout occurs when an agent is in a conference call unless it is an ACD call that came directly to the agent, or if the agent has an ACD call on hold. If either of these conditions exist, the agent is put into pending logout mode while remaining connected to the conference.
DAC	The Forced Agent Logout by Clock Time feature applies to an agent handling a Direct Agent Call (DAC) after the DAC is released.
Forced logout override	Forced logout override is initiated when an agent presses the forced logout override button, causing the lamp to light. If the button is on (lamp lit), the system will not log the agent out when the forced logout time arrives.
Improved Integration with Proactive Contact	You should not use the Call Center 4.0 Forced Agent Logout by Clock Timer feature with the Call Center 4.0 Improved Integration with Proactive Contact Outbound Calling capability. The Proactive Contact system places a PC agent in the AUX work-mode when the agent is making an outbound PC-imitated call. If the administered time for Forced Agent Logout by Clock Time is reached, the PC agent will be logged out immediately.
Manual-In Mode, Pending ACW, and Timed ACW	A pending forced agent logout takes precedence. When the ACD call is released, the agent is logged out and not put into the ACW state.
Multiple Call Handling	Agents not on an ACD call or with ACD calls on hold are logged out when the administered time for forced logout occurs. If the agent has ACD calls on hold, the agent is put into pending logout mode. The calls remain connected and the agent will not be logged out and reported to reporting adjunct until all of the ACD calls are disconnected.
Multiple Locations Feature	The forced logout is specified in the local time for the agent station. The local time for the agent is determined using the assigned multiple locations feature location number for time zone offset and daylight savings time (DST) rule. The time zone offset and DST rule assigned to the location number for the agent is applied to the main

Interaction	Description
	switch clock time (which also has an assigned DST rule) to determine the current time local to the agent. If the multiple locations feature is not active, the default location 1 is used. In this case, the time zone and DST rule assigned to the main location is used to determine local time.
Non-ACD call connection	A forced logout occurs at the assigned time when an agent is connected to any incoming or outgoing destination that is not an established ACD, ACDO, or DAC call. The reporting adjunct sees the agent as logged out and does not record any subsequent actions from the station. No other events are logged even if the agent is in the middle of dialing, being called by an ACD or non-ACD call, on a trunk, hearing an announcement, in vector processing, or queued to an ACD or non-ACD hunt group.
Pending logout mode	If the agent is still on a call when the forced agent logout time is reached, the agent is put into pending logout mode. The forced logout override button flashes and the agent, a Service Observer or anyone else connected to the agent in a conference hears a repeating tone. The caller cannot hear this tone. This tone has precedence over any other tones including: Service Observing tones, call-waiting tones, music on hold, conference tones, and so on. When the call is released or disconnects, the forced logout occurs.
Reason codes	A specific logout reason code as defined on the Feature-Related System Parameters screen is sent to the reporting adjuncts and BCMS/VuStats as the reason for the forced logout.
Supervisor assist	A forced logout does not occur if a supervisor is logged in as an agent and assisting an agent with an ACD call that the agent received. The logout does not occur because the forced logout is not directed at the supervisor's login.
Transfer	If an agent is in the process of transferring an ACD call when a forced logout time is reached, the agent is put in pending logout mode. The forced logout occurs when the transfer is completed. If the agent is transferring a non-ACD call when a forced logout time is reached, and there are no ACD calls on hold, the forced logout occurs at the assigned time.
VuStats	Active VuStat sessions reflect the system-assigned forced agent logout reason code when displaying the statistics for that reason code.

Forced Agent Logout from Clock Time example

The ABC Call Center has groups of agents in several different timezones. The shift log out times vary based on the local time for each of the groups of agents. Agents sometimes leave their position without logging out at their designated time.

The ABC Call Center wants to use the Forced Agent Logout by Clock Time to set a TOD for automatically logging out such agents. Forced Agent Logout by Clock Time will be set up for some of the agents who have in the past forgotten to logout.

In this example, the main system clock is in the Central timezone. The following table shows the assignments required for the Forced Agent Logout by Clock Time feature.

Agent	Login ID	Local Logout Time	"Forced Agent Logout After:" field setting (set on Login ID screen)	Physical Station	Location for Physical Station (used for timezone offset and DST rule)
1	1001	8:10 am	08:15	3001	2
2	1002	5:30 pm	17:30	3002	2
3	1003	12:15 pm	12:15	3003	1
4	1004	4:50 am	05:00	3004	7
5	1005	5:45 pm	17:45	3005	7

Set the Feature Related System Parameters field**Clock Time Forced Logout Reason Code:** to `2`. This sets logout reason code 2 to be used for this type of logout.

To help you understand how the local time will be determined, the following table shows an example of the locations screen setup for this configuration. Normally the locations screen setup will be part of the original system and station set configuration, and will not need to be configured as part of the Forced Agent Logout by Clock Time feature.

Location	Name	Timezone Offset	[DST] Rule
1	Main (in Central Time)	+ 00:00	1
2	Mountain time location (Phoenix)	- 01:00	0
...			
7	Eastern time location	+ 01:00	3

Note:
The timezone offset is defined relative to the main location.

Forced Agent Logout/Aux Work by Location/Skill

About Forced Agent Logout/Aux Work by Location/Skill

The Forced Agent Logout/Aux Work by Location/Skill feature is used to force all agents in a:

- location to logout
- location into Aux Work
- skill to logout
- skill into Aux Work

Note:
This feature does not apply to Auto-Available Splits/Skills (AAS). For more information on AAS, see About AAS on page 67.

Forcing agents to log out or to the Aux work mode is a two-step process that you can perform directly from a phone by dialing a Feature Access Code or a VDN:

1. A Supervisor, using an administered workstation, dials the administered Feature Access Code to do one of the following:

Feature Access Code (FAC)	Action
Forced Agent Logout by Location	Log out all the staffed agents in a particular location
Forced Agent Aux Work by Location	Put all the agents in a location into Aux work mode
Forced Agent Logout by Skill	Log out all the agents who are logged into a particular skill
Forced Agent Aux Work by Skill	Put all the agents in a skill into Aux work mode

Note:
When the feature access code for Forced Agent Aux Work by Location is entered, the system validates the location to be in the range of allowable locations. 1 is a valid location digit string when **Multiple Locations** is set as `n`; when **Multiple Locations** is set as `y`, the range 1-50 for small systems and 1-250 for others.

The location digit string is not validated against the list of administered location numbers.

2. After entering one of the Facility Access Codes (FACs) on a phone, the Supervisor gets another dial tone. The Supervisor then enters the location or skill number of the agents to log out or put in Aux Work mode. For example, if the FAC for Forced Agent Logout by Location is *46, and the location is 9, the Supervisor dials *46, waits for a second dial tone, then dials 9#. A confirmation tone is then heard.

An administrator needs to set up the FACs before they can be used. For more information, see Administering Forced Agent Logout/Aux Work by Location/Skill in *Administering Avaya Aura™ Call Center Features*.

To access an FAC that performs Force Agent Logout/Aux Work by Location/Skill:

1. Create a vector that contains a route-to <number> vector step that references the appropriate FAC. The vector can also contain, for example, vector steps to prompt the Supervisor for a password as a security precaution, and steps to prompt for and collect the location or skill number.
2. Create a VDN that references the administered vector.
3. The Supervisor calls the VDN.

When the forced logout operation occurs, if an agent is on an ACD or DAC call or has ACD/DAC calls on hold, the agent hears the forced logout tone and is put in pending logout mode until the calls are released. However, if the agent is on a non-ACD call and/or has only non-ACD calls on hold, forced logout occurs immediately, whereas a forced AUX mode change remains pending until any non-ACD calls are released. The agent remains connected to the non-ACD call after being logged out.

For the forced logout and forced Aux features, an agent is considered to be in a particular skill if she/he is logged into the skill at the time the feature request is made. Pending logout also applies only to those agents.

Note:

To facilitate use of this feature for skills, it is useful to assign all agents to whom the operation is to apply to a common skill number that can be utilized to force logout or change to the Aux Work mode.

You can see entries of successful Forced Logout/Aux mode by Location/Skill events on the Events Report screen. The category for forced logout/Aux mode events is feat_event, Event Data 1 is the location or skill number.

```
display events

                            EVENTS REPORT

Event Event                          Event     Event      First        Last      Evnt
Type  Description                    Data 1    Data 2     Occur        Occur      Cnt

3800  Frcd lgout by loc invoked 1         0        12/03/16:42 12/03/16:42  1
3801  Frcd lgout by skl invoked 501       0        12/03/16:42 12/03/16:42  1
```

```
3802   Frcd aux by loc invoked    8          0          12/03/16:42 12/03/16:42  1
3803   Frcd aux by skl invoked    42         0          12/03/15:47 12/03/15:47  1
```

Note:

When the feature access code for “Forced Agent Aux Work by Location” is entered, the system validates the location to be in the range of allowable locations (1 is a valid location digit string when Multiple Locations is set as ‘n’; when Multiple Locations is set as ‘y’, the range 1-250 is valid for XL systems and 1-50 for small systems. The location digit string is not validated.[Eric Roth] I agree, except that we are not using ‘XL’ anymore. Perhaps change “the range 1-250 is valid for XL systems and 1-50 for small systems” to “the range 1-50 for small systems and 1-250 for others”. Also, change “validated” to “validated against the list of administered location numbers”.

Logging out all the agents in a location

Prerequisites

The COR (Class of Restriction) for the station which the Supervisor is using to force an action must have the permission to do so, as indicated by the **Can Force a Work State Change?** field.

Each agent whose work state is being changed must have the COR (Class of Restriction) **Work State Can Be Forced?** field set appropriately.

Forced Logout by location applies only to the active server on which the feature was invoked. If the system is fragmented into multiple servers (Main/LSP/ESS), then the command must be invoked separately on each individual server, as required.

By completing the following steps, a Supervisor can log out all the agents in the location 3.

1. A supervisor, using an administered workstation or a route-to number Vector step, dials the FAC for forced logout of agents in a location: `*46`.
2. The Supervisor dials the location code with # (# terminates the digit string) suffix: `3#`.

Result

All the agents logged into location 3 are logged out.

Remotely forcing all agents in a location/skill to logout/Aux work mode

Prerequisites

1. An administrator needs to set up the Feature Access Codes (FAC) before they can be used. For more information, see Administering Forced Agent Logout/Aux Work by Location/Skill in the *Administering Avaya Aura™ Call Center Features* document.
2. Assign a common skill/location number to all the agents to be forced to logout or to the Aux work mode.
3. The COR (Class of Restriction) for the station which the Administrator is using to force an action must have the permission to do so, as indicated by the **Can Force a Work State Change?** field.

Using the following steps, a Supervisor can remotely force all agents in a location/skill to logout or into the Aux work mode:

1. Create a vector that contains a route-to <number> vector step that routes to the appropriate FAC. (The vector can also contain, for example, vector steps to prompt the Supervisor for a password as a security precaution and steps to prompt for and collect the location or skill number.)
   For example, if the FAC is *22, the vector step is: `route-to number *22`

   Note:

   For an alternate way to achieve this, see Remotely forcing all agents in a location/skill to logout/Aux work mode in one step on page 219
2. Create a VDN that references the administered vector.
3. Call the VDN.

   You will hear a 2nd dial tone as a prompt to enter the location or skill.
4. Dial the appropriate location or skill number of the agents followed by `#`.

Example

You can see entries of successful Forced Logout/Aux mode by Location/Skill events on the Events Report screen. The category for forced logout/Aux mode events is feat_event, Event Data 1 is the location or skill number.

```
display events

                                EVENTS REPORT

Event Event                          Event      Event       First        Last      Evnt
Type  Description                    Data 1     Data 2      Occur        Occur      Cnt
```

```
3800  Frcd lgout by loc invoked 1         0         12/03/16:42 12/03/16:42  1
3801  Frcd lgout by skl invoked 501       0         12/03/16:42 12/03/16:42  1
3802  Frcd aux by loc invoked   8         0         12/03/16:42 12/03/16:42  1
3803  Frcd aux by skl invoked   42        0         12/03/15:47 12/03/15:47  1
```

Note:

When the forced logout/Aux operation occurs, if an agent is on an ACD or DAC call or has ACD/DAC calls on hold, the agent is put in pending logout/AUX mode until the calls are released. However, if the agent is on a non-ACD call and/or has only non-ACD calls on hold, forced logout occurs immediately while the agent remains connected to the non-ACD call, whereas a forced AUX mode change remains pending until any non-ACD calls are released.

For the forced logout and forced Aux features, an agent is considered to be in a particular skill if she/he is logged into the skill at the time the feature request is made. Pending logout also applies only to those agents.

Remotely forcing all agents in a location/skill to logout/Aux work mode in one step

Prerequisites

1. An administrator needs to set up the Feature Access Codes (FAC) before they can be used. For more information, see Administering Forced Agent Logout/Aux Work by Location/Skill in the *Administering Avaya Aura™ Call Center Features* document.
2. Assign a common skill/location number to all the agents to be forced to logout or to the Aux work mode.
3. The COR (Class of Restriction) for the station which the Administrator is using to force an action must have the permission to do so, as indicated by the **Can Force a Work State Change?** field.

Using the following steps, a Supervisor can remotely force all agents in a location/skill to logout or into the Aux work mode:

1. Create a vector that contains a route-to <number> vector step that routes to the appropriate FAC. (The vector can also contain, for example, vector steps to prompt the Supervisor for a password as a security precaution and steps to prompt for and collect the location or skill number.)
   For example, if the FAC is *22 and the location/skill is 1, the vector step is: `route-to number *221#`
2. Create a VDN that references the administered vector.
3. Call the VDN.

All the agents logged into skill/location 1 are logged out.

Inbound Call Management

About ICM

Inbound Call Management (ICM) allows you to integrate features of the communication server with host-application processing and routing, and automate delivery of caller information to agents' displays. You can create a sophisticated system to handle inbound calls for applications such as telemarketing and claims processing. To implement ICM, you integrate features of the communication server such as Automatic Call Distribution (ACD), Expert Agent Selection (EAS) Call Vectoring, Direct Agent Calling (DAC), and Call Prompting with an application on a host processor.

The host application, or adjunct, can be a CallVisor/PC, an IVR voice system, Telephony Services Server serving a local-area network, or a vendor application using the CallVisor Adjunct/Switch Applications Interface (ASAI). A CallVisor ASAI link between the communication server and adjunct allows the adjunct to control incoming call processing and routing.

In addition, you can automate ACD agent telephone displays and associate them with new and transferred calls, and assist calls to a supervisor. You can display incoming call information such as Calling Party Number (CPN), Billing Number (BN), and Dialed Number Identification Service (DNIS). Or, you can set up the adjunct to retrieve caller information from a database and display it on a particular agent's screen, based on the service dialed.

See ICM detailed description for more information on applications.

Applicatiions of ICM

The following are some typical ICM applications:

- The system passes calling party/billing number (CPN/BN) information and the call is routed to an adjunct application for screen pop and supervisory transfers, with screen duplication.
- The system sends to the adjunct application both caller and prompter information about all incoming calls to a particular number. According to caller information in a database, the application directs the communication server to route the call. For example, the call could be routed to a preferred agent, to best customer treatment, or to accounts receivable.

- The system uses Call Prompting to obtain a customer account number and then passes this information to the adjunct for call routing or screen pop.
- The system connects the caller to a voice response unit (VRU), along with caller CPN/BN and DNIS information.The caller then interacts with the VRU to direct how the call is handled. The system can verify a caller's identity and provide access to database information such as claims status or account balance.
- With Direct Agent Calling (DAC), an adjunct application can transfer a call to a specific ACD agent and have the call treated as an ACD call and tracked on Call Management System (CMS).
- An adjunct application can attach information used by another application to an ICM call using User-to-User Information fields. The adjunct transfers the call, along with the application-specific information, over primary rate interface (PRI) trunk to a CallVisor ASAI application at another communication server. For example, an application at one communication server can determine a caller's account or claim number and pass this information to a special list on another communication server, where an application will transfer the call.

For additional application scenarios, see *Programming Call Vectors in Avaya Aura™ Call Center*.

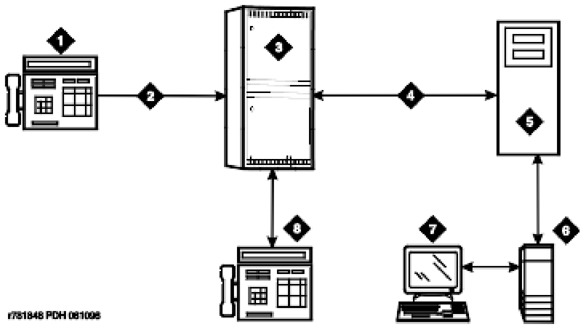
Agent data screen delivery applications

You can use an Interactive Voice Response (IVR) system to deliver appropriate display data about callers to agents. You can pass information such as CPN/BN, DNIS, and Look-Ahead Interflow information, digits collected from Call Prompting, and which agent is selected to an IVR system. The IVR system delivers the appropriate data screen to the agent who takes the voice call. The IVR system can transfer or duplicate data screens for transferred or conferenced calls.

A simplified configuration for the use of an IVR system for agent data screen delivery applications is shown in the following figure.

Note:

An IVR VIS is used as an example - other adjunct processors have similar capabilities but should be verified for a particular application. If the host supports ASAI, the IVR system is not needed.



1. Telephone
2. ISDN-PRI
3. Avaya switch
4. ASAI
5. IVR
6. Host
7. Agent data terminal
8. Agent telephone

General processing for this type of application occurs as follows.

1. An Interactive Voice Response (IVR) system or host requests notification for events such as call offered, call ended, call connected, call dropped, call transfer, and alerting.
2. The communication server notifies the IVR system with event reports when the call arrives, when the agent answers, when the call drops, and so on.
3. The IVR system sends information to the host application so that it can send a data screen to the agent's data terminal.

The IVR system can determine when a call drops before being answered and can track abandoned calls or use CPN/BN information for callbacks.

Integration with speech processing adjuncts

ICM can be used to provide integration with VRUs. The advantages of using ICM with CallVisor ASAI in addition to tip/ring interfaces are as follows:

- Data-screen integration is provided on transferred calls.
- Answer notification is provided on internal calls (CallVisor ASAI capabilities let you know what happens with the call).
- ISDN network information such as CPN/BN and DNIS is delivered to agents (call prompting for this information is not necessary).

A simplified configuration of this application is shown in the following figure.

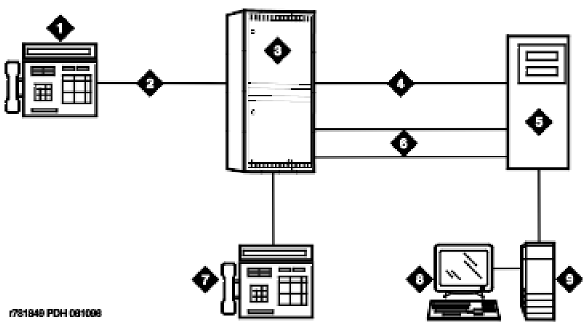


Figure 3: Simplified ICM configuration for speech processor integration

1. Phone
2. ISDN-PRI
3. Avaya switch
4. ASAI
5. Speech processor
6. Tip/ring lines
7. Agent phone
8. Agent data terminal
9. Host

General processing for this type of application occurs as follows:

1. The communication server uses CallVisor ASAI link to pass incoming call information to the IVR voice system.
2. The split or skill on the communication server distributes the call to an available voice line.
3. After digits are collected using a DTMF keypad, the IVR system transfers the call back to a split or skill or specific agent on the communication server using CallVisor ASAI.
4. If the call is transferred to an agent, the communication server uses CallVisor ASAI link to pass an event report on which agent receives the call.
5. The IVR system forwards the agent ID to the host application, which delivers a data screen to the agent.
6. Agents can display collected digits on their data terminals. Except for the dialed number, information from an IVR system cannot be carried with the call and displayed on a phone. For example, digits collected in an IVR system adjunct cannot be passed to the communication server for display.
7. If the collected digits are the extension where the call is being routed, these routing digits are passed to the communication server as the destination in the CallVisor ASAI third-party make-call request. The IVR system uses the request to set up various types of calls.

Host/adjunct call routing

The host or an IVR system adjunct uses incoming call information to route the call to a split or skill, vector, particular agent, or location off the communication server. The IVR system can also direct the system to handle the call as a priority call. Routing can be based on the caller's area code or country code, digits collected using Call Prompting, dialed number or service, agent availability, or information in a customer database.

To implement this type of call routing, make sure that calls come into a vector that contains an **`adjunct routing`** vector command. This command causes the communication server to initiate the route CallVisor ASAI capability. Vector processing occurs while the caller waits. A default split or skill or answering position can also be specified in the vector, in case the IVR system does not respond in the administered amount of time (determined by the announcement/wait steps). Announcement and wait steps are needed to give the host time to respond.

Note:
If the Display VDN for Route-to DAC option is enabled, and an **`adjunct`** vector step results in a direct agent call to an EAS agent, the VDN name is provided in the same manner as when a **`route-to digits`** or **`route-to number`** vector command is used.

For adjunct routing, if the call queues to a split or skill or leaves vector processing, a route-end request is sent to an IVR system.

ICM considerations

Administrators and planners must consider:

- ICM traffic
- Rated communication server capacity
- CallVisor ASAI interface traffic
- Rated capacity of the adjunct application processor

Avaya Technical Design Center can provide planning assistance.

In addition, you must consider the following:

- CallVisor ASAI and BX.25 CPN/BN-ANI are not supported simultaneously.
- Direct agent calls are allowed only if the caller and the receiving agent have a Class of Restriction (COR) that allows Direct Agent Calling (DAC).
- Direct agent calls cannot go through vectors.
- Direct agent calls cannot be made over a DCS link. If the receiving agent is not an internal extension, the call is denied.

ICM interactions

Call Prompting: Digits collected by Call Prompting are passed with current call information to an IVR system adjunct.

Direct Agent Calling: DAC allows an adjunct to direct a call to a particular ACD agent and have the call treated as an ACD call. Calls that enter the communication server as ACD calls and are routed to a particular agent using adjunct routing, or are transferred using a third-party make-call request, are treated as ACD calls for the duration of the call. See Automatic Call Distribution for more information on direct agent calls.

Priority Calling: CallVisor ASAI allows both Priority Calling and DAC for the same call.

Information Forwarding

Information Forwarding

The Information Forwarding feature sends call related information, such as UCID (Universal Call ID), ASAI data, collected digits, active VDN name, time spend waiting as well as ANI, (Automatic Number Identification, CLID (Calling Line Identification II (Information Indicator) Digits, and CINFO (Customer Information Forwarding), with calls over public and private networks using ISDN (Integrated Services Digital Networks) trunks or (Session Initiation Protocol) SIP trunks. Private networks that are enabled for Information Forwarding can also be configured for QSIG or non-QSIG protocols. Call data derived from the Information Forwarding feature can be used to enhance call processing, customer service, and data collection.

Note:
Asynchronous Transfer Mode (ATM) trunking and IP trunking can be set up to emulate ISDN PRI. For more information, see Administering Network Connectivity on Avaya Aura™ Communication Manager, and ATM Installation, Upgrades and Administration using Communication Manager.

About Information Forwarding

Whenever the communication server interflows a call over ISDN (for example, PRI or BRI) or SIP trunk facilities by means of a **`route-to`** (with Look-Ahead Interflow active), **`queue-to best`**, or **`check best`** command, the following information is sent with the call using user-to-user information transport and can be used by adjuncts or displayed at the receiving communication server:

- ASAI user information
- the name of the active VDN (LAI DNIS)
- other LAI information (a time stamp showing when the call entered the current queue, the call’s priority level in its current queue, and the type of interflow)
- any collected digits (this does not include dial-ahead digits). These digits are available for processing at remote vectors and/or displaying to the agent.
- the number of seconds that the call has already spent in vector processing (called *in-VDN time*)
- Universal Call ID (UCID)

Note:
Sending of information depends on priority settings and activated features. Also the communication server version must be V6 or later.

Information Forwarding detailed description

In the past, for ISDN trunks, look-ahead interflow transported the LAI Information Element (IE) in codeset 6 or 7, which functioned over non-QSIG private networks, but only over certain public networks.

Now, call centers can transport application information (including the LAI information) over many more public ISDN networks using User to User Signaling (UUS) Supplementary Services that incorporate user-to-user information (codeset 0 UUI). Information passes over QSIG private networks using manufacturer specific information (MSI - codeset 0 Facility IE) in various messages.

Beginning with Call Center 5.0, user-to-user information can also be transported over SIP trunks.

This feature:

- Enables multiple applications on the communication server to share the contents of the UUI IE or MSI
- Allows for backwards compatibility with software prior to the DEFINITY R6.3.

Information Forwarding interactions

Best Service Routing: Best Service Routing-related data is sent in addition to the associated ASAI user data and UCID.

Intraflow and Interflow: Intraflow and Interflow allow you to redirect ACD calls from one split or skill to another split or skill when the splits/skills are not vector-controlled. Intraflow redirects calls to other splits/skills within the system using Call Coverage or Call Forwarding All Calls. Interflow redirects calls to an external split or skill or location using Call Forwarding All Calls.

Information Forwarding considerations

1. Enhanced information forwarding has been tested with several major carriers.

   To find out if these capabilities work with your carrier, check with your account team for the most current information. If testing has not been done to verify operation over the public networks involved with the preferred specific configuration, use of private trunking between the nodes should be assumed until successful testing has been completed.
2. Any communication server that acts as tandem node must have priorities assigned to the Shared UUI features for non-QSIG trunk groups.

   Even if this communication server does not create anything, the priorities must be set correctly to pass the information along. For more information, see the Troubleshooting on page 231 section.
3. The `Send codeset 6/7 LAI` trunk group option operates independently of the `UUI IE Treatment` trunk group option.

   However, if you turn both of these options on, you'll send the same information twice and possibly exceed the maximum ISDN message size. The communication server provides a warning message when both options are administered. There are two ways to correct when the user data exceeds the maximum message size, either:

- put a blank in the priority fields for `VDN Name` and `Other LAI Information` on the Shared UUI Feature Priorities screen, or
- disable the `Send codeset 6/7 LAI` option.

4. For non-QSIG or QSIG trunk groups to the communication server that require information forwarding, the `UUI IE Treatment` should be shared and the `Send Codeset 6/7 LAI IE` should be n.
5. Information transported using the Shared UUI will not work with non-Avaya switches unless they adhere to the proprietary encoding.

Support of call center features

Information transport supports these call center features:

- Enhanced Look-Ahead Interflow - routes calls from busy call centers to centers that are less busy (see Interflow and Intraflow interactions on page 250).

Note:
Look-Ahead Interflow information can be forwarded using information transport or the traditional codeset 6/7 LAI IE.

- Best Service Routing - routes calls to the best available agents wherever they are (see Best Service Routing).
- Universal Call ID - provides a means to collect and trace call data from multiple call centers (see Universal Call ID).

Data handled by Information Forwarding

The following table shows the call-related information you can send through Information Forwarding depending on your trunk type:

Incoming call-related information	Supported trunks
ANI	ISDN SIP
II-Digits	ISDN
CINFO	ISDN
Adjunct Switch Application Interface (ASAI)-provided user information	ISDN SIP[9]

9 Adjunct Switch Application Interface (ASAI)-provided user information is the only information provided if the SIP trunk group is assigned as “service provider”.

Incoming call-related information	Supported trunks
Look-Ahead Interflow (LAI) information, such as the in-queue timestamp, VDN name, and network-provided caller information, including priority level and type of interflow. *This is still an open issue with SIP!!!	ISDN
Universal Call ID (UCID) - UCID provides a unique identifier for each call that is used to track the call. For more information, see Universal Call ID in Avaya Aura™ Communication Manager Feature Description and Implementation.	ISDN SIP
Interflowed Collected Digits and in-VDN time data.	ISDN SIP

For information about administering information transport, see Avaya Aura™ Communication Manager Feature Description and Implementation. For detailed information about trunk group setting interactions with Information Forwarding, UCID, and multi-site routing, see Advanced multi-site routing on page 533.

Information Forwarding benefits

The following table lists Information Forwarding benefits:

Function	Benefit
Improved agent efficiency and service to call	Forwarding of original caller service requirements and entered prompted digits speeds service to the caller and saves the agent time.
Improved network-wide call tracking	Forwarding of UCID, In-VDN-Time and collected digits allows tracking as a single call and provides a network-wide view for call statistics.
Improved CTI integration	Forwarding of UCID, In-VDN-Time, and collected digits provides screen pop and database access applications across sites.
Forwarding of original call service requirements (VDN Name or DNIS)	Faster and more efficient agent handling, better service to the caller, and improved CTI integration
Transport of UCID	Improved call tracking as a single call and CTI integration
Collected Digits Transport	Better service to the caller because the caller doesn’t have to repeat input of information, more information for the agent, better and faster call handling, improved call tracking because the collected digits are included with the call record, and improved CTI integration
Forwarding of In-VDN Time	Improved call tracking as a single call and end-to-end time-before-answer statistics

Function	Benefit
Support of ASAI user Information Forwarding	CTI integration

ISDN network requirements

Your network must meet the following requirements to support Information Forwarding:

- Both the private and public networks must support end-to-end transport of codeset 0 user data either as user-to-user information (UUI IE) or QSIG manufacturer specific information (MSI) in the SETUP and DISCONNECT ISDN messages. Private networks can be configured for either non-QSIG transport by way of a codeset 0 UUI IE or QSIG transport by way of MSI packaged in a codeset 0 Facility IE. Public networks do not currently support QSIG, and user data can only be transported by way of the UUI IE when supported by the network. Future public network offerings may support QSIG by way of a Virtual Private Network.
- The Communication Manager must support the ISDN country protocol.

  Important:

  If testing has not been done to verify operation over the public networks that are involved with the preferred specific configuration, use of private ISDN trunking between the nodes should be assumed until successful testing is complete.
- The network byte limit for the user data portion of user information contents must be large enough to carry the data that is needed for the customer application.

Note:

Some public network providers may require service activation and/or fees for user information transport.

SIP network requirements

Your network must meet the following requirements to support Information Forwarding:

- The public network must support end-to-end transport of SIP user data in a user-to-user information (UUI) header in the INVITE messages.
- The Communication Manager must support the SIP protocol and the UUI header.
- The network byte limit for the user data portion of user information contents must be large enough to carry the data that is needed for the customer application.

Troubleshooting

The following troubleshooting hints should be reviewed when information is not forwarded, even though you received no error messages while administering the Shared UUI feature, and all software and connections meet the minimum requirements:

- If DCS is used, make sure all ISDN trunks between the communication server used for DCS or remote AUDIX are configured in the D-channel mode.
- For each ISDN trunk administered with the Shared UUI option, make sure the UUI size does not exceed the UUI IE size that the network can support.
- For all non-QSIG ISDN trunks and SIP trunks, make sure the **UUI Treatment** field is set to `shared`.
- Make sure trunk group options are set correctly for the application and configuration.
- Applications may fail on networks supporting limited UUI transport. Administration determines which application's UUI will be transported in these cases. If a given application is failing, first check the administration to determine if the application in question has the highest priority. This applies to tandem nodes as well as originating nodes.

  Applications that originate UUI on tandem nodes can request that assigned priorities at the tandem node be applied to the resulting UUI. Therefore, it is possible for a tandem node to erase UUI information received from the originator. Passing UUI through a tandem node transparently, as required for UUS Service 1, does not apply to the proprietary shared UUI procedures of the communication server.

Information Forwarding support for BSR and LAI

When a call is interflowed to another Communication Manager by BSR or Look-Ahead Interflow, the following data types are supported for Information Forwarding:

- Collected Digits - Any digits that are collected for the call are passed with the interflowed call, and automatically collected when the call enters vector processing at the receiving Communication Manager.
- Elapsed in-VDN time - The elapsed time that the call has already spent at the sending Communication Manager is passed with the interflowed call and automatically sent to the Avaya Call Management System (CMS) when the call enters vector processing at the receiving Communication Manager.
- UCID - Universal Call ID.

The following sections describe handling and transport of Information Forwarding data in interflowed calls:

- Forwarding collected digits with interflowed call on page 232
- Forwarding accumulated in-VDN time on page 232
- Transport by way of globally-supported methods over ISDN on page 233
- Transport by way of globally-supported methods over SIP on page 234
- LAI backward compatibility issues on page 234

Forwarding collected digits with interflowed call

The following list describes how forwarded collected digits are handled in interflowed calls:

- The last set of up to 16 collected digits, not including the dial-ahead digits, are forwarded with a call interflowed over ISDN or SIP facilities.
- When processing for the call at the remote location reaches the VDN, the forwarded digits are inserted in the collected digits buffer. Therefore, a TTR is not needed. The objective is to immediately provide the collected digits to the CMS in a DIGITS message and to ASAI by way of the VDN event report in the same manner as incoming ANI.
- The collected digits are available for further routing by steps in the assigned and subsequent vectors, and eventual display to the answering agent.
- All interactions with the collected digits are the same as digits that are collected using a collect step. For example, a subsequent collect step will clear the digits.
- If the call is further interflowed or tandemed over ISDN or SIP facilities, the collected digits are tandemed with the call. If more digits are collected at the tandem Communication Manager, the latest collected digits are tandemed.

Forwarding accumulated in-VDN time

The following list describes how forwarded in-VDN time data is handles in interflowed calls:

- When a call is interflowed, the in-VDN time in seconds, from 0 to 9999, is included. The in-VDN time is the elapsed time starting from the VDN that was originally called until when the Information Forwarding message is created.
- If the call was interflowed to the local system and in-VDN time was received for the call, the previous in-VDN time is added to the local in-VDN time.
- If the accumulated time exceeds the largest value that can be transported, the maximum value is sent.
- The accumulated in-VDN time that is received on an incoming interflowed call is forwarded to the CMS in the DNEVENT message when the call starts VDN/vector processing at the remote location.
- In-VDN time does not pass to the Basic Call Management System (BCMS) for reporting by BCMS.

Transport by way of globally-supported methods over ISDN

Use of codeset 0 supports information transport over ISDN PRI/BRI facilities (QSIG or non-QSIG) as well as supporting operation over public networks. The following list describes information transport by way of globally-supported methods over ISDN trunks:

- When a call is LAI or BSR interflowed, the following information is forwarded with the call over public or private ISDN networks using QSIG or non-QSIG protocols:
  - LAI information.

    Note:
    The forwarded LAI information is the same as that sent in the LAI IE: VDN name (also called LAI DNIS), put in queue time-stamp, priority level and type of interflow.
  - Collected digits.
  - in-VDN time data in the ISDN SETUP message.
- Other call related information, including calling party number (ANI), calling party name, II-digits and CINFO digits, that is tandemed with the interflowed call in the SETUP message is forwarded in the normal manner.

  Note:
  II-digits and CINFO are forwarded as codeset 6 IEs which may be a problem in some networks.
- At the remote end, the transported data is separated into its component parts for storage with the call, call vectoring, call processing and display, further interflow or tandeming, and forwarding to adjuncts. For example, the LAI info is treated as though it was received as an incoming codeset 6 LAI IE including forwarding over ASAI as a code set 6 LAI IE in event reports.
- When a status poll call is placed to the remote location, the Communication Manager only forwards the UCID and caller information that was received from the original call.
- In response to a status poll, the Communication Manager forwards the reply-best status data in the ISDN DISCONNECT message over public or private ISDN PRI/BRI networks. In this case, the DISCONNECT message has a cause value of 31 *Normal-Unspecified* for wider international interoperability.
- The Multi-Site Routing related data is in addition to the associated ASAI user data, which was previously sent in a non shared UUI IE, and the UCID data.

Transport by way of globally-supported methods over SIP

SIP supports information transport with the UUI header. The service provider must support SIP information transport. The following list describes information transport by way of globally-supported methods over SIP trunks:

- When a call is LAI or BSR interflowed, UUI header information is forwarded with the call over SIP networks using non-QSIG protocols.
- Other call related information, including calling party number (ANI), and calling party name is forwarded in the normal manner.
- At the remote end, the transported data is separated into its component parts for storage with the call, call vectoring, call processing and display, further interflow or tandeming, and forwarding to adjuncts.
- The Multi-Site Routing related data is in addition to the associated ASAI user data, which was previously sent in a non-shared UUI, and the UCID data.

LAI backward compatibility issues

The following list summarizes LAI backward compatibility issues:

- An ISDN trunk group option is provided in the SETUP message for LAI interflowed calls to specify whether to include an LAI IE (codeset 6 or 7). When this option is set to “y” (default), an LAI interflow (using the existing or enhanced LAI vector command) will include a codeset 6/7 LAI IE to provide inter-operability in a mixed Communication Manager environment with pre-Definity R6 systems. The option must be set to “n” if the network does not support codeset 6/7 or this IE is not required.

  Important:

  Codeset 0 information transport by way of shared UUI is required for BSR polling calls.
- Administer the ISDN Trunk Group option: Send Codeset 6/7 LAI IE. This option is valid even if LAI at the remote site is not active for tandem situations. Use of this option for LAI does not depend on the setting of the Vectoring Best Service Routing customer option.
- If the ISDN trunk group option is set to send the LAI IE, this IE is sent in addition to the Information Forwarding by way of codeset 0 shared UUI transport when a call is LAI interflowed over a trunk in this trunk group. With shared UUI, you can set the LAI data to be excluded in the UUI IE.
- Administer the Shared UUI priorities for ISDN trunks. This is important when the network byte limit on the user data part of the UUI user information contents is not large enough to carry the data that is needed for the customer application. Note that Shared UUI priorities do not apply to QSIG. To determine customer application data sizes, see Determining user information needs. For instructions on how to administer Shared UUI, see Avaya Aura™ Communication Manager Feature Description and Implementation.

ASAI shared UUI data conversion

The outgoing trunk interface treatment controls whether the ASAI data format is shared or non-shared:

- If the outgoing trunk interface is non-shared, ASAI UUI data stored in shared format is converted to the non-shared (service provider) format.
- If the outgoing trunk interface is shared, ASAI UUI data stored in shared format is sent in shared format.

Determining user information needs

User information rules

The network byte limit on user information contents (the user data part of the UUI) must be large enough to carry the data needed for the customer application.

If you want to forward information over a network that does not support at least 82 bytes of user data, you must determine the space required for the application and adjust priorities accordingly.

The network byte limit on the user data part of the UUI user information must be large enough to carry the data that is needed for the customer application.

The UUI IE for ISDN uses 3 bytes for the IE header information and allows from 32 bytes to 128 bytes for the user data portion. For example, if the network specifies that it can transport 32 bytes of user data, the UUI IE header length is 35 bytes.

The UUI for SIP includes the same data as the ISDN UUI IE except for the first 2 bytes in the UUI IE header information. For example, if the network specifies that it can transport 32 bytes of user data, the UUI header length is 33 bytes.

The user information capacity need is determined by adding the space that is required for each data item to be transported based on the following rules.

Rule	Description
Minimum and maximum byte lengths	A maximum of 128 bytes of user data is supported by the Communication Manager for UUI. Some network providers might impose limitations on the amount of user data they support.
User data length	Each shared data item requires 2 bytes for the application header. The application header is the application identifier plus the application data length. The application data length depends on the configuration of the customer application, except for UCID, In-VDN time, and Other LAI. These applications have a fixed byte length. For more information, see Bytes length ranges for UUI user data on page 237.

Rule	Description
Byte length overruns	If the administered over run for the data length is exceeded, the lowest priority items are not included until the remaining data fits. If a specific data item at a higher priority exceeds the administered Maximum UUI Size setting, that item is not sent, leaving room for other lower priority items.
Priority settings	If the data item priority is set to blank in the Shared UUI Feature Priorities page in the Trunk Group administration form, the data item is not sent and no space is allocated for it.
QSIG considerations	QSIG signaling and networks do not have user information size limits. They will support sending MSI for user data items at their maximums. Determination of space allocation and administration of priorities does not need to be done for QSIG networks.
ASAI data length considerations	If the network supports 128 bytes and 78 bytes or less of ASAI user data is required, you do not need to determine space allocation or administer priorities. If your ASAI user data is greater than 78 bytes, you can have up to 96 bytes of ASAI user data (98 bytes with the application header). The need for other interflow shared data transport must be carefully considered when setting priorities and determining how much ASAI user data to support for the application. If the network supports the full 128 bytes and all interflow data at their maximums is transported (48 bytes), the maximum length for ASAI user data is 80 bytes (78 bytes plus the application header). If the full 96 bytes of ASAI user data is required (plus 2 bytes for the application header), then only 30 bytes is available for other interflow data. With the service provider trunk group for UUI transport (non-shared UUI), the UUI transports only ASAI UUI. The first byte of the UUI contains the UUI codepoint, which is 0x7E, followed by the byte that is set to the length of the following bytes. The next byte is the UUI protocol discriminator which, is unusually set to 0x00, that means the following data (ASAI UUI data) is in a user specified format. Another possible setting for the protocol discriminator is 0x04, meaning that the following data is in the IA5 (ASCII) character format. With service provider non-shared UUI format the protocol discriminator setting is retained by Communication ManagerCommunication Managerand included in the ASAI UUI IE sent to a CTI adjunct. The shared format for the trunk group UUI transport is a multi-data element format that can contain UCID, ASAI user data, collected digits, VDN name, etc. associated with the call. Each data element is designated by a unique op code defined by the Avaya shared UUI specification. The ASAI user portion of the shared UUI consists of the op code (0xC8) and the data length byte followed by the user data. The ASAI UUI portion of the shared UUI does not contain a protocol discriminator field that is included with the non-shared UUI. When an ASAI UUI IE is created for passing to an adjunct, Communication Manager inserts a 0x00 for the protocol discriminator.

Rule	Description
	Note: In most cases the application should not rely on or depend on the setting of the protocol discriminator since it usually is not meaningful other than to account for it as being part of the format. If the Call Vectoring set command initially creates the ASAI UUI data, the protocol discriminator is always set to 0x04 in both the service provider and shared cases by Communication Manager, otherwise the protocol discriminator setting is not changed. In the case of shared the UUI IE protocol discriminator is set to 0x04 regardless of what else is carried by the shared UUI. The set command defines the data bytes following the protocol discriminator which is limited to the decimal digits (0 - 9) subset of the ASCII (IA5) character set. The SIP UUI format is basically the same as the ISDN format but without the first two bytes (the 0x7E op code and length byte). The SIP UUI is in an ASCII string of hex characters (two for each byte) starting with the UUI protocol discriminator byte (00 or 04).

Bytes length ranges for UUI user data

The following table specifies minimum and maximum byte lengths used to send user data over call center networks.

Type of user data	Total user data bytes (with 2-byte header)	Description
ASAI	2 to 98 or 0 (calculated by 1 byte per byte of ASAI user information)	Required for certain CTI applications when the CTI application sends user information and the amount of space is determined by the application. For example, 34 bytes is required if the application sends 32 bytes of data. Sending more than 78 bytes of ASAI data (80 bytes with the application header) reduces capacity for other interflow data.
UCID	10 or 0	A unique tag that identifies the call that this message is being sent for and the other information included in the UUI applies to. The priority for this element can be set between 1 to 7 (1 is the highest) or a blank. The default for this element is priority 2. If blank, this field's information is not forwarded. Used by BSR to track calls across multiple sites. Trunk group setting and/or system feature settings control transport of UCID data, even when the priority is set to 1. When the data item is not included, it does not take up any space.
In-VDN Time	4	Used by BSR to determine time before answer and call tracking across sites. This data type can be eliminated when short waiting times are anticipated. If the priority field is not blank, it is always included.

Type of user data	Total user data bytes (with 2-byte header)	Description
VDN Name	2 to 17 (calculated by 1 byte per character in name) maximum of 15	Used by BSR, but can be eliminated if receiving sites use dedicated VDNs that display equivalent information to the answering agent. An interflowed call that is received without the originating VDN name uses the incoming VDN name. If the priority field is not blank, the 2-byte header is always included.
Collected Digits	4 to 11 or 0 (calculated by 1 byte per 2 digits plus 1) maximum of 16 digits	Requires a whole byte for an odd number of digits. For example, 1 digit requires 2 bytes (1 plus 1), 7 digits need 5 bytes (4 plus 1), and 16 digits need 9 bytes (8 plus 1).
Other LAI Info	6	Required for existing CTI applications that use any of the following obtained from the from the LAI IE: • in-queue time stamp • queue priority • interflow type
Held Call UCID (Universal Call ID)	10 or 0	The unique tag for the last call that was put on hold by the ACD agent placing this call to another system. This UCID can be used to identify the original or parent call that may be eventually be conferenced and/or transferred to that other system. The UCID included in the element with default priority 2 is the tag for this new call placed by the agent while the original call is on hold. The priority for this element can be set between 1 to 7 (1 is the highest) or a blank. The default for this element is priority 7. If blank, this field's information is not forwarded. For more information, see *Avaya Aura™ Communication Manager Screen Reference*.

Information Fowarding example

Assume that your public network supports only 32 bytes of user information. Your application requires 13 bytes of ASAI user information (15 bytes of user data), UCID (10 bytes of user data), and 8 collected digits (7 bytes of user data - 4 plus 1 plus 2 for the header). It does not require Other LAI Information. Also, call time at the sending Communication Manager is brief because calls are not queued before interflow takes place and tracking as a single call is not required.

By dedicating appropriately named VDNs at the receiving Communication Manager, the public network can support the application. Because the needed data items require the entire 32 bytes of user data, the priority fields for the **In-VDN Time**, **VDN Name**, and **Other LAI Information** must be set to blank.

Information Forwarding troubleshooting

In some circumstances, UUI data may not be forwarded, even though you received no error messages while administering the Shared UUI feature, and all software and connections meet the minimum requirements. The following list provides items that can be evaluated to troubleshoot the problem:

Tip:

When a new application is implemented, run the `display events` command on a periodic basis for the appropriate vector. The resulting report notifies you if any UUI data could not be sent.

Proposed Solution

1. If DCS is used, ensure that all ISDN trunks between Communication Manager that are used for DCS or remote AUDIX are configured in the D-channel mode.
2. For each ISDN or SIP trunk that is administered with the Shared UUI option, make sure that the UUI size does not exceed the UUI size that the network can support.

   For more information, see Determining user information needs.
3. Verify that trunk group options are set correctly for the application and configuration.
4. Applications may fail on networks supporting limited UUI transport.

   Administration determines which application's UUI will be transported in these cases. If a given application is failing, first check the administration to determine if the application in question has the highest priority. This applies to tandem nodes as well as to originating nodes.
5. Applications that originate UUI on tandem nodes can request that assigned priorities at the tandem node be applied to the resulting UUI.

   Therefore, it is possible for a tandem node to erase UUI information that was received from the originator. Passing UUI through a tandem node transparently, as required for UUS Service 1, does not apply to Communication Manager proprietary shared UUI procedures.

Advanced information forwarding for trunks

Advanced information forwarding for ISDN and SIP trunks

This section explains ISDN (BRI or PRI) and SIP trunk group setting interactions with Information Forwarding, UCID, and Multi-Site Routing.

About advanced information forwarding

User information included in the SETUP message for an outgoing call (at the sending switch) or DISCONNECT message sent back for an incoming call (at the receiving switch) is based on the trunk group settings at the sending or receiving sites.

The shared user information forwarding supports various data items (UCID, collected digits and In-VDN-Time) in addition to shared forwarding of LAI Info (VDN-Name and Other-LAI) and ASAI provided user data. Shared forwarding over non-QSIG ISDN trunks packs the data items in a codeset 0 UUI IE (called shared UUI), where each item consists of a two-byte header (application ID and data length). Shared forwarding over QSIG trunks transports the data items as Manufacturer Specific Information (MSI) in codeset 0 Facility IEs.

BSR and shared data forwarding (UCID and other new data items) requires QSIG or the shared**UUI Treatment** setting with non-QSIG trunk groups on both the sending (outgoing trunks) and receiving (incoming trunks) at the switch. Shared settings are also required on tandem trunk connections through the switch that routes these calls. LAI Info, UCID, collected digits, In-VDN-Time and ASAI provided user data can be forwarded with a call in the SETUP message (LAI or BSR interflowed call, a tandemed call, for UCID with any outgoing call and for ASAI user data any adjunct routed outgoing call). Only BSR reply-best data is returned with a BSR poll call and only ASAI user data is returned for a non-poll call in a DISCONNECT message (both types of data will not be included in the same DISCONNECT message). Shared UUI Priority settings do not affect what is put in the DISCONNECT message or data forwarded over QSIG trunks.

The protocol (QSIG or non-QSIG) is set on page 1 of the ISDN trunk group screen using the**Supplementary Service Protocol** field. QSIG type as defined for shared MSI is protocol type b (another protocol type d, ECMA QSIG is considered non-QSIG for Information Forwarding). The**Send Codeset 6/7 LAI** field on page 2 indicates whether or not to include an LAI IE in the SETUP message. The codeset used (6 or 7) is determined by the**Codeset to Send TCM, Lookahead** field on page 1. The**Send UCID** field on page 2 indicates whether or not the UCID data item should be included as user information with calls routed over this trunk group. The**Send Codeset 6/7 LAI IE** field is ignored for BSR polls over the trunk group (an LAI IE will never be included with BSR calls).

Non-QSIG protocol

UUI Treatment set to `service-provider` includes any application provided UUI in a codeset 0 UUI IE on a non-shared basis. That is, the data portion of the UUI IE only includes user info in the SETUP or DISCONNECT messages as provided by an application such as ASAI without the shared App-ID and length header fields. User data from only one application can be included in non-shared UUI. This setting would be used for non-QSIG trunk groups when service-provider functionality is wanted (for example, where shared forwarding of the new data items is not required or for trunk groups to other vendor switches or network services that need user information from the trunk group in a non-shared UUI IE such as provided by ASAI).

Incoming calls received with shared user information (shared UUI IE) that are routed outgoing over a non-QSIG service-provider trunk group will forward only ASAI provided user data in a non-shared UUI IE.

UUI Treatment set to `shared` allows all applications to include data items in the UUI IE on a shared forwarding basis. The Shared UUI Feature Priorities page settings along with the **Max. Size of UUI Contents** field on page 2 and the features configured for the system determines what actually is included in the UUI IE. This is the normal setting for non-QSIG trunk groups that route calls to the switch over private or public networks when information forwarding is required and must be used for BSR.

QSIG trunk group

UUI Treatment set to `service-provider` forwarded ASAI provided user data in a non-shared codeset 0 UUI IE and all other user data in codeset 0 Facility IEs as MSI. In this case the **Max. Size of UUI Contents** field is not relevant and the Shared UUI Feature Priorities page does not show nor apply. This setting would only be used for QSIG trunk groups to pre-R6.3 DEFINITY switches for compatibility with existing ASAI applications or when service-provider functionality is wanted (e.g., where shared forwarding of the new data items is not required or for trunk groups to other vendor switches that need user information from the trunk group in a non-shared UUI IE such as provided by ASAI). Incoming calls received with shared data (shared UUI IE) routed out over a QSIG service-provider trunk group, will separate any ASAI provided user data included in the shared UUI IE and forward it in a non-shared UUI IE.

UUI Treatment set to `shared` will forward all user information including ASAI provided user data in codeset 0 Facility IEs as MSI in the SETUP or DISCONNECT message. The UUI IE is never included over a shared QSIG trunk group. In this case, the **Max. Size of UUI Contents** field and the Shared UUI Feature Priorities page do not apply. This is the normal setting for QSIG trunk groups to the switch when information forwarding is required and must be used for BSR.

Send Codeset 6/7 LAI IE option interactions

The Send Codeset 6/7 LAI IE option is independent of the Supplementary Service Protocol and UUI Treatment settings to allow additional flexibility. The switch can have a mix of trunk groups set with non-QSIG or QSIG protocol and with service-provider or shared settings. Calls interflowed over the shared non-QSIG trunk groups will contain the data items to be forwarded with the call in the UUI IE while calls interflowed over the non-QSIG service-provider trunk groups will not (except for ASAI which can always be sent in UUI). Calls interflowed over the QSIG trunk groups will always have MSI user information (except for ASAI whose transport method depends on the UUI Treatment setting).

When a call is LAI interflowed over a non-QSIG service-provider trunk group, the Send Codeset 6/7 LAI IE option being active will result in just the LAI IE being forwarded with the call in a SETUP message. When interflowed over a non-QSIG shared trunk group, setting the Send Codeset 6/7 LAI IE to “yes” includes a codeset 6/7 LAI IE in the SETUP message in addition to the same LAI information included as shared data in the UUI IE. If necessary and appropriate, the LAI information fields (and others) can be set to blank on the Priorities page to exclude these data items from the UUI IE. For details, see Determining user information needs. When interflowed over a QSIG service-provider or shared trunk group with Send Codeset 6/7 LAI IE active, the LAI information will be included as both MSI and in the LAI IE. However, in this case there is no mechanism to eliminate the duplication of data if the codeset 6/7 LAI IE is required.

These combinations can be used when calls are LAI interflowed to the switches previous to the switch with existing ASAI applications using ASAI provided UUI that may or may not be using the LAI IE. Note that codeset 6/7 IEs are not defined for QSIG and other vendor switch treatment of calls with a LAI IE is undefined (could be ignored, blocked, or misinterpreted).

When the trunk group is set to non-QSIG and shared or to QSIG (service-provider or shared), it is recommended that the Send Codeset 6/7 LAI IE option should not also be set to y due to the overhead of sending duplicate information. In some cases, this configuration could exceed the SETUP message and/or user information byte count limits for the network and result in the user information being dropped. Also, transport could cost more in networks which charge for user transport by quantity of bytes transported. An administration warning message will be given when this combination is set for the trunk group. In fact this combination is not recommended except in very limited cases where a mix of early and later switches can be reached over the same trunk group (using a public or switched private network) using Look-Ahead Interflow, and where BSR or UCID is not active or being used and the data that needs to be forwarded with the call can be limited to that supported by the network.

The Send Codeset 6/7 LAI IE option must not be set to y with trunk groups (or in switches) where calls will be interflowed over public networks or virtual private networks that do not support codeset 6/7 transport. In these cases, the codeset 6/7 IE will not be forwarded or the calls may not be routed by the network (blocked due to protocol errors). This can happen in some international situations, notably over networks in Germany.

Table 25: Summary of what is included in the SETUP message

		Supplementary services protocol	
UUI Treatment	**Send Codeset 6/7 LAI IE**	**Non-QSIG (other than b)**	**QSIG (SS b)[10]**
service-provider	n	ASAI provided user info in codeset 0 UUI IE	ASAI provided user info in a codeset 0 UUI IE and all other user info in codeset 0 MSI
	y	ASAI provided user info in codeset 0 UUI IE & a codeset 6/7 LAI IE	ASAI provided user info in codeset 0 UUI IE, all other user info in codeset 0 MSI and a codeset 6/7 LAI IE [11]

10 MSI is sent in codeset 0 Facility IEs.

11 With this combination, the LAI information (LAI Name and Other LAI) will be sent both as MSI (in a Facility IE) and in the LAI IE. Note that LAI IE and shared MSI operation with other vendor switches is undefined.

		Supplementary services protocol	
UUI Treatment	**Send Codeset 6/7 LAI IE**	**Non-QSIG (other than b)**	**QSIG (SS b)[10]**
shared	n	All user info in a shared codeset 0 UUI IE	All user info in codeset 0 MSI
	y	All user info in a shared codeset 0 UUI IE & a codeset 6/7 LAI IE [12]	All user info as codeset 0 MSI and a codeset 6/7 LAI IE [12]

Table 26: When to use specific trunk group options

	Trunk group option settings		
	UUI treatment		Send Codeset 6/7 LAI IE
Situation	**Non-QSIG**	**QSIG**	
Trunk groups over which information forwarding is not required (for LAI, BSR or UCID transport).	service-provider	service-provider	n
Non-LAI interflow or tandem calls to service providers or other vendor switches that do not recognize shared UUI.	service-provider	service-provider	n
LAI to pre-R6.3 switches over networks that block codeset 6/7 IE calls.	service-provider	service-provider	n
LAI to pre-R6.3 switches over networks that allow codeset 6/7 (traditional LAI) with or without ASAI applications that use UUI and/or LAI Info	service-provider	service-provider [13]	y
LAI over public/virtual private network to mixed R6.3 and earlier switches, where the Avaya switches have shared information forwarding. The pre-R6.3 switches may use LAI Info in an ASAI application, but must not use UUI.	shared [14]	shared[14]	y
LAI over public/virtual private network to mixed R6.3 and earlier switches. The R6.3 and earlier switches may use LAI info or UUI in an ASAI application.	service-provider [15]	service-provider [14]	y
BSR and/or LAI to all R6.3 or newer switches [16]	shared	shared	n

12 With this combination, the LAI information (VDN-Name and Other-LAI) will be sent in both the UUI IE and in the LAI IE (setting the UUI Priorities for these items to blank can eliminate the duplication).
13 With this combination, the LAI information will be sent both as MSI (in Facility IEs) and in the LAI IE.
14 With this combination, the LAI information (LAI Name and Other LAI) will be sent in both the UUI IE and in the LAI IE.
15 The LAI IE and ASAI non-shared UUI is supported, but BSR, UCID and other new data items are not.

Interruptible Aux work

About Interruptible Aux

If a skill's designated service level is not met, unavailable EAS agents who are in Auxiliary (AUX) work mode and have an interruptible reason code can be made available. Using this feature, for example, during the call volume spikes, you can use agents in Auxiliary (AUX) work mode to maintain your desired service level.

You enable Interruptible Aux by:

- When a threshold for an interruptible hunt group (skill) is exceeded, setting the Interruptible Aux Threshold policy for any skill/hunt group to calls-warning-threshold, service-level-target, or time-warning-threshold.
- Setting the corresponding threshold field (on the hunt group form) with a value: `Calls Warning Threshold`, `Service Level Target (% in sec)`, or `Time Warning Threshold`

  Setting the **Interruptible?** field on the change reason-code-names screen to `y` (signifies that the corresponding reason code is interruptible).
- Identifying interruptible agents by setting the **Agent Login ID Reserve Level (RL)** field of an agent's skills to one of the "interruptible" values to either a notify type (`n`) or a forced type (`a`, auto-in-interrupt or `m`, manual-in-interrupt)

When a threshold for an interruptible hunt group (skill) is exceeded, agents with that interruptible skill who are in AUX work mode with an interruptible reason code are notified that they are needed. The notification consists of a display message ("You are needed"), flashing auto-in and/or manual-in buttons and an audible, full ring tone. Agents who move to an interruptible Aux mode after the threshold is exceeded are also notified. The duration of notification to "auto-in-interrupt and "manual-in-interrupt" ("forced interruptible")" agents is administrable using the Interruptible Aux Notification Timer (sec) field on page 13 of feature-related system-parameters form. Notification to "requested" agents continues until a further event, such as the agent becoming available or logging off, takes place or the threshold is no longer exceeded.

Forced interruptible agents in AUX work are automatically made available after the timer expires except if they are connected to or being alerted by a non-ACD call, or if the agent is

[16] All switches interflowed to must be R6.3 or newer with shared incoming and outgoing trunk group settings. Tandeming/interflowing through R6.3 or later switches requires shared settings. Switches tandemed through can be older than R6.3 (or other vendor switches that pass codeset 0 UUI or MSI transparently). This is the only combination that supports BSR and new data items information forwarding. In this scenario it is recommended to never set Send Codeset 6/7 LAI IE to "y" in order to save SETUP message space and to ensure operation over networks that do not allow codeset 6/7 IEs. This combination is the recommended setup for Multi-Site Routing.

logged in as an auto-answer agent. A forced interruptible agent administered with auto-answer (automatic call delivery with zip tone) is treated as “requested interruptible”, not forced, even if the RL setting is a forced interruptible type. This prevents the situation where a call is delivered automatically when the agent is not able to respond to the call (for instance, if the agent is not physically at the endpoint). Therefore the forced interruption type is only applicable to agents without the auto-answer administration; that is the agent operates with manual answer where the call is delivered by ringing the endpoint.

Auto-in-interrupt is used when the agent is to be forced available into the Auto-In mode where the agent becomes automatically available after each call. The manual-in-interrupt setting is used when the agent is to be forced available into the manual-in mode where the agent is put into ACW after each call. With the notify-interrupt setting (“requested interruptible” agents), the agent is notified but remains in AUX until agent manually becomes available using an Auto-in or Manual-in button or dial code. The notify-interrupt type can be used with either auto-answer or manual answer operation. Whether the agent is forced available or manually becomes available, the agent becomes available for all of their assigned skills, not just the one that reached the threshold.

The following table explains the treatment of reserve level of the agents. In the following example, the interruptible Aux threshold is set as the Service Level Target. After completing their calls, based on the reason codes, reserve levels, and the actions they take, the agents are notified and placed in the Aux mode as follows:

<table>
<tr><th>Agent</th><th>Reason code</th><th>Reserve level</th><th>When the skill reaches its activation threshold, the agent…</th><th>Agent action (where applicable)</th></tr>
<tr><td>5002</td><td>9 (not interruptible)</td><td>Auto in</td><td>…gets no notification</td><td></td></tr>
<tr><td>5004</td><td rowspan="3">8 (interruptible)</td><td rowspan="3">Notify</td><td>… is notified and moved from Aux to auto-in.</td><td></td></tr>
<tr><td>5005</td><td rowspan="2">...receives a notification to become available.</td><td>Presses the auto-in button with the flashing lamp and becomes available to take calls for all assigned skills.</td></tr>
<tr><td>5007</td><td>The agent is away from the phone and thus does not press the auto-in button. The</td></tr>
</table>

Agent	Reason code	Reserve level	When the skill reaches its activation threshold, the agent…	Agent action (where applicable)
				agent remains in Aux mode.
5008		Manual in	...is moved to manual-in mode.	

The forced interruptible type assignments only apply to agents who are logged into an endpoint that is administered for manual answer where the call rings the endpoint. With manual answer the skill can be assigned RONA (Redirection on No Answered) so that if the agent doesn't answer the call, the call can be redirected back into queue and the agent put into AUX work. Even with manual answer, agents should be trained to remain physically available to answer calls when using a forced AUX work reason code. The forced operation does not apply if the agent is logged into an endpoint with auto-answer because if the agent does not have the headset on and is ready for a call, the call will be delivered and the agent will not know to respond to the caller. RONA does not apply to auto-answer delivered calls.

Other features that help meet service level targets are Service Level Maximizer, which limits agents to availability in the skills when they are below targets and Business Advocate Service Level Supervisor which adds reserve level 1 or 2 agents to a skill based on Expected Wait Time (EWT) threshold targets. Interruptible Aux shares the Reserve Level (RL) field on the Agent Login ID screen with the Business Advocate Service Level Supervisor feature. The agent login RL field for a particular skill can be used for either the Business Advocate Service Level Supervisor feature or for the Interruptible Aux feature for the agent.

Interrupted agents become available for all their skills, not just the skill for which they are interrupted. This functionality is unlike the Service Level Maximizer Auto-reserve agents, which limits the agents' availability only for the skills which are not meeting their service level target.

Interruptible Aux Thresholds

Activation thresholds for Interruptible Aux :

Interruptible Aux notifies agents and makes them available for calls when a skill is not meeting a threshold:

If Interruptible Aux Threshold is…	Administer Threshold…	Interruptible Aux Activation Criteria	Allowed Values
calls-warning-threshold	Calls Warning Threshold	When at least X calls are in a hunt group queue	0-998

If Interruptible Aux Threshold is…	Administer Threshold…	Interruptible Aux Activation Criteria	Allowed Values
service-level-target	Service Level Target	When less than X% of calls are answered in Y seconds	1-99 for % of calls 1-999 for seconds
time-warning-threshold	Time Warning Threshold	When the oldest call has been in queue for at least Y seconds	0-998

The following table explains the three activation thresholds:

Activation thresholds	Description
Calls Warning Threshold	Calls Warning Threshold activates Interruptible Aux if the number of calls in the queue for a hunt group exceeds a specified number. If Calls Warning Threshold is set to 20, interruptible agents in Aux start getting interrupted as soon as the number of calls in the queue goes to 21 or beyond.
Service Level Target	Service Level Target is administered as a percentage of calls answered within specified number of seconds. If, for example, 90% (specified percentage) or more of calls are answered within 15 (specified number of) seconds, then the skill is meeting the Service Level Target. If fewer than 90% of calls are answered within 15 seconds, then the skill is not meeting the Service Level Target and interruptible agents in Aux start getting interrupted.
Time Warning Threshold	Time Warning Threshold activates Interruptible Aux if the oldest call has been in the queue for longer than the specified number of seconds. If Time Warning Threshold is set to 60, interruptible agents start getting interrupted as soon as the duration of the oldest call in the queue for a hunt group exceeds 60 seconds.

Deactivation thresholds for Interruptible Aux:

Based on the Interruptible Aux Threshold policy and the associated threshold value you have set, you also need to set up a deactivation threshold to turn off agent notification.

If Interruptible Aux Threshold is…	Interruptible Aux Deactivation Criteria	Allowed Values for Deactivation threshold
calls-warning-threshold	Fewer than X calls are in the hunt group queue	0-998
service-level-target	At least X% of calls are answered in Y seconds	1-99 for % of calls 1-999 for seconds
time-warning-threshold	The oldest call has been in queue for less than Y seconds	0-998

A threshold for deactivation, in addition to the activation threshold, is used to help keep a buffer (differential) between the two levels at which Interruptible Aux is activated and deactivated.

This buffer avoids a situation of the Interruptible Aux feature going on and off frequently when the Calls Warning, Service Level, or Time Warning values fluctuate around the activation threshold. The values of the activation and deactivation thresholds must differ by at least ‘1’ (one).

For example, if the service level threshold is 80% calls in 15 seconds, a deactivation threshold of 90% calls in 15 seconds will keep the notifications on until the service level rises back to 90% in 15 seconds. The system does not deactivate Interruptible Aux until the Service Level gets up to the specified level. If the service level drops below the Service Level Target again, the system reactivates Interruptible Aux notifications.

For Calls Warning and Time Warning, the deactivation threshold needs to be lower than the activation threshold. For example, if the activation threshold for the Calls Warning Threshold is 100, the deactivation threshold could be set to 90. The Interruptible Aux notifications start when the number of calls in the queue reaches 100, but the notifications continue until the number of calls in the queue drops to 90.

Interruptible Aux considerations

- CTI-based applications using ASAI:
    - Can change an agent’s work mode to an interruptible Aux mode
    - Can receive events indicating that an agent was made available
- When interrupted, agents become available for all skills, not just the one that exceeded its service level. Subsequent calls delivered to a newly available agent are not necessarily for the critical skill.
- Several Reason Codes cannot be designated as interruptible. These are the Default Reason Code, the Auto-answer IP Failure Aux Work Reason Code, and the Maximum Agent Occupancy Aux Work Reason Code.
- When a large number (1000 or more) forced interruptible agents are in Aux mode with an interruptible reason code, simultaneously moving all of them to an available mode can degrade the performance of the Communication Manager. To alleviate this degradation, only 5 agents are notified every ½ second until all needed agents are notified. Similarly, the agents are forced to an available mode at the same rate: 5 agents every ½ second.

Intraflow and Interflow

About Intraflow

Use Call Coverage with Intraflow to redirect ACD calls from one split or skill to another conditionally, according to the coverage path’s redirection criteria. For example, you can define a split or skill's coverage path to automatically redirect incoming ACD calls to another split or skill when a telephone is busy or unanswered. You can redirect calls to less busy splits/ skills, for more efficient call handling.

Use Call Forwarding with Intraflow to unconditionally forward calls for a split or skill.

About Interflow

Interflow allows you to redirect ACD calls from a split or skill on one communication server to a split or skill on another communication server or external location. Use Call Forwarding All Calls with Interflow to unconditionally forward calls directed to a split or skill to an off-premises location. Calls can be forwarded to destinations off the communication server (that is, phone numbers on the public telephone network). You cannot use Call Coverage with Interflow. If a coverage point station or split or skill is forwarded/interflowed, it is taken out of the coverage path.

For details on how to forward calls to an external extension and on Call Coverage redirection criteria, see *Avaya Aura™ Communication Manager Feature Description and Implementation*. See Call Vectoring and Interflow and Intraflow interactions on page 250 for information on advanced Interflow capabilities.

Intraflow and Interflow detailed description

Assign an inflow threshold for each split or skill receiving Intraflow and Interflow calls. This threshold prevents a split or skill from receiving new ACD calls if the oldest call in the queue has been there longer than the threshold. If an ACD call is forwarded or redirected through Call Coverage, but cannot be routed to another split or skill or coverage path point, it remains in queue at the original split or skill even though coverage tone may be heard.

For a split or skill with Intraflow and Call Coverage assigned, you can also assign Priority on Intraflow. When an ACD call intraflowing from a split or skill with Priority on Intraflow to a covering split or skill enters the queue, that call is placed ahead of non priority calls but behind other priority calls already in the queue. All priority calls are answered before any non priority calls.

Calls intraflowed using Call Coverage to a covering split or skill are never connected to the first delay announcement at the covering split or skill. Calls redirected using Call Forwarding receive the delay first announcement at the forwarded-to split or skill, but never receive a forced first announcement.

As an illustration of how Intraflow works, assume the following:

- A call is intraflowed from split 1 to split 2 using Call Coverage.
- Split 1 is assigned priority on intraflow.
- Split 2 has a queue with three priority calls and four non priority calls.

- Split 2 has an inflow threshold of 90 seconds and the oldest call in queue at split 2 has been in queue for 60 seconds.
- Split 2 has been assigned a second delay announcement and has a second delay announcement interval of 45 seconds.
- Music-on-Hold is provided.

When the call is intraflowed from split 1 to split 2, the call is placed in the split 2 queue as the fourth priority call, ahead of the four non priority calls. The call stays in the queue for 45 seconds and is still not answered. Then the call is connected to the second delay announcement for split 2. After the announcement, the caller hears music until an agent answers the call.

You can assign a Coverage ICI button to an agent's multiappearance phone. The agents use the button to identify a call that is intraflowed from another split or skill. When an agent receives such a call, the button lamp lights.

Interflow and Intraflow considerations

The same coverage path can be used for as many splits/skills as desired. You should administer redirection criteria for a split or skill coverage path so that calls are redirected under Busy or Don't Answer conditions. Do not use All or Send All Calls as redirection criteria.

Interflow and Intraflow interactions

Interaction	Description
Call Coverage	All splits/skill with the same coverage path are automatically assigned the same Don't Answer Interval. The default Don't Answer Interval is 2. If Intraflow using Coverage is active, the Coverage Don't Answer Interval associated with Call Coverage begins when a call enters the split or skill queue. If the Coverage Don't Answer interval expires before either of the two delay-announcement intervals expires, a call is redirected to coverage. If either of the delay-announcement intervals expires before the Coverage Don't Answer interval, the call is connected to a delay announcement, if available. If no coverage point is available to handle a call, a call remains in queue and may then be connected to a delay announcement.

Interaction	Description
Temporary Bridged Appearance	If an ACD call is routed to an agent but is intraflowed to another split or skill before being answered, the Temporary Bridged Appearance at the agent's telephone or console is no longer maintained.
Look-Ahead Interflow	Use Look-Ahead Interflow (LAI) to balance the load of ACD calls across multiple locations. For more information, see About LAI.

Location Preference Distribution

Description of Location Preference Distribution

Location Preference Distribution tries to route incoming Automatic Call Distribution (ACD) calls to agents located in the same location as the incoming trunk on which the call originated whenever possible. Note that this feature can be used to avoid unnecessarily using trunks between locations, thus reducing trunking costs or saving trunks for additional incoming calls or other purposes. If an incoming caller cannot be matched with an agent in the same location, calls are routed to agents at different locations. In this case, the routing is determined by administered distribution algorithms without regard to location.

When there is more than one choice for call delivery, Local Preference Distribution matches the trunk and the agent location numbers. The Multiple Locations feature defines the location number. Delivery preference is given to the agent whose location number matches the incoming trunk location number. Location Preference Distribution takes precedence over most other caller-agent selection features except for direct agent and reserve agent calls.

Reasons to use Location Preference Distribution

Customers can use this feature to:

- Lower customer networking costs by reducing the amount of intra-switch network traffic
- Improve audio quality

Prerequisites for using Location Preference Distribution

You can set Location Preference Distribution only if all of the following conditions are true:

- Expert Agent Selection (EAS) is enabled and active.
- The**Multiple Locations** field is set to `y`on Page 4 of the Systems-Parameters Customer-Options screen.
- The Call Center release is 3.0 or later.

About location numbers

Agents obtain a location number when they log in. The location number remains with the agent until the agent logs out. Local Preference Distribution uses location numbers to match the incoming trunk and the agent.

This section includes the following topics:

- The Multiple Locations feature on page 252
- Changes to the location number on page 252
- How trunks, stations, and agent endpoints obtain location numbers on page 253
- How to set up a location number on page 253

The Multiple Locations feature

The location number used by Location Preference Distribution is defined by the Multiple Locations feature. The Multiple Locations feature was originally developed to display local time on station sets that are located in a different time zone from their connecting switch. Call Center added the ability to have the Call Management System (CMS) track and report on agents and trunks using a location number, or location ID. The location number for the agent is provided to CMS when the agent logs in. The location number for a trunk is provided to CMS when a measured incoming call is received.

Changes to the location number

During call center activity, location numbers can be changed through administration, or the Multiple Locations feature can be disabled. If this happens, the Location Preference Distribution feature uses the agent's ID obtained when the agent logged in and the call's ID when the call was originally received. During such a change, the Location Preference

Distribution matching may not be appropriate until all agents log out and log back in again. The software does not check for these changes.

How trunks, stations, and agent endpoints obtain location numbers

Trunks, stations, or agent endpoints obtain location numbers as follows:

- Non-IP phones and trunks inherit the location number from their connected hardware. For example, a non-IP phone inherits its location number from a cabinet, remote office, or media gateway.
- IP phones indirectly obtain their location numbers when the location numbers are administered on the Network Region screen. This screen applies location numbers to all phones in that IP region.

  If the **location** field is left blank on the Network Region screen, the IP phone derives its location from the CLAN board located on the cabinet or gateway where the phone is registered.
- IP and SIP trunks obtain their location from the cabinet containing the CLAN/NIC that the trunk is signaling through.

If none of the above applies, location 1 is used as the default.

How to set up a location number

Important:

For details on how to use these forms and the commands associated with these forms, see Administrator Guide for Communication Manager.

Establishing location numbers

Use the Locations screen to establish location numbers. This will define the characteristics of the location that can include:

1. Time zone offset between local standard time and the remote server location
2. Daylight savings rules used by any expansion port networks (EPNs) located in different time zones
3. Number plan area codes
4. An ARS prefix that is required for 10-digit calls.

The ARS prefix defines calls that are routed to the relevant location, such as E911 local call routing

Assigning location numbers to EPN cabinets or to the Media Gateway

Do one of the following tasks:

1. Use the Cabinet Description screen to assign location numbers to the appropriate EPN cabinets.

   Use the **`change cabinet xx`** command.
2. Use the Media Gateway screen to assign location numbers to the Media Gateway.

Result

You can assign the same location number to more than one cabinet or gateway that is located in the same time zone. Note that you can assign all Avaya DEFINITY and Media Server configurations, except the S8100 Media Server configuration, to locations other than 1. The DEFINITY Server CSI and SI configurations default to location 1. Digital and analog station sets get their defined location number based on the port location of the cabinet or gateway. The circuit switch trunks also obtain their gateway number in the same manner.

Assigning the location by IP network region

Use the Change IP-Network-Region screen to administer the location by IP network region.

This sets the following conditions:

1. The correct date and time information and trunk routing based on the IP network region.
2. The correct date and time worldwide displayed for IP phones registered with a server, but located at a remote site or a site with a S8300 Media Server with a G700 or equivalent gateway.

   The IP phone can be administered in a different network region from other Communication Manager endpoints, and in the same location as the S8300 Media Server or remote office users. This allows IP endpoint users the ability to move from location to location and always have correct display information. Remote users are identified in a network region and location that routes them to correct 911 services or notifies them through announcements that they are in a different 911 jurisdiction than where they are registered.

Call-handling conditions

You can use Location Preference Distribution to administer how the system handles agent-surplus conditions and call-surplus conditions.

Agent-surplus conditions

An agent-surplus condition is when available agents are waiting for incoming ACD calls. The Location Preference Distribution algorithm routes new incoming ACD calls to an idle agent located within the same location number as the calling party's trunk or station. If there is no match for an idle agent, the incoming ACD call is routed to the agent at the top of the skill's free-agent list based on the administered selection criteria.

Call-surplus conditions

A call-surplus condition is when there are ACD calls in queue waiting for an available agent. The Location Preference Distribution algorithm routes the next best queued call to a multi-skilled EAS agent that has the same location number as the call. The next best queued call is determined by the appropriate Avaya Business Advocate or non-Advocate algorithm. If there is no match between the queued ACD call and the skills administered for the agent, the normal best queued ACD call selection is made by the appropriate Avaya Business Advocate or non-Advocate algorithm. The selection is made without any location number preference of the queued ACD call or agent based on the administered selection criteria.

Location Preference Distribution selects calls only from the top of the queue for each skill. Location Preference Distribution does not try to match the agent's location from the skill queue. For example, when an agent with five skills becomes available and has calls in queue for three of his skills, Location Preference Distribution looks at the call at the top of each queue. If one or more calls match the location of the agent, Location Preference Distribution uses the administered selection criteria to pick a call for the agent from the same location.

Local Preference Distribution interactions

Only the features that are impacted by Local Preference Distribution are described in this section.

Feature	Description
Avaya Business Advocate	Avaya Business Advocate provides call handling preferences based on: • A service objective that is assigned on the Hunt Group screen. • Agent percentage allocation that is assigned on the Agent Login ID screen on a per skill basis. Local Preference Distribution takes precedence over any Avaya Business Advocate call handling preferences.
Best Service Routing (BSR)	Local Preference Distribution is used to select an available agent within the Call Center during consider and queue-to best step operations. Local Preference Distribution is not used across system sites. In this case, there is no notion of a multi-site network region.
Call Admission Control	Location Preference Distribution does not interact directly with the Call Admission Control feature. However, when Location Preference Distribution selects a trunk-agent combination at the same location, not as much overall bandwidth is needed between locations. Location Preference Distribution cannot circumvent an inter-node bandwidth blockage between two Wide Area Network (WAN) switch or gateway sites when any of the following conditions exist: • All agents at an incoming trunk switch or gateway location are busy. • The WAN bandwidth has reached capacity between the incoming trunk location and a remote location. • An agent is unavailable at the remote location. In order to bypass the blocked WAN call path, BSR - or any other multi-site feature - routes an incoming ACD trunk call to an available agent at the remote location. The call is routed over the Public Switched Telephone Network (PSTN), Integrated Services Digital Network (ISDN) tie trunk, as well as other types of networks.
Call handling conditions	For information about how call handling conditions are used with Location Preference Distribution, see Call-handling conditions on page 255
Conference and Transfer	When an incoming trunk call is transferred to an ACD hunt group and the agent is available when the transfer occurs, the location number is for the agent that transferred the call. If the transferred call queues and the transferring agent drops before an agent is available, the location number is for the incoming trunk.
Direct Agent Calling	Direct Agent calls take precedence over Location Preference Distribution.

Feature	Description
Dynamic Advocate	Dynamic Advocate provides call-handling preferences based on: • A Percentage Allocation Distribution (PAD) group type preference assigned on the Hunt Group screen • A Percentage Allocation (PA) assignment for the skill assigned on the Agent Login ID screen. • The Service Objective field on the Hunt Group screen overrides the service objective assigned for Service Level Supervisor (SLS) on the Hunt Group screen. Local Preference Distribution takes precedence over Dynamic Advocate call-handling preferences.
EAS	Expert Agent Selection (EAS) must be enabled and active before you can assign Local Preference Distribution.
Inter-Gateway Alternate Routing (IGAR)	IGAR provides the ability to alternately use the Public Switched Telephone Network (PSTN) to carry the bearer portion of a call when the IP-WAN is incapable of carrying the bearer location. Local Preference Distribution does not interact directly with IGAR. However, when Location Preference Distribution selects a trunk-agent combination in the same location, the need for IGAR between locations is reduced.
IP hard and soft phones	For more information about how IP phones interact with Location Preference Distribution, see How trunks, stations, and agent endpoints obtain location numbers on page 253.
Multiple Locations	For information about how the Multiple Locations feature interacts with Location Preference Distribution, see The Multiple Locations feature on page 252.
Path replacement	When an incoming trunk call receives path replacement before the call is delivered to an agent, the original incoming trunk retains the location number for the call.
Percent Allocation	See Reserve agents.
Reserve agents	You can assign reserve agents using any of the following features: • Service Level Maximizer (SLM) • Avaya Business Advocate Service Level Supervisor (SLS) • Avaya Business Advocate Percent Allocation In most cases, the selection of an agent or a call based on Location Preference Distribution takes precedence over SLM, SLS, or Percent Allocation. Nevertheless, SLM, SLS, and Percent Allocation take

Feature	Description
	precedence when the system chooses a reserve agent for the following reasons: • The skill is above the Estimated Wait Time (EWT) threshold with SLS. • The service level is below the threshold with SLM or Percent Allocation. **Note:** If more than one reserve agent is eligible for the call, Location Preference Distribution is used to choose the agent. For more information about reserve agents, see Avaya Business Advocate User Guide.
Separation of Bearer and Signaling (SBS)	The location number of an incoming SBS call is obtained from the bearer trunk assignment.
Service Level Maximizer (SLM)	See Reserve agents.
Service Level Supervisor (SLS)	See Reserve agents.

Look-Ahead Interflow (LAI)

About LAI

Look-Ahead Interflow (LAI) enhances Call Vectoring for call centers with multiple ACD locations. LAI allows these centers to improve call-handling capability and agent productivity by intelligently routing calls among call centers to achieve an improved ACD load balance. This service is provided by ISDN D-channel messaging over QSIG or non-QSIG private networks, virtual private networks, or public networks. The receiving switch is able to accept or deny interflowed calls sent by the sending switch.

LAI has the following basic attributes:

- Produces First in First Out (FIFO) or near-FIFO call processing
- Includes enhanced information forwarding, that is, codeset 0 user information transport

LAI requirements

The following items are criteria for basic LAI call control operation over a virtual private network or a public switched network:

- The sending and receiving call center locations must have ISDN (PRI or BRI) trunk facilities.

  Note:
  ATM trunking and IP trunking can be set up to emulate ISDN PRI. For information on setting this up, see Administering Network Connectivity on Avaya Aura™ Communication Manager, and ATM Installation, Upgrades and Administration using Communication Manager.

- The switch must support the ISDN country protocol.
- LAI has been tested with several major carriers. To find out if these capabilities work with your carrier, check with your account team for the most current information. If testing has not been done to verify operation over the public networks that are involved with the preferred specific configuration, use of private ISDN trunking between the nodes should be assumed until successful testing is complete.
- The ISDN SETUP and DISCONNECT messages are transported between sending and receiving locations, for example, SS7 or equivalent public network connectivity.
- A receiving-end generated DISCONNECT message must transmit back to the sending the switch call center without changing the cause value.

  Conversion of the DISCONNECT message to a progress message (with a Progress Indicator Description set to 1 and a cause value other than 127 included) is a valid reject message and compatible with LAI.
- Progress messages that are generated towards the sending end by intervening network switches must have the Progress Indicator Description set to 8 so that the switch does not consider the call accepted or rejected.
- ISDN codeset 0 user information transport supports LAI information forwarding. As an alternative, LAI can use dedicated VDNs at the receiving location to provide an equivalent display of the forwarding application identity and set trunk group options to not send either the codeset 6/7 LAI IE or codeset 0 information transport.

Note:
Best Service Routing (BSR) cannot use these LAI alternatives. BSR must use ISDN codeset 0 user information transport.

Example of a two-switch configuration

Look-Ahead Interflow (LAI) is enabled through the use of call vectors and their associated commands. For a two-switch configuration, these vectors are included in both the sending switch, which processes vector outflow, and the receiving switch, which processes vector inflow.

LAI command set

LAI enhances call vectoring so that calls interflow only to those remote locations that can accept the calls.

LAI is achieved through a set of vector commands. The following table lists the call-acceptance vector commands that are used in LAI.

Table 27: Call-acceptance vector commands

Command	Qualification
announcement	Announcement available Queued for announcement Retrying announcement
check split	Call terminates to agent Call queued to split
collect digits	Always (except for ced and cdpd digits, which are neutral)
converse-on split	VRU answers the call Call queued to converse split
disconnect	With announcement and announcement available With announcement and queued for announcement With announcement and retrying announcement
messaging split	Command successful Call queued
queue-to split	Call terminates to agent Call queued to split
route-to	Terminates to valid local destination Successfully seizes a non-PRI trunk Results in a LAI call attempt, and the call is accepted by the far-end switch
wait-time	Always (except **wait-time hearing i-silent**, which is neutral)

If the receiving switch decides it is unable to accept the LAI call, call denial is accomplished by executing one of the vector commands that are listed in the following table.

Note:
It is recommended that you use `busy` instead of `disconnect` to allow for compatibility with similar network services such as Alternate Destination Redirection (ADR).

Table 28: Call-denial vector commands

Command	Qualification
`busy`	Always
`disconnect`	Without announcement With announcement but announcement unavailable
`reply-best`	Always; used with BSR

The vector commands that are shown in the following table are considered neutral because they do not generate either call acceptance or denial messages.

Table 29: Neutral vector commands

Command	Qualification
`adjunct routing link`	Always
`announcement`	Announcement unavailable
`check split`	Call neither terminates nor queues
`collect ced/cdpd digits`	Always
`consider`	Always - used with BSR
`converse-on split`	Call neither terminates nor queues
`goto step`	Always
`goto vector`	Always
`messaging split`	Command failure
`queue-to split`	Call neither terminates nor queues
`route-to`	Unsuccessful termination Trunk not seized LAI call denied by the far-end switch
`stop`	Always
`wait-time hearing i-silent`	Always **Note:** This command is used following an `adjunct routing link` command in applications

Command	Qualification
	where the adjunct decides whether to accept or reject the Look-Ahead calls.

Forms and fields used to administer LAI

The following forms and fields are required to administer the LAI feature.

Screen	Field
System Parameters Customer-Options	• **Vectoring (Basic)** • **ISDN-PRI** • **Look Ahead Interflow**
Trunk Group (ISDN)	• **Outgoing Display** • **Codeset to Traveling Class Mark, Look Ahead** • **Supplementary Service Protocol** • **UUI Treatment**
Feature-Related System Parameters	**Interflow-Qpos EWT Threshold**
ISDN Numbering - Public/ Unknown	• **Ext Len** • **Ext Code** • **CPN Prefix**
Call Vector	Complete a screen for each Look-Ahead Interflow vector

See Call Vectoring for associated Call Vectoring administration.

- System-Parameters Customer-Options - For full functionality, options must be enabled at both the sending and receiving communication servers. If Look-Ahead Interflow is not optioned on the receiving communication server, interflow still results on a look-ahead basis. However, the forwarded Dialed Number Identification Service (DNIS) (sending communication server VDN name) information is ignored and tandem Look-Ahead Interflow is not provided.
- Trunk Group Screen (ISDN) - If you do not want the call originator's display to update on each Look-Ahead Interflow call attempt, look-ahead calls should be routed over trunk groups with the Outgoing Display field set to n.
- Feature-Related System Parameters Screen - Administer the **Interflow-Qpos EWT Threshold** field when working with enhanced Look-Ahead Interflow. Any calls that will be answered before this threshold will not be interflowed (therefore saving CPU resources on the Avaya Server that is driving the Avaya Communication Manager).
- ISDN Numbering - Public/Unknown Screen - Administer a CPN Prefix for each Vector Directory Number (VDN) that maps to a vector used to place Look-Ahead Interflow calls. If

you do not, a Look-Ahead Interflow DNIS of all blanks appears on the answering agent's phone.

For private network non-QSIG connectivity with direct facilities between the communication servers, administer Look-Ahead Interflow DS1/E1 circuit packs with Country Protocol Option 1 independent of the country where the system is located.

LAI considerations

Consideration	Description
Carrier compatibility	LAI has been tested with several major carriers. To find out if these capabilities work with your carrier, check with your account team for the most current information. If testing has not been done to verify operation over the public networks involved with the preferred specific configuration, use of private ISDN trunking between the nodes should be assumed until successful testing has been completed. ISDN routing with LAI enabled: All calls routed over ISDN facilities by a route-to number with cov n or route-to digits with cov n vector command on a communication server where Look-Ahead Interflow is enabled are treated as Look-Ahead Interflow call attempts. A vector may route a call over an ISDN facility to a destination that is not a VDN. The sending communication server processes this call as a Look-Ahead Interflow call even though it is not. ISDN processing at the receiving communication server causes the call to always be accepted. However, the DNIS and any other information in the Look-Ahead Interflow information forwarded with the call are ignored. Interim call handling before LAI is accepted by receiving communication server: Until the look-ahead attempt is accepted by the receiving communication server, the caller continues to hear any feedback applied by the sending communication server vector and will remain in any split or skill queues.
Call handling with Route-to number or Route-to digits handling with coverage	Route-to number with coverage y or route-to digits with coverage y commands never result in a Look-Ahead Interflow call attempt.

Consideration	Description
	The sending end assumes the call is always going to be accepted. This command always completes the call. Moreover, the command should not be used if the vector at the receiving communication server might deny the call, since the caller in this case would be given a busy signal or would be disconnected. Use this command with coverage y only when you want unconditional interflow (with Look-Ahead Interflow active) and the terminating communication server is set up accordingly.
Continuity during call transfer between communication servers	Audible feedback may be provided to the caller before interflow is attempted. Therefore, another audible feedback from the receiving communication server may confuse the caller. For example, a caller hearing ringback on the sending communication server may be confused if music is applied suddenly when the call interflows to the receiving communication server.
Backward compatibility of LAI applications	For backward compatibility of LAI applications between Avaya communication servers, leave the Send “Codeset 6/7 LAI IE” option on the Trunk Group screen set to its default y. Existing LAI applications will continue to operate as before, even after you upgrade. You can use enhanced LAI available in the communication server without any network or trunk administration changes, by adding the “interflow-qpos” conditional to original LAI vectors (the conditional applies only to calls in queue).
AAR/ARS	ISDN facilities used to provide Look-Ahead Interflow to a VDN on another communication server in a private network can use the AAR feature if private facilities are to be used for call routing.
Agent Telephone Display	If collected digits are forwarded with an interflowed call, the forwarded digits are displayed to the answering agent (unless they’re overridden with newly collected digits) on the telephone display.

Consideration	Description
Attendant Control of Trunk Group Access	Calls will not route over a trunk with Attendant Control of Trunk Group Access set.
Authorization Codes	Authorization Codes must not be required for interflow routing. Assign a high enough FRL to the VDN so that the route desired for routing interflow calls can be used without requiring an Authorization Code entry. If a route choice is encountered that requires a higher FRL, the interflow is considered an invalid destination (rejected for Look-Ahead Interflow or not available for standard interflow) without the application of recall dial tone.
BCMS	BCMS does not log LAI attempts, nor does it report accumulated in-VDN time.
Call Detail Recording - Sending Server	No Ineffective Call Attempt or Outgoing Call CDR records are generated for vector **`route-to`** commands that are unsuccessful including denied Look-Ahead Interflow attempts. If a local (on-communication server) call to a VDN generates a Look-Ahead Interflow call attempt that is accepted, and answer supervision is returned from the receiving communication server, then one Outgoing Call CDR record is generated with the originating extension as the calling number. If an incoming (off-communication server) call to a VDN generates a Look-Ahead Interflow call attempt that is accepted, and no answer supervision is returned from the receiving communication server, then one incoming CDR record is generated. The VDN is the called number, and the duration is from the time answer supervision was provided to the incoming trunk. If an incoming (off-communication server) call to a VDN generates a Look-Ahead Interflow call attempt that is accepted, and answer supervision is returned from the receiving communication server, then two incoming CDR records are generated: • An incoming record with the VDN as the called number and the duration as the time since answer supervision was provided to the incoming trunk. This is generated if the

Consideration	Description
	call is initially answered in the sending communication server before interflow takes place. • An outgoing record containing the incoming trunk information as the calling number and the dialed digits and the outgoing trunk information as the called number.
Call Detail Recording - Receiving Server	On the receiving communication server, an incoming Look-Ahead Interflow call is treated like any other incoming vector call. If answer supervision is returned by the vector, and the call is never terminated to another destination, then the VDN extension is recorded as the called number in the CDR record. If the call terminates to a hunt group, then the VDN, hunt group, or agent extension is recorded as the called number. If the `Record VDN in Record` field of the Feature Related System Parameters is y, then the VDN extension overrides the `Call to Hunt Group - Record` administration option for vector calls.
Call Prompting	Digits collected at the sending communication server, no matter how they are collected (caller-entered, ASAI provided, CINFO provided, etc.) are forwarded with interflowed calls and available at the remote communication server using information forwarding. For more information, see Information Forwarding. **Note:** Dial-ahead digits are not forwarded with the call. There is a maximum of 16 forwarded digits.
Centralized Attendant Service	A centralized attendant can be a Look-Ahead Interflow destination.
VDN Name Display	The VDN name (part of the LAI information forwarded with calls) can be up to 15 characters long. Any characters over this limit will be dropped. On Communication Manager, a VDN name can be administered with as many as 27 characters.

Consideration	Description
Distributed Networking - Manufacturers Specific Information (MSI)	LAI (whether enhanced or not) may not function with systems from other vendors (unless that vendor develops a corresponding capability that works with the Avaya communication server).
Facilities Restriction Level and Traveling Class Marks	The FRL for interflow over ARS/AAR route choices is assigned to the original VDN used for the incoming call.
Incoming Call Management	The adjunct routing capabilities of vectoring can be used at the sending communication server to determine if a call should be interflowed. Adjunct routing at the receiving communication server can be used to tandem the call to a far-end communication server. If the call terminates to a tandem trunk, two CDR records are generated: • An incoming record with the VDN as the called number and the duration as the time since answer supervision was provided to the incoming trunk. • An outgoing record containing the incoming trunk information as the calling number and the dialed digits and the outgoing trunk information as the called number.
Network Access	LAI operates over public, private, or virtual private (for example, SDN) ISDN-BRI and -PRI networks that meet minimum network requirements. The sending of a Look-Ahead Interflow codeset 6/7 information element is counted toward Message Associated User-to-User Information (MA-UUI) counts.
Path replacement for QSIG/DCS ISDN calls	For calls that are waiting in queue or in vector processing, even if the call is not connected to an answering user, path replacement using QSIG can be attempted to find a more optimal path for this call. This results in more efficient use of the trunk facilities. The QSIG ISDN or DCS ISDN trunk path-replacement operation can be triggered for ACD calls by the Look-Ahead Interflow route-to number vector step, BSR queue-to best vector step, and the Adjunct Routing vector steps.

Consideration	Description
	The ability to track a measured ACD call after a path replacement has taken place is available for CMS versions r3v9ai.o or later. Starting with the r3v12ba.x release, CMS reports a path replacement as a rename operation rather than a path replacement. The rename operation properly reports scenarios where a path replacement takes place from a measured to an unmeasured trunk facility. Avaya recommends that you upgrade CMS to r3v12a.x or later and administer all trunks associated with path replacement as measured by CMS to ensure better CMS tracking of path-replaced calls.
QSIG	LAI and information forwarding function over QSIG trunk facilities if the remote locations are Avaya communication servers. You may get LAI call control functionality with other vendors if an Avaya communication server is the starting point.
Redirect on No Answer (RONA)	Calls redirected to a VDN by RONA can be subsequently processed and routed by LAI applications.
Service Observing	You can observe a call in LAI processing using VDN observing throughout the life of the call (as long as the call is still connected through the local communication server). All current restrictions on Service Observing still apply. Incoming calls can be service observed at the remote communication server.
Trunk-to-Trunk Transfer	Interflowed calls may be transferred by a receiving communication server to another trunk connection.
VDN Override	The name of the active VDN for a call is displayed at the remote answering agent.

How traditional LAI works

Traditional LAI is recommended when the preferred call flow performs LAI attempts before queuing the call.

This section includes the following topics:

LAI commands

LAI uses the commands that are included within the Basic Call Vectoring and Call Prompting features:

- **`route-to number with coverage n`** or **`route-to digits with coverage n`** command on a switch that has LAI enabled and that successfully seizes an ISDN trunk automatically results in a normal LAI call attempt being placed. The call attempt can be rejected or accepted by the remote end.
- **`route-to number with coverage y`** or **`route-to digits with coverage y`** command never results in a LAI call attempt. The sending end assumes that the call is always going to be accepted. This command always completes the call. Moreover, the command should not be used when the vector at the receiving location ends up denying the call, since the caller in this case is given a busy signal, or the call is disconnected. Use this command with coverage set to **`y`** only for those cases when an unconditional interflow is wanted (with LAI active) and the terminating switch is set up to handle this type of call.

When a LAI call attempt is made, Call Vectoring at the sending location checks a potential receiving location to determine whether to hold or send the call. While this is done, the call remains in queue at the sending location. As such, the call can still be connected to the sending-location agent if one becomes available before the receiving location accepts the call.

Call Vectoring at the receiving location decides whether to accept the call from the sending location or to instruct the sending location to keep the call. In the latter case, the sending location can then either keep the call, check other locations, or provide some other treatment for the call. Conditions for sending, refusing, or receiving a LAI call attempt can include a combination of any of the following:

- Expected wait time for a split
- Number of staffed or available agents
- Number of calls in queue
- Average speed of answer or the number of calls active in a VDN
- Time of day and day of week
- Any other legitimate conditional

If the call is accepted by the receiving switch, the call is removed from any queues at the sending switch, and call control is passed to the receiving switch. If the call is denied by the receiving switch, vector processing continues at the next step at the sending switch. Until the call is accepted by either switch, the caller continues to hear any tones applied by the sending switch. If the call is denied, the call vector can apply alternate treatment, such as placing another LAI call to an alternate backup switch.

Note:

The LAI operation is completely transparent to the caller. While a LAI call attempt is being made, the caller continues to hear any audible feedback that is provided by the sending switch vector. The caller also maintains his or her position in any split queues until the call is accepted at the receiving switch.

LAI passes Call Prompting digits collected in the sending switch to the receiving switch by codeset 0 user information transport. For more information, see Information Forwarding on page 225.

Example of traditional LAI

The vectors in the sending switch use the **goto** command to determine whether the call should be sent to the receiving switch. Recall that the **goto** command tests various outflow threshold conditions such as expected wait time. If the expressed condition is met, a branch is made to the appropriate **route-to** command. This command sends the call to the receiving switch, which, as already noted, can accept or deny the call.

The following example shows an outflow vector that might be included in a sending switch.

Using LAI with route-to commands to outflow calls

```
1. wait-time 0 secs hearing ringback
2. goto step 5 if expected-wait for split 3 pri m < 30
3. route-to number 5000 with cov n if unconditionally
4. route-to number 95016781234 with cov n if unconditionally
5. queue-to split 3 pri m
6. announcement 3001
7. wait-time 30 secs hearing music
8. goto step 6 if unconditionally
```

If split 3 has an expected wait time of less than 30 seconds (step 2), step 5 queues the call to the split's queue at a medium priority.

If the expected wait time is 30 seconds or more, LAI attempts are made in steps 3 and 4. If the call is accepted by one of the receiving switches call control passes to the receiving switch.

If the receiving switches deny the call, the call queues to split 3 and announcement 3001 plays. The caller then hears music (interrupted by announcement 3001 every 30 seconds).

Receiving switch operation

When the receiving switch receives the LAI request, the call first routes to a VDN. The VDN then maps the call to the receiving switch's inflow vector, and vector processing begins, starting with inflow checking. Inflow checking is enabled by conditional goto commands in the inflow vector. The decision to accept or deny a call can be based on checks such as any of the following:

- Expected Wait Time
- Number of staffed agents
- Number of available agents
- Time-of-day/day of the week
- Number of calls in split's queue
- Average Speed of Answer
- Active VDN Calls
- ANI
- II-Digits
- CINFO ced and/or cdpd digits
- Collected digits forwarded from the sending switch

Once inflow checking is complete, acceptance of the LAI call is accomplished by executing any of the vector commands listed in Table 27: Call-acceptance vector commands on page 260.

Note:

For each of the commands listed in Table 27: Call-acceptance vector commands on page 260, Table 29: Neutral vector commands on page 261 and Table 28: Call-denial vector commands on page 261, only one of the corresponding qualifications needs to be true for the command to effect the desired result, which is call acceptance, call denial, or no effect on such acceptance or denial.

The following example shows an inflow vector that might be used by a receiving switch.

Using inflow checking for LAI requests

```
1. goto step 6 if expected-wait in split 1 pri h > 30
2. queue-to split 1 pri h
3. announcement 4000
4. wait-time 2 seconds hearing music
5. stop
6. busy
```

Step 1 of this inflow vector checks the inflow thresholds. The **`goto step`** command in step 1 checks the expected wait time in split 1. If the expected wait time is greater than 30 seconds, a branch is made to the **`busy`** command in step 6. If executed, the **`busy`** command denies the call, and the receiving switch returns a call denial message to the sending switch. The sending switch, in turn, drops the LAI call attempt and then continues vector processing at the next vector step.

If the expected wait time in split 1 is less than or equal to 30 seconds, the receiving switch returns a call acceptance message to the sending switch, and call control is passed to the receiving switch. Thereafter, the call is queued to split 1 in the receiving switch (step 2). Once queued, the caller receives the appropriate announcement in step 3 and is then provided with music until the call is answered by an agent or abandoned by the caller (steps 4 and 5). Remember that the **stop** command halts vector processing but does not drop the call.

If the sending switch does not receive a call acceptance or call denial message within 120 seconds after the LAI call request, the LAI attempt is dropped. The sending switch continues vector processing at the next step.

How enhanced LAI works

About enhanced LAI

Enhanced LAI uses the same basic vectoring commands as traditional LAI, but adds the conditional **interflow-qpos**. Enhanced LAI is recommended when the preferred call flow performs LAI attempts after queuing the call.

Using enhanced LAI **interflow-qpos** conditional:

- Produces First in First Out (FIFO) or near FIFO call processing
- Uses less processing during LAI

The simple way to achieve FIFO

You can use the *interflow-qpos* conditional in a **route-to** or **goto** command to achieve FIFO results.

For example, you can use the following **route-to** command with the conditional to achieve FIFO results:

route-to number 9581234 with cov n if interflow-qpos=1

If you have a lot of remote agents, you may want to set the **route-to** command as follows:

route-to number 9581234 with cov n if interflow-qpos<=2

Detailed information about the interflow-qpos conditional

You can use this feature without understanding the differences between split queues and eligible queues or between interflow-qpos and queue position. There are features that are built into enhanced LAI so that when you write a step such as route-to number 9581234 with cov n if interflow-qpos=1, the system operates smoothly under all conditions.

The interflow-qpos conditional

The **interflow-qpos** conditional only applies interflow processes to a dynamic eligible queue and to calls that are queued locally before the **route-to** is attempted.

The eligible queue is that portion of the split/skill queue that:

- Includes only calls that are not expected to be answered locally during the interflow process at that moment relative to the call being processed
- Does not include direct agent calls because these calls are excluded from any interflow process.

The following is an example of the **interflow-qpos** conditional used in a **route-to** command:

```
route-to number _____ with cov _ if interflow-qpos CM x
```

where

- **CM** is the comparator. It is one of three symbols: =, <, <=
  - With **if interflow-qpos = x**, the call is interflowed if it is at the **x** position from the top of the eligible queue.
  - With **if interflow-qpos < x**, the call is interflowed if it is among the top **x-1** of the eligible queue.
  - With **if interflow-qpos <= x**, the call is interflowed if it is among the top **x** eligible calls.
- **x** indicates the call's position in the eligible queue. Valid queue positions are 1 through 9. The top queue position is 1. The eligible queue is made up of calls from the first local split/skill that the call has been queued to due to previous steps in the vector.

Note:
Calls that are likely to be serviced locally before an LAI can be completed are not eligible for interflow since they are excluded from the eligible queue. Calls that are likely to be answered are identified based on conditions of the split/skill to which the call is queued and, under certain conditions, an administered minimum EWT threshold value.

The following is an example of the **interflow-qpos** conditional used in a **goto** command:

```
goto step/vector ____ if interflow-qpos CM x
```

where

- **CM** is the comparator. It is one of six symbols: =, <>, <, <=, >, >=
- **x** indicates the call's position in the eligible queue. Valid queue positions are 1 through 9. The top queue position is 1.

Calls that are likely to be serviced locally before an LAI can be completed are not eligible for interflow since they are excluded from the eligible queue.

When does a call not interflow?

A call does not interflow under the following circumstances:

- If the interflow-qpos conditional is not met.

  As with other conditionals, the route-to number... if interflow-qpos step or the goto step/vector branch is executed only if the conditional is met, otherwise vector processing goes to the next step.

- If the call is not in a split/skill queue or not in the eligible portion of the queue when the conditional step is executed.

  If the call is not in queue when the “route-to number... if interflow-qpos” step is executed, a vector event is logged and vector processing continues at the next step.

  If the call is not in queue when a “goto... if interflow-qpos” step is executed, the queue position of the call is considered to be infinite in determination of the conditional.

  Note:
  A vector event is not logged if the call is in queue, but is not in the eligible portion of the queue.

- Interflow failure or LAI rejection

  Interflow failure or LAI rejection will also go to the next step. Route-to operation and feature interactions will be the same as other configurations of the route to number command, for example, `route to number ___ with cov _ if digit CM x`.

The following table outlines what action is taken for different cases of interflow eligibility.

Table 30: Actions taken for cases of interflow eligibility

Case	Action at route-to step	Action at goto step
The call not eligible for interflow.	The call is never routed.	Treat as if the interflow queue position is infinite.
The call is not in any split queue.	The call is treated as if the interflow queue position is infinite.	Treat as if interflow queue position is infinite.
The call is eligible for interflow.	Act according to the conditional.	Act according to the conditional.

Setting the minimum EWT

The minimum expected wait time (EWT) threshold that is used to help determine which calls are more likely to be answered locally is administered on the Feature-Related System

Parameters screen. Minimum EWT is used when the local agents, that is, in the first split/ skill to which the call is queued, are handling a significant number of the calls. If these agents are not handling a significant number of calls, the call is eligible for LAI even if its EWT is lower than the threshold.

Note:

When enhanced LAI vectors or the look-ahead EWT threshold are administered inappropriately, remote agents may experience phantom calls or a delay between becoming available and receiving an ACD call.

The instructions below assume that you use a SAT terminal or terminal emulator to administer the switch.

To set the minimum EWT threshold:

1. In the command line, type `change system-parameters feature` and press `Enter`.

   The system displays the Feature-Related System Parameters screen.

2. Find the page of the Feature-Related System Parameters screen that has the **Interflow-Qpos EWT Threshold** field.

   If Look-Ahead Interflow is active, the **Interflow-Qpos EWT Threshold** field can be administrated.

3. In the **Interflow-Qpos EWT Threshold** field, enter the number of seconds, as a number from 0 to 9, that you want for the EWT threshold.

   The default of 2 seconds is recommended.

   Note:

   When the **look-ahead EWT threshold** field is set too low, remote agents may experience phantom calls.

4. Press `Enter` to save your changes.

Example of single-queue multi-site operation

In this scenario, all new calls for a given customer application are routed by the public network to only one of the switches in the network, where the calls are put in the queue.

Local agents service the calls from the queue in the normal fashion; however, remote agents service calls by means of enhanced look-ahead.

The switch with the call queue does rapid enhanced look-ahead attempts to all other switches in the network that can service this call type, looking for an available agent.

Normally, the look-ahead attempts are placed only on behalf of the call that is at the head of the queue (**`interflow-qpos = 1`**). However, in scenarios where there are large numbers of agents at a remote switch, it may be necessary to do interflows on behalf of more than one call in order to outflow a sufficient volume of calls to keep all agents busy (**`interflow-qpos <= 2`**).

Vector to back up split

```
1. announcement 3501
2. wait-time 0 secs hearing music
3. queue-to skill 1 pri m
4. route-to number 93031234567 with cov n if interflow-qpos = 1
5. route-to number 99089876543 with cov n if interflow-qpos = 1
6. wait-time 5 secs hearing music
7. goto step 4 if unconditionally
```

In this example, interflow call attempts are placed on behalf of the call that is at the beginning of the queue every 5 seconds to the two other switches in the network.

If queuing times are very long, 5 minutes, for example, and the call is not near the beginning of the queue, it is wasteful to go through the vector loop from step 4 to step 7 every 5 seconds. For this reason, the FIFO processing vector on page 276 is more efficient.

Example of maintaining FIFO processing with LAI

One of the advantages of enhanced LAI is the ability to provide FIFO or near-FIFO call processing. The following example shows a vector that is used to achieve such call processing.

FIFO processing vector

```
1. announcement 3501
2. wait-time 0 secs hearing music
3.  queue-to skill 1 pri m
4. goto step 7 if interflow-qpos < 9
5. wait-time 30 secs hearing music
6. goto step 5 if interflow-qpos >= 9
7. route-to number 93031234567 with cov n if interflow-qpos = 1
8. route-to number 99089876543 with cov n if interflow-qpos = 1
9. wait-time 5 secs hearing music
10. goto step 7 if unconditionally
```

In this vector:

- The rapid look-ahead loop is only entered when the call reaches one of the top 8 positions in queue.
- The number of executed vector steps is reduced dramatically when call waiting times are long.

It is important to write vectors so that calls at the head of the queue have advanced to the rapid look-ahead loop by the time their turn to interflow has been reached. In the vector example shown above, if 8 calls can be serviced from queue in less than 30 seconds (which is the loop time on step 5), there can be a delay in outflowing calls to available agents at the remote sites.

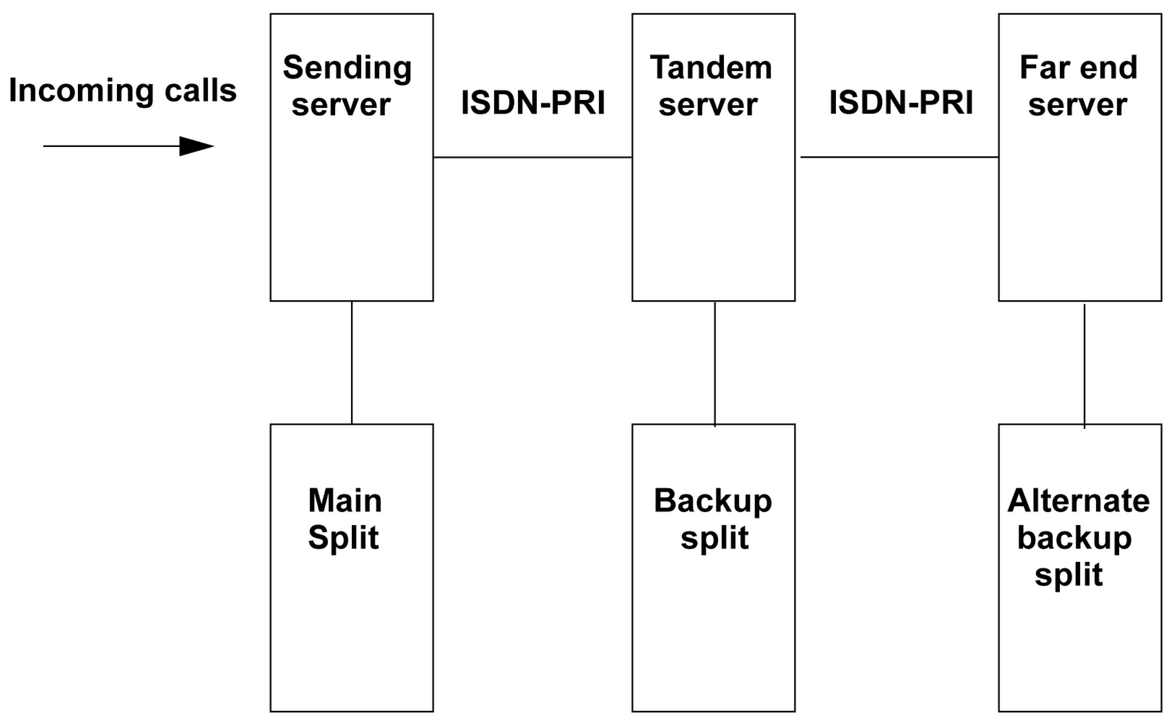
Single-queue FIFO considerations

The following issues need to be taken into consideration for FIFO in a single queue:

- When there are available agents, calls are always delivered to available agents at the queuing switch before available agents at the remote switches.
- When there are calls in the queue and agents serve calls from multiple applications, the agents always service calls from the applications that are queued locally before calls from applications that are queued at another switch.
- Backup VDNs and vectors are recommended in order to provide continuous operation in the event of a failure at a queuing switch.
- EWT predictions cannot be made if the split/skill in which the calls are queued has no working agents.
- EWT predictions may be temporarily inaccurate if there are sudden, major changes in the number of working agents in the split/skill in which the calls are queued.

Example of LAI in a tandem switch configuration

Tandem LAI is implemented by using `route-to` commands that contain external destinations that route over ISDN facilities. This configuration is shown in the following figure.



Sending switch operation

The sending switch is unaware that its LAI call is being tandemed to an alternate switch. The operation of the sending switch in the tandem switch configuration is the same as that in the two-switch configuration.

Tandem switch operation

If the receiving switch executes a **`route-to`** command that routes the call over an ISDN facility before call acceptance, the **`route-to`** command is performed on a look-ahead basis in the same manner as a sending switch. If the call is accepted at the far-end switch, acceptance is passed to the sending switch, and call control is passed to the far-end switch, along with tandeming of the original calling party information and the original DNIS name. If the call is denied, the next step of the tandem switch vector is executed.

The following example shows a tandem switch vector.

Tandem switch vector example

```
1. goto step 6 if expected-wait in split 30 pri h > 30
2. queue-to split 30 pri h
3. announcement 200
4. wait-time 2 seconds hearing silence
5. stop
6. route-to number 4000 with cov n if unconditionally
7. busy
```

Step 1 of this vector checks the inflow threshold. If the inflow criteria are acceptable, the vector flow drops to step 2, where the **`queue-to split`** command provides acceptance to the sending switch. Thereafter, steps 3 through 5 provide a typical queuing-wait scheme.

If the inflow criteria are not acceptable, a branch is made to step 6. The **`route-to`** command in this step checks another switch that is enabled with LAI on a look-ahead basis. If this far-end switch rejects the call, a denial message is relayed back to the tandem switch, which then drops the LAI call attempt. On the other hand, if the far-end switch accepts the call, an acceptance message is relayed all the way back to the sending switch.

No ringback is provided in this tandem switch vector. This is necessary so that an acceptance message is not returned to the sending switch. This operation is appropriate for the caller because the sending switch has already returned an announcement before a LAI attempt is made to the receiving switch.

Be sure that the sending switch is not used as a backup location for the tandem switch or for any of the far-end switches. If the sending switch is administered in this manner, all trunk facilities could be tied up by a single call.

Far-end switch operation

The far-end switch is also unaware that tandeming has taken place. The far-end switch functions in the same manner as the receiving switch within the two-switch configuration.

LAI-initiated path replacement for calls in vector processing

About path replacement for calls in vector processing

Path replacement for calls in queue and vector processing can be accomplished using QSIG or DCS with Reroute using ISDN SSE. For calls that are waiting in queue or in vector processing, even if the call is not connected to an answering user, path replacement can be attempted to find a more optimal path for this call. This results in more efficient use of the trunk facilities.

The `route-to` command is used in LAI to initiate a QSIG path replacement for a call. The following scenario can take place. At the terminating Communication Manager, if a Path Replacement Propose operation is received for a call that is in queue or vector processing, the switch can immediately initiate path replacement using the Path Replacement Extension if the**Path Replace While in Queue/Vectoring** field is set to`y` and the **Path Replacement Extension** field has a valid entry. These fields are located on the ISDN parameters page of the Feature-Related System Parameters screen.

The ability to track a measured ACD call after a path replacement has taken place is available for CMS versions r3v9ai.o or later. Starting with the r3v12ba.x release, CMS reports a path replacement as a**rename** operation rather than a path replacement. The**rename** operation properly reports scenarios where a path replacement takes place from a measured to an unmeasured trunk facility. Avaya recommends that you upgrade CMS to r3v12a.x or later and administer all trunks associated with path replacement as**measured** by CMS to ensure better CMS tracking of path-replaced calls.

Example LAI vector

The following example shows how an LAI vector can be written to trigger path-replacement at the terminating switch.

Note:

In order for a path-replacement to be attempted, the incoming and outgoing trunks that are used for the call must be administered with the **Supplementary Service Protocol** field set to `b`.

LAI-initiated path-replacement vector

```
1. wait 0 seconds hearing music
2. queue-to skill "
n"
 if available-agents < 6
3. route-to number "
ARS number for ISDN trunk"
 with cov n
4. wait 999 seconds hearing ringback
```

At the receiving Communication Manager, the vector that processes the incoming call must use an **`announcement`**, or **`wait hearing music`** vector command to enable path-replacement.

DNIS and VDN override in an LAI environment

About DNIS and VDN override

LAI handles Dialed Number Identification Service (DNIS) and VDN Override in various ways, depending on a number of different characteristics of the call. DNIS, as described in Call Vectoring fundamentals, allows any agent with a display-equipped telephone to receive visual displays that specify the name of the called VDN. VDN Override in its basic form allows the name of a subsequently routed to VDN to be displayed to the answering agent instead of the name of the originally called VDN.

The following sections discuss how LAI handles DNIS and VDN Override.

DNIS information displayed to answering agent

For LAI, the DNIS name, which is the called VDN name from the sending switch, is presented on the display for the answering agent on the receiving switch if all of the following are true:

- The LAI option is enabled.
- The call routes to a VDN.
- The DNIS name field is not blank.

The type of DNIS information that is displayed depends upon a number of different scenarios. This information is presented in the following table.

Table 31: DNIS information displayed for LAI scenarios

Scenario	Information displayed
Tandem LAI call	Look-Ahead Interflow DNIS information from the original LAI call.

Scenario	Information displayed
No redirection at the sending switch	VDN name according to Override rules at the sending switch (active VDN).
Redirection at the sending switch (VDN in coverage path)	Original VDN name, or If multiple VDNs are accessed, the name of the VDN that was last accessed by a **`route-to`** command.
Sending switch sends a blank DNIS Name field (that is, a name is not assigned to the sending switch called VDN) or the trunk group is administered to not send the LAI name (see Information Forwarding on page 225).	Name associated with the receiving VDN. This name can be changed according to the rules of VDN Override at the receiving switch.

Note:

VDNs that map to vectors that place LAI calls must have their ISDN Calling Party Number (CPN) prefixes administered. If an ISDN CPN prefix is not administered, the assigned VDN name is not sent. Instead, a DNIS of all blank space characters is sent and displayed on the answering agent's terminal.

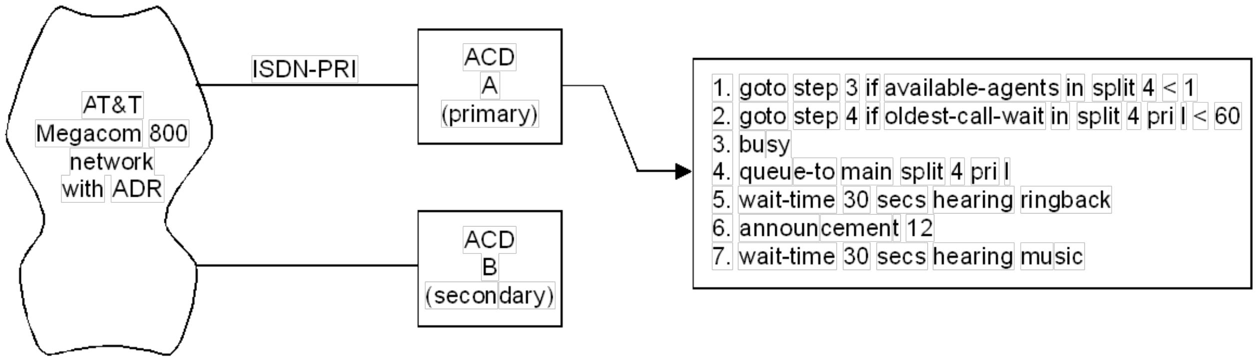
Originator's display

For internal calls, the originator's display contains the same information as for Basic Call Vectoring, but it is possible that the originator might receive unwanted display updates during LAI call attempts. In this case, LAI calls should go out over trunk groups that have the **Outgoing Display** field set to `n`. When the **display** field is set to `no`, internal callers who call that trunk group see the digits that they dialed on their display.

LAI with network ADR

Call Vectoring and LAI are compatible with and supplement the network services Alternate Destination Redirection (ADR) rerouting feature or equivalent service from other network providers. ADR uses ISDN-PRI connectivity with the switch in the same manner as LAI to allow the receiving system to indicate whether a call is to be accepted or rejected. The same type of vector that is used as a receiving ACD for LAI is used at the ADR-receiving ACD. If the call is accepted, it is connected to the system. If the call is rejected, the network routing number is translated to another number that routes the call to the alternate location within dialing-plan constraints. ADR allows for only one alternate location. LAI can be used at the alternate location to test other locations for less-busy conditions.

The following figure shows the configuration for a multilocation application.



The network requires ISDN-PRI connectivity to primary location A. Connection to secondary location B may or may not be ISDN-PRI. ADR attempts to route the call to location A over the ISDN-PRI link using a routing number that selects a VDN that is assigned to the receiving vector shown.

When the routing attempt is made, Call Vectoring starts processing the vector. The example then proceeds at location A as follows:

1. Step 1 checks for staffing of the ACD split, and branches to step 3 if it is not staffed.
2. If the ACD split is staffed, step 2 checks the oldest call waiting time in the split, and branches to step 4 if it is less than 60 seconds.
3. If the ACD split is unstaffed or if the oldest call waiting time is 60 seconds or more, step 3 rejects the call and returns a busy indication to the network.
4. If the oldest call waiting time is less than 60 seconds, step 4 accepts the call and queues it. ADR then connects the call through to the receiving system.
5. Steps 5 through 7 provide ringback, announcement, and music to the caller.

If the vector at location A rejects the call by sending a busy indication back to the network over the ISDN-PRI link, ADR reroutes the call to location B which must accept the call. If location B is closed or too busy to take the call, location B can use Call Vectoring and LAI to check other locations. If other locations exist and can take the call, location B can forward the call. If other locations do not exist or cannot take the call, location B can use Call Vectoring to route the call to location A. If location A is not open, location B can use Call Vectoring to provide an announcement or a busy tone to the caller.

Multi-site applications for Enhanced LAI

Enhanced LAI has two principal applications in a multi-site environment.

- It is possible to implement single-queue FIFO operation for any application. However, in many cases, Avaya recommends the use of BSR instead of LAI for maximum efficiency and flexibility. For more information, see Best Service Routing (BSR).
- LAI can be used in combination with BSR for those switches in the network with extremely low call volumes.

For more information about using BSR and LAI together, see Advanced multi-site routing on page 533

LAI considerations

The following are considerations for working with LAI:

- Never interflow to a remote vector that in turn might interflow back to the same local vector. This could cause a single call to use up all available trunks.
- Do not use the oldest-call-wait test condition in LAI vectors. OCW corresponds to the very next call to be answered and, as such, this test condition gives no information on the current state of call overload. For example, if OCW = 30 seconds, all we know from this is that the queue was overloaded 30 seconds ago. In place of oldest-call-wait, use the EWT conditional. For more information, see Expected Wait Time (EWT) on page 448.
- If an LAI call attempt is accepted by a step that contains a **`queue-to`**, **`check split`**, or **`route-to`** command, there is a small but finite interval during which the call could be answered by an agent at the sending switch before notification of acceptance is received by the sending switch. In this case, the caller is connected to the agent at the sending switch, while the agent at the receiving switch might receive a phantom call. For this reason, consider using a short wait-time or announcement step at the receiving switch to allow the call to be accepted and taken out of the queue at the sending switch. If call acceptance is to be based on available agents, use of a **`wait-time > 0`** seconds or an **`announcement`** is not recommended. A wait-time with 0 seconds of silence might be useful in this case.

  Note:

  For enhanced LAI operation, there are capabilities built into the feature to eliminate or reduce the occurrence of phantom calls. If phantom calls are a problem in an enhanced LAI operation, the **Interflow-Qpos EWT Threshold** field has been set too low.

- When an LAI call attempt is made, the TTR (if attached) is disconnected, and any dial-ahead digits are discarded. This implies that a subsequent **`collect digits`** command would require that the TTR be connected.
- Be sure that the feedback provided by the receiving switch after a successful LAI attempt is consistent with what the caller has already received.
- It is perfectly acceptable for a vector to route a call over an ISDN-PRI facility to a destination that is not a VDN. In this case, the sending switch treats the call as if it were a LAI call. Generic ISDN processing at the receiving switch causes the call to be accepted. The DNIS name is ignored.
- If a LAI call terminates to a VDN on a receiving switch where the LAI option is not enabled, intelligent interflow still results. However, any relevant DNIS information is ignored, and intelligent interflow to far-end switches is not possible.

- The LAI time-out in the sending switch occurs after 2 minutes.
- T-1 equipment might modify the ISDN D-channel that is used for LAI. If multiplexors are introduced into the ISDN-PRI circuit, bit compression and echo cancellation must be turned off for the D-channel.

Troubleshooting for LAI

The following are troubleshooting suggestions when working with LAI:

Proposed Solution 1

1. If remote agents are experiencing a high volume of phantom calls, the **Interflow-Qpos EWT Threshold** may be set too low or too high.
2. If remote agents are experiencing a delay between becoming available and receiving a call, the following may be the cause:
    a. The **Interflow-Qpos EWT Threshold** may be set too low.
    b. An insufficient number of LAI attempts have been made from the sending switch.
       In this case, change the interflow-qpos conditional at the sending switch. For example, change interflow-qpos=1 to interflow-qpos <= 2.
    c. An insufficient number of tie trunks are available.
3. If remote agents are receiving no calls, the maximum number of vector steps that are executed at the sending switch vector may have been reached before calls reached the head of the queue.
   In this case, rewrite the vector on the sending switch.

Maximum Agent Occupancy (MAO)

Maximum Agent Occupancy (MAO) overview

The Least Occupied Agent (LOA) and Most Idle Agent (MIA) methods attempt to maintain equitable agent occupancy rates based on time spent in call service. In contrast, SLM operations are driven solely by the needs of a skill in terms of meeting a specified target service level, and overall occupancy rates for individual agents are not a factor in the agent selection process. Instead, a Maximum Agent Occupancy (MAO) threshold can be used to achieve equitable agent occupancies and avoid agent burnout issues.

Note:

MAO can be used even when SLM is not active on the system, but MAO *must* be used with an EAS system or an EAS system using Business Advocate.

The MAO threshold is a system-administered option with a system-assigned maximum occupancy percentage value that is applied across all administered agents and is based on the total percentage of agent time in servicing calls. MAO data is derived from the same calculations that are used to derive the Least Occupied Agent (LOA).

When an agent who exceeds the specified MAO threshold attempts to become available, he or she is automatically placed in the AUX work mode for the reason code administered for this purpose. When the occupancy for such *pending* agents drops below the MAO, they are released from AUX work mode and made available.

When to use MAO

MAO is designed to provide short work breaks for agents who have high occupancy rates and is recommended only for call centers that use SLM, Avaya Business Advocate, or otherwise have some agents with high occupancy rates. High occupancy agents tend to be those agents with the highest skill level, or single-skill SLM agents.

MAO is not intended for call centers whose agents are administered in a highly similar manner. The MAO threshold should be set to a value that is sufficiently high to avoid situations where large numbers of agents are simultaneously put into the Auxiliary work mode.

Screens and fields used to administer MAO

This section lists the administration screens and settings that are required for MAO administration.

Table 32: MAO administration for setting Maximum Agent Occupancy

Administration command:	`change system-parameters features`	
Page name:	`Hunt Group`	
Required field(s):	Group Type:	EAS skill
	Maximum Agent Occupancy Percentage	___ [17]
	Maximum Agent Occupancy Aux Reason Code	___ [18]

[17] The default value for this field is set to 100%.

[18] Aux work reason code 9 is set as the default. A different reason code can be used for this purpose, but Avaya recommends that you do *not* use reason code 0.

Determining when an agent is pending availability due to MAO

When an agent is put in to the Aux Work mode while pending availability because the agent's occupancy is above the system assigned limit, the agent's Aux Work button is lit indicating the agent is in Aux Work. This Aux Work condition is reported to the CMS adjunct reporting system using the system assigned reason code for MAO. If the agent has more than one Aux Work button assigned on their station set because the Aux buttons are associated with specific reason codes, the button that has the MAO reason code assigned will be lit. If none of the buttons have that reason code assigned, the Aux Work button that doesn't have an assigned reason code will be lit.

Manual-in mode

For agents in manual-in mode, when the agent exceeds the maximum-administered occupancy threshold, the agent is first put into the after-call work mode after the current call drops.

The agent is then put into the auxiliary work mode for the administered MAO reason code if the agent attempts to become available again by pressing the Manual-In button or dialing the FAC while occupancy is still above the maximum. When the occupancy for the agent drops below the administered maximum, the agent is put back into manual-in mode.

Auto-in mode

For agents in auto-in mode, when a call drops, the Communication Manager automatically attempts to make the agent available again. If the occupancy for this agent has exceeded the maximum-administered level, the agent is put into auxiliary work mode for the MAO reason code instead of auto-in until the agent occupancy level has dropped below the administered maximum.

Manual override of MAO aux mode

If an agent wants to manually override the AUX code and leave the pending available state to make or receive an ACD call, they must do either of the following:

- Press the **auto-in**/**manual-in** button twice
- Enter the FAC code twice

Default AUX work reason code for MAO pending state

On the System-Parameters Feature screen, the default value in the **Maximum Agent Occupancy AUX Reason Code** field is set to 9. If reason code 9 is already being used to track other AUX time activities, agent time spent in the MAO pending state will be combined with time spent in those other activities. For this reason, you should designate an AUX work reason code solely for time spent in the MAO pending state, if it is possible to do so. For information about MAO administration, see Screens and fields used to administer MAO on page 285.

Important:

Avaya recommends that you do not use reason code 0 to track MAO Aux time.

Evaluating MAO using CMS reports

Avaya CMS includes database items that you can use to verify that MAO is functioning properly. In CMS Supervisor R12 or later, the historical Agent Summary report includes two fields that can be used to evaluate MAO performance. Depending on how you treat after call work in agent occupancy calculations, inspect one of the following fields:

- % Agent Occup w/ACW
- % Agent Occup w/o ACW

If the agent occupancy percentage is less then or equal to the specified MAO percentage, then this is one possible indication that MAO is functioning properly. You can also review agent AUX Work time for the reason code that you assigned for time spent in the MAO pending AUX Work state.

It is possible for an agent to have an occupancy that exceeds the MAO threshold if a recent call caused the agent to exceed the MAO threshold and insufficient time has elapsed for the agent occupancy to adjust.

MAO feature interactions

- Skills that are administered as auto-available will ignore the maximum occupancy system parameter. In other words, MAO does not apply.
- If the messaging system and the VRU skill are not assigned as Auto-Available Skills (AAS), the ports enter the pending availability Aux Work mode.
- An agent pending availability because of Maximum Occupancy is put at the bottom of the idle queue when he becomes available, if the hunt group is MIA.

- If the occupancy of an activated reserve agent is above the maximum, his availability is pending until his occupancy drops below the maximum with Avaya Business Advocate.
- When an agent that has been pending availability because of maximum occupancy becomes available, he is auto-reserved if he meets the auto-reserve criteria.
- Maximum Occupancy uses the same options and implementation as Least Occupied Agent (LOA) for determining an agent's occupancy. The occupancy includes the agent's time with ACD calls ringing, calls active, calls on hold at their voice terminal, and logged in After Call Work if the system option or specific agent login ID option considers ACW as agent work time. In other words, the option is set to y.
- (ACW) if the system-wide decision is made to consider ACW as agent work time.
- Agents that have been pending availability because of MAO are placed into the idle queue based on his occupancy, if the hunt group is LOA.
- When an agent becomes available or leaves ACW from a direct agent call, the agent's availability is pending if his occupancy is above the maximum.
- An agent does not receive new Multiple Call Handling calls if his occupancy is above the system administered maximum. An agent's availability is pending when he has dropped all active ACD calls.
- If an agent manually enters AUX with the maximum occupancy reason code, he is not treated as an agent with pending availability. The agent is treated as an agent in AUX Work and needs to manually change work mode.
- After the Timed After Call Work timer has expired, an agent's availability is pending if his occupancy is above the maximum. While availability is pending, the agent is placed in AUX Work instead of Auto-In until the agent occupancy drops below the maximum.
- An agent whose occupancy is above the system maximum is forced to enter Forced Stroke Counts or Forced Call Work Codes before pending availability. After the Stroke Count or CWC has been entered, if the agent attempts to go to an available mode, he will be put into AUX Work mode for MAO if his occupancy has exceeded the maximum.

Multiple Call Handling

MCH applications

Use Multiple Call Handling (MCH) in applications where you want agents to take additional calls without dropping the active call. Examples of applications include:

- An agent and a caller may need to wait on a call for information. MCH allows the agent to put the call on hold and handle other ACD calls until information becomes available.
- ACD calls may be more important to your business than non-ACD calls. Use MCH to interrupt agents who may already be on non-ACD calls with an potentially more highly valued ACD call.
- In an EAS environment, calls from one skill, such as Sales, may considered to be more important than calls from another skill such as Service. MCH can be used to interrupt an agent who has a call from the less-important skill with a call from the more-important skill.

You can use MCH in an Expert Agent Selection (EAS) or non-EAS environment.

- With EAS, you can administer any combination of MCH and non-MCH skills for an agent. If an EAS agent is a member of both MCH and non-MCH skills, they can handle multiple simultaneous ACD or Direct Agent calls only in their MCH skills.
- Without EAS, agents can be logged into only one split if it is an MCH split. Similarly, an agent logged in to a non-MCH split cannot log into an MCH split.

MCH settings

On request

In on-request splits/skills, the following is true:

- If an agent goes into auto-in or manual-in work mode, but there are no calls in the queue, the agent is placed at the bottom of the MIA queue or at the bottom of their skill level in the EAD queue, or is made available in the DDC queue.
- Agents must select auto-in or manual-in work mode for each new ACD call they take while a call is on hold.
- The agent can take additional ACD calls as long as there is an available line appearance.

Use on-request MCH in conjunction with a feature such as VuStats, which agents can use to see when the queue is getting full and take additional calls.

One forced

An agent who is idle or active on a non-ACD call is automatically interrupted with an ACD call from this split or skill when no other ACD call for any of the agent's splits/skills are alerting, active, or held. In addition, the following must also be true:

- The agent is in manual-in or auto-in work mode.
- The agent is the most idle or next available.
- An unrestricted line appearance is available.
- AUX work or Move from CMS are not pending.

As long as an ACD call is active or held, the agent does not automatically receive an additional call from the one-forced split or skill. An agent in a one-forced split or skill in auto-in or manual-in work mode is unavailable for that split or skill from the time that an ACD call rings until all ACD calls are abandoned, redirected, or dropped. However, the agent can request another ACD call from a one-forced split or skill by placing the active call on hold and selecting Manual-In or auto-in work mode.

If an agent with multiple skills is active on an ACD call for a group with one-forced MCH, the agent could be forced to take an ACD call for one of his or her other skills, depending on that skill's MCH settings.

Because one-forced MCH forces an ACD call to alert an agent who is not on an ACD call, use it when you want ACD calls to take precedence over other calls.

One per skill

You must have EAS to use one-per-skill MCH. An agent with no ACD calls for this skill is automatically interrupted with a single ACD call from this skill under the same conditions listed for one-forced.

If a one-per-skill call is active or held, the agent does not automatically receive additional calls from that skill. However, the agent can request another ACD call from a one-per-skill in the usual way.

If an agent with multiple skills is active on an ACD call for a one-per-skill group, the agent could be forced an ACD call for one of his or her other skills if those skills are many-forced or one-per-skill MCH.

Use one-per-skill MCH when calls from one skill are higher priority than other ACD calls.

Many forced

Agents are automatically interrupted with an ACD call under the same conditions listed for one-forced. As soon as an agent answers an alerting ACD call, the agent immediately becomes available to receive another ACD call from a many-forced split or skill.

Agents in many-forced groups in auto-in or manual-in work mode are unavailable only when an ACD call is ringing.

Use many-forced MCH when agents must answer important or urgent calls, even when they must put equally important calls on hold. It can also be used to force direct agent calls to an agent.

MCH example

In this example, an agent is logged into four skills, each with a different MCH option. The following table shows how calls are delivered when an unrestricted-line appearance is available and the agent is in auto-in or manual-in work mode (AUX work mode is not pending).

	Calls Delivered?			
Condition	Skill 1 (MCH=one-request)	Skill 2 (MCH=one-forced)	Skill 3 (MCH=one-per-skill)	Skill 4 (MCH=many-forced)
No calls on set	yes	yes	yes	yes
One active extn call	no	yes	yes	yes
Skill 1 call active	no	yes	yes	yes
Skill 2 or 4 call active	no	no	yes	yes
Skill 3 call active	no	no	no	yes
Extn call held, no other action	no	yes	yes	yes
Skill 1, 2, or 4 call held, no other action	no	no	yes	yes
Skill 3 call held, no other action	no	no	no	yes
Extn call held, then AI/MI selected	yes	yes	yes	yes
Skill 1,2,3, or 4 call held, then AI/MI selected	yes	yes	yes	yes

Agents and supervisors in on-request MCH splits/skills can use Queue Status, VuStats, and BCMS/CMS reports to determine if a waiting call must be answered immediately.

MCH considerations

- Agents can receive multiple calls only when in auto-in or manual-in work mode. All forced MCH calls are delivered with ringing at the agent's station, not with zip tone. Requested MCH calls are delivered with ringing or zip tone.
- Agents can toggle between auto-in and manual-in work mode.
- If an agent selects ACW or AUX work mode with calls on hold, the work mode is pending until all calls complete or until an manual-in call completes. New ACD calls are not delivered when AUX work is pending. When an ACD or direct agent call with pending ACW completes, the agent enters ACW. When an agent is active on a non-ACD call with ACW pending, the agent can receive forced MCH calls.
- If an agent is either in auto-in work mode and active on an ACD or direct agent call, or in auto-in or manual-in work mode and active on a non-ACD call and a Manual-In ACD or direct agent call abandons from hold, the agent is pending for ACW work mode and the after-call button lamp flashes.
- If an agent reconnects to an ACD or direct agent call on hold, his or her work mode changes to the call's work mode (auto-in or manual-in).
- Do not use forced MCH with DDC distribution because the first agent continues to receive calls until all line appearances are busy.

MCH interactions

Interaction	Description
Automatic Hold	To answer a ringing ACD call, an agent in a many-forced, one-forced, or one-per-skill split or split or skill pushes the line-appearance button. If automatic hold is administered, the active call is automatically placed on hold. Otherwise, the agent must first push hold.
Call Work Codes and Stroke Counts	Agents who handle multiple ACD calls simultaneously with MCH can enter CWCs and Stroke Counts. When an agent does so with multiple calls on the station, the code/count is associated with the last call the agent handled. If an agent enters a code/count during an active call with calls on hold, the code/count is associated with the active call. If an agent with on-request MCH is active on a call that requires forced entry of CWC or stroke counts and places the call on hold without entering a code/count, he or she cannot request another call. If agents with many-forced MCH are in a split or skill with forced entry of CWC or stroke counts, they are forced to handle an ACD call even if they have not entered a code/count.

Interaction	Description
Direct Agent Calling	Agents can handle multiple direct agent calls if their direct agent skills have MCH. The queue-status indicator is not lit when a direct agent call queues to a split or skill. Agents are notified that calls are waiting with a ring ping and a flashing current-work-mode lamp.
Forced Agent Logout from ACW	An agent in ACW is logged out because the Forced Agent Logout from ACW timer has expired, even if the agent has ACD calls on hold.
Move Agent While Staffed	An agent with a move pending can place a call on hold and request another ACD call. All calls and ACW must complete before the pending move occurs.
Non-ACD calls	If an agent activates auto-in or manual-in work mode with calls on hold, he or she can answer or originate a non-ACD call. With on-request MCH, the agent is temporarily unavailable for ACD or direct agent calls. With forced MCH, a call can be delivered. If an agent in ACW reconnects to an AUXIN/AUXOUT call, the agent remains in ACW.
Queueing	When an agent is available, the agent is placed at the end of the queue for Uniform Call Distribution (UCD) hunt groups or at the bottom of the skill type for Expert Agent Distribution (EAD) hunt groups, or is made available for Direct Department Calling (DDC) hunt groups. When the agent becomes the most available according to group type (UCD, EAD, or DDC), he or she receives a queued ACD or direct agent call. If the last agent on a forced MCH split or skill is pending for AUX work mode in a non vector-controlled split, the agent must empty the queue before going to AUX work mode. The agent continues receiving ACD calls until the queue is emptied.
Redirection on No Answer	If an agent has a call active or on hold and the RONA timer expires for another ringing ACD call, RONA redirects the alerting call back to the split or skill or administered VDN. The agent is not taken out of service when the call redirects, but is placed at the bottom of the Most Idle Agent (MIA) or Expert Agent Distribution (EAD) queue.
Restricted line appearance	If you administer last-available line appearance as Restricted Last Appearance for an agent’s telephone, the agent does not receive additional ACD calls because the appearance is reserved for making conference or transfer calls.

Network Call Redirection (NCR)

Network Call Redirection (NCR) provides an Avaya Communication Manager call routing method between sites on a public network or a Virtual Private Network (VPN) that can reduce trunking costs. These cost reductions are particularly valuable in enterprises or multi-site call center environments where trunk costs are high.

When an incoming call arrives at an Avaya Communication Manager (CM) that has the NCR feature enabled, call redirection is managed by the Public Switched Telephone Network (PSTN) or VPN switch instead of the local Avaya Server that is driving the CM. As a result, trunks that the server would otherwise retain to accomplish a necessary trunk-to-trunk transfer are released after the call redirection takes place.

The cost reductions associated with reduced trunk use can be significant particularly when Avaya virtual routing features, such as Best Service Routing (BSR) with Expected Wait Time (EWT), are implemented. The cost-savings are achieved by the Avaya customer requiring fewer trunks to handle the same number of incoming/outgoing calls after the NCR feature is implemented within the local Communication Manager.

Network Call Redirection

About NCR

Network Call Redirection (NCR) provides an Avaya Communication Manager call routing method between sites on a public network or a Virtual Private Network (VPN) that can reduce trunking costs. These cost reductions are particularly valuable in enterprises or multi-site call center environments where trunk costs are high.

When an incoming call arrives at an Avaya Communication Manager (CM) that has the NCR feature enabled, call redirection is managed by the Public Switched Telephone Network (PSTN) or VPN switch instead of the local Avaya Server that is driving the CM. As a result, trunks that the server would otherwise retain to accomplish a necessary trunk-to-trunk transfer are released after the call redirection takes place.

The cost reductions associated with reduced trunk use can be significant particularly when Avaya virtual routing features, such as Best Service Routing (BSR) with Expected Wait Time (EWT), are implemented. The cost-savings are achieved by the Avaya customer requiring fewer trunks to handle the same number of incoming/outgoing calls after the NCR feature is implemented within the local Communication Manager.

NCR options supported by PSTNs

This section describes the various NCR redirection options that are supported by Public Switched Telephone Networks (PSTNs). All of the NCR protocols described in this section support Information Forwarding using UUI transport to the redirected-to location over the PSTN or VPN network.

Protocols not supported by NCR

The PSTN call-redirection protocols that are currently available but are *not* supported by the NCR feature include the following:

- AT&T 4ESS Out-of-Band Transfer and Connect
- Nortel RLT

Network Call Transfer-type options

About NCT-type feature operations

A key advantage of NCT-type protocols invoked by call vectoring or by manual station call-transfer or call-conference/release operations is that the redirecting server retains control over the call and can continue to use a trunk-to-trunk connection if the PSTN switch does not accept the request to merge B-channels for both legs of the call. If the PSTN switch returns a `failure` message to the PSTN, the originating server maintains a trunk-to-trunk connection for the call. For vector call processing, NCR call processing still considers the NCR attempt to be successful, but the following outcomes occur:

- A vector event is logged to indicate that the NCT operation as attempted with the PSTN failed.
- If Avaya CMS is used to track incoming calls to an externally measured VDN, the call is not counted as deflected.

For NCR invocation by call vectoring, the local Avaya Communication Manager sets up the second leg of the call, waits for the second site to be connected, and then requests the PSTN or VPN switch to merge the first leg of the call with the second leg. If this request is accepted, the PSTN or VPN switch joins the original caller to the redirected-to endpoint, sends a `success` message back to the redirecting server and then drops both legs of the call at the redirecting server.

For NCR NCT or TBCT invocation by a station, ACD agent, VRU, or CTI-controlled call doing a call-transfer or call-conference/release operation, if the second leg of the call is set up over an outgoing trunk in the same signaling group as the incoming call, then call-redirection takes place when the call-transfer or call-conference/release occurs. For the NCR ISDN ETSI ECT protocol, the call redirection will take place when the outgoing trunk B-channel either has the same or a different D-channel than the incoming call.

Specific NCT-type protocols

Specific NCT-type protocols include the following protocols:

MCI Network Call Transfer over ISDN PRI

Network Call Redirection and PSTN switch operations associated with the MCI NCT protocol are consistent with those described in About NCT-type feature operations on page 295.

MCI Network Call Redirection/Network Call Transfer is compliant with ANSI Explicit Network Call Transfer (ENCT) T1.643 (1995), the MCI Nortel variant of ANSI ECT (1995).

Note:
MCI NCT is offered in the United States by MCI for their Nortel DMS-250 and Alcatel DEX-600 PSTN switches.

Two B-Channel Transfer (TBCT) over ISDN PRI

Network Call Redirection and PSTN switch operations associated with the TBCT protocol are consistent with those described in About NCT-type feature operations on page 295.

The Network Call Redirection/Telcordia Two B-Channel Transfer (TBCT) protocol is compliant with the Telcordia Two B-Channel Transfer and ANSI Explicit Call Transfer (1998) standards. For more information, see any of the following:

- Telcordia GR-2865-CORE
- ANSI T1.643 (1998)
- Lucent 99-5E-7268

Note:
TBCT is offered in the U.S. by AT&T for their DMS-100 PSTN switches configured with the NI2 network protocol. TBCT is offered in Canada by Bell/Canada for their DMS-100 PSTN switches; and by AT&T/Canada, for their DMS-500 PSTN switches.

ETSI Explicit Call Transfer

Network Call Redirection and PSTN switch operations associated with the European Telecommunications Standard Institute (ETSI) Explicit Call Transfer (ECT) protocol are consistent with those described in About NCT-type feature operations on page 295.

The Network Call Redirection/ETSI Explicit Call Transfer protocol is compliant with ETSI standard EN 300 369-1.

Note:
ETSI ECT is offered in Europe by France Telecom and other in-country PSTN service providers for their Ericsson AXE-10 PSTN switches. ETSI ECT is offered in the United Kingdom by MCI for their DMS-100 PSTN switches.

Selection of outbound call leg for NCT-type NCR protocols

For the MCI NCT and Telcordia TBCT NCR protocols, the PSTN switch requires that the outbound call leg of a redirected PRI call is in the same trunk group and has the same Direct Access Line (DAL) D-channel as the inbound call. For vector-initiated invocation of the NCR feature by either a BSR queue-to best or route-to number vector step, the Avaya Communication Manager enforces this requirement by automatically selecting an outbound B-channel that has the same signaling group as the incoming call's D-channel. This results in sending the NCR invocation request on the same D-channel used for the first call leg's associated signaling or for the same associated D-channel when the Non-Facility Associated Signaling (NFAS) D-channel backup configuration is used.

For the ETSI ECT NCR protocol, there is no restriction that the outbound PRI call leg must have the same Direct Access Line (DAL) D-channel used for the first call leg's associated signaling. If the PRI trunk group has more than one associated D-channel, NCR processing sets up the second call leg for call redirection using any B-channel in the trunk group independent of its associated D-channel.

Network Call Deflection (NCD)

The Network Call Deflection (NCD) operation by a PSTN switch can occur only if the incoming call to the Avaya Communication Manager is not answered (that is, an ISDN CONNECT message is not sent to the PSTN switch from the incoming server).

The NCR NCD feature is compliant with ETSI Supplementary Services Network Call Deflection ETS 300 207-1 (partial call rerouting in the public network).

Important:

Some call vectoring commands cause CONNECT messages to be sent to the PSTN switch. If call vectoring methods are used to implement NCR and the PSTN switch supports the NCD protocol, call vectors used to invoke NCR must not include any of the following vector commands:

`announcement`

`collect x digits`

`converse-on split/skill`

`wait hearing music`

`wait hearing (announcement extension) then (continue, music, ringback or silence)`

When the Avaya server invokes the NCD feature, the PSTN switch sets up the second leg of the call instead of the redirecting Avaya Communication Manager. There are two PSTN options for NCD specified by the ETSI standards: *retain call until alerting/connect* and *clear call upon* *invocation*. The *clear call upon invocation* option is commonly referred to as a *partial call reroute*.

When the *clear call on invocation* option is used, a successful NCR/NCD attempt is indicated when the PSTN or VPN switch has validated the NCR request and sends a *call reroute return* DISCONNECT message to the originating server. In this case, the server loses control of the call after it is transferred to the PSTN or VPN redirection endpoint, and no alternate transfer method is possible if the PSTN or VPN switch fails to transfer the call to the second location.

The *retain call until alerting/connect* option is not available because there are presently no known PTSN or VPN offers with this protocol. With this option, the PSTN or VPN switch sets up the second leg of the call, waits until an ALERTING message is received, and then sends a *call reroute return* FACILITY message followed by a DISCONNECT message to the originating server. In this case, if the second leg of the call fails, the server can redirect the call with a trunk-to-trunk connection so that the call is not lost.

NCD is offered in Europe by British Telecom for their Marconi/Plessey System X and Ericsson AXE10 PSTN switches; and by Deutsche Telecom for their Siemens EWSD and Alcatel S12 PSTN switches. NCD is offered in Australia by Telstra for their Alcatel S12 PSTN switches.

AT&T In-Band Transfer and Connect

This section describes PSTN redirection operations associated with the AT&T In-Band Transfer and Connect service. Details of the service are described in AT&T Technical Reference 50075.

NCR provides Information Forwarding support for the AT&T In-Band Transfer and Connect network service ISDN D-channel data-forwarding capability. The Information Forwarding feature forwards the UUI that is associated with the call to the redirected-to location. When call vectoring and AT&T In-Band Transfer and Connect are used to transfer a call, and NCR is enabled for the system, the disconnect vector step causes UUI IE information to be inserted into the ISDN DISCONNECT message generated by a successful AT&T In-Band Transfer and Connect operation.

Note:

For information about NCR administration and other administration measures that are required when the AT&T In-Band Transfer and Connect service is used, see Administering NCR with AT&T In-Band Transfer and Connect in *Administering Avaya Aura™ Call Center Features*.

AT&T In-Band Transfer and Connect operations can be initiated by call vectoring after first doing the following switch administration:

1. Administering a route-to number vector step with an announcement extension, where the associated announcement is recorded with Dial Tone Multi-Frequency (DTMF) tones that include a *T followed by a PSTN endpoint number.
2. Administering a BSR location**VDN Interflow** field on the Best Service Routing Application Plan screen as an announcement extension, where the associated

announcement is recorded with DTMF tones that include a *T followed by a PSTN endpoint number.

3. Administering a BSR location**VDN Interflow** field on the Best Service Routing Application Plan screen as a local switch VDN number associated with a vector that contains anannouncement step, where the associated announcement is recorded with DTMF tones that include a *T followed by a PSTN endpoint number.

When the route-to number vector in action 1 is executed, or when a queue-to best vector step is executed and the BSR location described in action 2 or action 3 above is selected as the BSR best location for call interflow, the AT&T In-Band Transfer and Connect operation succeeds, but no UUI IE information is sent to the redirected-to PSTN endpoint by the Avaya Communication Manager. However, for Step 3 on page 0 above, NCR can be administered for use with the AT&T In-Band Transfer and Connect feature and a disconnect hearing announcement none vector added after the announcement step such that UUI information associated with the call is passed to the routed-to endpoint when the call redirection is completed. This UUI information can be used to do agent screen pop-ups at the redirected-to PSTN endpoint where the call is interflowed.

BSR call-flow resulting in AT&T In-Band Transfer and Connect UUI IE

A typical BSR call-flow that results in UUI IE information being inserted in the ISDN DISCONNECT message during a successful AT&T In-Band Transfer and Connect operation is as follows:

1. A PRI call from the PSTN switch arrives at the local Avaya Communication Manager and is routed to a VDN that uses a vector to do subsequent BSR vector processing.
2. The BSR polling vector steps on the local server receive status information from various local skills and remote BSR locations, and identifies a remote call center site as the BSR best location.
3. Call control passes to the interflow VDN selected as the BSR best location specified on the Best Service Routing Application Plan screen.

   For information specific administering a BSR application plan, see Call vectoring methods used with AT&T In-Band Transfer and Connect service in *Administering Avaya Aura™ Call Center Features*, or for general information about BSR application plans, see Selecting or administering application plans on page 122.

   Important:
   The **Net Redir?** field in the BSR application plan for the remote location must be set to `n`.

4. The vector associated with the interflow VDN for the BSR best location includes the following:

   - An announcement vector step that specifies an extension for which a special sequence of DTMF digits has been recorded. The recorded DTMF digits return in-band information about the redirected-to endpoint back to the PSTN. The

DTMF digits provided in the announcement are entered from a Touch-tone keypad, and use the format:

*T + PSTN number

T corresponds to the number 8 button on a DTMF keypad, and PSTN number represents the PSTN endpoint number where the call is redirected.

Note:
The phone equipment required to create the announcement is described in Methods for setting up DTMF announcements for AT&T In-Band Transfer and Connect in *Administering Avaya Aura™ Call Center Features*.

- A wait-hearing silence step provides a brief interval to allow sufficient time for the PSTN switch to process the DTMF digits.
- A disconnect after announcement none vector step. This vector step sends an ISDN DISCONNECT message that includes a UUI Information Element. The UUI IE contains Avaya Information Forwarding for the call that is sent to the PSTN switch.

5. The PSTN switch makes the connection to the specified redirected-to endpoint and releases the B-channel connection to the Avaya Communication Manager.

NCR considerations

Limitations on call redirection

You should understand the following items that pertain to limitations on the NCR feature:

Limitation	Description
NCR feature support	PSTN support for NCR varies with geographical location and may be limited or absent in some areas. Consult your Avaya account team to determine PSTN Service Provider availability of one of the NCR protocols in your area.
NCD redirection protocol support	At this time, no PSTNs offer the Network Call Deflection *retain call until alerting/connect* operation. Therefore, only the Network Call Deflection *clear call upon invocation* offer is available from PSTNs. Both methods are described in this document. It is advised that you negotiate with your PSTN as the NCR feature will work on either platform. NCR is limited by which PSTN platform is available to you.
Allowable number of redirection per call	There may be limits placed on the number of times a call may be redirected over the public network. These limits are imposed by the public network service provider. For example, in the United States, MCI currently allows only one redirection per call. In the United Kingdom, there is a limit of 20

Limitation	Description
	call deflections per call. In addition, there may be additional charges associated with redirected calls.
User-to-User information forwarding support	Some public network service providers do not support forwarding of User-to-User Information (UUI), including Adjunct Switch Application Interface (ASAI) user data, collected digits, VDN name, the VDN in-time (as reflected by the NETINTIME database items), and the UCID. In such situations, Information Forwarding will be lost and the second leg of the redirected call will look like an entirely new call to the redirected-to server at the second location. One of the data items lost is the VDN name, which is rerouted to the originally called service (DNIS) information. The indication that the call has been forwarded can be achieved by using dedicated VDNs for call forwarding, but this strategy loses the benefits of Information Forwarding inherent with NCR and limits use of CTI applications. PSTN service providers typically charge by call or by a monthly rate for the redirect and UUI transport services. For more information about such charges, contact your Avaya account team.

NCR operational considerations

Reserving outbound trunk B-channels to ensure NCR operations succeed

When the trunk group service type is set to call-by-call, the trunk group Usage Allocation capability can be used to reserve a minimum number of trunk channels for outgoing PRI B-channel calls within the same trunk group and same D-channel.

For more information, see Reserving trunk group B-channels for NCT-type redirection operations in *Administering Avaya Aura™ Call Center Features*.

Call vectoring configuration required for successful MCI NCT operations

When NCR is used with the MCI NCT protocol, the VDN call vector the call is redirected to by a successful MCI NCT operation must immediately return an ISDN CONNECT message to the PSTN switch. To meet this requirement, either a “wait 0 secs hearing music” or an “announcement” vector step should be the first step executed in the redirected-to call vector.

Ericsson AXE-10 configuration required for successful ETSI ECT operations

Following is Ericsson AXE-10 release and configuration information required for successful NCR ETSI ECT operations:

- Verify that AXE-10 has *VN7 Translocal 4.2* or later software. This is also called *GOAS 2.1* by Ericsson.
- Configure the AXE-10 for the *pure ETSI* level.

- Configure all PRI trunks used with the Communication Manager 2.0 NCR/ETSI ECT feature to subscribe to the AXE-10 *ETSI ECT* mode. On the AXE-10 trunk configuration screen, configure the ECT category to *ON*.
- Do *not* configure the AXE-10 PRI trunk to expect a *HOLD* ISDN message to be sent by the NCR ETSI ECT feature as part of the ETSI ECT invocation sequence.

NCR and Information Forwarding

The Avaya Information Forwarding feature is supported with NCR when the PSTN supports ISDN UUI IE transport in conjunction with the specific network redirection protocol used by the switch.

UUI data included in Information Forwarding for ISDN calls

Information Forwarding forwards the following call center-related data (as User-to-User Information) with an ISDN call:

- Adjunct Switch Application Interface (ASAI) user data
- Universal Call ID (UCID)
- Collected digits
- In-VDN time
- VDN name.

UUI data forwarding

When an NCT-type option is used for NCR, the UUI is forwarded by the Avaya Communication Manager in the ISDN SETUP message sent with the call to the second site.

When the NCD option is used for NCR, the UUI is included in the ISDN FACILITY invoke message sent from the Avaya Communication Manager to the PSTN. The PSTN then forwards the UUI to the second site.

When the AT&T In-Band Transfer and Connect service is used for NCR, the UUI is returned in an ISDN DISCONNECT message that includes the data in a codeset 0 or 7 UUI IE element.

PSTN terms used for UUI transport service

For NCT-type options and the NCD option, the PSTN service provider must configure the PRI trunks used with the Avaya NCR feature to transport the UUI data associated with the Avaya Information Forwarding feature. The various PSTN terms used in different countries for UUI IE transport are listed in the following table.

Country	UUI Transport Term	Providers
Australia	UUS Service 1	Telstra
Canada	UUS Service 1	AT&T/Canada Bell Canada
France	Mini-Message	France Telecom
Germany	(included in basic ISDN package)	Deutsche Telecom

Country	UUI Transport Term	Providers
Singapore	Not supported	
UK	Not supported	
USA	N-Quest Type 1 Service MA UUI Type 1 Service	MCI AT&T

NCR feature interactions

NCR interacts with the following call center features:

- Attendant Vectoring — Attendant Vectoring can use the route-to number vector step to route calls to attendants located at another Communication Manager node. The operation of the NCR feature using the NCT-type or NCD networks features to accomplish the call redirection is exactly the same as for redirecting ACD calls.

  For more information, see Using the route-to command for NCR in the *Programming Call Vectors in Avaya Aura™ Call Center* document.
- Advice of Charge — No new capabilities are added for the NCR feature for the Advice of Charge PSTN feature. The Advice of Charge feature should be used with the same trunk facilities used for the NCR feature.
- BCMS — No change is made to BCMS for support of NCR. Redirected calls are tracked as completed calls since the PSTN disconnects the incoming facility of the original call when the call is redirected to another site.
- Enhanced Information Forwarding — For the NCR feature, Enhanced Information Forwarding transports User-to-User information (UUI) for the incoming ISDN call to the PSTN endpoint that receives the redirected call. The use of the Enhanced Information Forwarding capability with NCR (the recommended configuration) requires that the incoming call trunk group be assigned as shared (i.e., the **UUI IE treatment** field is set to `shared`). However, if the trunk group is set up as service provider, only the ASAI user information (or user information provided by the incoming ISDN call) will be included in the UUI IE sent on a non-shared basis to the redirected-to PSTN endpoint. NCR supports Information Forwarding for AT&T In-Band Transfer and Connect service.
- Look-Ahead Interflow — NCR activation using the route-to number vector step does not require Look-Ahead Interflow to be active to provide multi-site capabilities, which are required for considering remote locations and access to the BSR Application Plan screen.
- Service Observing by VDN — If the Service Observing by VDN feature is used to service observe a VDN, where the NCR feature is used to redirect incoming ISDN calls, the service-observer will hear the same tones, music, and/or announcements heard by the incoming caller before the NCR feature reroutes the call to another PSTN endpoint. When the NCR operation is completed, the service-observer will be dropped as an observer of the incoming call and placed in the service-observing queue associated with the VDN.
- Trunk-to-Trunk Transfer — If the NCR feature is optioned and the ASAI Third-Party make Call/transfer operation is used to redirect an incoming ISDN to a PSTN endpoint, the

Trunk-to-Trunk Transfer field on the System-Related Customer Options form must be set to `y` for the call redirection to succeed. If the route-to number or BSR queue-to best vector step uses the NCR feature to redirect an incoming ISDN call to a PSTN endpoint, the **Trunk-to-Trunk Transfer customer** option does not need to be set to `y`.

For more information, see Using the route-to command for NCR in the *Programming Call Vectors in Avaya Aura™ Call Center* document.

- VDN Return Destination — If the VDN Return Destination feature is administered for the VDN that is associated with a vector that causes the NCR feature to be invoked, the VDN Return Destination feature will be canceled when the call is redirected by NCR.
- CMS database items — The following Avaya CMS database items are affected by NCR:
    - DEFLECTCALLS: In the VDN CMS database tables, the DEFLECTCALLS item includes the number of calls that are redirected using NCR through the BSR feature by using the **`route-to number`** or **`queue-to best`** commands. Successful NCR attempts are pegged as DEFLECTCALLS.
    - INTERFLOWCALLS: In the VDN CMS database tables, the INTERFLOWCALLS item includes successful BSR interflows using NCR redirections.
    - LOOKATTEMPTS: In the VDN CMS database tables, the LOOKATTEMPTS item includes the number of times the Look-Ahead Interflow or BSR interflow was attempted for calls in the vector. Successful Look-Ahead Interflow or BSR attempts ar also counted. NCR invoke attempts (NCD or NCT) are also reflected in LOOKFLOWCALLS.
    - LOOKFLOWCALLS: In the VDN CMS database tables, the LOOKFLOWCALLS item includes the number of INTERFLOWCALLS that were redirected by the Look-Ahead Interflow or BSR features. LOOKFLOWCALLS is a subset of INTERFLOWCALLS and includes LOOKATTEMPTS for the Look-Ahead Interflow or BSR interflows. With BSR interflow using trunk-to-trunk transfer or NCR, every LOOKATTEMPT will also be counted as a LOOKFLOWCALLS unless a failure occurs.

NCR implementation methods

NCR activation using call vectoring methods

Summary of call vectoring-activated NCR operations

The processes by which NCR is implemented by a call vectoring method is summarized in the following steps:

Note:

The following description does not apply when the AT&T In-Band Transfer and Connect service is used with NCR. For a description of NCR operations associated with that service, see AT&T In-Band Transfer and Connect on page 298.

1. The PSTN switch sends an incoming ISDN call to the Avaya Communication Manager, where the call enters vector processing.
2. One of the following occurs:
   - If the Avaya Communication Manager trunk group and PSTN or VPN switch are configured to use an NCT-type redirection protocol, the redirecting Communication Manager must return an ISDN CONNECT message to the PSTN switch. Any of the following vector commands can be used to return the message:
     - **`announcement`**
     - **`collect x digits`**
     - **`converse-on split`**
     - **`wait hearing music`**
     - **`wait hearing (announcement extension) then ("continue", "music", "ringback" or "silence")`**

     Note:

     If the redirecting Communication Manager does not execute one of the vector steps listed above, a CONNECT message is automatically returned to the PSTN switch.
   - If the server trunk group and PSTN or VPN switch are configured to use the NCD redirection protocol, a CONNECT message must *not* be sent to the PSTN switch. Therefore, when the NCD protocol is applied, none of the vector commands listed above should be included in call vectors that implement NCR.
3. Call processing proceeds to either a route-to number ~r or BSR queue-to best vector step. Depending on which type of redirection is administered for the incoming trunk group, either NCT-type or NCD processes are initiated. In either case, a FACILITY message is sent to the public network over the D-channel associated with the incoming trunk to invoke redirection of the call.

Note:

You should understand the following items that pertain to the PSTN or VPN endpoint number and receiving vector for the interflow location.

4. The PSTN or VPN switch indicates redirection success or failure consistent with the protocol-specific operations described in NCR options supported by PSTNs on page 294. An unsuccessful NCR attempt results in one of the following outcomes:
   - If an NCT-type protocol is used, the redirecting Communication Manager establishes a trunk-to-trunk connection.
   - If the NCD protocol is used and the Avaya DEFINITY version is earlier than load 37 of Release 10, vector processing continues to the next vector step that follows the queue-to best vector step without any best local BSR call treatment.
   - If the NCD protocol is used, the call may be redirected to the best location by means of a trunk-to-trunk connection. However, the ability of the originating server to establish such a trunk-to-trunk connection depends on the specific features of the NCD protocol in use. For more information, see Network Call Deflection (NCD) on page 297.

Using BSR queue-to best vector step to activate NCR

NCR is especially useful for multi-site call center operations in which the Best Service Routing feature is enabled, since the number of PRI B-channels needed for call interflows is reduced. The **`queue-to best`** vector step can be used to interflow ISDN calls between Communication Managers over the PSTN. This method provides the best approach for balancing loads across a multi-site environment and is more cost effective and accurate than pre-delivery routers. For more information about BSR, see Best Service Routing (BSR).

NCR is activated by the **`queue-to best`** vector step when the BSR feature determines a BSR location is the best BSR location and that location is administered with the **Net Redir?** option set to `y` on the BSR Application Table screen. Note that the administered **Interflow VDN** field on the Best Service Routing Application screen must be a PSTN or VPN endpoint number without a trunk/ARS/AAR access codes included. For some PSTN switch dialing plans, the long-distance access code (for example, a “1” in the United States) must be prefixed to the PSTN number for the call to be succesfully routed by he PSTN switch.

As shown in the following example, the Best Service Routing Application Plan screen must include locations that have the **Net Redir?** field set to `y`.

```
                    BEST SERVICE ROUTING APPLICATION

Number: 1     Name:        Maximum Suppression Time: 60       Lock? y
Num  Location Name   Switch Node     Status Poll VDN  Interflow VDN    Net Redir?
1    Omaha                            95552011       3035551211            y
2    Paris                            95552022       18005551234           y
3    Sydney                           95552033       18665553456           y
```

An appropriate vector is then used to identify a BSR best location and NCR is activated by the queue-to-best vector step.

```
wait 2 seconds hearing ringback
```

```
consider skill 1 pri l adjust-by 0
consider location 1 adjust-by 20
consider location 2 adjust-by 40
consider location 3 adjust-by 20
queue-to best
```

Using route-to number ~r vector step to activate NCR

This method can be used to invoke NCR when a **`route-to number`** vector step that specifies a number that begins with the **~**r character. This method can be used to invoke NCR with or without the **LAI**option set to `y` or with Attendant Call Vectoring active.

Note that the administered route-to number vector step number field must be a PSTN or VPN endpoint number without a trunk/ARS/AAR access codes included. For some PSTN or VPN switch dialing plans, the long-distance access code (for example, a “1” in the United States) must be prefixed to the number for the call to be succesfully routed by the PSTN or VPN switch.

Example route-to number ~r vectors: The following examples show vectors that include **`route-to number`** commands to activate NCR, either with or without use of the Attendant vectoring feature.

```
wait 0 seconds hearing ringback
goto step 4 if skill oldest-call < 30 secs
route-to number ~r13035403001
queue-to skill 35 priority m
...
```

```
goto step 6 if time-of-day is all 17:00 to 09:00
wait 0 seconds hearing ringback
queue-to attd-group
wait 999 secs hearing music
stop
route-to number ~r13035551002
```

Using vector/VDN variables with route-to number ~r to activate NCR

The **number** field of the **`route-to number ~r`** vector command can be administered with a global vector variable A-Z or AA-ZZ instead of a PSTN endpoint number after the leading *~r* characters. The **number** field of the **`route-to number ~r`** vector command can also be administered with a VDN variable V1-V9 instead of a PSTN endpoint number after the leading *~r* characters.

An example of using the **`route-to number ~r`** vector command with a vector variable in the **number** field is shown in the following example. For this example, it is required in the Variables for Vectors screen that the following administration is done:

- Vector variable A is defined as of type collect for digit-buffer and L for local
- Vector variable T as of type tod to contain the current system clock time-of-day value

```
1.  goto step 5 if T < 0700      [if time-of-day is less than 7:00 a.m., set up NCR
                                   call-redirection to out of hours PSTN endpoint]
2.  goto step 5 if T > 1800      [if time-of-day is after 6:00 p.m., set up NCR
                                  call-redirection to out of hours PSTN endpoint]
3. set A = none CATR 18005555555   [set digit-buffer to in-office
                                                 hours PSTN endpoint number]
4. goto step 6 if unconditionally [jump to step 6 to do NCR call-redirection ]
5. set A = none CATR 18661111111 [set digit buffer to out-of-office hours
```

```
                                                  PSTN endpoint number]
6. route-to number ~rA [initiate NCR call-redirection operation]
```

For information about using variables with the ~r vector step, see route-to command with vector variables or route-to command.

NCR activation using ASAI Call Transfer and third-party Merge/Release operations

NCR NCT-type operations are activated by ASAI call processing when the Call Transfer or Third-Party Merge/Release operation is performed by a CTI application. This occurs in the following manner:

1. This is typically initiated by the CTI application user selecting an icon, menu item, or button to transfer an answered incoming ISDN call to another party over the PSTN.

   Since the incoming ISDN call must be connected to a station user before the Call Transfer or Third-Party Merge/Release operation is requested, NCR can only initiate the call redirection if an NCT-type protocol is optioned on the trunk.
2. If a call arrives at an ASAI-monitored VDN, ASAI will send appropriate information in the ASAI disconnect event to notify the CTI application that the call has been redirected by NCR.

For the ASAI operations listed above to succeed, the following conditions must be in effect:

- The **Network Call Redirection** field is set on the System Parameters Customer Options screen.
- An NCR NCT-type protocol is administered for both the incoming and outgoing call ISDN trunk group.
- The PSTN number that the CTI application uses to redirect an incoming ISDN call to another PSTN endpoint must be added to the ARS digit analysis screen in such a way that for the NCR MCI NCT and TBCT protocols, the second leg of the call transfer uses the same trunk group with a trunk that has the same D-channel as the incoming call. For the NCR ETSI ECT protocol, the CTI-initiated second call leg can be over a different trunk group with a different signaling group than the incoming call leg.

Note:

NCR-related AAR/ARS routing table administration is required for station transfer or conferencing with MCI trunks. For more information, see Station or ASAI transfer or conference/release administration.

Other things to know about using NCR with ASAI

Using ASAI data for call tracking

ASAI event reporting allows tracking of ISDN ACD calls that were redirected by NCR in a multi-server call center environment. These calls can be tracked by the UCID assigned to each call, or by the UUI information inserted by the application through either the Third Party Make Call or Adjunct Routing features.

ASAI drop event

Successful NCR call redirection causes an ASAI drop event to be sent to the CTI application with a CV_REDIR cause value of decimal (30) after the redirection is completed. Only one

NCR drop event is received for a successful NCR operation when the NCT PSTN feature is used, even though two trunks are dropped by the PSTN.

ASAI third-party merge/call transfer

The CTI application requests a third-party merge/call transfer ASAI operation to transfer the call to the second Communication Manager. This is only used if Network Call Transfer is not available. Once the two calls merge, then ASAI sends a third-party acknowledgement, and when the call is completed, ASAI sends a drop event report, and the third-party call ends.

NCR activation using station call transfer or call conference/release operations

When an incoming ISDN call over a trunk with NCT-type PSTN service is answered at a station or voice response unit (VRU), the station user or VRU places the call on hold, and dials the number for a PSTN of VPN endpoint where the outgoing trunk B-channel is determined by AAR or AAS routing. The station user initiates a call transfer using the Transfer feature button or a switch hook flash, or the VRU initiates a call transfer by using an analog or line-side E1/T1 switch-hook flash.

The switch automatically sends an “invoke NCT” FACILITY message when the transfer is completed if the following conditions are met:

- The **Network Call Redirection** field is set to `y` on page 3 of the System Parameters Customer Options screen.
- An NCT-type protocol is administered for both the incoming and outgoing call ISDN trunk group.
- The second leg call is eligible for redirection by means of an NCT-type protocol, which requires for the MCI NCT and TBCT protocols the second leg of the call is in the same trunk group and has the same signaling group as the incoming call. For the NCR ETSI ECT protocol, the second leg of the call can be over a different trunk group with a different signalling group than the incoming call leg.

If the station user or IVR initiates and completes a three-way conference instead of doing a call transfer operation as above, and releases or hangs up from the conference with the following condition also being met, the switch automatically sends an “invoke NCT” ISDN message to the PSTN or VPN switch if also the following condition is met:

The number of parties in the conference including the conference originator must be no greater than three parties.

Note:

NCR-related AAR/ARS digit-analysis and routing table administration is required for correctly setting up the second call leg over NCT-type trunks associated with the station or IVR call transfer and call conference/release operations. For more information, see Station or ASAI transfer or conference/release administration.

NCR activation using ASAI adjunct route operations

NCR can be invoked by specifying the activate NCR option for the ASAI route message in a route select ASAI message sent by a CTI application to the Avaya Communication Manager after an adjunct routing vector step is executed during call vector processing. This Communication Manager 2.0 capability provides greater flexibility for CTI applications to directly route calls to PSTN or VPN endpoints without the need to specify a VDN extension in the route select ASAI message to route the call instead to a VDN and vector step that activates NCR via a route-to number ~r or queue-to-best vector step. The invocation of NCR by the adjunct routing vector step route select ASAI messagefor various NCT-type protocols follows the same rules as used for the route-to number ~r or queue-to-best vector step operations.

For more information, see the following ASAI documents:

- For information about the Call Options codepoint for NCR Routing or the ASAI Call Route Selection message, see ASAI Protocol Reference.
- For information about possible feature interactions, see ASAI Technical Reference.

Troubleshooting NCR

You can use the following methods and resources to analyze NCR problems:

Proposed Solution 1

1. When NCR and BSR are both implemented, your first troubleshooting step should be to verify that no problems exist with BSR polling and interflow operations when NCR is *not* administered on the BSR Best Routing Application screen.

   After any problems are identified and resolved, set the *Net Redir?* files to `y` on this screen for all locations where NCR is used, and then verify that NCR works properly.

2. The ISDN message trace information provided by the Message Sequence Tool (MST) for the ISDN trunk D-channel associated with NCR invocation attempts.

   The steps to configure MST for NCR troubleshooting are as follows:

   a. Enter the **`ch MST`** Switch Administration Terminal command, then on page 1 set the **ISDN-PRI?** field to `Y`, and on page 2 set the ISDN-PRI Filter Data **Port Type** field to `d-channel` and the **Port** field to the DS1 D-channel switch equipment location associated with the PRI trunk being used with the NCR feature.

   b. Use the **`enable mst`** and the **`list mist cont`** Switch Administration Terminal commands to see NCR-related MST trace data.

   c. When a NCR NCT-type invocation is initiated by vector processing operation or by a manual call-transfer or call-conference/release operation, a D-channel message is sent to the PSTN switch by the Communication Manager to initiate the merging of the two B-channels associated with the first and second call-legs of a trunk-to-trunk call.

      The following MST trace example is for a NCR Two B-Channel Transfer D-channel invocation message that has the same general format as for the MCI NCT, ETSI ECT, or NCD protocols:

```
<msg #> 62 <time stamp> 40 01 18 0F 08 02 80 02 62 1C 09 91 A1
 06 02 01
                04 02 01 06
```

Look for the *91 A1* data-byte sequence shown in bold characters above to verify that a NCR invocation D-channel message is being sent by the Communication Manager.

d. If the NCR NCT-type invocation is successful, the PSTN switch will return a D-channel message to the Communication Manager that has the following general format:

```
<msg #> 60 <time stamp> 00 00 4B 17 08 02 93 E5 62 1C 06
91 A2
03 02 01
                 01
```

Look for the 91 A2 data-byte sequence shown in bold characters above to verify that the PSTN switch accepted the NCR invocation request. A D-channel message instead sent by the PSTN switch that has 91 A3 or 91 A4 data-byte sequence indicates the NCR invocation attempt was rejected. Use the **`display events`** System Administration Terminal command to see vector events that will explain why the NCR invocation failed.

e. For the NCR ETSI ECT protocol, a NCR `Request LinkID` D-channel message is first sent to the PSTN switch by the Communication Manager to determine which D-channel to use for this NCR ETSI ECT invocation: This will result in the PSTN sending a `Returned LinkID` D-channel message to the Communication Manager, where an example of an Ericsson AXE-10 single-byte LinkID MST message is as follows:

```
<msg #> 60 <time stamp> 40 01 18 0F 08 02 00 57 62 1C 13 91 A2
10 02 01
               0B 30 0B 06 06 04 00 82 71 01 04 02 01 FE
```

f. The Communication Manager next will send an `Invoke Explicit ECT` D-channel message to the PSTN switch using the LinkID returned by the PSTN switch, where an example Ericsson AXE-10 single-byte LinkID MST message is as follows:

```
<msg #> 62 <time stamp> 40 01 18 0F 08 02 01 92 62 1C 11 91 A1
0E 02 01
               0C 06 06 04 00 82 71 01 01 02 01 FE
```

g. For any of the NCR NCT-type protocols, a successful invocation results in both legs of the trunk-to-trunk connection being dropped by the PSTN switch after the B-channels are merged.

An example of the PSTN switch first dropping the second call-leg by sending a `Disconnect`, the Avaya switch sending back a `Release`, and the PSTN switch sending a `Release Complete` D-channel message is as follows:

```
<msg #> 60 <time stamp> D  40 01 18 0F 08 02 81 92 45 08 02 82 90 1C
23 91
                           D  A1 20 02 02 00 80 02 01 22 30 17 A1 0F 30
```

```
06 02
                         D  01 00 02 01 01 30 05 05 00 02 01 02 82 01
00 83
                             D  01 00 1C 06 91 A2 03 02 01 0C
<msg #> 62 <time stamp> 40 01 18 0F 08 02 01 92 4D
<msg #> 60 <time stamp> 40 01 18 0F 08 02 81 92 5A
```

An example of the PSTN switch completing the NCR call-redirection operation by dropping the first call-leg by sending a `Disconnect`, the Avaya switch sending a `Release`, and the PSTN switch sending a `Release Complete` D-channel message.To verify the called number information associated with the NCR setup of the second call-leg is correct and to see trunk-related denial events that may be generated if the NCR fails, use the **list trace tac <trunk group number>** Switch Administration Terminal command

3. To see the behavior of a particular VDN or vector, use the **list trace vdn** and **list trace vector** Switch Administration Terminal commands to check for NCR errors.
4. To check for NCR errors using BSR processing:
   a. If you are logged in at the Switch Administration Terminal (SAT) using the init login, enter **go tcm**
   b. When the `tcm1>` prompt is received, enter the **rdd:dp_mgr Bsr_applloc** command to see the total NCR attempts, internal errors, network errors, successful redirections, and disconnects peg counts that are associated with BSR call interflows where NCR was invoked.

   These peg counts are free-running and are only reset when the BSR Best Service Routing Application screen is accessed using the **ch best SAT** command for a particular BSR application number.
5. If NCR vector invocation by call vectoring has failed for previous calls, use the **display events** SAT command to obtain a real-time display of vector events that may be logged for call redirection attempts.

   The possible NCR vector events are as follows:
   a. 68: Adjunct Route via NCT failed
   b. 310 NCR: Invoke trunk not ISDN
   c. 311 NCR: Bad NCR trunk admin
   d. 312 NCR: No NCT PSTN service
   e. 313 NCR: No NCT outgoing trk
   f. 314 NCR: NCT outgo trk drop
   g. 315 NCR: PSTN NCT invoke err
   h. 316 NCR: PSTN NCT netwrk err
   i. 317 NCR: Used NCT trk-to-trk
   j. 318 NCR: No NCD PSTN service

k. 319 NCR: NCD invalid PSTN nmbr
l. 320 NCR: NCD call connect err
m. 321 NCR: PSTN NCD invoke err
n. 322 NCR: PSTN NCD netwrk err
o. 323 NCR: PSTN NCD max redirs
p. 324 NCR: PSTN NCD no disc
q. 325 NCR: Internal system err

Network Call Redirection with SIP

Network Call Redirection with SIP

About NCR support with SIP

Network Call Redirection (NCR) provides an Avaya Communication Manager call routing method between sites on a public network or a Virtual Private Network (VPN) that can reduce trunking costs. These cost reductions are particularly valuable in enterprises or multi-site call center environments where trunk costs are high.

When an incoming call arrives at an Avaya Communication Manager that has the NCR feature enabled, call redirection is managed by the SIP service provider or VPN switch instead of the local Avaya server. As a result, trunks that the server would otherwise retain to accomplish a trunk-to-trunk transfer are released after the call redirection takes place.

The cost reductions associated with reduced trunk use can be significant particularly when Avaya virtual routing features, such as Best Service Routing (BSR) with Expected Wait Time (EWT), are implemented. The cost-savings are achieved by the Avaya customer requiring fewer trunks to handle the same number of incoming/outgoing calls after the NCR feature is implemented within the local Communication Manager.

SIP Network Call Redirection protocols

NCR can occur over SIP trunks. The SIP REFER or SIP 302 Moved Temporarily messages contain all the information needed for Network Call Redirection. When the call is answered by an agent or call vectoring event, a SIP REFER message is sent. If a SIP call is not answered by an agent or a call vector process, a SIP 302 Moved Temporarily message is sent.

SIP limitations on call redirection

You should understand the following items that pertain to SIP limitations on the NCR feature:

NCR feature support

SIP service provider support for NCR varies with geographical location and may be limited or absent in some areas. Consult your Avaya account team to determine SIP service provider availability of one of the NCR protocols in your area.

Allowable number of redirection per call

There may be limits placed on the number of times a call may be redirected over the public network. These limits are imposed by the public network service provider. There might be additional charges associated with redirected calls.

User-to-User information forwarding support

Some public network service providers do not support forwarding of User-to-User Information (UUI), including Adjunct Switch Application Interface (ASAI) user data, collected digits, VDN name, the VDN in-time (as reflected by the NETINTIME database items), and the UCID. In such situations, Information Forwarding will be lost and the second leg of the redirected call will look like an entirely new call to the redirected-to server at the second location.

One of the data items lost is the VDN name, which is rerouted to the originally called service (DNIS) information. The indication that the call has been forwarded can be achieved by using dedicated VDNs for call forwarding, but this strategy loses the benefits of Information Forwarding inherent with NCR and limits use of CTI applications.

SIP NCR and Information Forwarding

The Avaya Information Forwarding feature is supported with NCR when the SIP service provider supports UUI transport in conjunction with the specific network redirection protocol used by the switch.

This section includes the following topics:

- UUI data included in Information Forwarding for a SIP call on page 314
- UUI data forwarding with SIP on page 315

UUI data included in Information Forwarding for a SIP call

Information Forwarding forwards the following call center-related data (as User-to-User Information) with a SIP call:

- Adjunct Switch Application Interface (ASAI) user data
- Universal Call ID (UCID)
- Collected digits

- In-VDN time
- VDN name.

UUI data forwarding with SIP

When NCR is used, the UUI is forwarded by the Avaya Communication Manager in the SIP REFER or 302 Moved Temporarily messages.

The UUI is forwarded by the Avaya Communication Manager in a SIP REFER message if the call has been answered. A call can be answered for example, if an agent answers the call, a vector plays an announcement or music, or the call is processed by a command that provides an answer such as collect digits. The REFER is sent back to the caller, causing the first call to be dropped after a second call is issued and established with the the next location.

The UUI is forwarded by the Avaya Communication Manager in a 302 Temporarily Moved message if NCR is inoked before the call is answered. The first call is redirected to the next location.

SIP NCR feature interactions

NCR interacts with the following call center features:

- Attendant Vectoring — Attendant Vectoring can use the **`route-to number`** vector step to route calls to attendants located at another Communication Manager node. The operation of the NCR feature using the network call redirection features to accomplish the call redirection is exactly the same as for redirecting ACD calls.

  For more information, see Using the route-to command for NCR in the *Programming Call Vectors in Avaya Aura™ Call Center* document.
- BCMS — No change is made to BCMS for support of NCR. Redirected calls are tracked as completed calls since the SIP service provider disconnects the incoming facility of the original call when the call is redirected to another site.
- Enhanced Information Forwarding — For the NCR feature, Enhanced Information Forwarding transports User-to-User information (UUI) for the incoming call to the SIP service provider endpoint that receives the redirected call. The use of the Enhanced Information Forwarding capability with NCR (the recommended configuration) requires that the incoming call trunk group be assigned as shared (i.e., the **UUI treatment** field is set to `shared`). However, if the trunk group is set up as service provider, only the ASAI user information (or user information provided by the incoming call) will be included in the UUI sent on a non-shared basis to the redirected-to SIP service provider endpoint.
- Look-Ahead Interflow — NCR activation using the route-to number vector step does not require Look-Ahead Interflow to be active to provide multi-site capabilities, which are required for considering remote locations and access to the BSR Application Plan screen.
- Service Observing by VDN — If the Service Observing by VDN feature is used to service observe a VDN, where the NCR feature is used to redirect incoming calls, the service-observer will hear the same tones, music, and/or announcements heard by the incoming caller before the NCR feature reroutes the call to another SIP service provider endpoint.

When the NCR operation is completed, the service-observer will be dropped as an observer of the incoming call and placed in the service-observing queue associated with the VDN.

- Trunk-to-Trunk Transfer — If the NCR feature is optioned and the ASAI Third-Party make Call/transfer operation is used to redirect an incoming call to a SIP service provider endpoint, the **Trunk-to-Trunk Transfer** field on the System-Parameter Features form must be enabled for the call redirection to succeed. If the route-to number or BSR queue-to best vector step uses the NCR feature to redirect an incoming call to a SIP service provider endpoint, the Trunk-to-Trunk Transfer customer option does not need to be set to y.

  For more information, see Using the route-to command for NCR in the *Programming Call Vectors in Avaya Aura™ Call Center* document.

- VDN Return Destination — If the VDN Return Destination feature is administered for the VDN that is associated with a vector that causes the NCR feature to be invoked, the VDN Return Destination feature will be canceled when the call is redirected by NCR.
- CMS database items — The following Avaya CMS database items are affected by NCR:
  - DEFLECTCALLS: In the VDN CMS database tables, the DEFLECTCALLS item includes the number of calls that are redirected using NCR through the BSR feature by using the route-to number or queue-to best commands. Successful NCR attempts are pegged as DEFLECTCALLS.
  - INTERFLOWCALLS: In the VDN CMS database tables, the INTERFLOWCALLS item includes successful BSR interflows using NCR redirections.
  - LOOKATTEMPTS: In the VDN CMS database tables, the LOOKATTEMPTS item includes the number of times the Look-Ahead Interflow or BSR interflow was attempted for calls in the vector. Successful Look-Ahead Interflow or BSR attempts are also counted. NCR invoke attempts are also reflected in LOOKFLOWCALLS.
  - LOOKFLOWCALLS: In the VDN CMS database tables, the LOOKFLOWCALLS item includes the number of INTERFLOWCALLS that were redirected by the Look-Ahead Interflow or BSR features. LOOKFLOWCALLS is a subset of INTERFLOWCALLS and includes LOOKATTEMPTS for the Look-Ahead Interflow or BSR interflows. With BSR interflow using trunk-to-trunk transfer or NCR, every LOOKATTEMPT will also be counted as a LOOKFLOWCALLS unless a failure occurs.

Summary of call vectoring-activated NCR operations for SIP

The processes by which NCR is implemented by a call vectoring method with SIP is summarized in the following steps:

1. The SIP service provider switch sends an incoming call to the Avaya Communication Manager, where the call enters vector processing.
2. Call processing proceeds to either a route-to number ~r or BSR queue-to best vector step.

BSR queue-to best vector step activation of NCR

NCR is especially useful for multi-site call center operations in which the Best Service Routing feature is enabled, since the number of trunk resources needed for call interflows is reduced. This method provides the best approach for balancing loads across a multi-site environment and is more cost effective and accurate than pre-delivery routers. The queue-to best vector step can be used to interflow calls between Communication Managers over the SIP service provider network. You can use SIP to interflow BSR calls, but BSR polling is not supported over SIP trunks. Instead BSR polling can be done using other methods, such as Polling Over IP without B-Channel. For more information about BSR, see Best Service Routing (BSR).

NCR is activated by the queue-to best vector step when the BSR feature determines a BSR location is the best BSR location and that location is administered with theNet Redir? option set to`y` on the BSR Application Table screen. Note that the administered**Interflow VDN** field on the Best Service Routing Application screen must be a SIP service provider or VPN endpoint number without a trunk/ARS/AAR access codes included. For some SIP service provider switch dialing plans, the long-distance access code (for example, a “1” in the United States) must be prefixed to the SIP service provider number for the call to be succesfully routed by the SIP service provider switch.

As shown in the following example, the Best Service Routing Application Plan screen must include locations that have the **Net Redir?** field set to `y`.

```
                  BEST SERVICE ROUTING APPLICATION

Number: 1    Name:        Maximum Suppression Time: 60      Lock? y

Num  Location Name   Switch Node    Status Poll VDN  Interflow VDN    Net Redir?
1    Omaha                           95552011       3035551211           y
2    Paris                           95552022       18005551234          y
3    Sydney                          95552033       18665553456          y
```

An appropriate vector is then used to identify a BSR best location and NCR is activated by the queue-to-best vector step.

```
wait 2 seconds hearing ringback
consider skill 1 pri l adjust-by 0
consider location 1 adjust-by 20
consider location 2 adjust-by 40
consider location 3 adjust-by 20
queue-to best
```

Troubleshooting NCR for SIP

You can use the following methods and resources to analyze NCR problems:

Proposed Solution 1

1. When a NCR NCT-type invocation is initiated by a vector processing operation or by a manual call-transfer or call-conference/release operation, a SIP REFER or 302 Moved Temporarily message is sent to the SIP service provider by the Communication Manager to initiate the redirection operation.

   You can determine what User-to-User data was sent in the SIP REFER or 302 Moved Temporarily messages, but the absense of User-to-User information does not mean that an invocation failed, there might not be any User-to-User info at the time the message was generated.

   The following examples are for SIP NCR invocation messages:

   a. A successful SIP NCR invocation with a REFER message:

   ```
   REFER sip:
   30341@135.9.72.61
   ;transport=tcp SIP/2.0^M
   From: "
   3322"
    <sip:
   3322@avaya.com
   >;tag=0b6ce3afb70dc14f047524900^M
   To: "
   ISDN 2"
    <sip:
   30341@avaya.com
   >;tag=0b6ce3afb70dc19a0474ff5700^M
   Call-ID: 0b6ce3afb70dc19b0474ff5700^M
   CSeq: 1 REFER^M
   Max-Forwards: 70^M
   Route: <sip:
   123.4.72.61;lr;transport=tcp>^M
   Via: SIP/2.0/TCP
   123.4.72.136:5062;branch=z9hG4bK8010313dfb70dc150047524900^M
   User-Agent: Avaya CM/R015x.00.0.822.f^M
   Contact: "
   test vdn"
    <sip:
   3322@135.9.72.136
   :5062;transport=tcp>^M
   Refer-To: <sip:
   825030340@avaya.com
   ?User-to-User=00C8103132333435363738393031323
   343536F7020007F803042143%3Bencoding%3Dhex>^M
   Content-Length: 0^M
   ^M
   ```

   A successful invocation is indicated by the User-to-User information for the `Refer-To` header.

The REFER method of NCR invocation is used for the case where the call has already been answered either in the vector or by an answering station.

Failure of the SIP NCR invocation with vector processing will results in continuation of vector processing at the following step.

b. A successful SIP NCR invocation with a 302 Moved Temporarily response:

```
SIP/2.0 302 Moved Temporarily^M
From: "
ISDN 2"
 <sip:
30341@avaya.com
>;tag=0f0a1affb70dc1ac0474ff5700^M
To: "
3322"
 <sip:
3322@avaya.com
>;tag=0f0a1affb70dc15e047524900^M
Call-ID: 0f0a1affb70dc1ad0474ff5700^M
CSeq: 1 INVITE^M
Via: SIP/2.0/TCP
123.4.72.61;branch=z9hG4bK0f0a1affb70dc1ae0474ff5700^M
Server: Avaya CM/R015x.00.0.822.f^M
Contact:
<sip:
825030340@avaya.com
?User-to-User=00C8103132333435363738393031323334353 6F7020002%
3Bencoding%3Dhex>^M
Content-Length: 0^M
^M
```

A successful invocation is indicated by the User-to-User information for the `Contact` header.

2. The 302 Moved Temporarily method of NCR invocation is used for the before answer case that only applies within vector processing.

   Failure of the SIP NCR invocation in this case will result in the stopping of vector processing.

3. To verify the called number information associated with the NCR setup of the second call-leg is correct and to see trunk-related denial events that may be generated if the NCR fails, use the **`list trace tac <trunk group number>`** Switch Administration Terminal command

4. To see the behavior of a particular VDN or vector, use the **`list trace vdn`** and **`list trace vector`** Switch Administration Terminal commands to check for NCR errors.

5. If NCR vector invocation by call vectoring has failed for previous calls, use the **`display events`** SAT command to obtain a real-time display of vector events that may be logged for call redirection attempts.

   The possible NCR vector events are:

   a. 68: Adj Route via NCR failed

   b. 311: Bad NCR trunk admin

c. 312: NCR: No NCT service
d. 316: NCR: NCT netwrk err
e. 318: NCR: No service
f. 319: NCR: invalid num
g. 322: NCR: netwrk err
h. 325: NCR: Internal system err
i. 327: NCR: Caller not SIP trk

Proactive Contact outbound calling improved reporting

About Proactive Contact outbound calling improved reporting

Avaya Call Center customers using a Proactive Contact integration can track and analyze the time agents in AUX Work mode spend on outbound calls using the same real-time and historical Call Management System (CMS) reports used by customers with other Outbound Call Management (OCM) applications. This feature:

- Delivers outbound Proactive Contact calls to agents that are in the AUX Work mode as though the calls were Automatic Call Distribution (ACD) calls associated with a skill specified for reporting.
- Includes these calls in Least Occupied Agent (LOA) occupancy calculations since they are treated as ACD calls by the Communication Manager software.
- Tracks switch-classified and non switch-classified outbound calls on CMS as ACD-OUT calls instead of as AUX-IN calls.
- Allows Avaya IQ to track these calls using the Proactive Contact event stream instead of via the Management Information System (MIS) link SPI (Switch Protocol Interpreter) events off the connected Avaya Communication Manager.

Reasons to use Proactive Contact outbound calling improved reporting

For customers using Proactive Contact, this feature improves reporting capabilities and provides fair treatment of agents.

Prerequisites for using Proactive Contact outbound calling improved reporting

You can use the improved integration with Proactive Contact outbound calling only if all of the following Avaya Call Center conditions are true:

- The **Expert Agent Selection (EAS)** field is set to `y` on the System-Parameters Customer-Options screen.
- The **ASAI Link Plus Capabilities** field or the **Computer Telephony Adjunct Links** field is set to `y`. Both fields are located on the System Parameters Customer-Options screen.
- The **Call Center Release** field is set to 4.0 or later on the System Parameter Customer-Options screen.

Switch-classified and non switch-classified calls

Switch-classified outbound calls are outbound calls placed by the Proactive Contact dialer and connected to the agents after the Communication Manager Call Classifier determines that the call has been answered.

Non switch-classified outbound calls are outbound calls that are automatically launched by Communication Manager and connected to an available agent during call setup. This configuration is also referred to as agent-classified calling.

The Proactive Contact outbound calling process

Acquiring agents for outbound calling

The process described in this section applies to both switch-classified and non switch-classified calls. Proactive Contact selects agents specified for inbound calls and acquires them for outbound calls as follows:

1. Agents are assigned both inbound skills and a skill defined for outbound calling.
2. Agents log in to both Communication Manager and Proactive Contact and take inbound calls in Auto-In or Manual-In mode.
3. Agents select an outbound campaign (application) using the Proactive Contact terminal.

4. Proactive Contact acquires agents who have selected an outbound campaign when Proactive Contact determines that current staffing is more than adequate for handling inbound calls.

   The details are as follows:

   a. Proactive Contact obtains an available agent by placing a call to the outbound skill using an ASAI Third-Party Make Call operation with a phantom number as the originator.

      The call is made to a VDN whose vector has a queue-to outbound skill step. This setup is used to acquire agents for outbound calling.

   b. The queue-to step selects an available agent and Proactive Contact then changes the work state of the acquired agent to AUX using the ASAI Change Agent Work Modes request feature and then drops the connection.

   c. Proactive Contact then uses the Third-Party Make Call operation to send the call to an announcement extension using the acquired agent as the originator.

      The agent hears a recording that says, “You are acquired for outbound calling”. The connection then drops.

5. Proactive Contact launches an outbound call.

   For more information, see:

   - Launching switch-classified outbound calls on page 322
   - Launching non switch-classified outbound calls on page 324

Launching switch-classified outbound calls

1. Proactive Contact launches a switch-classified outbound predictive call request through an ASAI Third-Party Make Call operation using a VDN as the originator.
2. When the call is classified as answered, Communication Manager connects the call to the originating VDN.

   The adjunct routing step in the assigned vector requests a route.

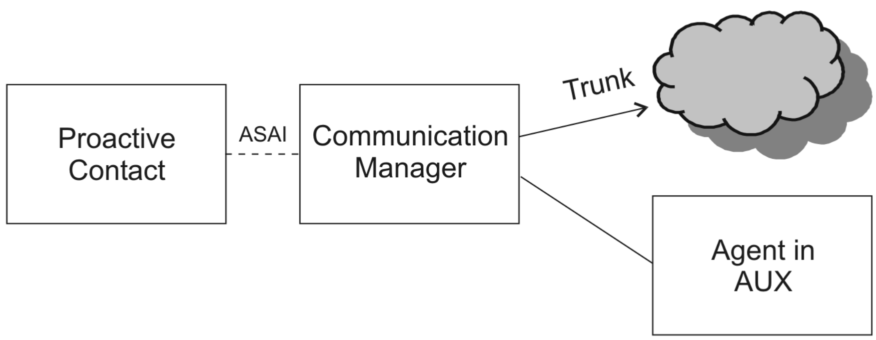
   For more information about switch-classified calls, see Avaya Call Center Automatic Call Distribution (ACD) Guide.
3. Proactive Contact does one of the following steps:
   - If there is an acquired agent in AUX Work mode that is available for that campaign, Proactive Contact immediately has Communication Manager connect the call to that agent using a route-select message.
   - If there is no available agent, Proactive Contact waits to send the route-select message until an agent is available. The vector provides a message to

the called party, waits until the called party drops, or provides other programmed treatment.

4. The call delivery is reported to the reporting adjuncts.
   - CMS receives events for an ACD-OUT OCM (ACDO) call that is associated with the assigned reporting skill instead of an AUX-IN call.
   - Avaya IQ (if connected) receives a message to ignore these events.

Sample of Proactive Contact launching an outbound switch-classified call to an agent in AUX Work mode

The following example shows how Proactive Contact launches an outbound switch-classified call to an agent in AUX Work mode.



ASAI Third-Party Make Call operation for VDNx → Communication Manager places outbound call and classifies it as answered → Call is connected to VDNx
Vector z is processed
Route request sent to Proactive Contact
Route-select message sent to route the call to the available agent

```
VDNx - vector z
1. adjunct routing link 1
2. wait-time 2 secs hearing silence
3. announcment 42015
4. wait-time 10 secs hearing silence
5. goto step 3 unconditionally
6. disconnect after announcement 42020
```

Launching non switch-classified outbound calls

1. Proactive Contact launches an outbound call using an ASAI Third-Party Make Call operation to an available acquired agent in AUX Work. The Priority Call parameter for the Third-Party Make Call operation must be set to yes.
2. Communication Manager connects the call to the agent at the same time the outbound connections are being established.
3. The call delivery is reported to the reporting adjuncts.
   - CMS receives an ACD-OUT OCM (ACDO) call that is associated with the assigned reporting skill instead of an AUX-IN call based on the setting of the Third-Party Make Call priority flag. The call is reported to CMS only if Proactive Contact non switch-classified calls are administered. For more information, see Administering PC non switch-classified calls for improved reporting on page 325.
   - Avaya IQ receives a message to ignore these events.

Administering PC switch-classified calls for improved reporting

To administer improved reporting for Proactive Contact switch-classified calls:

1. Enter `change vdn xxxxx`.
2. On the Vector Directory Number screen (page 2), set the options for the following fields:
   - **Reporting for PC Predictive Calls?**

- **PC Predictive Reports Skill:** (appears after the **Reporting for PC Predictive Calls?** field is set to `y`.

  For more information about these fields, see Vector Directory Number field descriptions in *Administering Avaya Aura™ Call Center Features*.

3. Press **Enter**.

Administering PC non switch-classified calls for improved reporting

1. Enter `system-parameters features`.
2. On the Feature-Related System Parameters screen (page 13), set the options for the following fields:
   - **Report for PC Non-Predictive Calls?**
   - **PC Non-Predictive Reports Skill:** (appears after the **Report for PC Non-Predictive Calls?** field is set to **y**.)

   For more information about these fields, see Call Center Miscellaneous fields in *Administering Avaya Aura™ Call Center Features*.
3. Press `Enter`.

Proactive Contact improved reporting interactions

Interaction	Description
BCMS	You can use Basic Call Management System (BCMS) to track Communication Manager switch-classified and non switch-classified calls. BCMS reports for Proactive Contact (PC) include ACD-OUT triggered and ACD time tracking. There is no ring-time associated with Proactive Contact calls since Proactive Contact agents are logged in to auto-answer stations.
Forced Agent Logout by Clock Time	You should not use the Call Center 4.0 Forced Agent Logout by Clock Timer feature with the Call Center 4.0 Improved Integration with Proactive Contact Outbound Calling capability. The Proactive Contact system places a PC agent in the AUX work-mode

Interaction	Description
	when the agent is making an outbound PC-imitated call. If the administered time for Forced Agent Logout by Clock Time is reached, the PC agent will be logged out immediately.

Queue Status Indications

About Queue Status Indications

Queue Status Indications allows you to assign queue-status indicators for Automatic Call Distribution (ACD) calls based on the number of split or skill calls queued and time in queue. You can assign these indications to lamps on agent, supervisor, or attendant telephones or consoles to help users monitor queue activity.

In addition, you can define auxiliary queue warning lamps to track queue status. On telephones and consoles with displays, you can display the number of calls queued and time in queue of the oldest call in the split or skill.

Queue Status Indication detailed description

There are two types of Queue Status Indications:

- Number of queued calls (NQC) - The system reports the total number of calls, excluding direct agent calls, in queue at a hunt group.
- Oldest queued time (OQT) - The system reports the time in queue of the oldest call in a split or skill queue.

You can also use auxiliary queue warning lamps to provide both types of indications. Install the lamps at any location convenient to agents and supervisors.

If a queue status threshold is reached, the lamp next to the associated button flashes. If calls are queued but the threshold is not reached, the lamp lights steadily. If no calls are queued, the lamp goes dark.

If the OQT or NQC button on a telephone or console with display is pressed, the following information is briefly displayed:

- Split or skill name (or extension, if name is not assigned)
- Oldest queued time
- Number of queued calls

You can use Queue Status Indications to provide status information for attendant groups or other hunt group types (DDC and UCD). With attendant groups, the button names (AQT and AQC) are different than for split or skill queues, the display shows OPERATOR instead of the split or skill name or extension, and all status information applies to the attendant group queue.

If you need to know how many queue status buttons have been administered, or how many your system will allow you to administer, check page 5 of the System Capacity screen.

Queue Status Indication interactions

Queue Status Indication	Description
Attendant and Telephone Display Timers	The timer and the queue status information can be displayed at the same time. On 1-line displays, the timer is displayed in the last eight display positions and the number of queued calls is not displayed. On 2-line displays, the timer is displayed on the first line and the queue status information is displayed on the second line.
CMS	When you use CMS to move an agent from one split to another, all buttons associated with the first split, including NQC and OQT buttons, become associated with the second split. With EAS skills the move of an agent just adds or removes skills the agent is logged into and the skill assigned to the Queue Status Indicators do not change.

Reason codes

About reason codes

Reason codes allow agents to enter a numeric code that describes their reason for entering Auxiliary (AUX) work mode or for logging out of the system. Reason codes give call center managers detailed information about how agents spend their time. Use this data to develop more precise staffing forecasting models or use it with schedule-adherence packages to ensure that agents are performing scheduled activities at the scheduled time.

You can administer the codes so that entry of the code is forced or optional. Beginning with 3.0, you can have up to one hundred AUX reason codes, including a default code (0).

You can use VuStats to display the reason code name or number. Use VuStats or CMS to gather historical and real-time reason-code statistics.

You must have Expert Agent Selection (EAS) enabled to use reason codes.

Reason code detailed description

Reason Code	Description
Forced reason codes	If you have administered forced reason codes, agents cannot enter AUX work mode or log out until they enter a code. Agents can enter codes 1 - 99, but not default code 0. If an agent enters an invalid code or fails to enter a code within the 10-second timeout interval, the change is denied and the agent remains in the current work mode. If the agent dialed a FAC, the agent hears an intercept tone. If the agent pressed the AUX button, the AUX lamp flutters and then goes dark or lights steadily if the agent was already in AUX with a different reason code.
Requested reason codes	If you have administered requested reason codes, agents need not enter a code to enter AUX work mode or log out. Agents can enter the codes 0 - 99. If an agent enters an invalid code or fails to enter a code within the timeout interval, the agent enters AUX work mode or logs out with default code 0.
Entering AUX work mode	An agent can enter an AUX reason code in one of three ways: • Pressing an AUX work button with an assigned code • Pressing an AUX work button with no assigned code and responding to the prompt for a reason code • Dialing an AUX work FAC and responding to the dial tone prompt for a reason code If there are no calls ringing, active, or held at the telephone, agents enter AUX work mode

Reason Code	Description
	immediately and the AUX lamp lights steadily. Otherwise, the AUX lamp blinks until the agent completes all calls at the telephone. If a button for AUX work is associated with the reason code that the agent entered, the button lamp lights. If no such AUX button exists, the system lights the first AUX button lamp with no administered reason code. You can assign an AUX button without a reason code to an agent's phone. This allows agents with a limited number of buttons to use all 100 reason codes.
Logging out	To log out with a reason code, the agent dials the logout FAC, hears a second dial tone and enters a reason code. The agent hears confirmation tone and is logged out.
Default code	Default code 0 is used when the system puts an agent into AUX work mode or logs the agent out without the involvement of the agent. For example: • When an agent logs in and is put into AUX mode • When an agent makes or receives a non-ACD call from the available state • When a call is redirected as a result of Redirection on No Answer (RONA) and the agent is logged out or put into AUX mode • When agent skill assignments are changed while an agent is staffed (the system automatically logs the agent out and back in) • When an agent forces a logout without entering a code (for example, by pulling the headset) • When an agent who is requested to enter a reason code fails to enter a valid code within the 10-second timeout period • When an agent with requested reason codes enters # or *

Reason Code	Description
Reason codes not interruptible:	The following 3 reason codes cannot be made interruptible: • Auto-answer IP Failure Aux Work Reason Code • Maximum Agent Occupancy Aux Work Reason Code • Default Reason Code listed at the bottom of screen one of the Reason Code Names form **Note:** For more information on Interruptible Aux, see Interruptible Aux work.

Reason code considerations

- If an agent in auto-in or manual-in work mode dials the logout FAC but fails to enter a reason code and logout reason codes are forced, the agent is returned to the available state. ACD calls are delivered even if the agent has left the phone. To prevent this, be certain that agents enter AUX or ACW work mode before logging out.
- When an agent changes to AUX work mode and the AUX Work Reason Code Type is set to none, the agent is put into AUX work mode with the default reason code even if you have administered a different reason code for the AUX button. Setting AUX Work reason code in this way allows you to complete button administration before activating the feature.
- Do not administer AUX buttons without a reason code for hybrid station sets.
- When an agent in AUX work mode is active on a non-ACD call, the agent cannot immediately change the reason code. A change is pending until the call drops.
- There is a limit to the number of agents who can simultaneously be entering either a reason code or a Call Work Code.

Reason code interactions

Reason Code	Description
Abbreviated Dialing	You can program FACs for AUX work mode or logout with or without an associated reason code on automatic-dial buttons or in abbreviated-dial lists. At the reason code prompt, when an agent

Reason Code	Description
	selects an abbreviated-dial or automatic-dial button, the first digit of the button is taken as the reason code.
Agents in Multiple Skills	When an agent who is assigned to multiple skills enters AUX work mode with a reason code, the agent enters AUX work for all of his or her skills with the same reason code.
ASAI	ASAI allows a host to log an agent out and place an agent in AUX work mode with a reason code. The host can query the agent's current work mode and receive the reason code associated with the AUX work mode.
Auto-Available Split/ Skill	The system logs AAS agents out for Redirection on No Answer with the default reason code.
Basic Call Management System	Statistics about AUX work mode by reason code are not available in BCMS reports.
CMS	CMS tracks time in AUX work mode by reason code and displays reason codes for agents currently in that mode. When an agent is moved from CMS while the agent is staffed, the system logs the agent out using the default code, and then logs the agent back in again. If an agent is in AUX work mode when moved, the agent is returned to AUX work mode with the same reason code when the move is completed.
Direct Agent Calling	When a direct agent call is queued for an agent in AUX work mode with a reason code, the appropriate AUX button lamp flutters to alert the agent to the queued call. If there is no AUX button lamp, agents receive an audible alert (ring-ping or call-waiting tone). If there is an AUX button with no assigned reason code administered, then that lamp flutters.
Redirection on No Answer	When a call is redirected using RONA, an agent is placed into AUX work mode with the default code or is logged out with the default code if the agent is in an auto-available skill.
Redirection on IP Failure and Redirection on OPTIM Failure	The ROIF and ROOF features use the same reason code.

Redirection on No Answer

About Redirection on No Answer

Redirection on No Answer (RONA) redirects a ringing ACD split or skill call or direct agent call after an administered number of rings. RONA prevents an unanswered call from ringing indefinitely especially for IVRs/VRUs that may have one or more ports fail. The call can redirect either to the split or skill to be answered by another agent or to a VDN for alternative call handling. Direct agent calls route to the agent's coverage path, or to a VDN if no coverage path is administered.

You must have ACD enabled to use RONA. Administer RONA for each ACD hunt group as required. RONA can be used in Auto-Available Splits/Skills (AAS), or in splits/skills with agents operating in Auto-In/Manual-In work mode. You can administer RONA for vector-controlled or non vector-controlled splits/skills. RONA only applies to manual answer station operation where the station or port is rung waiting for answer. RONA will not work with auto-answer configurations.

Do not administer RONA for splits/skills controlled by adjuncts or AUDIX or for auto-answer agents assigned splits/skills because calls must ring at a telephone to be redirected.

You can specify whether to retain the active VDN context when RONA redirects a call to an alternate VDN defined as the redirect VDN due to an agent that has not answered. When you administer the **Retain Active VDN Context?** field as `y`, the VDN context from the previous active VDN is retained and used after the call is redirected to the specified redirect VDN. If you administer the **Retain Active VDN Context?** field as `n` and RONA occurs, the system uses the context of the applicable redirected to VDN. The active VDN for a call is based on VDN Override rules, normally being the first VDN called unless overridden by a routed to VDN. For more information, see VDN Override in the *Programming Call Vectors in Avaya Aura™ Call Center* document.

The VDN context includes the following information:

- VDN Name
- Tenant Number (TN)
- VDN of Origin Announcement (VOA) Extension
- VDN Skills (1st, 2nd, 3rd)
- VDN Return Destination

Note:

The VDN Return Destination is set before being RONA/ROIF/ROOF redirected and is not changed by subsequent routing. The staffed agent receiving the redirect call sees

"CR" at the right end of the display indicating that this is a RONA/ROIF/ROOF redirected call so that they can provide more appropriate answering treatment.

- VDN Timed ACW (After Call Work) Interval
- BSR (Best Service Routing) Application
- BSR Available Strategy
- BSR Tie Strategy
- Display VDN for Route-to DAC (Direct Agent Calling)
- Trunk ASAI (Adjunct Switch Application Interface) Messages
- BSR Local Treatment
- VDN Variables
- VDN Time Zone Offset

If you choose to retain the active VDN context, you can set up a generic VDN-vector combination that caters to calls redirected from multiple VDNs with specialized treatment based on the context parameters of the previous active VDN. For more information, see Generic VDNs for redirected calls handling on page 336.

If the call is redirected to a VDN when routed directly to a hunt group rather than through a VDN, the redirect to VDN is the active VDN regardless of the setting of the **Retain Active VDN Context?** field.

RONA detailed description

When RONA is invoked for a call, the system:

- Places an agent in AUX work mode, and thus unavailable to receive calls from other splits/skills. In an AAS, the agent is logged out.
- Redirects split or skill calls back to the split or skill or administered VDN. Redirected calls are re-queued at the highest priority so that they are distributed before any other split or skill calls. See RONA routing sequences on page 335 for more information about call redirection.
- Sends a message to CMS. When a RONA timeout occurs, the Noans-alrt lamp for the split or skill lights steadily. The supervisor presses the Noans-alrt button to display the login ID or the extension and name of the last agent timed out with RONA.
- Records the redirection in BCMS or CMS. See Using BCMS/CMS reports with RONA on page 336 for additional information.

RONA application examples

VRU applications

Typically, RONA is used with IVR/VRU applications in AAS configurations. RONA detects VRU failures and provides alternate operation. For example, an adjunct port failure is not detected by ACD call processing. RONA detects the failure, takes the port out of service, and provides notification of the failure.

Use Call Vectoring for flexible call handling in case of a VRU failure. Assign RONA to a converse split or skill connected to the IVR system or to equivalent VRU ports. Whenever RONA times out on a ringing call delivered using the **converse-on** command to an AAS VRU port, the port is logged out and the call is redirected back to the converse split or skill.

With a complete VRU failure, all VRU ports are eventually logged out and vector processing for the **converse-on** command bypasses that step for new calls.

The following vector example shows how to provide automatic backup for a complete VRU failure.

Example vector - Providing automatic backup for a complete VRU failure

```
                                 CALL VECTOR

01 wait-time 0 secs hearing ringback

02 converse-on split... (VRU returns the digit “
1”
 as a return code

        followed by additional digits for the application)

03 collect 1 digits after announcement none

04 goto step 6 if digits = “
1”

05 goto vector xxx (for backup when the VRU fails)

06 collect 2 digits after announcement none

07 ...
```

In the example vector shown above, the application works as expected as long as the VRU returns the digit string, which includes a return code of 1. In this case, the condition in Step 4 is satisfied and the program branches to Step 6, which provides normal application processing.

On the other hand, if all VRU ports in an AAS split or skill are logged out by a RONA timeout, the **`converse-on`** command step (Step 2) is skipped, and no digits are collected by Step 3 (after the 10-second timeout). The condition in Step 4 is not satisfied and vector processing proceeds to Step 5, which branches to vector xxx to connect the call to an agent.

Other applications

You can use RONA for applications that involve human agents with manual answering and other adjunct applications, such as Home Agent. For example, a call may not be answered because an agent left without entering AUX work mode or logging out. You can use RONA to make the non answering agent unavailable and redirect calls to another agent or to the RONA VDN.

RONA routing sequences

The following tables describe how RONA redirects split or skill calls and direct agent calls.

Table 33: RONA routing sequence for direct agent calls

Redirection Destination	Explanation
Coverage path	Direct agent calls redirect to a coverage path, if one exists. Priority calls do not route to coverage.
RONA VDN	Calls redirect to the VDN specified when RONA occurs. If no coverage path exists when a VDN is administered for RONA, direct agent calls redirect to the VDN. If the **Retain Active VDN Context?** field is `y`, the system retains the VDN context from the previous active VDN after the call redirects. Otherwise, the system uses the context of the redirect to VDN.
VDN return destination	For external calls, if neither a coverage path nor a RONA VDN are administered, then IC email calls redirect to the VDN Return Destination extension.
None	Calls continue ringing.

Table 34: RONA Routing Sequence for Split Or Skill Calls

Redirection Destination	Explanation
RONA VDN	If a RONA VDN is administered, calls redirect to the VDN.If the **Retain Active VDN Context?** field is `y`, the system retains the VDN context from the previous active VDN after the call redirects. Otherwise, the system uses the context of the RONA VDN.

Redirection Destination	Explanation
Requeue to split or skill	If a RONA VDN is not administered, the call redirects back to the split or skill at a priority that is above the highest priority.
Coverage path	In non vector-controlled splits, if calls cannot requeue to the split, they redirect to the split's coverage path if one is administered.
VDN return destination	For external calls, if a coverage path or a RONA VDN is not administered and calls can not requeue, they redirect to the VDN Return Destination extension.

Using BCMS/CMS reports with RONA

You can use BCMS and CMS reports to determine which agents had RONA timeouts and how calls were redirected.

With R3V2 and later releases of CMS, the exception report lists agents who were timed out and made unavailable. With BCMS and earlier releases of CMS, you can determine which agents were in AUX work mode or logged-out with AAS.

With R3 CMS, you can use the real-time Split Status report to see which agents are in AUX work mode, but you need a custom report to see logged-out agents.

With BCMS, use SAT to create a list of unstaffed agents for the split to see which agents are logged out (for AAS applications). With EAS, list agent-loginid specifying unstaffed and AAS = yes.

With BCMS, agents' changes to AUX work mode appears in the BCMS Split (AGENT) Status report. In an AAS split, agents log out, so they do not appear in the Split Status report. When the call is requeued, the System Status report shows only the AVG ANSW SPEED time and AVG ABAND TIME time for the requeued call. The Historical Split and system reports show both a FLOWOUT (primary split) and FLOWIN (redirected split) for requeued calls, while the VDN report shows only a FLOWOUT.

Direct agent calls are recorded as ACD split or skill calls but the flowout is recorded only if an agent's coverage path requeues the call to a split or skill.

Since BCMS does not report exceptions, RONA events are not reported. If you have BCMS, use the RONA split or skill lamp indicator for RONA event indication.

Generic VDNs for redirected calls handling

For the RONA, ROIF, and ROOF calls, you can set up a generic VDN and assigned vector that:

1. Utilizes the VDN/Call parameters carried over by the “VDN Override function” to specialize the treatment and further routing for the call.
2. Allows specialized treatment for redirected calls from many different applications based on each of their original active VDNs.

Note:

For information on setting up a generic VDN for redirected calls handling, see Setting up a generic VDN to handle redirected calls on page 337.

Setting up a generic VDN to handle redirected calls

To use the VDN context /Call parameters carried over by the VDN override function:

1. Administer the following fields for the hunt group as specified below:
   a. Administer **Redirect On No Answer to VDN** and/or **Redirect on IP/OPTIM Failure to VDN** fields as `2110200` (The generic VDN number where you want to redirect the calls).
   b. Administer the **Retain Active VDN Context?** field as `y`.
2. Set up a generic vector to take care of RONA, ROIF, and ROOF call handling.

```
                                          CALL VECTOR
1 announcement V1 (Using a VDN variable for the application to give a
personalized announcement)
2 queue-to skill 1st pri t (puts the call in a queue personalized for the
application)
3 queue-to skill 2nd pri t (adds another queue specific for the
application)
4 announcement V2 (a further personalized wait announcement)
```

In the example, vector step 1 plays a personalized announcement identified by the extension stored in the active VDN’s variable V1, that the caller gets if the original call delivery fails. The announcement is specified by a VDN variable (V1) assigned to the active VDN instead of the actual extension number so redirected calls form other VDNs can hear the announcement defined by the V1 assigned to the VDN the call was redirected from. Step 2 queues the call to the skill assigned to the 1st skill parameter assigned to the active VDN with top priority. Step 3 then queues the call to the 2nd skill that is assigned to the active VDN Step 4 is for giving a personalized announcement for the waiting caller utilizing the 2nd VDN Variable (V2) assigned to the active VDN.

For a call routed to the generic vector for the redirected calls, the answering agent gets the following display: `303-555-0003 to Billing CR`

Where:

a. 303-555-0003 = caller’s number
b. Billing = The name of active VDN (application) the call was redirected from

c. CR = Indication that the call was redirected and not a directly routed call

Returning AAS agents to service

When RONA redirects a call that was directed to an AAS, the agent is logged out. To return an AAS agent to service, readminister the agent as a member of the AAS split or skill to be logged in again in one of the following ways:

- For ACD splits, remove the agent from the split and then resubmit the split Hunt Group screen with the agent added to it. Alternatively, administer the agent in a different location in the split members list on the Hunt Group screen. Use the **`list unstaffed-agents`** command to get a list of all AAS agents that have been logged out, not just AAS agents that were logged out because of a RONA timeout.
- For EAS skills, readminister the Agent LoginID screen so that the AAS agent is automatically logged in. To determine which EAS agents are logged out, use the **`list agent-loginid`** command.
- For ACD splits and for EAS skills, you can busy-out the AAS agent station with the **`G3-MT busyout station`** command and release it with the **`release station`** command. Releasing the AAS agent station automatically logs the agent in. If all AAS agent ports on the circuit pack had a RONA timeout, busy-out and release the entire circuit pack.
- Use CMS Move Agents to move up to 32 agents at a time into a dedicated unused split or skill and then move the agents back into the AAS split or skill. You can set this up using the timetable on a manual-scheduled basis to activate when the VRU has been restored to service after a failure.
- Use ASAI to log the logged-out agents back in using ASAI login request messages.

RONA considerations

- RONA can timeout while an agent is actually at the station if the agent does not answer soon enough or has selected another work mode while a call is ringing. RONA handles the call as usual, making the agent unavailable. With ACD splits, agents at multifunction telephones know that they have been made unavailable when they see the aux-work lamp lit. They can press the auto-in or manual-in button to become available.
- Specify a coverage path or VDN for redirection for non vector-controlled splits or for Logical Agent IDs with EAS direct agent calls to ensure that calls are always redirected.
- A noans-alrt button can be assigned to non-SIP agent or supervisor phones to indicate that a call has been redirected. When the noans-alrt button is pressed, the phone display shows the login ID or extension and name of the last agent timed out with RONA.

RONA interactions

Interaction	Description
AAS	Use AAS with RONA for VRU ACD non-ASAI adjunct-controlled split or skill applications. Assign AAS only to ACD hunt groups. When all lines in a vector-controlled AAS split or skill are logged out, the split or skill is considered unavailable, and vector processing skips the step in the vector for new calls. If RONA occurs on the last VRU port in an AAS split, the call is not requeued to the converse split, but is processed by the next vector step. Any calls queued to a split or skill that has been taken out of service may be left at this split or skill. When the system reinitializes, all busied-out ports are automatically logged back into the AAS splits. New calls cause a RONA timeout if the adjunct or agent still does not answer after the system reinitializes.
Abandoned Call Search	Abandoned Call Search, if defined for a trunk, is reapplied to call on that trunk that RONA requeued whenever the calls are routed to another agent.
Agents in multiple splits	When a RONA timeout occurs, an agent is placed in AUX work mode with notification to CMS for all splits that the agent is logged into. The agent is responsible for becoming available in each split. In an AAS, agents are logged out of all splits that they are logged into. You must log agents back into the AAS splits.
Agent logout	An agent can log out from a multifunction set while an ACD call subject to RONA is ringing the set. However, if the agent logs out before RONA times out, RONA timing is canceled, and RONA redirection and notification occur immediately.
Agent work modes	If an agent presses the ACW button with an ACD call ringing, the change request is pending. If the agent has a pending change to ACW before a RONA timeout occurs on a ringing ACD call, RONA timing continues. At timeout, the call is redirected, CMS is notified, and the agent is placed in AUX work (overriding the pending ACW request). If an agent presses the aux-work button with an ACD call ringing, the change request is pending. With ACD splits/skills, since the RONA time-out changes the state to aux-work, there is no conflict with the pending aux-work change request. With AAS splits/skills, an agent-initiated aux-work change is denied per existing operation.
ASAI	RONA applies to vector-processed calls that are routed by an adjunct to a split or agent as a direct agent call. You can assign RONA to ASAI adjunct-monitored splits and adjunct-monitored calls. An event report is not sent to the ASAI adjunct when a RONA timeout puts an agent into AUX work mode. The adjunct makes an agent query (as part of the value query capability group) to determine the agent's state. Once the call is requeued to the

Interaction	Description
	split, the adjunct receives a call-queued event report if event reporting is active for the domain (VDN or non vector-controlled split or skill). An adjunct-monitored split or skill can be assigned as an auto-available split or skill. The logout event for an AAS split or skill is sent to the adjunct when RONA timeout logs an agent out. You cannot assign RONA to an adjunct-controlled split or skill. An adjunct-controlled split or skill cannot be an AAS. ASAI IVR VRU applications are configured with non vector-controlled splits/skills using manual-answer operation on analog lines to the IVR ports. The ASAI link provides event notification for the ACD split or skill for enhanced services. In addition, you can log in and log out the ports as required. (AAS splits/skills are not used for this application because the ASAI link controls the login or logout). You can assign RONA to these splits/skills to detect failure conditions in the same manner as non-ASAI VRU applications. RONA does not notify the IVR system of AUX work mode changes. An ASAI IVR system cannot query to determine the states of its ports. You must restore ports manually after a failure using the IVR system management screens. Complete failure is automatically restored when the IVR system reinitializes. The following table describes ASAI events that the communication server sends the adjunct for various stages of the RONA call. Also included are the ASAI associations (assuming that they are active) for which the events are provided. For the split or skill to have Notification association active, the split or skill must not be vector-controlled or adjunct-controlled. **Note:** When a call is redirected using ASAI Redirect Call, the RONA timer is canceled.

Stage of Call	ASAI Event	ASAI Associations
1. RONA timeout	Logout (for AAS)	Domain (agent) control
2. Call redirected to split	Call redirected	Domain (station) control (for agent ext call is leaving)
	Call queued (only if the call queues)	Domain (station) control, (for new agent & for internal originator) call control, notification
3. Call redirected to VDN processing	Call redirected	Domain (station) control (for agent ext that call is leaving)
	Call redirected (only if call is redirected to a	Notification

<table>
<tr><th>Interaction</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Stage of Call</th><th>ASAI Event</th><th>ASAI Associations</th></tr>
<tr><td></td><td>VDN with Notification active)</td><td></td></tr>
<tr><td></td><td>Call offered to Domain</td><td>Notification (VDN)</td></tr>
<tr><td>4. Call delivered to agent</td><td>Alerting</td><td>Domain (station) control, (for new agent & for internal originator) call control, notification</td></tr>
<tr><td>5. Call routed to split’s coverage path</td><td>Call redirected followed by existing operation of ASAI Events</td><td>Domain (station) control (for agent ext that call is leaving)</td></tr>
<tr><td>6. Infinite feedback to caller</td><td>Call redirected</td><td>Domain (station) control (for agent ext that call is leaving)</td></tr>
<tr><td>7. Continue vector processing</td><td>Call redirected followed by existing ASAI events</td><td>Domain (station) control (for agent ext that call is leaving redirecting to VDN)</td></tr>
<tr><td>8. Call routed to direct agent’s coverage path</td><td>Call redirected followed by existing operation of ASAI events</td><td>Domain (station) control (for agent ext that call is leaving)</td></tr>
</table>
</td></tr>
<tr><td>Attendant return call</td><td>If an attendant extends a call to an ACD split or VDN for which the return call timer is not activated, the call does not interact with RONA. The Attendant Return Call Timer is not set if an attendant extends the call to another attendant.</td></tr>
<tr><td>AUDIX Transfer</td><td>RONA applies to a call transferred by AUDIX to an ACD split. A redirected call to AUDIX does not go to split or agent coverage after it is transferred out of AUDIX. If RONA times out on this type of call, the call cannot be redirected.</td></tr>
<tr><td>Automatic answering</td><td>If an agent with automatic answering receives a call with zip tone instead of ringing, RONA timing is canceled.</td></tr>
<tr><td>Call Coverage</td><td>Direct agent calls are redirected to the agent’s coverage path if a path is administered. A temporary bridged call appearance is not maintained for a call directed to an ACD hunt group or VDN, or for a direct agent call.</td></tr>
</table>

Interaction	Description
	When a call is redirected to a split or skill, the Coverage Subsequent Redirection/CFWD No Answer timer is started on the call. Covered calls go to the next point in the split or skill coverage path. If no other point is available to accept the call, the call remains queued or continues to ring the current coverage point. When RONA times out at the coverage point, the following occurs: • RONA does not reset the Subsequent Redirection/CFWD No Answer timer. The timer that expires first controls the call. • If the coverage point for a covered call is a direct agent logical agent ID whose skill has RONA, and if RONA times out first, the call is sent to the next point in the skill coverage path, not to the agent's coverage path. The Subsequent Redirection/CFWD No Answer timer is reset when the call is redirected to the next coverage point. • If RONA was applied to an ACD call that was a previously redirected coverage call (that is, the RONA split was a point in the coverage path), RONA is used to requeue the call as specified for a non covered call. However, the call is not designed to go to split coverage or forwarding. The Subsequent Redirection/CFWD No Answer timer is reset if RONA requeues the call to the RONA split. Both the RONA timer and Subsequent Redirection/CFWD No Answer timer are reapplied. • If RONA applies to an ACD call that was a previously-redirected coverage call (for example, the RONA split was the second point in the coverage path), the call is redirected to the next coverage point in the principal's coverage path if the call cannot be requeued to the RONA split. The Subsequent Redirection/CFWD No Answer timer is reset. • If no other point in the coverage path exists or other points are unavailable, the split-covered call that cannot be requeued or the direct-agent-covered call receives call-cannot-be-redirected handling.
Call Detail Recording (CDR)	When an agent is assigned to be recorded on the CDR record as the called number, the RONA redirected-to answering destination is recorded as the final called number. You can administer CDR to record the VDN, the hunt group, or the answering agent as the called number.
Call Forwarding All	If an adjunct direct agent call is made to an agent's extension that has Call Forwarding All assigned and it is redirected by RONA, the call follows the agent's coverage path. A call forwarded using Call Forwarding to a split or logical agent ID with RONA is sent to the principal's coverage path instead of going to the split's coverage path (if the call cannot be requeued) or to the agent's coverage path (for a direct agent call) on RONA redirection.
Call Pickup	A member of an agent's pickup group can pick up an ACD call that is being timed for RONA. RONA is cancelled.

Interaction	Description
Call Vectoring	RONA applies to vector-controlled ACD splits when calls are queued using the **`queue-to split`**, or **`converse-on split`**, or **`check split`** commands. Also, RONA applies to non vector-controlled and vector-controlled ACD splits when calls are routed to the split using a **`route-to`** or a **`messaging split`** command. Basic Call Vectoring handles an AAS with all agents logged out as unavailable and skips the relevant step. With an **`adjunct routing`** or **`route-to with coverage`** step that routes to a vector-controlled split with all agents logged out, the call is given a busy tone just as when the call cannot queue to a non vector controlled split according to the existing operation. Vector events are generated for a RONA timeout when **`converse-on`** processes a call or results in a RONA redirection failure, and when a vector step is skipped because all AAS agents are logged out. Do not assign vector-controlled splits coverage, forwarding, or night service, because Call Vectoring provides these functions. These functions do not apply to RONA-redirected calls involving vector-controlled splits.
Calling/Called Number Display	A call to a split or skill that RONA redirects is similar to a direct call to the split or skill. If the call goes to coverage, the destination display looks like it does for a normal covered call. An internal or DCS caller to an ACD hunt group or VDN sees displayed the hunt-group or VDN name and extension. This display remains when the call rings an agent. A direct agent call (with EAS) initiated at a phone displays the agent name and logical ID when the call rings the agent station. If the ACD split call or direct agent call goes to coverage, the name remains, but the extension or logical ID portion changes to *cover*. This also happens when RONA redirects a call.
Delay announcements	Delay announcements assigned to non-vector-controlled splits are applied to requeued RONA calls as usual for redirected calls.
Direct Agent Calling	RONA applies to direct agent calls from splits with RONA assigned. RONA timing applies when a direct agent call (from an adjunct or phone) is delivered to and rings an agent with manual answering. Agents are placed in AUX work mode or logged out even if they are the last agent in the split and ACD split calls are queued. Direct agent calls that are queued for an agent remain queued and are not delivered because the agent is unavailable. Don't-answer (DA) coverage continues for the queued calls. If an agent with a coverage path is made unavailable by a RONA time-out on a non-covered direct agent call, the call follows the agent's coverage path. With EAS, the agent's logical extension coverage path for direct agent calls is used. If the agent has no coverage path or if the path is unavailable, the call cannot be redirected and the caller hears previously-provided feedback. If a direct agent call comes from a split that has forwarding or night service, the call is forwarded, precluding RONA timing. If the agent has

Interaction	Description
	forwarding or Send-All-Calls, the direct agent call is forwarded (ACD calls only) or goes to coverage, precluding RONA timing.
Direct Department Calling	RONA applies to DDC-type hunt-group ACD calls.
Home Agent	RONA applies to Home Agent lines that terminate on the IVR Home Agent system as a means to detect port failures. Home Agent lines use Manual Answer and are not present in AAS. Once RONA notification is made, you can correct the failure and restore service manually on the IVR system.
Inbound Call Management (ICM)	RONA applies to ICM-managed calls that ring an agent in an ACD split with RONA assigned.
Message Center/ Server Service	You can assign RONA to Message Center/Server ACD splits.
Multiple Call Handling (MCH)	If an MCH agent has a call active or on hold and the Redirection on No Answer timer expires for another ringing ACD call, the ringing call is redirected to the split or skill or administered VDN. When the call redirects, the agent is not made unavailable, but is placed in the queue of available agents.
Music-on-Hold access - Music on Transferred trunk call	Trunk callers who are transferred to another destination continue to hear administered music (or silence), not ringback, while the call rings. This applies while the transferred call queues to a split. If the trunk call (an ACD call or direct agent call) is transferred to a split with RONA, timeout applies to the call, but the caller continues to hear the previous feedback instead of ringback.
Night Service	When Night Service is activated, calls (including RONA calls) for the hunt group redirect to the night station extension. If the night service split has RONA assigned, RONA timing is reapplied to the redirected call.
Queue status indications	Calls that RONA requeues are counted in the queued calls total. When a RONA call is queued, the call's call-wait time is reset, so RONA does not affect the oldest call waiting (OCW) time.
Queuing	When redirected to a split, RONA timed-out ACD calls in a non vector-controlled split are queued at the highest priority. These calls are distributed before any other calls, except direct agent calls.
Stations	RONA applies to ACD split or direct agent ACD calls that ring at multifunction or hybrid stations with Manual Answering in an ACD hunt group. RONA applies to Off-Premises Station (OPS) lines in an ACD split.
Voice Response Integration (VRI)	You can assign RONA to converse splits. RONA timing applies to calls that a **`converse-on`** command queues and delivers. RONA timing is

Interaction	Description
	canceled if a call is delivered to an agent in another split to whom the system previously tried to queue a call. RONA interacts with a converse split that is an AAS like any other AAS. If RONA must redirect a call to an agent port in a converse split and the queue is full or all AAS agents are logged out, the call is processed by the next vector step while the caller continues to hear the previous vector feedback.

Interactions with other ringing call timers

Several features time the ringing when an ACD call is delivered to an agent. You can use the RONA timer in conjunction with other timers.

Note:

The timer that expires first applies to the call. RONA is canceled if any of the other timers expires first, except in the case of coverage timers.

When a coverage timer expires, RONA timing is canceled only when the call goes to coverage. If RONA times out first, the other timers continue timing or are stopped and may later be reset. RONA interactions with other timers are summarized in the following table.

Table 35: Summary of RONA-Timer interactions

Timer	Description	RONA-timer interaction	
		RONA timeout	Restarted after redirection?
Split DA	Split Call Coverage Don't Answer (non vector-controlled)	Stopped	If re queued or delivered to another agent
Covering DA	Covering Point DA - Subsequent Redirection No Answer	Stopped	If redirects to covering point
Agent DA	Agent DA Coverage (Direct Agent Calls)	Stopped	If covers to direct agent with coverage
NATO	DID/CO Trk No Answer Timeout	Continues	N/A
WAST	Wait Answer Supervision Timer	Continues	If ringing destination or RONA redirection fails

If you want RONA notification and redirection, set the number of rings (or equivalent time) for a RONA timeout to shorter than other timeout periods. DA timers start when a call is placed in queue and continue when the call rings the station. Since RONA starts only when the call is ringing, the RONA interval is usually set to two or three rings, while the DA interval is set to 10 or more rings.

Since queue time is variable, assign a coverage timeout period that is greater than the longest expected queue time plus three or four rings (the time the call could ring the agent).

The NATO timer starts when the call seizes the incoming trunk. The timer could thus be timing before the call is queued by vector processing. Therefore, set the NATO timer to greater than the longest expected time before the call rings the agent (including time before and after being queued) plus three or four rings.

The WAST timer starts when the call rings the agent. Set the RONA timer to a slightly shorter interval (fewer than 10 rings) than the WAST 50-second interval.

Redirection on IP Failure

ROIF description

The Redirection on IP Failure (ROIF) feature applies to agents configured with ACD auto-answer or manual answer and applies to IP hard- or softphones that utilize H.323 IP network connectivity.

For releases prior to 2.1, calls could sometimes be lost when delivered by Communication Manager to auto-answer ACD agents equipped with IP phones. ROIF provides redirection of calls back into queue or to the specified VDN when calls to auto-answer ACD stations cannot be connected due to loss of IP connectivity.

How ROIF works

ROIF works as follows:

- When the system option (Switch Hook Query Response Timeout) is active, ROIF checks IP connectivity before delivering a call to the auto-answer or manual answer agent using an IP phone. Operation with manual answer agents was added an enhancement to ROIF. With manual answer, the timing for the switch hook query starts before the phone is actually rung. The caller hears ringback during this period and during the actual ringing period if there isn’t a connectivity failure.
- Invokes redirection if loss of IP connectivity is detected (the switch hook query is not acknowledged within the administered timeout period). The agent is taken out of service and the call is put back in queue or forwarded to a “Redirect on IP/OPTIM Failure to VDN:-specified VDN on the hunt group screen.

- Prevents a lost call during the period when IP connectivity failure has not yet been detected by Communication Manager maintenance.
- Puts the non Auto-Available Split/Skill (AAS) agent into Aux Work mode, then redirects the call to the split or skill queue or “Redirect on IP/OPTIM Failure to VDN:” -specified VDN if IP connectivity failure is detected while that call is being delivered. If the Reason Codes feature is active, the change to Aux Work is reported with the ROIF-ROOF reason code.
- Logs out the AAS agent instead of putting the agent into AUX Work.

You can specify to retain the active VDN context when a call that queued to the skill defined by the hunt group redirected due to an IP connectivity failure. When you administer the **Retain Active VDN Context?** field as `y`, on the hunt group screen the VDN context from the original active VDN is retained and used while the call is redirected to the specified VDN. If you administer the **Retain Active VDN Context?** field as `n` and ROIF occurs, the system uses the context of the applicable redirect to VDN assigned to the **Redirect on IP/OPTIM Failure to VDN:** hunt group field.

The VDN context includes the following information:

- VDN Name
- Tenant Number (TN)
- VDN of Origin Announcement (VOA) Extension
- VDN Skills (1st, 2nd, 3rd)
- VDN Return Destination

Note:
The VDN Return Destination is set before being RONA/ROIF/ROOF redirected and is not changed by subsequent routing. The staffed agent receiving the redirect call sees "CR" at the right end of the display indicating that this is a RONA/ROIF/ROOF redirected call so that they can provide more appropriate answering treatment.

- VDN Timed ACW (After Call Work) Interval
- BSR (Best Service Routing) Application
- BSR Available Strategy
- BSR Tie Strategy
- Display VDN for Route-to DAC (Direct Agent Calling)
- Trunk ASAI (Adjunct Switch Application Interface) Messages
- BSR Local Treatment
- VDN Variables
- VDN Time Zone Offset

If you choose to retain active VDN Context, you can set up a generic VDN-vector that caters to calls redirected from multiple VDNs. For more information, see Generic VDNs for redirected calls handling on page 336.

If the call is redirected to a VDN when routed directly to a hunt group rather than through a VDN, the redirect to VDN is the active VDN regardless of the setting of the **Retain Active VDN Context?** field.

ROIF considerations

About setting reason codes for ROIF

Set the IP Failure Aux Work reason code to a non zero number not currently being used in order for the system to distinguish between Aux Work changes that have been caused by miscellaneous changes and those caused by a loss of IP connectivity. ROOF uses the same reason code as ROIF.

For more information, see About reason codes on page 327.

Auto-in or manual-in button for ROIF

If an agent has not received a call within the usual timeframe, the agent should press the auto-in or manual-in button to ensure IP connectivity. The lamp update for the Aux Work button is not always received by the IP station due to the loss of IP connectivity.

Call redirection alert button

A **noans-alrt** button can be assigned to non-SIP agent or supervisor phones to indicate that a call has been redirected.

ROIF interactions

The ROIF call and agent interactions are the same as for RONA, with the following additions:

- ROIF is applied system-wide. The default for the system is not active.
- ACD calls delivered from the split or skill queue and Direct Agent Calling (DAC) work the same as with RONA. ROIF first attempts to redirect DAC to the agent's coverage path. If the call cannot go to coverage, the call is redirected to “Redirect on IP/OPTIM Failure to VDN:” if it is assigned to the direct agent skill group. If “Redirect on IP/OPTIM Failure to VDN:” is not assigned, the call is re-queued to the same skill at a high priority. If there are no queue slots available, the caller will hear a busy signal. If all fails, the caller receives

ringback until the system receives a caller disconnect. This also applies to priority direct agent calls when the “Redirect on IP/OPTIM Failure to VDN:” is not specified.

- The agent will not be aware that the line is in Aux Work during an IP connectivity failure. If connectivity is restored during the TCP retry period, the lamp will indicate that the line is in the Aux Work mode.
- The only indication that CMS receives after an ROIF has occurred is a state change and the resultant flow out, flow in, and DFWD-unknown indications for the call. Unlike RONA, the action is not specifically identified, other than by the reason code.
- As with RONA, the caller will hear ringback during the ROIF timer period and calls that are redirected are given ringback when re-queued. Calls that are redirected to a VDN will hear the feedback determined by the assigned vector. If the call cannot be re-queued, because no queue slots are available, the caller will hear a busy signal until the caller abandons the call. In this case, a DFWD-unknown message is sent to CMS to decrease the tracking of call ringing.
- As with RONA, if the **Retain Active VDN Context?** field is administered as `y`, the calls are redirected to the specified VDN with the previous active VDN context information.
- ROIF does not provide a lamp indication to the call center supervisor as is done for RONA.
- ROIF applies to AAS agents/VRU/IVR ports if they are connected through IP, and auto-answer or manual answer is active. AAS lines are logged out if an IP failure is detected during call delivery.

For more information, see RONA interactions on page 339.

Redirection on OPTIM Failure

ROOF description

You can specify whether to retain the active VDN context while redirecting the calls for a VDN. When you administer the **Retain Active VDN Context?** field as `y`, the VDN context from the previous VDN is retained and used while the call is redirected to the specified VDN. If you administer the **Retain Active VDN Context?** field as `n` and redirection occurs, the system uses the context of the applicable redirected to VDN.

The VDN context includes the following information:

- VDN Name
- Tenant Number (TN)
- VDN of Origin Announcement (VOA) Extension
- VDN Skills (1st, 2nd, 3rd)

- VDN Return Destination

Note:
The VDN Return Destination is set before being RONA/ROIF/ROOF redirected and is not changed by subsequent routing. The staffed agent receiving the redirect call sees "CR" at the right end of the display indicating that this is a RONA/ROIF/ROOF redirected call so that they can provide more appropriate answering treatment.

- VDN Timed ACW (After Call Work) Interval
- BSR (Best Service Routing) Application
- BSR Available Strategy
- BSR Tie Strategy
- Display VDN for Route-to DAC (Direct Agent Calling)
- Trunk ASAI (Adjunct Switch Application Interface) Messages
- BSR Local Treatment
- VDN Variables
- VDN Time Zone Offset

If you choose to retain the active VDN Context, you can set up a generic VDN that caters to calls redirected from multiple VDNs. For more information, see Generic VDNs for redirected calls handling on page 336.

If the call is redirected to a VDN when routed directly to a hunt group rather than through a VDN, the redirect to VDN is the active VDN regardless of the setting of the **Retain Active VDN Context?** field.

The Redirection on OPTIM Failure (ROOF) feature applies only to ACD agents using SIP hard- or softphones administered as Off-PBX Telephone Integration and Mobility (OPTIM) endpoints.

As with ROIF, ROOF provides redirection of calls back into queue or to the specified VDN when calls to ACD stations cannot be connected due to loss of IP connectivity and ROOF applies to agents using auto answer or manual answer. ROOF can also retain the active VDN context while redirecting the calls to the specified VDN.

ROOF applies to SIP IP endpoints used as ACD agent stations operating as auto-answer or manual answer. ROOF applies to the SIP interfaced phones. ROIF applies to H.323 IP endpoints, for example, the 4622SW IP telephone.

How ROOF works

ROOF works as follows:

- Before delivering a call to an OPTIM endpoint, the system checks for IP connectivity.
- Invokes RONA if IP connectivity is not acknowledged. The agent is placed in AUX work mode and the call is put back in queue or forwarded to a RONA-specified VDN.
- Prevents a lost call during the period when IP connectivity failure has not been detected by Communication Manager maintenance.

You can specify whether to retain the active VDN context while redirecting the calls for a VDN. When you administer the **Retain Active VDN Context?** field as `y`, the VDN context from the previous VDN is retained and used while the call is redirected to the specified VDN. If you administer the **Retain Active VDN Context?** field as `n` and redirection occurs, the system uses the context of the applicable redirected to VDN.

The VDN context includes the following information:

- VDN Name
- Tenant Number (TN)
- VDN of Origin Announcement (VOA) Extension
- VDN Skills (1st, 2nd, 3rd)
- VDN Return Destination

  Note:
  The VDN Return Destination is set before being RONA/ROIF/ROOF redirected and is not changed by subsequent routing. The staffed agent receiving the redirect call sees "CR" at the right end of the display indicating that this is a RONA/ROIF/ROOF redirected call so that they can provide more appropriate answering treatment.

- VDN Timed ACW (After Call Work) Interval
- BSR (Best Service Routing) Application
- BSR Available Strategy
- BSR Tie Strategy
- Display VDN for Route-to DAC (Direct Agent Calling)
- Trunk ASAI (Adjunct Switch Application Interface) Messages
- BSR Local Treatment
- VDN Variables
- VDN Time Zone Offset

If you choose to retain the active VDN Context, you can set up a generic VDN that caters to calls redirected from multiple VDNs. For more information, see Generic VDNs for redirected calls handling on page 336.

If the call is redirected to a VDN when routed directly to a hunt group rather than through a VDN, the redirect to VDN is the active VDN regardless of the setting of the **Retain Active VDN Context?** field.

ROOF considerations

About setting reason codes for ROOF

Set the IP Failure Aux Work reason code to a non zero number not currently being used in order for the system to distinguish between Aux Work changes that have been caused by miscellaneous changes and those caused by a loss of IP connectivity. ROOF uses the same reason code as ROIF.

For more information, see .

Auto-in or manual-in button for ROOF

If an agent has not received a call within the usual time frame, it is possible that a network disruption has occurred, causing the communication server to change the agent's state to Aux Work. The agent should press the auto-in or manual-in button to ensure IP connectivity. The lamp update for the Aux Work button may not be received if the SIP station has lost IP connectivity.

ROOF interactions

The ROOF call and agent interactions are the same as for RONA, with the following additions:

- ROOF is applied system-wide. The default for the system is *active*.
- ACD calls delivered from the split or skill queue and Direct Agent Calling (DAC) work the same as with RONA. ROOF first attempts to redirect DAC to the agent's coverage path. If the call cannot go to coverage, the call is redirected to “Redirect on IP/OPTIM Failure to VDN” if it is assigned to the direct agent skill group. If “Redirect on IP/OPTIM Failure to VDN” is not assigned, the call is re-queued to the same skill at a high priority. If there are no queue slots available, the caller will hear a busy signal. If all fails, the caller receives ringback until the system receives a caller disconnect. This also applies to priority direct agent calls when the “ “Redirect on IP/OPTIM Failure to VDN” is not specified.

- The agent will not be aware that the line is in Aux Work during an IP connectivity failure. If connectivity is restored during the TCP retry period, the lamp will indicate that the line is in the Aux Work mode.
- The only indication that CMS receives after a ROOF has occurred is a state change and the resultant flow out, flow in, and DFWD-unknown indications for the call. Unlike RONA, the action is not specifically identified, other than the reason code.
- As with RONA, the caller hears ringback during the timer period and calls that are redirected are given ringback when re-queued. Calls that are redirected to a VDN will hear the feedback determined by the assigned vector. If the call cannot be re-queued, because no queue slots are available, the caller will hear a busy signal until the caller abandons the call. In this case, a DFWD-unknown message is sent to CMS to decrease the tracking of call ringing.
- ROOF does not provide a lamp indication to the call center supervisor as is done for RONA.

For more information, see RONA interactions on page 339.

Remote Logout of Agent

Description of Remote Logout of Agent

The Remote Logout of Agent feature allows a user to logout an idle ACD or EAS agent without being physically present at the agent's station. The user who is logging out the agent can be locally or remotely located.

The Remote Logout of Agent is similar to the Add/Remove Skills feature.

Reasons to use Remote Logout of Agent

If an agent walks away from his or her station without logging out, ACD calls are sent to the station without being serviced. Without the Remote Logout of Agent feature, supervisors or other agents had to walk over to the agent's station, enter the logout FAC to change the agent's work mode or log out. Customers could also busy-out the station from the switch room or use the RONA feature to put the station in aux work mode without logging out the agent.

The Remote Logout of Agent feature makes it simpler to logout the agent from the user's station.

Prerequisites for using Remote Logout of Agent

The Remote Logout of Agent feature can only be used if user permissions are administered appropriately for the person or VDN attempting to use the feature. The communication server administrator must ensure that the appropriate users have permissions administered so that they can use this feature.

The feature user must:

- Be in the same Tenant Partition as the agent as set on the Tenant screen
- Have Remote Logout COR permissions set on the Class of Restriction screen
- Have console permissions set on the COS screen for local users
- Have a Call Center release set to 9.1 or later set on the System Parameters Customer-Options screen by the RFA license file

For a description of how to check these values, see Administrator Guide for Communication Manager.

Locally logging out an agent

To log out an agent from a local site:

1. Use a local station assigned with the COS and COR to logout an agent locally within the communication server.
2. Enter the FAC that was established to activate this feature followed by the agent's login ID or physical station extension.
    - You must be in the same Tenant Partition as the agent.
    - The physical extension is used only in non-EAS systems.

Administration for Remote Logout of Agent using a VDN

Many call centers are geographically dispersed, but the reporting and tracking of agent activity takes place from one main location. Or, agents can log into the system remotely and take calls using the Home Agent capability.

With the Remote Logout of Agent feature, a vector is administered with a "route-to number "step that contains the FAC for remote logout. The FAC can be followed by the agent's loginID or the supervisor can dial the loginID after the VDN with an appropriately programmed vector.

Note that Service Observing and remote logout are the only features that can use a VDN vector in this manner.

If the user is remote and calling into the communication server through a trunk, the user may reach the activation VDN to logout the agent. In this case, the entire FAC-with-EXT is required in the vector. Or, the vector can prompt the user to enter the extension and then route-to digits. An activation vector can also prompt the user for a password for additional security. Note that a remote trunk user might not hear confirmation tone (it varies with trunk type and trunk administration) and the user's phone will continue to hear feedback until the user hangs up.

To set up this capability, the communication server administrator must create an activation VDN and set the incoming destination of a CO trunk or DID dialed number to be the activation VDN. The VDN to which the call terminates must be assigned a COR and a TN that include the appropriate settings for use of the Remote Logout of Agent feature. COS assigned with console permissions is not required. In this example, the activation VDN includes the following vector.

```
01 wait time 0 seconds hearing ringing
02 collect 5 digits announcement 3501 ("enter password")
03 goto step 5 if digits = 39744
04 disconnect after announcement 3502 ("bye")
05 collect 1 digits announcement 3503 ("enter 1 to logout agent 89923, 2 to logout
agent  89924...")
Note: The names of the agents can also be requested in the appropriate switch-setup.
06 route-to number *6389923 with cov n if digit = 1
07 route-to number *6389924 with cov n if digit = 2)
08 goto step 5 if unconditionally
```

Note:

In this example, *63 is the FAC assigned for Remote Logout of Agent. This example is one of many ways in which the vector can be written to activate the VDN.

Remotely logging out an agent using the assigned VDN

Note:

This procedure assumes you have used the example in Administration for Remote Logout of Agent using a VDN on page 354.

To log out an agent from a remote location using the assigned VDN:

1. Dial into the communication server from an outside line that reaches the activation VDN.
2. Enter the password as programmed in the vector.

   See Step 2 in the vector example described in Administration for Remote Logout of Agent using a VDN on page 354.

3. Enter the physical or logical agent extension or the digit corresponding to the desired agent you want logged out.
4. Enter `1`.

   The login ID associated with that prompt is Agent A's login ID or name.

Remote Logout of Agent failures

In addition to failures that occur due to permission requirements, see Prerequisites for using Remote Logout of Agent on page 354. The logout fails if a Remote Logout of Agent is attempted for an agent who is on an ACD call, has an ACD call on hold, or who is not logged in.

Remote Logout of Agent interactions

Interaction	Description
Auto-Available Split/Skill	If an agent login ID is assigned to an Auto-Available split or skill, then the Remote Logout of Agent feature cannot be used to log the agent out. RONA can be used to automatically logout a port that is not answering calls.
AUDIX	If an agent is a member of an AUDIX hunt group and has no other splits/skills assigned to the agent login ID, then the Remote Logout of Agent feature will not successfully log out the agent, even though the user attempting the logout hears a confirmation tone.
Non-ACD hunt groups	If an agent is a member of ACD splits/skills and is using a physical extension that is a member of a non-ACD hunt group, then use of the Remote Logout of Agent feature will log the agent out of the splits/skills but allow the agent to continue receiving non-ACD calls.
Non-EAS agent operation	A non-EAS agent is logged out of all splits even while active on an ACD call. This call is not dropped, but all Call Center reporting of the call is stopped.
Timed ACW	If an agent answers an ACD call for a hunt group with Timed After Call Work administered and then hangs up the call, the

Interaction	Description
	Remote Logout of Agent feature can be used to log out the agent during the ACW time.
Service Observing	An agent can be logged out using the Remote Logout of Agent feature while being service observed.

Representation of trunk locations

When calls are reported by Communication Manager to the reporting adjuncts, the trunk equipment location of the trunk the call is received on is sent to the reporting adjunct for use in trunk reports. This section discusses how the different types of trunk locations are identified on Communication Manager and how they are represented by the reporting adjunct as a trunk equipment location.

Display of physical port locations

Physical port network trunk locations that terminate on the G650 gateway are represented by a cabinet, carrier, slot and circuit format. For example a location can be 1, A, 1, 22. This equipment location will show as 01A01022 on CMS.

Port IDs for physical circuit-switched trunks that terminate on an H.248 Media Gateway such as the G700 or G450, are displayed in a different format than that used for normal port network equipment locations such as the G650. This difference is also reflected in the way those port IDs are listed in the reporting adjunct.

The following table compares the standard trunk equipment location format for port network port IDs to the H.248 Media Gateways format.

Regular port network trunk equipment location format like the G650	Media Gateway-terminated trunk equipment location format
bbXssccc where: bb = cabinet (1-64) X = carrier (A-E) SS = slot (1-25) CCC = circuit (1-256)	gggVscc where: ggg = gateway (replaces cabinet) (1-250) V = indicates H.248 media gateway (replaces carrier) s = slot (1-4) cc = circuit (1-32)

In reports, media gateway-terminated trunk equipment locations are displayed in a slightly different format than that used for port network gateways.

Depending on the reporting adjunct version, the trunk equipment location is shown in a fixed 8-character or 9-character format according to the following rules:

Gateway number: The gateway number will show as either 2-digits (the leading digit shown in the communication server display is dropped by CMS) or 3 digits.This variation in the gateway number format is the result of the following factors:

- The numerical designation assigned to a gateway can be any number from 1 to 250, but earlier Avaya communication server releases only supported up to 99.
- R3V11 CMS and CMS Supervisor do not provide a third-digit space for the gateway numbers.

Therefore, for R3V11 installations, when the gateway numbers that are greater than 99, the leading (hundred) digit of the gateway number is shown as the leading (tens) digit in the slot number and the following rules apply to display of slot numbers for H.248 Media Gateway-terminated trunk equipment on R3V11 versions of CMS and CMS Supervisor:

- For gateways 1-99, slot numbers range from 01 to 04.
- For gateways 100-199, slot numbers range from 11 to 14.
- For gateways 200-250, slot numbers range from 21 to 24.

Carrier number: The carrier number shows as the letter *V* followed by two digits (01-04) for the slot number.

Circuit number: The circuit number shows as 3 digits (001-032).

The following table shows how H.248 Media Gateway-terminated trunk equipment location formats are listed on Avaya communication servers and reporting adjuncts.

Table 36: H.248 Media Gateway port ID representations on communication servers and reporting adjuncts

Avaya communication server	On CMS (pre-R3V11ag) and CMS Supervisor (pre-R3V11FJ.04)	On CMS (R3V11ag or later) and CMS Supervisor (R3V11FJ.04 or later)	On CMS (R3V12 or later), CMS Supervisor (R3V12 or later), and Avaya IQ
Example 1: gateway=12, slot number=2, circuit number=16			
012V216	12702016	12V02016	012V02016
Example 2: gateway=130, slot number=2, circuit number=16			
130V216	30702016	30V12016	130V02016
Example 3: gateway=240, slot number=2, circuit number=16			
240V216	40722016	40V22016	240V02016

Display of IP or SIP trunk locations

IP or SIP trunk members are defined using a virtual trunk equipment location number that is sent from the communication server to the reporting adjunct for use in trunk reports. The virtual port numbers are defined in Communication Manager during administration using the letter “T” followed by 5 digits with leading 0's. The first virtual port is defined by T00001 and ranges to the maximum port location of T24000.

When trunk information is sent to the reporting adjunct, the "T" virtual port number is mapped into a 9-digit number with leading zeros using the bbbXssccc format (the same as the G650 ports but with the carrier always set to 0 to indicate that the equipment location is a virtual port and an extra 0 ["b"] in the "cabinet" field). To display IP trunk member port-IDs to the reporting adjunct as virtual trunk equipment location numbers, a 9-digit number starting with leading zeros is used.

For example, an IP trunk member with a port-id of T00001 is sent to CMS/IQ as "000 0 00 001" (cabinet, carrier, slot, circuit), and is displayed on CMS/IQ as 000000001. An IP/SIP trunk member with port-id of T00400 is sent to CMS/IQ as "000 0 00 400" and displayed on the CMS reporting adjunct as "000000400." Due to the conversion process used by the software, the numbering above Txxx500 for each 1,000 port IDs are mapped to numbering 000 to 499 with an increment to the next 1000 in the series (a modulo 500 function). For example, T00500 is sent as "000 0 01 000" and the range of T00501 to T00999 is displayed on CMS/IQ as 000001001 to 000001499.

Use the following table to correlate IP trunk member port IDs on the communication server and the reporting adjunct.

Communication server representation	Reporting adjunct representation
T00001 through T00499	000 0 00 001 through 000 0 00 499
T00500 through T00999	000 0 01 000 through 000 0 01 499
T01000 through T01499	000 0 02 000 through 000 0 02 499
T01500 through T01999	000 0 03 000 through 000 0 03 499
T02000 through T02499	000 0 04 000 through 000 0 04 499
T02500 through T02999	000 0 05 000 through 000 0 20 499
T03000 through T03499	000 0 06 000 through 000 0 06 499
T03500 through T03999	000 0 07 000 through 000 0 07 499
T04000 through T04499	000 0 08 000 through 000 0 08 499
T04500 through T04999	000 0 09 000 through 000 0 09 499
T05000 through T05499	000 0 10 000 through 000 0 10 499
T05500 through T05999	000 0 11 000 through 000 0 11 499
T06000 through T06499	000 0 12 000 through 000 0 12 499
T06500 through T06999	000 0 13 000 through 000 0 13 499
T07000 through T07499	000 0 14 000 through 000 0 14 499
T07500 through T07999	000 0 15 000 through 000 0 15 499
T08000 through T08499	000 0 16 000 through 000 0 16 499

Communication server representation	Reporting adjunct representation
T08500 through T08999	000 0 17 000 through 000 0 17 499
T09000 through T09499	000 0 18 000 through 000 0 18 499
T09500 through T09999	000 0 19 000 through 000 0 19 499
T10000 through T10499	000 0 20 000 through 000 0 20 499
T10500 through T10999	000 0 21 000 through 000 0 21 499
T11000 through T11499	000 0 22 000 through 000 0 22 499
T11500 through T11999	000 0 23 000 through 000 0 23 499
T12000 through T12499	001 0 00 000 through 001 0 00 499
T12500 through T12999	001 0 01 000 through 001 0 01 499
T13000 through T13499	001 0 02 000 through 001 0 02 499
T13500 through T13999	001 0 03 000 through 001 0 03 499
T14000 through T14499	001 0 04 000 through 001 0 04 499
T14500 through T14999	001 0 05 000 through 001 0 05 499
T15000 through T15499	001 0 06 000 through 001 0 06 499
T15500 through T15999	001 0 07 000 through 001 0 07 499
T16000 through T16499	001 0 08 000 through 001 0 08 499
T16500 through T16999	001 0 09 000 through 001 0 09 499
T17000 through T17499	001 0 10 000 through 001 0 10 499
T17500 through T17999	001 0 11 000 through 001 0 11 499
T18000 through T18499	001 0 12 000 through 001 0 02 499
T18500 through T18999	001 0 13 000 through 001 0 13 499
T19000 through T19499	001 0 14 000 through 001 0 14 499
T19500 through T19999	001 0 15 000 through 001 0 15 499
T20000 through T20499	001 0 16 000 through 001 0 16 499
T20500 through T20999	001 0 17 000 through 001 0 17 499
T21000 through T21499	001 0 18 000 through 001 0 18 499
T21500 through T21999	001 0 19 000 through 001 0 19 499
T22000 through T22499	001 0 20 000 through 001 0 20 499
T22500 through T22999	001 0 21 000 through 001 0 21 499
T23000 through T23499	001 0 22 000 through 001 0 22 499
T23500 through T23999	001 0 23 000 through 001 0 23 499

Communication server representation	Reporting adjunct representation
T24000	002 0 00 000

Service Level Maximizer

Service Level Maximizer

Service Level Maximizer (SLM) is an optional Communication Manager Call Routing feature introduced in Release 2.0 that is used with Expert Agent Selection (EAS), and without Avaya Business Advocate.

SLM ensures that a defined service level of *X*% of calls are answered in *Y* seconds. When SLM is active, the software verifies that inbound calls are matched with agents in a way that makes sure that the administered service level is met.

This section includes the following topics:

Service Level Maximizer

Service Level Maximizer (SLM) is an optional Communication Manager Call Vectoring feature that is used with Expert Agent Selection (EAS), and without Avaya Business Advocate.

SLM ensures that a defined service level of X% of calls are answered in Y seconds. When SLM is active, the software verifies that inbound calls are matched with agents in a way that makes sure that the administered service level is met.

For more information, see Service Level Maximizer on page 361.

Auto-reserve agents

Auto-reserve is an added feature that can be used to ensure that the service level is met in critical skills. When a critical skill is not meeting its service level, auto-reserve puts agents in standby for their other skills to ensure that there is an available agent when the next call arrives for the critical skill. When an agent becomes available, all of his or her assigned skills are checked to see if any auto-reserve skills are not meeting their target service level. If so, the agent is made available only in those skills.

For more information, see Auto reserve agents on page 365.

MAO

When SLM is used, an optional feature called Maximum Agent Occupancy (MAO) can be used to set thresholds on the amount of time an agent spends on a call. MAO is used to avoid agent burnout.

For more information, see Maximum Agent Occupancy under the chapter ACD Call Center features.

SLM requirements

SLM works on all platforms and operating systems that are supported by an Avaya Communication Manager. SLM has the following licensing and system requirements:

- The Call Center Elite package.
- The Call Center Release field on the System-Parameters Customer-Options screen must be set to 12.0 or later.
- To obtain CMS reports that include information related to SLM, you must use CMS Release 12 or later. For more information about how to use CMS reports to evaluate SLM operations, see SLM reporting on page 368.
- SLM and Avaya Business Advocate cannot be simultaneously enabled on the System-Parameters Customer-Options screen. Therefore, SLM and Advocate can not both be used on the same system. Avaya Business Advocate provides a more flexible and functional form of achieving service level targets.

SLM operations

SLM agent selection

Agent selection methods are used when a call is queued to a skill and two or more agents are available to take the call. This is known as an agent surplus condition.

The algorithm for agent selection:

- Selects the agent with only one skill first.
- For agents with more than one skill, the agent's skills that are *not* needed for the incoming call are reviewed and the agent with skills that are less likely to be needed for future calls is selected. This is another way of saying that the algorithm selects the agent with the lowest opportunity cost.
- In a tie, the most idle agent is selected.

For details, see SLM target service levels and agent opportunity costs on page 363.

The agent selection algorithm does not consider the agent's skill level, or agent occupancy.

SLM call selection

Call selection applies when an agent becomes available and there are calls waiting in queue in two or more of the agent's skills. This is known as a call surplus condition.

Call selection compares all of the agent's assigned skills that have a call in queue and picks the one with the lowest ART value. If any ART values are negative, the call in the skill with the highest negative ART value is chosen.

For more information about ART values, see SLM target service levels and agent opportunity costs on page 363.

Note:

The SLM call selection method is applied to agents having at least one skill administered as “slm”.

The call selection algorithm does not consider an agent’s skill level or call priorities.

SLM target service levels and agent opportunity costs

The SLM agent selection method is based on user-defined *target service levels* for SLM-administered skills and the concept of *agent opportunity costs*.

Target service level: You define specific target service level goals for each SLM skill based on the following format:

`SLM target service level = x percent calls answered in y seconds`

For purposes of SLM reporting, estimates of service level compliance for a skill are expressed as the Actual service level Relative to the Target service level (ART). At any point in time, an SLM skill can be below, equal to, or above its specified target service level. For example, if a skill has a target service level of 80% of all calls answered within 20 seconds and the current service level is 75% of all calls to the skill answered within 20 seconds, then the current ART value is -5%. Alternately, if the current service level indicates that 90% of all calls are being answered within 20 seconds, then the current ART value is +10%.

For information about how to administer service target levels for a skill, see SLM administration in *Administering Avaya Aura™ Call Center Features* and for information about evaluation of skill service level data, see Evaluating target service level compliance on page 368.

Opportunity costs: SLM compares actual call service levels to target service levels for each SLM skill, so that when an incoming call arrives at a skill, service level data can be used as the basis to develop agent *opportunity cost* estimates. The opportunity cost for an agent at a given point in time is represented as a weighted estimate that considers the status of the agents skills relative to the target service levels of each skill.

The process that SLM uses to derive agent opportunity cost estimates can be summarized as follows:

- An incoming call arrives for an SLM skill and agents that are both assigned to that skill and currently available are identified.
- All skills to which the available agents are assigned are also identified. For each of the assigned skills (excluding the skill associated with the incoming call), a current service level estimate is calculated and compared to the target service level.

  Note:
  The opportunity costs for a single-skill agent is always equal to zero, since they can always be selected for an incoming call in their assigned skill with no impact on the service level status of any other skills.
- Based on the current overall service level for the skills of each available agent, SLM derives a weighted estimate that identifies which of the available agents is currently the least needed for their other assigned skills, where the *need* of a skill is (approximately) defined as the difference between the current service level and the target service level. This agent has the lowest overall opportunity cost.

Because of the way that SLM estimates agent opportunity costs in the agent selection process, available agents whose skills are currently closest to matching their specified target service levels are selected first, while agents whose skills are furthest from matching their specified target service level are selected last. This strategy maximizes the possibility that an agent will be available when a call arrives at a skill whose target service level is at risk.

For example, consider a simplified scenario in which agents A and B, are both assigned to Skill 4 as well as two other skills. When an incoming call arrives at Skill 4 and both agents are available, SLM compares the current service level to the target service level for each of the skills to which the agents are assigned. The agent who currently has the lowest opportunity cost is identified and selected to receive the incoming call in Skill 4.

The following table shows how the agent with the lowest opportunity costs is selected in two different call service level scenarios:

Note:
To simplify this example, the service level states for each skill are represented as ART values. The actual agent selection algorithms used by SLM are complex and do not rely directly on ART data.

Skill Assignments	SLM Skill 1	SLM Skill 2	SLM Skill 3	Skill 4[19] (Incoming Call)
Agent A	X	X		X
Agent B		X	X	X
	For Skill 1, if...	For Skill 2, if...	For Skill 3, if...	Then...

[19] SLM agent opportunity cost estimates do not include service level data of the receiving skill for the incoming call.

Skill Assignments	SLM Skill 1	SLM Skill 2	SLM Skill 3	Skill 4[19] (Incoming Call)
				Agent with lowest opportunity cost for incoming call is:
Scenario 1	ART[20] = -5%	ART = +2%	ART = +2%	Agent B
Scenario 2	ART = -1%	ART = +5%	ART= -6%	Agent A

In scenario 1 in this table, Agent B has the lowest opportunity cost compared to Agent A because the skills other than skill 4 assigned to Agent B (skills 2 and 3) are both above target service level. At the same time, of Agent A's skills (skill 1 and skill 2), skill 1 is below target. Agent A is selected for skill 1. Therefore, of Agents A and B, it is better to select Agent B for the incoming call to handle skill 4.

In scenario 2, of Agent B's other skills (2 and 3) skill 2 is above target level but skill 3 is below target by 6%. At the same time, of Agent A's other skills (1 and 2), skill 1 is only below target by 1%. Therefore, in this scenario, Agent A has the lowest opportunity cost compared to Agent B, since Agent B has a skill in worse shape than Agent A.

SLM benefits

Because SLM is able to differentiate skills in terms of their current call service demands, it provides the following advantages over other agent selection methods:

- Since agent resource needs for each skill are assessed in real-time, you can use SLM to allocate agent resources to those skills that have the greatest call service demand in a dynamic manner, thereby reducing overall call response times.
- Potential problems associated with staffing exceptions, or fluctuating, intra-day call service demands are also reduced.
- SLM is especially useful for call center operations that are bound by contract or other legal obligation to meet specific service level requirements.

Auto reserve agents

This section includes the following topics:

For information about administration of the auto reserve option, see SLM administration in *Administering Avaya Aura™ Call Center Features*. For information about how to use CMS reports to evaluate auto reserve operations, see Evaluating auto reserve rates on page 369.

---

[20] ART = Actual service level relative to Target service level, where the service level is defined as x% calls answered in y seconds. For more information, see Evaluating target service level compliance on page 368.

About auto reserve agents

Auto-reserve is an added feature you can use to ensure that the service level is met in critical skills. When a critical skill is not meeting its service level, auto-reserve puts agents in standby for their other skills to ensure there is an available agent when the next call arrives for the critical skill. When an agent becomes available, all of his or her assigned skills are checked to see if any auto-reserve skills are not meeting their target service level. If so, the agent is made available only in those skills.

How auto reserve works

SLM also allows you to specify auto reserve agents for a skill to ensure that the desired service level is met in critical skills. When an agent becomes available, the agent can be reserved for SLM skills that have a weighted service level below their assigned targets. When the agent is reserved in one or more of his or her assigned skills, that agent is made available to receive calls only from those skills.

With SLM, an agent becomes reserved for an SLM skill, which has a Group Type of slm, when the agent becomes available. At that time, the SLM software checks all the agent's assigned skills to determine if any have a weighted service level below the target service level. Before the agent is automatically reserved for one or more of those skills:

- The skills must have a maximum auto-reserve setting of greater than 0 - as set on the Hunt Group screen for the skill
- The limit of reserved agents has not been exceeded for the skill

In other words, the agent is only available in that skill. The agent is made unavailable in the above-target skills, thus reserving each agent for the neediest skills.

Considerations for allocating auto reserve agents

Since auto reserve agents are kept unavailable in other skills, auto reserve agents should only be used in skills for which achievement of service level targets is considered to be critical. The addition of even a single auto reserve agent to a skill can have a significant impact on the service level that is realized. Therefore, Avaya recommends that you initially set the number of auto reserve agents on the Hunt Group screen to 0 or 1, observe the impact on the service level, and if necessary, gradually increase the number of auto reserve agents by increments of one at a time until you have determined that your service level goals are reliably achieved.

Rules for auto reserve agents

For agents that are assigned to any skills that use the auto reserve option, the following rules apply when an agent becomes available:

- If any of the auto reserve-enabled skills to which an agent is assigned are currently below their specified target service level, the agent is available only in those skills.
- The designation of auto reserve agents for a skill is continuously assessed as agents become available. If the maximum number of auto reserve agents has already been

reached, a single-skill agent who becomes available replaces the multi-skilled agent who has the highest opportunity cost.

- If one or more of an agent's auto reserve-enabled skills are currently below his specified target service level, a *multi-skill* agent is put into the auto reserve state if one of the following conditions are met:
    - The maximum number of auto reserve agents for the skill is not yet filled.
    - The maximum number of auto reserve agents for the skill *is* filled, but the opportunity cost for an idle, multi-skilled agent is lower than the opportunity cost of a multi-skilled agent who is currently in the auto reserve state. In this case, the agent with the highest opportunity cost is released from the auto reserve state.

Agent selection rules in mixed skill environments

SLM skills can be co-resident on the system with skills that use other agent selection methods, such as LOA. However, situations may arise in which a skill is not administered as an SLM skill, but includes agents that are also assigned to one or more SLM skills. In such a mixed skill environment, the following rules apply:

- If a non-SLM administered skill does not include any agents who are also assigned to SLM skills, then agent selection is based on the agent selection method that is administered for that skill.
- If a non-SLM administered skill includes one or more agents who are also assigned to SLM skills, a current service level value of 100% is applied to the non-SLM skill for purposes of SLM service level and agent opportunity cost calculations.

Important:

In a mixed skill environment, the service level for non-SLM hunt groups should be administered so that it reflects the importance of the hunt group to your business. For example, if it is permissible for inbound callers to wait for longer amounts of time, you might set the service level to be 75% (of calls answered) in 180 seconds. In other cases, when an extended wait time is not expected, but target service level compliance is not critical, you might set the service level to be 45% in 15 seconds.

SLM algorithms

Starting with Communication Manager Release 3.1.2 (load 632), you can choose an alternative algorithm for selecting agents and delivering calls to maximize service level targets. The original Weighted Service Level (WSL) algorithm used for maintaining the service level targets has been changed to an Actual Service Level (ASL) algorithm that works better with low staff or low traffic level conditions. The ASL algorithm also handles the higher staff and traffic conditions and is recommended. ASL is determined as a percentage on a hunt group basis using the number of *accepted* calls in the current interval divided by the total calls in the current interval. A call is counted as *accepted* if it is answered within the target service level time period. You can still select the WSL algorithm on a system basis when desired. The WSL algorithm is based on a weighting calculation that uses the difference between the target time and the estimated wait time.

Criteria for choosing algorithms

The ASL algorithm is an improved algorithm for maintaining service level targets. Use ASL for most situations, in addition to low staff levels or low traffic level conditions unless actual experience indicates that WSL provides better performance for your installation.

Administering the ASL algorithm

To use the ASL algorithm:

1. From the System Administration Terminal (SAT), enter **`display system-parameters customer options`**.
2. On the System Parameters Customer-Options screen, make sure the**Service Level Maximizer** field is set to `y`.
3. Enter **`change system-parameters features`**.
4. On the Feature-Related System Parameters screen, make sure the**Service Level Maximizer Algorithm** field is set to `actual`.
5. Enter **`change hunt-group`**.
6. On the Hunt Group screen, set the**SLM Count Abandoned Calls** field to`n` or`y` (the default), depending on whether or not you want to include abandoned calls in the ASL algorithm calculations for SLM.
7. On the Hunt Group screen, set the**Service Level Interval** field to the time interval when you want ASL calculations to run.

   The default value is daily. The other choices arehourly or weekly.

SLM reporting

This section provides an overview of new Avaya CMS Supervisor report features that allow you to evaluate various aspects of SLM performance.

For detailed information about:

- CMS database items that are related to SLM or MAO, see Avaya CMS Database Items and Calculations
- ART reports, see the Avaya CMS Supervisor online help

This section includes the following topics:

Evaluating target service level compliance

CMS includes database items that you can use in CMS Supervisor to evaluate how well your target service levels are met by SLM operations.

Note:

The service level used by the Communication Manager to route calls is based on a prediction of a call being answered in the target service level. The service level calculated by CMS is the actual service level being achieved.

ART reports

Supervisor provides several types of Actual Relative to Target (ART) reports that compares actual service levels to target service levels and expresses the difference on a percent basis in a graphical format.

Note:

If your service level targets are based on contractual agreements, verify that your assessment of service level performance is based on a time frame (days, weeks, months) that is appropriate for the terms of your contract.

A percent value that exceeds zero means that actual service levels exceed the target, while percent values less than zero mean that the service level is not being achieved. When actual and target service levels correspond closely, the percent difference between the two data sets that are displayed in ART reports will tend to be close to zero, which is an indication that staffing levels are consistent with call service goals.

Service level calculations

Service level calculations can also be used to evaluate service level compliance. In R12 new database items have been added to track the number of calls answered (TARGETACDCALLS), abandoned (TARGETABNS) and outflowed (TARGETOUTFLOWS) within the service level administered on the Communication Manager.

CMS uses the target service level that is administered on the Communication Manager to generate these items. The advantage to using these items is that if the target service level is changed, CMS receives the new service level value and automatically adjusts how these items are computed. These items can be included in custom reports.

Note:

The existing CMS service level calculation can be used only if the acceptable service level on CMS Split/Skill Call Profile matches the Target Service Level administered on the Communication Manager. If the target service level is modified on the Communication Manager, the CMS service level must be manually modified to match that value.

Evaluating auto reserve rates

Avaya CMS Supervisor includes a %Skills Available column in historical Agent Summary Reports. The %Skills Available value is 100% when an agent spends no time in the auto reserve state. All values less than 100% indicate agent time spent in the auto reserve state.

SLM feature interactions

Before you use SLM, you should understand the feature interactions described below.

Interaction	Description
Avaya Business Advocate	SLM and Avaya Business Advocate cannot both be enabled on the System-Parameters Customer-Options screen.
BCMS Reporting Desktop VuStats	If BCMS Reporting Desktop VuStats is used to display acceptable service level report data, the displayed value is identical to the seconds value that is set in the **Target Service Level (% in sec)** field on the Hunt Group screen. For more information about administration of SLM skills, see SLM administration in *Administering Avaya Aura™ Call Center Features*.
Best Service Routing	With BSR, the best resource choice (among the local skills and best skills of the remote sites) is based on the lowest adjusted EWT or assigned available agent strategy rule. This rule does *not* consider service level targets that may be assigned to individual skills. However, when an SLM skill is selected as the best resource, the available agent selection *is* based on the specified service level target for the skill. Therefore, service level objectives are maintained within the local or remote skills but not across sites.
Direct agent calls	For agents assigned to SLM skills and eligible to receive direct agent calls, direct agent calls have priority over ACD calls.
Least Occupied Agent	SLM does not use LOA as an agent selection method.
Location Preference Distribution	You can assign reserve agents using SLM. In most cases, the selection of an agent or a call based on Location Preference Distribution takes precedence over SLM. However, SLM takes precedence when a reserve agent is needed because the service level is below the threshold. **Note:** If more than one reserve agent is eligible for the call, Location Preference Distribution is used to choose the agent.
Non-SLM Skills	Agents that have at least one assigned SLM skill will have their administered call handling preference (CHP) ignored and will be treated as if their call handling preference is set to “slm”. The non-SLM skills will be treated as if they are always at service level when it comes to agent and call selection. For more information, see Agent selection rules in mixed skill environments on page 367.
Greatest Need	Greatest Need is not used when SLM is enabled, since call selection is driven by the target call service levels that are administered for each SLM skill.
RONA	Redirected calls are considered in the service level calculations of any SLM skill to which they are sent.

Service Observing

About Service Observing

Service Observing allows a specified user, such as a supervisor, to observe or monitor another user's calls. In this section, observer refers to the supervisor who is observing calls. Agent refers to the extension, attendant, or logical agent being observed. A vector directory number (VDN) call can also be observed. Observers can observe in listen-only or listen-and-talk mode.

Note that you set up Service Observing to observe a particular extension, not all calls to all extensions at a station.

Service Observing may be subject to federal, state, or local laws, rules, or regulations or require the consent of one or both of the call parties. Familiarize yourself and comply with all applicable laws, rules, and regulations before using this feature.

Service Observing detailed description

About using Service Observing

To begin observing, the observer presses the Service Observing button plus the agent's extension number. Initially, the observer is in listen-only mode. The observer presses the Service Observing button to toggle between listen-only and listen/talk mode. The lamp indicates which mode the observer is in.

To deactivate Service Observing, the observer hangs up, selects another call appearance, or presses the Release button.

An observer can observe an agent who is not active on a call. The observer is in the wait state until the agent receives a call, then the observer is bridged onto the call.

You can administer a warning tone on each system to let agents and callers know when someone is observing a call. Before connection, the conference tone may add 2-3 seconds delay if enabled. The parties hear a 2-second, 440-Hz warning tone before an observer connects to a call, followed by a half-second burst of this tone every 12 seconds during observation.

Service Observing with Exclusion

Starting with Release 2.2, an option is available on the Feature Related System Parameters screen called **Service Observing Allowed with Exclusion?**.

<table>
<tr><th>Release</th><th>Service Observing Allowed with Exclusion field</th><th>Result</th></tr>
<tr><td rowspan="2">Release 2.2 and later</td><td>Set to <code>y</code></td><td>Allows Service Observing of a station with Exclusion active, either by COS or by manual activation of Exclusion.</td></tr>
<tr><td>Set to <code>n</code> (default)</td><td rowspan="2">Observing towards a station with Exclusion active is denied, or if Exclusion is activated by a station while being observed, all bridged parties including the observer are dropped.</td></tr>
<tr><td>Earlier than Release 2.2</td><td>NA</td></tr>
</table>

No-talk FAC for Service Observing

Starting with Call Center Release 3.1, a station user or a Service Observing route-to number vector operation can optionally activate Service Observing in a listen-only/no-talk mode that does not reserve a second time slot in port network gateway configurations. This does not apply to H.248 Media Gateway configurations. With this option, the ability to switch to the Service Observing talk mode while observing stations or ACD agents is denied. By not reserving the extra time slot, call recording applications that use Service Observing in a listen-only/no-talk mode have greater recording capacity since time slot usage is reduced.

This feature is available only through the Call Center release 3.1 Service Observing No Talk Feature Access Code (FAC), where a Service Observing listen-only/no-talk session cannot be activated or toggled to different modes using a Service Observing station button.

To use this option, first assign an appropriate dial code in the **Service Observing No-Talk Access Code** field on the Feature Access Code (FAC) screen. Then activate observing associations for call recording using the defined No-Talk FAC.

Observing Logical-Agent IDs

With EAS, an observer can observe agents based on their logical-agent ID rather than their physical phone. The observer enters the logical-agent ID extension number of an agent, who must be logged in to a phone. The observer can monitor every ACD, personal, and direct agent call delivered to or placed by the agent, including calls placed to the physical extension.

Without Multiple Observers active (See Multiple observers in*Avaya Aura™ Call Center Feature Reference*) , only one observer can observe an extension at one time. An observer cannot observe a logical agent ID extension at a physical telephone that is already being observed.

Likewise, an observer cannot observe a physical extension that is being observed as a logical-agent ID extension.

Observing VDNs

To observe a VDN, the observer enters a specific VDN extension and bridges onto calls (one call at a time) that have started vector processing for that VDN. The observer hears all tones, call prompting, caller dialing, announcements, music, and speech that the agent and caller hear. If an observer is in a COR administered to hear VDN of Origin announcements and has a VOA Repeat button, he or she can hear and replay VDN of Origin announcements.

Service observing of VDNs is enhanced to (optionally) start observation of a call to the VDN when the call is delivered to the agent or station. When this VDN option is active, VDN service observing activation still associates the observer with calls to the VDN, but the observer does not hear a call during vector processing. After initial activation, the first call to be observed must first pass through vector processing before the observing is enabled. When the observing connection is completed for the first call (the call is released), the observer is bridged on a subsequent call to the VDN (which has also been through vector processing) when the call is answered by an observable agent/station. This ability saves time for the observer because, after observing of the VDN has been activated, the observer does not have to wait (and listen) for each subsequent call to go through vector processing and for the agent to answer.

The ability to observe VDNs when the call is delivered to an agent/station is activated by setting the**Observe on Agent Answer** field on the VDN screen to `y`.

The observer sees the name of the VDN, agent, or trunk as each is accessed in sequence by the VDN. For example, during vector processing the VDN name is displayed, but when the call connects to an agent, the agent name is displayed.

When the observer connects to a call in vector processing, the system maintains the connection until the call is disconnected or the observer hangs up, even if the call is routed or transferred externally. If the observer does not disconnect after one observed call is disconnected, the observer is connected to another call on the same VDN. Observing is listen-only as long as the call is in vector processing. Once the call is out of vector processing, an observer with listen/talk capability can talk as well as listen.

Observing Remotely or by FAC

Observers can observe calls from a remote location or locally using Service Observing FACs. When observing remotely, observers must use FACs. Different FACs are required for listen-only and listen/talk modes. When observing locally or remotely by FAC, the observer cannot toggle between modes. Physical extensions, logical-agent ID extensions, and VDNs can be observed remotely.

Remote observing is initiated through Remote Access or Call Vectoring.

- With Remote Access, an observer accesses a communication server using a trunk group dedicated to Remote Access or using a DID to the Remote Access extension. Remote observing works with all types of DID trunks, including ISDN-PRI and tie trunks, and DCS over analog, T1, or PRI.
- With Call Vectoring, an observer accesses a communication server by dialing a VDN extension or a Central Office (CO) trunk that has a VDN extension as its incoming destination. Using route-to commands, you can design a Service Observing vector to allow a VDN call to directly access a specific extension to be observed or a Service Observing dial tone. At the dial tone, observers can enter any extension that they are authorized to observe. The following is a simple example of a Service Observing vector.

```
1.wait-time 0 seconds hearing ringing
2.collect 5 digits announcement 2300
  ("please dial your 5- digit security code")
3.goto step 5 if digits = 12345
4.disconnect after announcement 2000
5.collect 1 digits announcement 2310  ("enter 1 to observe sales, 2 to observe
billing")
6.route-to number 113001 with cov n if digit = 1  (11=listen-only observe,
3001="Sales" VDN)
7.route-to number 113002 with cov n if digit = 2  (11=listen-only observe,
3002="Billing" VDN)
8.goto step 5 if unconditionally
```

You can combine Call Prompting and Call Vectoring to provide security and to limit observation.

Service Observing indicators

General indications to observer

The following table shows general Service Observing indicators that observers receive.

Condition	Button lamp	Tone
Not active	Dark	None
Denied activation	Broken flutter	Intercept/busy/reorder
Activated	Steady/Winking	Confirmation tone followed by silence or connection to call.
Observing (listen only)	Steady	Hear call
Observing (listen/talk)	Winking	Hear/talk on call
In wait state	Flash	None
Denied observing	Flash (wait state)	Silence/ineligible tone followed by silence

Tables showing what observers receive when using Service Observing

The following tables show the indicators that observers receive when they activate and use Service Observing. In these tables:

- Wait state means that the observer has activated Service Observing but there are no calls or a call cannot be observed. A call appearance is not reserved. The observer must have an idle call appearance available to be used by Service Observing when an observable call comes in.
- Ineligible tone is heard when an observed call becomes ineligible for observation. See Service Observing considerations for conditions that make a call ineligible. This tone is the *hold confirmation tone* - a rapid series of 5 short 440-Hz beeps. The observer does not hear this tone if the agent receiving the ineligible call hears zip tone.

Table 37: Feedback to observers when activation denied

Condition	State	Lamp	Tone
No such extension	denied	broken flutter	intercept
Extension not observable	denied	broken flutter	intercept
[21] Not allowed COR[21]	denied	broken flutter	intercept
Extension has Data Restriction	denied	broken flutter	intercept
Extension has Exclusion Active	denied	broken flutter	busy
Extension has Data Privacy Active on call	denied	broken flutter	busy
Extension already observed	denied	broken flutter	busy
Extension is an observer	denied	broken flutter	busy
Extension being busy-verified	denied	broken flutter	reorder
Extension has a 6-party conference	denied	broken flutter	reorder
COR doesn’t allow SO activation	denied	broken flutter	intercept
Observe VDN not optioned	denied	broken flutter	intercept
Logical ID not logged In	denied	broken flutter	busy
Activation to logical with physical observed	denied	broken flutter	busy
Activation to physical with logical ID observed	denied	broken flutter	busy
Maximum VDNs being observed	denied	broken flutter	reorder

[21] Extension COR cannot be observed or COR for observer calling permission does not allow observing the COR of extension to be observed.

Table 38: Feedback to observer when activation allowed - at time of activation

Condition	State	Lamp	Tone
Active-eligible call	observing	steady/ winking	confirmation tone followed by connection to call
No active call	wait state	flash	confirmation tone followed by silence
Call ineligible	wait state	flash	confirmation tone followed by silence
Call has No Observe COR	wait state	flash	confirmation tone followed by silence
VDN call already being observed	wait state	flash	silence

Table 39: Feedback to observer when activation allowed - after observe activated

Condition	State	Lamp	Tone
No active/eligible Call	wait state	flash	silence
Call in 6-party conference	wait state	flash	silence
Call already being observed	wait state	flash	silence
Call is being busy-verified	wait state	flash	silence
Call has Data Privacy active	wait state	flash	silence
Call has Data Restriction	wait state	flash	silence
Call has Exclusion Active	wait state	flash	silence
Active-eligible call (in listen-only mode)	SO listen	steady	hear call
Active-eligible call (in listen/talk mode)	SO listen/ talk	winking	hear/talk on call
Press button while observing in listen-only mode	SO listen/ talk	winking	hear/talk on call
Observer presses Release	not observing	dark	none
Call has No Observe COR	wait state	flash	silence
VDN call already being observed	wait state	flash	silence
No active eligible call	wait state	flash	silence
Eligible VDN call	observing	steady/ winking	hear call

Condition	State	Lamp	Tone
Eligible VDN call (in vector processing)	SO listen	steady	hear call
Eligible VDN call (out of vector processing in listen-only)	SO listen	steady	hear call
Eligible VDN call (out of vector processing in listen/ talk)	SO listen/ talk	winking	hear/talk on call
Press button while observing in vector processing	SO listen	steady	no change to mode
Press button while not in vector and in listen-only	SO listen/ talk	winking	hear/talk on call
Call being observed becomes ineligible	wait state	flash	ineligible tone followed by silence
Active call disconnects	wait state	flash	silence
Logical agent logs out	denied	broken flutter	busy, then silence
Observer (without button) hangs up	deactivates observing	n/a	n/a

Table 40: The Service Observing button and IP Telephones

Action by observer	What happens	Condition
No action.	The Service Observing button is not highlighted.	The telephone is idle.
The observer presses the Service Observing button.	The Service Observing button is highlighted.	The system is waiting for dialing to begin.
The observer dials a station extension, Agent ID extension, or a VDN extension.	The Service Observing button remains highlighted and displays two down arrows.	The observer has successfully established a session and is “awaiting a call to observe”.
No action.	The Service Observing button remains highlighted. The call display reflects the call being observed. Two arrows are displayed if the observer is in the Listen/ Talk mode. No arrows indicate the Listen Only mode. Initially, Service Observing starts out in the Listen Only mode.	An observable call is offered to the observer.

Action by observer	What happens	Condition
The observer presses the Service Observing button again.	The Service Observing mode of operation is toggled between Listen Only and Listen/Talk and vice versa. The mode is indicated by the appearance or lack of two down arrows on the Service Observing button.	The observer has actively changed their Service Observing mode.

While observing, the observer should press only the following buttons:

- Call Appearance
- Service Observing
- Position Busy
- Auto-ckt Assure
- Release (ACD) (This will end Service Observing)
- Bridged Appearance
- Auxiliary Work
- Queue Status (NQC, OQT, AQC, and AQT)
- System Night Service
- Hold (ignored)

General security

Use the following COR restrictions to prevent unauthorized observing.

- For the observer, set the **Can Be An Observer** field on the COR screen to `y`.
- For the agent to be observed, set the `Can Be Observed` field on the COR screen to `y`.
- For the observer, grant permissions to all CORs to be observed on the Service Observing Permissions COR table.

VDN-call security

Use the following COR restrictions for VDN-call observing.

- For the VDN extension to be observed, set the **Can Be Observed** field on the COR screen to `y`.
- For the VDN destination, set the **Can Be Observed** field on the COR screen to `y`.
- Enter the VDN extensions to be observed in the observer's Service Observing Permissions COR table.

Vector-initiated security

Use the following guidelines for vector-initiated observing.

- Use Call prompting commands in Service Observing vectors to provide passcode protection and limit access to specific destinations or vector-verified, caller-entered digits.
- Use Time of Day/Day of Week checks in Service Observing vectors.
- Create a vector used exclusively for Service Observing.
- If you use route-to commands to observe a VDN extension, ensure the extension has an observable COR.
- If the observer is observing locally, grant calling permission to the observer on the VDN's COR.

In vector-initiated Service Observing, the COR assigned to the VDN used to initiate Service Observing, the COR assigned to the internal caller extension, and the COR assigned to agent to be observed are used to determine if Service Observing will be allowed. If the agent's COR is not observable, observation fails regardless of the VDN or caller COR. When a call routes through multiple VDNs, the COR of the last VDN is used for calling/observing permissions regardless of VDN Override settings.

If you have administered the optional warning tone, the caller and the observer hear the tone only when the system connects the call to the answering or routed-to destination after vector processing is finished. The periodic tone is heard during the call even if the call is transferred off-communication server. Use a warning announcement at the beginning of vector processing to inform the caller of observation since the system cannot give a warning tone until the call is out of vector processing.

Remote-access security

Use the following guidelines for remote observing.

- Use barrier codes and authorization codes to limit the use of Remote Access to authorized users. Refer to *Avaya Aura™ Communication Manager Feature Description and*

Implementation for information about these codes and other remote access security measures.

- Use different authorization codes for different service observing permissions.
- Use Facility Restriction Levels (FRLs) and restrictions such as the authorization code COR to restrict Remote Access service observer access to other destinations (for example, stations or trunks).
- Use Call Prompting to create additional access security.

Assign the VDN, barrier code, and authorization code calling and service observing permissions and set Can Be Observer to yes on the associated COR screen. The last COR encountered is used to determine observer permissions.

Service Observing considerations

Observability

Although an agent can be a member of multiple splits/skills an agent can be observed by only one observer at a time. If two agents with different supervisors are observed and one agent calls the other, the originator's supervisor observes the call, and the other supervisor is placed in the wait state.

An attendant can be observed but cannot be an observer.

Ineligibility

A call to an agent extension or VDN is ineligible for observing when the call:

- Is already being observed
- Is being busy-verified
- Has Data Privacy active
- Has Data Restriction active, is conferenced with an extension that has Data Restriction active, or is a VDN call that reached an extension that has Data Restriction active
- Has Privacy - Manual Exclusion active, is conferenced with an extension that has Privacy - Manual Exclusion active, or is a VDN call that reached an extension that has Privacy - Manual Exclusion active

Note:

If Service Observing with Exclusion is active, observing is allowed when manual exclusion is active.

- Is in a conference where adding the observer results in more than six parties (see Conferenced calls on page 381 for more detail on conferences)
- Is a VDN-observed call that reaches an unobservable extension or VDN. (Note that the COR of the hunt group split or skill used to distribute the call to the station/agent is not checked. The CORs of stations/agents conferenced with the call are not checked.)

Trunk calls

If an agent being observed makes an trunk-call, observation starts after the agent finishes dialing. For Central Office (CO) trunks, dialing is considered complete when answer supervision is returned or when answer supervision timeout occurs.

Multiple observers

Multiple observers can observe a single VDN simultaneously, but only one VDN observer is observing a given call to the VDN. There is no limit to the number of observers observing a single VDN as long as the total number of observers actively observing VDNs does not exceed 50.

When the Allow Two Observers in the Same Call option is active, the VDN observed call could connect to:

- An agent being observed by an agent
- A station observer
- A conference with another observer.

See Service Observing with Multiple Observers for details.

Conferenced calls

An observer cannot initiate a conference while observing.

If an observed agent conferences a call and the number of conferenced parties is less than six, the observer is placed in the wait state until the call is connected. Then the observer observes the conference. In addition, the observer is bridged onto any call on which the agent becomes active before the conference is complete. When the conference is complete, the observer is again bridged onto that call.

If an observed agent conferences a call and the number of conferenced parties (including the observer) is six, the conference is denied.

A call to an observed VDN cannot be monitored if the observer, caller, and other parties bridged onto the call constitutes more than six parties.

If a conference is being observed because an observed agent entered the conference, when the agent hangs up, the conference is no longer observed. If a conference is being observed because an observed VDN call entered the conference, observing continues until the call is routed to an unobservable destination.

Conference members are observed during a conference regardless of their COR setting.

If a VDN call being observed is conferenced to an agent call being observed, the VDN observer continues to observe and the agent observer goes into wait state. If two observers (of either VDN or agent calls) are conferenced to a call, the first observer conferenced-in continues to observe and the second observer goes into the wait state. VDN or agent call observers hear the ineligible tone before going into wait state.

The same rules apply when multiple observers monitor transferred calls.

Transferred calls

Observers cannot initiate a transfer while observing.

When a service-observed agent starts a call transfer by pushing the Transfer station-button, all parties connected to the call are placed on hold, and the service observer is placed in the service observing wait state hearing silence. The service observer continues to hear silence while the agent dials a second party and hears ringback. When the second party answers the agent's call, the service observer is reconnected to the transferring agent. When the Transfer station-button is pushed again to complete the call transfer, all held parties are reconnected to the call, the transferring agent is disconnected from the call, and the agent's service observer is placed in the service-observing wait state hearing silence.

A VDN observer continues to monitor the transferred call until it is transferred or routed to a unobservable destination.

Service Observing interactions

Interaction	Description
ASAI	A call to an observed VDN continues to be observed after it routes to an adjunct. A call can be routed to a Service Observing FAC by the adjunct routing command in the same way that it can be with the route-to command.
Assist	A VDN observer continues to observe a call during an assist operation. The observer observes the caller on hold and the conference, when the agent conferences the assist call with the VDN call.
Avaya Call Recording (Witness)	The Call Center Release 3.1 listen-only/no-talk mode for Service Observing reduces the time slot usage required by the recording device to connect to monitored stations and ACD agents, allowing greater call recording capacity. Since the Service Observing No-Talk Feature Access Code used by the recording device to activate a recording session does not provide a

Interaction	Description
	talk-path, a warning tone cannot be inserted by the recording device when the recording session begins. If you need to notify the parties in a call that the call may be recorded, design your vector to execute an `announcement` vector step before queuing or routing the call that provides an audible warning message. As another option, you can have the Communication Manager provide the warning tones by enabling the Service Observing warning tones on the system-parameter features screen.
BCMS	BCMS does not report on Service Observing. BCMS reports show normal measured-call and agent activity related to Service Observing calls. When a physical agent (non-EAS) is observed, the BCMS Report By Login ID shows the physical extension along with the login ID.
Bridged appearances	If an observer observes agent extension 3082, the observer is bridged onto calls only to 3082. If the agent with extension 3082 has a bridged appearance for extension 3282, calls to extension 3282 are not observed. Although extensions 3082 and 3282 have a call appearance on the same telephone, the observer cannot observe both extensions at the same time.
Busy-verification	An observer cannot observe an agent call that is bridged onto by busy-verification. Also, an agent's call that is being bridged onto by an observer cannot be busy-verified.
Call Coverage/Call Pickup	An observer cannot observe a call answered by a covering agent or member of a pickup group until the called agent bridges onto the call. The observer continues observing a call to an observed VDN call if the call is routed to a destination that forwards the call (using Call Coverage, Call Forwarding, or Call Pickup).
Call Park	An observer cannot park a call while observing the call. An observer observing a VDN continues observing after a call is parked.
Call Waiting	A call cannot wait on a single-line phone that is being observed.
Call Work Codes/ Integrated Directory	The observer does not hear agent dialing with these features because the digits are passed to the communication server in S-channel messages.
CMS	When an observer is bridged onto a VDN call, CMS is notified.
Conference and Transfer	A VDN observer who is bridged on a call follows the call on a conference and/or transfer operation.
Converse Command	Converse-split extension ports can be observed as physical extensions. A call to an observed VDN continues to be observed if the call is answered by a VRU through the converse command.
Converse-on Vector Command	Calls connected by the converse-on command are not observed by the VDN observer when the Observe on Agent Answer option is set to y. If the call is subsequently answered at an agent station or other

Interaction	Description
	destination using the route-to command, the VDN observer is bridged on the call.
DCS	To observe stations on another node (a DCS station extension), you must set up remote-access service observing. A DCS station can only observe another node using remote service observing. Service observing displays are not supported across DCS.
Dialed Number Identification Service	Observing by VDN provides monitoring by DNIS since the VDNs represent the DNIS of the service dialed.
Direct Agent Calling	A direct agent call to a logical-agent ID is monitored by observing the Logical Agent not by monitoring the physical extension.
Hold	Observers cannot place calls on hold while observing. If an observed agent places a call on hold, the observer is put in wait state. A VDN observer continues to monitor the caller placed on hold.
Leave Word Calling	Parties on an observed call cannot use LWC.
Look Ahead Interflow	If an observed VDN call routes to another location using Look Ahead Interflow, the call continues to be observed. The observer hears a warning tone, if administered at the sending communication server, when the call arrives at the receiving communication server. The observer continues to hear the periodic tone while observing the VDN call.
Manual Answer	VDN observers are bridged on to the call when the agent answers the call that has been ringing the ACD agent extension with the Observe on Agent Answer set to y.
Move Agent/ Change Skills	Moves or changes of physical of logical agents being observed occur according to the move or change rules. Observing continues.
Multiple Call Handling	While an agent extension or logical ID is observed, only the active call is monitored. If all calls are put on hold, the observer hears silence.
Music-on-Delay/ Music-on-Hold	If an observer is in listen/talk mode, neither caller nor observer hears music-on-hold. If an observer is in listen-only mode, the caller hears music-on-hold, but the observer does not. A VDN observer hears music provided to the caller.
Night Service	A VDN observer continues to observe when a call routes to night service.
Recorded Announcement	A VDN observer continues to monitor a call connected to an announcement. A Verify Announcement call placed by an observed physical or logical agent can also be observed.
Redirection on No Answer	A VDN observer continues observing a call after it is redirected or rings in limbo.

Interaction	Description
Route-to Number Vector Command	Calls connected by the route-to number command are observed by the VDN observer after *answer* is received or assumed when the Observe on Agent Answer option is set to y. this includes routing to internal destinations (stations, hunt groups, ACD splits/skills, the attendant, etc.) or to external destinations (using trunk facilities.
Trunks without disconnect supervision	Service observing cannot be activated over no-disconnect-supervision trunks. The caller hears denial indication.
VDN of Origin Announcement (VOA)	VDN observers with the Observe on Agent Answer option set to y are not bridged on the call until after the VOA is given to the agent. Therefore, the observer does not hear VOAs.
VDN Return Destination	You can create a prompting VDN with a return destination assigned so that, if you activate observing and it fails or the denial indication times out, the prompting VDN allows you to retry activation. This is true only if the denial and disconnection occur after the call leaves vector processing. If a vector step fails, the system proceeds to the next vector step. Disconnect or busy commands cause calls to be dropped and do not trigger return destination. When return destination is triggered, the call is monitored through each return destination operation until the caller disconnects. The observer bridged on the call follows the call when the VDN Return Destination feature, active on the VDN, redirects the call back through vector processing after the agent releases the call.
Telephone displays	The display for local observers matches exactly what is displayed on the observed physical or logical agent's telephone display. For example: a="3035001234 to Sales SO" While observing a VDN, an observer sees displayed the name of the VDN being observed while in vector processing. After the call leaves vector processing, the name of the agent or trunk group that the call is connected to is displayed.
VuStats	Non remote observers using 2-line displays can activate VuStats for an agent. An observer must activate VuStats before using Service Observing. The agent's statistics appear on the second line of the observer's display.
Zip tone	VDN observers do not hear the zip tone that the answering agent hears.

Service Observing with Multiple Observers

Service Observing with Multiple Observers description

Service Observing with Multiple Observers means that:

- Up to two observers can monitor the same agent Login ID or station extension using the Service Observing station button or using any of the following Feature Access Codes (FACs):
  - Service Observing Listen-Only
  - Service Observing Listen/Talk
  - Service Observing No-Talk
- Two separate calls, each with an associated service observer, can be conferenced together with both service observers included in the merged conferenced call except when both observers are VDN observers. In this case one VDN observer will be dropped.
- Customers who use call recording products, such as the Avaya Witness Call Recording or NICE can connect a voice-storage server to a station or Login ID extension in order to record agent-to-customer transactions acting as an observer.
- Customers who use call recording products can also allow an observer to monitor a station or Login ID extension and record the transaction at the same time.

Note:
This feature does not allow multiple observers on the same call for the Service Observing by VDN feature.

Prerequisites for using Service Observing with Multiple Observers

You can use the Service Observing with Multiple Observers only if all of the following fields are set as follows:

- The**Call Center Release** field is set to `4.0` or later on the System Parameter Customer-Options screen.
- The**Service Observing (Basic)** field is set to`y` on the System Parameter Customer-Options screen.
- The **Can Be A Service Observer?** field is set to`y` on the Class of Restriction screen to be an observer.

OR

- The **Can Be Service Observed?** field is set to `y` on the Class of Restriction screen to be an observee.

Service Observing with Multiple Observers interactions

Interaction	Description
Agent Assist	If the supervisor an agent calls for help via the Assist button is currently observing the agent, when the supervisor picks up the call appearance to talk to the agent, the observing call appearance is dropped. Once the assist is over, the supervisor is no longer observing the agent.
Conference Tone	A second service observer added to a call does not have any effect on the current Conference Tone operation. If the second service observer drops or is dropped from the call, the tone is continued only if there are more than two parties left on the call.
Malicious Call Trace	If an incoming PRI trunk call with the Malicious Call Trace (MCT) feature activated has been routed to an agent being monitored or on a conference call, the two observers are included in the MCT reporting for this call.
No-Hold Conference	No-hold call conference works the same way with two observers as it does when one observer monitors a call.
QSIG Path Replacement	If the QSIG Path Replacement operation takes place when multiple observers are either monitoring the call-legs at the originating Avaya communication server or are merged together when the path replacement takes place, the total number of observers left on the call will not exceed two observers.
Service Observing of VDNs	The Multiple Observers with Service Observing feature allows only one VDN observer on a call in vector processing. If a service-observing route-to number < FAC> vector step associated with the VDN is currently being observed by a VDN observer, a second observer can be added to the call

Interaction	Description
	only if the second observer is a station observer or a login ID observer.

Service Observing with Multiple Observers example

A typical user of the Service Observing with Multiple Observers feature is Call Center in a financial institution like a bank where all the calls need to be recorded for security and quality control purposes. The Call Center may use a call recording facility like Avaya Witness Call Recording to record the calls to each agent. Additionally a Call Center manager or supervisor may simultaneously use temporary service observing of an agent to monitor how the agent handles calls.

The benefit provided by the Service Observing with Multiple Observers feature is exemplified when a call being recorded and observed simultaneously is conferenced with another call that is also being recorded. This feature allows the recording (service observing association) of both calls to continue.

Without Service Observing with Multiple Observers one of the recording associations is temporarily removed during the duration of the conference because only one service observer, the recorder in this case, can be observing the same call.

About UCID

UCID definition

Universal Call ID (UCID) is a unique tag assigned to a call.

In simple call scenarios, the tag stays with that call within a network that is based on communication servers connected by ISDN or SIP trunks. In complex call scenarios, the tag often merges with other tags.

Note:
The UCID data element is universal because it does not just identify a call on one particular communication server; a UCID uniquely identifies a call across a network of communication servers.

What is UCID's purpose?

The purpose of UCID is to tag a call with a unique identifier.

UCID provides a way to track calls across multiple communication servers and Voice Response Units.

Call centers can use UCID to track call history. Because UCID can uniquely identify every call in a network of any size, it is possible to track call-related data from multiple sources and multiple sites. For example, you can combine data from many locations and print reports that enable you to track a call throughout its lifecycle. For information about how to create reports, see *Avaya CMS Supervisor Reports*.

Note:
Although UCID is intended for call centers, a communication server configured to create UCIDs will assign one to every call - not just to Automatic Call Distribution (ACD) calls.

What does UCID look like?

The Universal Call ID is an 8-byte data element that displays as a 20-character number.

How does UCID work?

For every new call that comes into or is originated by the communication server or an IVR product, the product creates a UCID. Depending on the call scenario, the UCID will either remain unique to that call or merge with other UCIDs.

What creates UCIDs?

Both the communication server and the IVR system can create UCIDs once the capability has been enabled. In other words, neither product automatically creates UCIDs until the feature is enabled.

When are UCIDs created?

Once the communication server or the IVR system is administered to create UCIDs, these products assign a UCID to each call. For incoming calls over ISDN or SIP trunks, the communication server determines whether or not the call already has a UCID. If so, the communication server preserves the existing UCID and does not create a new one. If the call does not have a UCID, the communication server creates one when call processing begins.For incoming calls over trunks other than ISDN or SIP, the communication server does not create a UCID for the call because these trunks do not support the transmission of UCID.

For outgoing calls, the communication server creates a UCID when the caller goes off-hook.

How are UCIDs transmitted?

How communication servers transmit UCIDs depends on the sending and receiving equipment. The following table summarizes UCID transmission features.

Sender	Receiver	Connection	UCID contained in
Switch	Switch	ISDN (BRI or PRI) trunks using QSIG service protocol	[22] codeset 0 Facility IE as manufacturer specific information (MSI) IE[22]
		ISDN (BRI or PRI) trunks using Shared UUI service protocol	codeset 0 shared user-to-user information (UUI) IE[1]
		SIP trunks using Shared UUI service protocol	User-to-User header of an INVITE message
Switch	IVR	ASAI	various ASAI messages
IVR	Switch	ISDN-PRI	codeset 0 shared UUI IE[1]
Switch	CMS	X.25	SETUP5 CMS message
Switch	CTI adjunct	ASAI	various ASAI messages

How are UCIDs tracked?

Overview of UCID tracking

The way a network maintains and tracks a UCID depends on the call path. To illustrate UCID transport throughout a call's life cycle, this section describes several call scenarios:

- Station-to-station calls
- Incoming Trunk Calls
- Outgoing Trunk Calls

[22] Refer to Information Forwarding.

- Simple Transfer or Conference
- Complex Transfer and Conference

Station-to-station calls

This scenario describes what happens when Phone I calls Phone II (both phones are on the same communication server).

The communication server creates a new UCID (such as UCID a) for any call originated by an internal station user.

Incoming trunk calls

UCID is assigned to an incoming call.

The communication server does one of the following:

- Receives UCID x information from an incoming call over an ISDN trunk.
- Creates UCID y for incoming calls that do not already have a UCID.

There is one CMS call history record for each incoming call.

Outgoing trunk calls

UCID is associated with the outgoing trunk call from Phone I.

The communication server creates a UCID (such as UCID x) for an outgoing trunk call and then sends it over an outgoing shared UUI or QSIG ISDN trunk.

The communication server creates a UCID (such as UCID x) for an outgoing trunk call even if the trunk does not support the transmission of a UCID.

Simple transfer or conference

This scenario describes a simple transfer or conference call scenario.

When an incoming trunk or station call is received by the station user at Phone I and transferred to or conferenced with another station user or outside party:

1. The communication server creates a UCID for the incoming call if it needs one.
2. The communication server creates a new UCID for the temporary conference or transfer portion of the call.
3. The communication server merges the temporary portion of the call with the original call when the conference or transfer is completed within the communication server.

This is when the overriding UCID (such as UCID a), becomes the UCID for all parties within the communication server.

Note:

If the outgoing trunk does not support the sending of UCIDs, then the UCID of the outgoing call at the receiving communication server will be null.

If the call is transferred to another communication server, only the UCID for the transfer (UCID b) gets passed on. This is because the communication server cannot merge UCIDs if the call is not completed within the communication server.

Note:

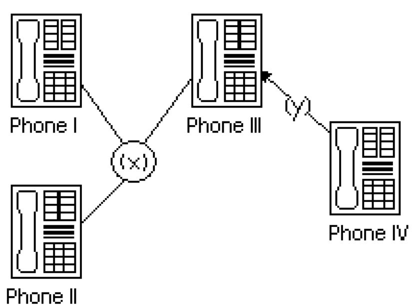
If, during the conference or transfer, the incoming call drops before the operation is complete, the two UCIDs will not appear to be associated because no merge of the two parts of the call was done.

Complex conference

The following complex call scenario illustrates when a station user adds an incoming call to an existing conference.

In this scenario,

1. Phones I, II, and III are in the same conference call with UCID “x”.
2. The person at Phone III receives an incoming call from Phone IV (this call has UCID “y” associated with it).

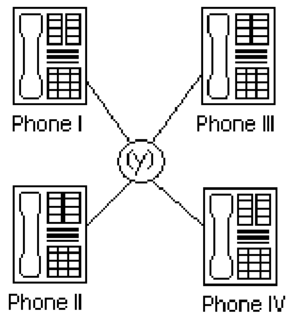


3. The person at Phone III puts the conference call on hold and answers the incoming call from Phone IV.
4. The person at Phone III decides to add Phone IV into the conference call.
5. The person at Phone III
    a. presses the Conference button
    b. presses the call appearance button to return to the conference call

c. presses the Conference button again.

This brings the conference call into the call between Phones III and IV.

6. UCID “y” overrides UCID “x” because the communication server views Phone IV as the primary party in the conference initiated by Step 5.



7. The UCIDs associated with each segment of the complex conference are sent to CMS if the parties in the call are measured (for this example, if the parties are ACD agents in a measured split or skill).

Configuration - communication server before the IVR system

The following scenarios describe what happens to UCID information when a call comes in to the switch before it goes to the IVR system. In this configuration, the IVR system serves as a Voice-Response Unit (VRU) that controls the routing of incoming ACD calls.

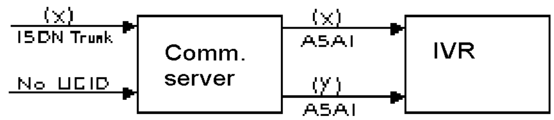
Note:
This configuration is more common than a call coming in to the IVR system before reaching the communication server.

This section describes two scenarios:

- Simple call tracking
- The IVR system transfers a call

Simple call tracking

The following call scenario describes when a call comes in to the communication server before the IVR system.



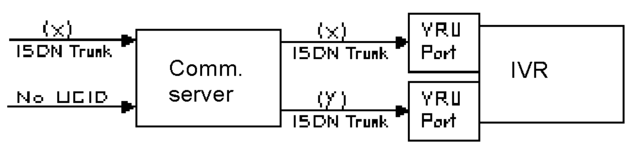
When the communication server is before the IVR system:

1. The communication server receives an incoming call over an ISDN trunk.
2. The communication server does one of two things:
    - If the incoming call has a UCID (such as UCID x), then the communication server passes it along.
    - If the incoming call does not have an associated UCID, the communication server creates a new one (such as UCID y).
3. The communication server passes the UCID to an Interactive Voice Response (IVR) voice system through an ASAI connection (using the activation of split or skill or VDN event notification by the IVR system.
4. UCID information is sent to the CMS if trunk, VDN(s), and/or split or skill(s) involved in the call are measured.

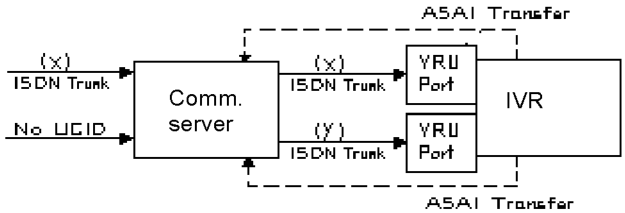
An IVR system transfers a call

The following call scenario involves an IVR system behind the communication server configuration when the IVR system initiates a call transfer after the call is answered by a port on the IVR system that serves as an ACD agent.

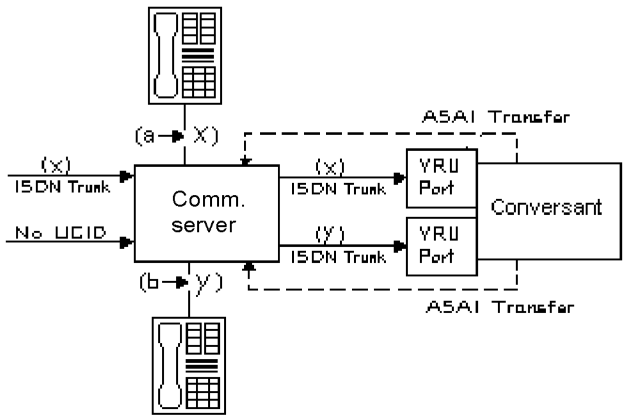
1. Call is directed to the IVR system VRU port (typically by call vectoring) with UCID information (UCID x or UCID y).



2. The IVR system determines the call's destination and transfers the call (using an ASAI third-party transfer operation).



3. The communication server temporarily creates a new UCID (such as UCID a or UCID b) for the transfer portion of the call (the original UCID is quickly merged into the call).



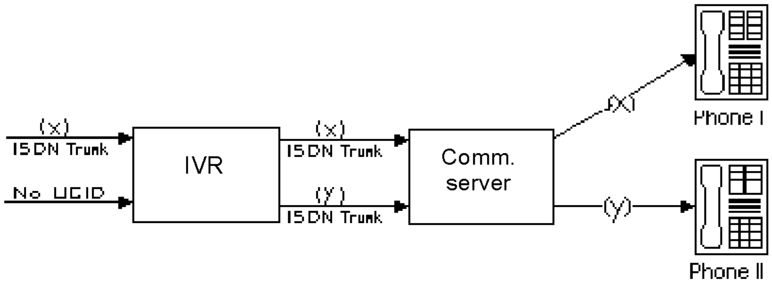
4. The UCIDs of the transfer segment and merged call are returned to the IVR system in ASAI acknowledgment messages.
5. The communication server sends UCID information to CMS if trunk, VDN(s), and/or split or skill(s) involved in the call are measured.

Configuration - IVR before the communication server

This scenario illustrates a system configuration where a call comes in to (Interactive Voice Response (IVR) before reaching the communication server. In this configuration, IVR provides voice response services or call screening so that the number of incoming calls to the communication server is reduced.

Note:

This configuration is less common than the communication server before the IVR configuration.



When IVR is before the communication server:

1. IVR receives an incoming call with UCID `x`.

   or

   IVR creates a new UCID y and associates it with the incoming call (if the call has no UCID already associated with it).

   Note:
   For IVR to recognize an incoming UCID (such as UCID x) from an ISDN trunk, special IVR scripting is required. When IVR receives a call from the public network, it automatically creates a new UCID because it cannot recognize whether or not the call already has a UCID.
2. IVR sends UCID to the communication server over an ISDN-PRI trunk.
3. The communication server receives UCID and reuses it for the incoming call.
4. The communication server reports UCID to the CMS if the trunks, VDNs, or splits/skills associated with the call are measured.

UCID interactions

Interaction	Description
Distributed Communications System (DCS)	If DCS is used in a network of communication servers where UCIDs are tracked, the DCS feature must be configured with ISDN or SIP trunks having the Shared UUI service protocol. Otherwise, calls that are handled through one of the many DCS features (such as DCS Coverage) will not retain the UCID initially assigned to the call.
Remote messaging system	For a remote messaging system over DCS, the DCS trunks used to accomplish the remote messaging system operation must be configured (as described previously in Distributed Communications System) to retain the UCID associated with a call.

Interaction	Description
Tandem Calls	When a tandem call is made through the communication server, the UCID information may be blocked or passed through the tandem communication server. To pass a UCID through a tandem communication server, both the incoming and outgoing trunks at the tandem communication server must be configured to handle UCIDs. See Information Forwarding for proper private and public network information forwarding administration.

VDN of Origin Announcement

About VOA

VDN of Origin Announcement (VOA) provides agents with a short message about a caller's city of origin or requested service based on the VDN used to process the call.

Use VOA messages to help agents to respond appropriately to callers. For example, if you have two 800 numbers, one for placing orders and one for technical support, you can administer two VDNs to route calls to the same set of agents. When an incoming call is routed to a VDN with a VOA assigned (for example, new order or tech help), the VDN routes the call to a vector, which can place the call in an agent queue. When an agent answers the call, he or she hears the VOA message and can respond appropriately to the caller's request.

VOA detailed description

The agent cannot hear the caller while the VOA message is playing. The caller is not connected to the agent until after the message completes and cannot hear the message or the agent during the message. The caller hears ringback while the agent is listening to the VOA.

Agents logged in at multiline telephones see the call-appearance button for an incoming call flash until after the VOA completes. An agent can press the flashing call-appearance button to stop the VOA.

To repeat the VOA, an agent presses the VOA Repeat button. The VOA Repeat button lamp lights during the VOA. The VOA Repeat button lamp remains lit if the repeat request is queued. If an agent presses the VOA Repeat button while the lamp is lit, the VOA is stopped. If an agent presses the VOA Repeat button but there is no VOA or the system cannot play the VOA within three seconds, the lamp flutters.

You assign VOAs for each VDN. However, the VOA applies to a COR, so you must administer a COR for agents who will receive VOAs.

You can set up VOAs in four ways:

- Agents can hear a unique announcement based on the dialed number identification service (DNIS) received from the service office or carrier communication server. Assign each DNIS as the VDN of a vector. Set up the VOA to announce the services associated with the DNIS.

  Note:
  The announcement associated with the current VDN only plays if the VDN Override for the previous VDN is set to y. If VDN Override for the previous VDN is set to n, the VOA associated with that VDN plays.

- Use vector steps, an integrated prompting, or converse-on step to route calls to a VDN. Set up the VOA to announce the service the caller requested or to announce a condition that caused the call to route-to the VDN.
- You can route calls to a voice response system, directly or through a vector. Use voice prompting to direct the caller to enter a touchtone response, and route the call to a specific VDN based on the caller's response. Set up the VOA to indicate the service the caller selected.
- If agents require a caller's city of origin, assign the trunk group to a particular VDN. Set up the VOA to provide the location of the origin of the trunk group. Subsequent VDNs can be used to handle the call, or multiple VDNs can be assigned to a single vector.

  Note:
  VDN Override applies to VOA in the same way that VDN Override applies to display information. If a VDN with a VOA has VDN Override enabled, the system overrides the original VOA with VOAs in subsequent VDNs to which the call is routed.

VOA considerations

- Because callers are kept waiting while a VOA plays, messages should be kept very brief - no more than 1.5 seconds in length. Agents should use a speakerphone or headset, so they do not miss the VOA while they are picking up the handset. If agents cannot use a speakerphone or headset, administer phones with a VOA Repeat button.
- If you have multiple announcement boards, you should place shorter VOAs on one board and longer recorded announcements on the other to avoid delaying delivery of VOAs. If you have only one announcement board, place VOAs on the integrated board and consider installing an auxiliary announcement device for longer announcements.

- Agents must be on the same communication server as the VOA.
- A VOA can be assigned to multiple VDNs, but a VDN can have only one VOA.
- If you use the TN750 circuit board for integrated announcements, the system maintains a separate logical queue for VOAs. If the VOA cannot be delivered to the agent within 1 second because of traffic or inoperative equipment, the system does not provide the announcement. VOAs are higher priority than other announcements on the TN750. A burst of VOAs can delay other announcements. Therefore, record non-VDN of Origin Announcements as auxiliary or analog.
- Auxiliary announcements are connected for a duration of 1 to 2 seconds on a barge-in basis, immediately after the agent answers (or is assigned the call for auto-answer) and the incoming call is extended to the agent. Integrated and non-barge-in auxiliary announcements are connected for the duration of the announcement. The communication server does not ensure that the integrated announcement is shorter than the allowed playback time.
- VOA supports Auxiliary Trunks (aux-trunk) with barge-in, queue, or without queue. For aux-trunk with or without queue, when the trunk is idle, a VDN call seizes the trunk to start the VOA and the system plays the entire announcement (not just 1 to 2 seconds). However, if the announcement is busy and if aux-trunk has barge-in, the call does not queue but bridges onto the announcement for 1 to 2 seconds. When the VOA completes, the trunk is released along with the listeners, and the next call requiring the VOA starts the process over again. For this reason, your aux-trunk announcements should consist of one short announcement that repeats during the full announcement time. For example, you might want to record *New Order* as many times as possible, so that when a call bridges to the announcement, the agent hears *New Order* no matter where the agent bridges into the announcement.
- If you use aux-trunk or integrated announcement without queue and a port is busy when a VDN call comes in, the system cannot play an announcement. If you use aux-trunk or integrated announcement with queue, the system plays the current announcement for an agent and then connects the next agent in the queue.

VOA interactions

Interaction	Description
Agent Call Handling - Automatic Answer	Includes the following interactions: • ACD agents at phones in Auto Answer mode hear a zip tone, then the VOA. You can also administer a zip tone after the VOA completes, to alert agents that an announcement is complete and a caller is connected. • Non-ACD agents can receive a VOA if a call is routed to them using vector processing. When non-ACD agents at phones in Automatic Answer mode receive calls, they hear a call ID tone

Interaction	Description
	then the VOA. Agents hear a second zip tone after the VOA indicating connection to the caller.
Agent Call Handling - Manual Answer	When non-ACD agents at phones in Manual Answer mode receive calls they hear ringing, answer the call, and hear the VOA.
ASAI Adjunct Routing	If a vector step includes Adjunct Routing, the VOA is played for the agent to whom the call is routed.
Auto-Available Split/ Skill (AAS)	AAS is intended to be used for splits/skills containing only nonhuman adjuncts such as a voice messaging system or an IVR system. However, VOAs can be directed to Auto-Available splits/ skills.
Call Forwarding	VOAs apply to forwarded calls, including those forwarded to a hunt group. The answering station must be on the same communication server. If a VOA is forwarded, the message is played only if the destination extension is administered with a COR that allows VOA.
Call Pickup	Call Pickup allows an agent to pick up a ringing call on another extension. If the pick-up extension has COR permissions for VOA, the agent can receive a VOA.
Conference	If an agent receives a call and then conferences in additional stations, any station on the connection can use VOA Repeat button to replay the VOA. Only the person using the button can hear the VOA unless the call is being service observed.
Converse-on split or skill	A converse-on split or skill is one used in a converse-on vector step. When a converse-on vector step is executed, a VOA is not applied. After returning to the vector, the call can be routed to a station or VDN where the answering agent receives the VOA (as if the converse-on step had not been processed).
Coverage	VOA applies to coverage paths.
Data Restriction	Data Restriction prevents tones from being applied to line or trunk circuits during a data call. VOAs are not played for data-restricted calls.
Direct Agent Calling	Direct Agent Calling (DAC) allows a vector to route a call to particular ACD agent and have the call treated as an ACD call. The VOA only applies to direct agent calls if the calls reach an agent through vector processing. Direct agent calls from a phone on a communication server are not vector-processed and cannot cause a VOA to be played.
Enhanced Automatic Wake-up	If you are using enhancements to Automatic Wake-up with integrated announcements, there can be contention for integrated announcement ports. VOAs have priority over Automatic Wake-Up announcements.

Interaction	Description
EAS	When you are using Expert Agent Selection (EAS), the logical agent COR definition determines the assignment of VOAs for each extension. EAS uses the COR of the logical agent instead of the COR for the telephone the agent is using.
Hold	Agents cannot use the VOA Repeat button if their calls are all on hold. The VOA Repeat button only applies to active calls.
Home Agent	You can assign an initial VOA to a home-agent port on the communication server. However, home agents cannot use a VOA Repeat button because home agents need a dial access code (DAC) to reach features and VOA replay does not use a DAC.
Hunt Groups	VOAs apply to calls routed to a hunt group. The COR for the answering station's extension determines whether the station can receive a VOA.
Look-Ahead Interflow	VOAs apply only to the communication server where the VDN is defined. If a call interflows to another communication server, the VOA is lost. You can have the interflow to another communication server access a VDN with the same VOA message as on the original communication server.
Redirection on No Answer (RONA)	If a call re-queues to a split or skill because the RONA timer expired, the VOA applies to the call when an agent answers the call.
Service Observing	The system handles Service Observing calls as conference connections. If the observer presses the VOA Repeat button only he or she hears the announcement. However, if another party on the call presses the VOA Repeat button, the user and the observer hear the VOA.
Supervisor Assist	If an agent requests supervisor assistance and conferences the supervisor into a call, either the agent or the supervisor can use their VOA Repeat button to replay the VOA, but only the person who presses the button hears the VOA.
Transfers	If an agent receives a VDN call and transfers the call, the answering party can use the VOA Repeat button to replay the message.
VOA distribution	If you use long VOAs or multiple VOAs, there may be a delay between the zip tone and the announcement. The system provides multiple announcement circuit packs to help prevent announcement delays. Contact your Avaya representative for more information.

Voice Response Integration

About VRI

Voice Response Integration (VRI) integrates Call Vectoring with the capabilities of voice response units (VRUs). You can:

- Run a VRU script while retaining control of a call in vector processing
- Run a VRU script while a call is queued, retaining its position in the queue
- Pool IVR ports for multiple applications
- Use a VRU as a flexible external-announcement device
- Pass data between the system and a VRU
- Tandem VRU data through a communication server to an ASAI host

The **`converse-on`** command, which is part of Basic Call Vectoring, provides these capabilities. Use a converse-on call-vector step to integrate a VRU with Automatic Call Distribution (ACD). VRI allows you to use VRU capabilities while controlling a call in ACD.

Include VRUs with vector processing to take advantage of the following:

- Access to local and host databases
- Validation of caller information
- Text-to-speech capabilities
- Speech recognition
- Increased recorded announcement capacity
- Audiotex applications
- Interactive voice-response (IVR) applications
- Transaction-processing applications

VRI allows users to make productive use of queuing time. For example, while a call is queued, a caller can listen to product information using an audiotex application or can complete an interactive voice-response transaction. It may be possible to resolve the caller's questions while the call is queued, which helps reduce queuing time for other callers during peak times.

VRI detailed description

A call queued to a split or skill retains position in the queue while a VRU script is being run. When an agent becomes available, the line to the VRU is dropped and the caller connects to the agent.

The **converse-on** command delivers a call to a predetermined converse split or skill. A converse split or skill is administered and operates exactly like other splits/skills. Non converse splits/skills are splits/skills that are accessed by queue-to and check vector steps.

Members of a converse split or skill are the ports connected to the VRU. If all VRU ports are busy, a call queues to the converse split or skill with the administered priority. After the VRU answers the call, the **converse-on** command may pass up to 2 data items to the VRU, depending on command parameters specified. You can pass data required by a VRU script or data that selects the VRU script to be run.

Whether or not you pass data, a caller is connected to the VRU, which runs the VRU script. Audible feedback provided by the vector is not heard and no further vector steps are run until the VRU script completes. The VRU may return data to the system and then drops the line to the system. Vector processing continues at the step following the **converse-on** command.

If the call was queued to a non converse split or skill before the **converse-on** command was run, the call retains its queue position. If an agent becomes available while the VRU script runs, the system drops the line to the VRU and connects the caller to the agent. The VRU detects the disconnect and terminates the VRU script.

Call Prompting allows you to collect and use digits that the VRU returns. These digits are handled as dial-ahead digits. Rules for collecting and processing VRU digits are the same as for Call Prompting.

You can use digits returned from the VRU in the following ways:

- To display for the answering agent's (automatically for 2-line displays or with the callr-info button for other displays)
- As an extension in a route-to digits vector step. For example:

```
converse-on split. . . . (VRU returns 4 digits)

collect 4 digits after announcement none

route-to digits coverage y
```

- For vector-conditional branching in an **if digits equals** vector step. For example:

```
converse-on split . . . (VRU returns 1 digit)

collect 1 digit after announcement none

goto vector 101 if digits = 1
```

```
goto vector 102 if digits = 2

goto vector 103 if unconditionally
```

- Tandem to an ASAI host - Collected digits are passed to ASAI hosts in Call Offered to Domain Event reports and in route request messages, thus caller digits or database information returned from the VRU can be sent tandem through the system to ASAI hosts. For example:

```
converse-on split ... (VRU returns 9 digits)

collect 9 digits after announcement none

adjunct route link Y
```

In this vector, the digits returned from the VRU are forwarded to the ASAI host in the adjunct routing route request message.

When you use a VRU application that returns data for a collect-digits step, the opportunity for toll fraud exists when the VRU application does not return any data. Take the following precautions:

- If the collected digits are used to route calls internally, ensure that the Class of Restriction (COR) for the vector directory number (VDN) does not allow calls to route externally.
- If the collected digits are used to route calls externally, use a password to verify that the collected digits have been passed by the VRU application. For example, in the following vector, the VRU application returns a 3-digit password followed by the 8-digit external number. The vector routes calls without the correct password to a vector 23.

```
converse-on split 10 pri m passing none and none (VRU returns 11 digits)
collect 3 digits after announcement none
goto vector 23 if digits <> 234
collect 8 digits after announcement none
route-to digits with coverage n
```

VRI interactions

Converse splits interact like other vector-controlled splits unless noted here.

Interaction	Description
Adjunct Switch Applications Interface (ASAI)	When a converse-on vector step places a call to an ASAI-monitored domain, ASAI event messages are sent over the ASAI link. When a converse-on step places an ASAI-monitored call, the ALERT message sent to the ASAI adjunct includes a cause IE, Coding Standard 3 value 23 (CS3/23), which informs the adjunct that the call has not been dequeued from any non converse splits.

Interaction	Description
	If a converse-on step is run while an adjunct routing request is outstanding, the request is canceled. ASAI cannot transfer or conference calls, but can direct the system to do this.
Agents	Although not recommended, you can use a converse-on step to deliver a call to a group of human agents. To agents, the call looks like an ACD call, except they cannot use certain features, such as Transfer, Conference, and Supervisor Assist. The agent can return data to vector processing by pushing the transfer button (or flash hook on analog) and dialing the converse-on data return code and required digits.
Answer supervision	Answer supervision is returned only once during a call. If a call is answered because of a converse-on step, answer supervision is sent if it hasn't previously been sent. If digits are passed to the VRU, answer supervision is sent after digits are sent.
AUDIX	If a converse-on step calls AUDIX, the call is handled as a direct call to AUDIX. The caller hears the AUDIX welcome message and can retrieve messages as usual. If a call is forwarded to a VDN and then delivered to an AUDIX hunt group by a converse-on step, the call to AUDIX is treated as a redirected call, and the caller may leave a message.
Auto-Available Split/Skill (AAS)	A converse-on vector step can place a call to an AAS. Use auto-available converse splits/skills for VRI except when ASAI controls the converse split or skill.
Automatic answering	When you administer ports on your IVR system as agents of a converse split or skill, do not administer agents as automatic answer. The system-provided zip tone may interfere with the interaction between the IVR system and the calling party.
BCMS/CMS	BCMS tracks calls that a converse-on step places to a BCMS-measured hunt group. CMS tracks calls that a converse-on step places to a CMS-measured hunt group, split, or skill.

Interaction	Description
	The VDN tracks such calls as waiting in the vector. A call is considered answered when answered by a non converse split or skill agent, not when answered by a converse split or skill agent. The converse split or skill tracks this as a separate answered call when the VRU answers. Though trunk and split or skill totals may no longer match, VDN and trunk totals match.
Call Detail Recording	The duration of a call to a VDN is recorded from when answer supervision is returned after a successful converse-on step. Unsuccessful converse-on steps do not generate ineffective call-attempt records. Converse-on steps cannot place calls; these steps simply direct a call to a hunt group.
Call Park	Calls that a converse-on step placed cannot be parked.
Call Pickup	Do not use Call Pickup with converse-on steps.
Class of Restriction	The system does not check CORs when a converse-on vector step routes a call to a split.
Conference	You cannot conference a call routed by a converse-on step.
Direct Department Calling	You can administer a converse split or skill as a DDC split or skill.
Distributed Communications System	If an incoming DCS call is placed to a vector with a converse-on step, the caller's DCS extension is sent to the VRU.
Expert Agent Selection	converse-on steps can place calls to a skill hunt group.
Hold	An agent answering a converse call can put the call on hold, but the caller does not hear music on hold. If a call is queued to a backup split or skill before it was sent to the VRU and a non converse split or skill agent answers the call on hold, the agent who placed the call on hold is dropped, and the caller connects to the answering agent.
Hold - Automatic	Automatic hold applies to converse-on calls.

Hunt Groups

A **`converse-on`** step can deliver a call to a vector-controlled or AUDIX hunt group, ACD split, agent skill, or message center.

Interaction	Description
ISDN	You can administer a converse-on step to send a caller's calling party/ billing number (CPN/BN) to the IVR system using the caller keyword.
Intraswitch CDR	If a converse-on call is answered and either the caller or the VDN associated with the call is administered for intraswitch recording, timing for the call is started and the CDR record shows "calling party to VDN" as the originating and answering parties.
Line-side T1 connectivity	T1 connectivity between the switch and the IVR system is supported for VRI. The DS1 board must be a TN767E (or later) or TN464F (or later). Administer all converse agents as DS1FD-type stations. Operation of the converse step using Line-side T1 is identical to that over a tip/ring line. In particular, delay-timing and outpulsing speed is the same as for analog lines. T1 connectivity to the IVR system is supported only in the United States and Canada.
Look-Ahead Interflow	If an incoming call or a call routed by a converse-on vector step is answered by a VRU, or is queued to the converse split or skill while a Look-Ahead Interflow call attempt is outstanding, the attempt is accepted.
Message Center	Converse-on steps can deliver calls to message hunt groups. Such calls are handled as direct calls to the message hunt group. If a call is forwarded to a VDN and a converse-on step delivers it to a message split, it is handled as a redirected call. A converse-on step can queue a call to three different skills and then to a converse skill group or split.
Music-on-Hold	During the data return phase of a converse-on step, the caller is placed on hold, but does not hear music.
Non vector-controlled splits	A converse-on step cannot route a call to a non vector-controlled split.

Interaction	Description
Queuing	Converse-on calls queue when they are delivered to busy hunt groups. Call Vectoring audible feedback is not disconnected while a converse-on call is queued. If a converse-on step is run while a call is queued to a non-converse split or skill, the call remains in queue, even after being answered by the VRU. converse-on steps can queue calls at one of four priority levels: low, medium, high or top. You administer the queue priority of a call on the converse-on step.
R2-MFC signaling	R2-MFC signaling trunks can send ANI to VRUs using the ani data item on theconverse-on step.
Recorded announcement	Use VRI to increase the system's recorded announcement capacity by offloading some recorded announcements to a VRU. Using the converse-on step, redirect callers to a group of VRU ports by passing the number of the announcement to be played. The IVR system can play any announcement on any port. Although only one caller can be connected to each port, up to 48 callers can be connected simultaneously to the IVR system. The maximum number of callers that can be connected to a VRU simultaneously varies with each VRU.
Redirection on No Answer (RONA)	If a converse-on step calls a hunt group with "no answer timeout" administered, and the call rings an agent/port for longer than the timeout interval, the call redirects and the agent/port is put into AUX work mode (or logged out if the agent is an AAS member). With RONA, the call is requeued to the split or skill. The call cannot requeue to the split or skill if it is an AAS with all agents logged out or if the queue is full. If the call cannot be requeued, the converse-on step fails, a vector event is logged, and processing restarts at the next vector step.
Service Observing	Calls delivered by a converse-on step can be observed. To prevent the observer from hearing tones associated with data being sent to the VRU, the observer is not connected to the call until after data is

Interaction	Description
	passed. If the VRU returns data, the observer is put in service-observing-pending mode and the caller is put on hold while the data is sent. When the converse-on session ends and the VRU drops the line, the observer remains in service-observing-pending mode and waits for the next call. In addition, the observer observing a VDN does not hear data being sent. After data is sent, the observer rejoins the call. Do not administer a service observing warning tone because the warning tone may interfere with the interaction between the IVR system and the caller.
System measurements	System measurements track converse-on calls to hunt groups.
Touch-tone dialing	A caller can use touch-tone dialing while digits are passed in a converse-on session. The data is not corrupted. The system does not collect the dialed numbers as dial-ahead digits. After the system sends digits to the IVR system, a caller can enter touch-tone digits at the IVR's prompt. After the IVR system has returned data to the system and an additional collect <#> digits vector step is run, a caller can enter a touch-tone response to a system prompt.
Transfer	A call delivered by a converse-on step cannot be transferred. If an attempt to transfer a converse-on call is made, a vector event is logged, the line to the IVR system is dropped, and processing restarts at the next vector step. If a human agent tries to transfer a call, the transfer fails and the agent reconnects to the call.
Transfer out of AUDIX	If a converse-on step delivers a call to an AUDIX hunt group and the caller tries to transfer out of AUDIX, the transfer fails and processing continues at the next vector step.
Uniform Call Distribution (UCD)	You can administer a converse split or skill as a UCD split or skill.
VDN display override	If a call that accesses multiple VDNs encounters a converse-on step that passes

Interaction	Description
	vdn, normal display override rules determine which VDN number is sent to the VRU.
Vector-controlled splits/skills	Converse-on steps can deliver calls only to skills or vector-controlled splits.

VuStats

About VuStats

VuStats presents call center statistics on phone displays. Agents, supervisors, call center managers, and other users can press a button and view statistics for agents, splits or skills, VDNs, and trunk groups.

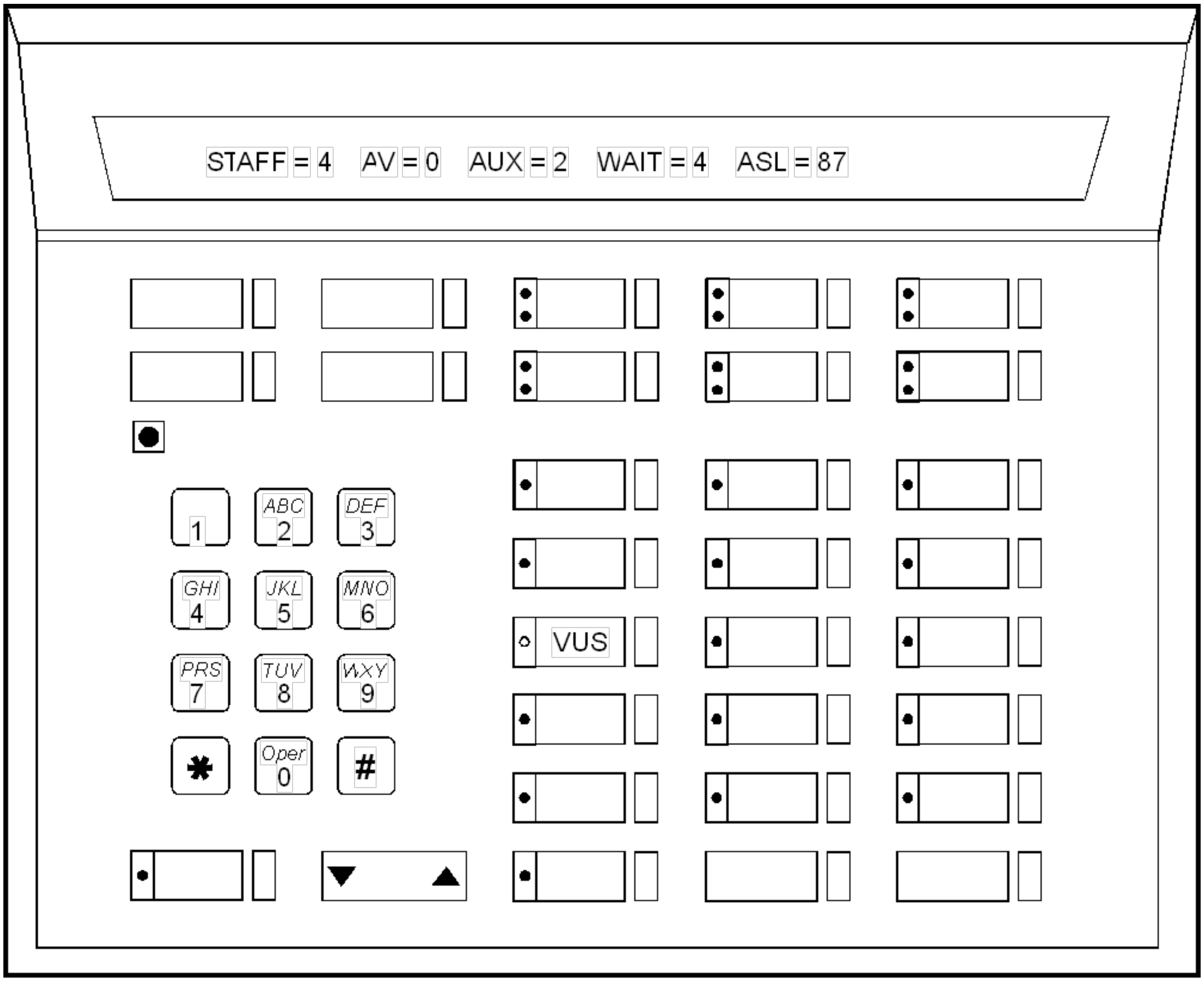
These statistics reflect current information collected during the current BCMS interval, information collected since the agent logged in or since the day began, or historical data accumulated over an administered number of intervals. The information is limited to 40 characters displayed at a time. VuStats can display on demand or update periodically.

With VuStats, anyone who is using a telephone with digital display can view BCMS statistics, which are otherwise available only on BCMS reports or management terminals. These statistics can help agents monitor their own performance or can be used to manage splits, skills, or small call centers.

Note:

Although VuStats can run with either BCMS or CMS enabled, neither is required.

The following figure illustrates a Callmaster with a VuStats display.



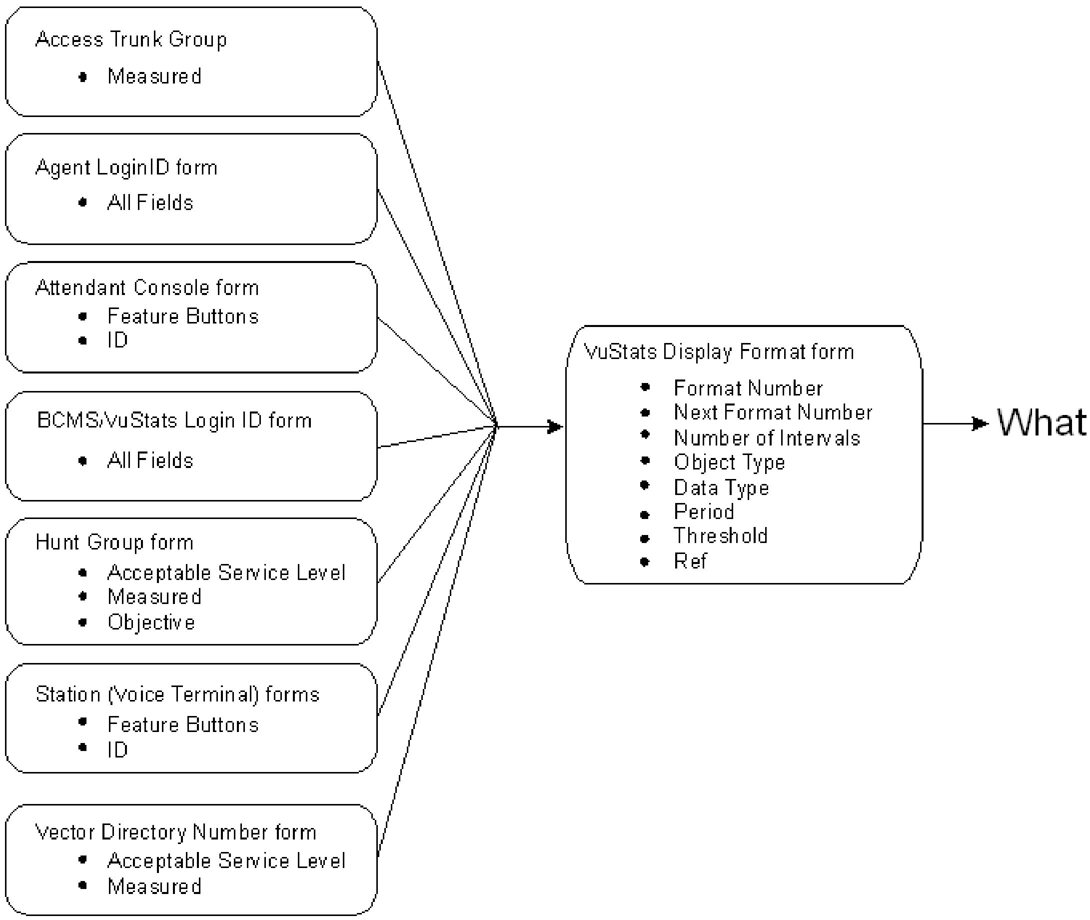
callmstr CJL 061896

VuStats detailed description

VuStats forms and fields

The following forms and fields determine information that VuStats displays.

Forms that determine what information appears on the VuStats display



Data type

Data type defines what data is displayed for an object type. For example, for an agent object type, VuStats can display information agents are interested in, such as the total number of calls the agent has answered since login, the average time the agent has spent on ACD calls, the number of agents available to receive calls for a split or skill, and the percent of calls within the acceptable service level.

For split or skill object types, VuStats can display split or skill description and performance information, such as average speed of answer, number of calls waiting, and agent work states. VuStats can also display an objective, acceptable service level, or percent of calls answered within the acceptable service level for a split or skill.

For more information, see the data types tables in ACD Call Center screens.

Period

VuStats can show statistics that have accumulated for the day, or for an administered number of intervals. For example, if you administer VuStats to display the number of ACD calls for the past 4 completed intervals, it displays the number of ACD calls received in the past 2 hours (1/2-hour intervals) or 4 hours (1-hour intervals) plus those completed during the current interval. Using historical data can affect processor occupancy, depending upon the number of active users, their update rates, and the number of historical data types.

With agent or agent-extension object types, shift data is available for the number of ACD calls answered, the average ACD talk time, and AUX work mode time by reason code for an agent. You can clear shift data at midnight or the next time an agent logs in.

Threshold

Many data types can be administered with a threshold comparator and value. When the condition defined by the threshold is true, and the data type is shown on the display, the VuStats button lamp flashes. For example, suppose a format is created in which the oldest call waiting data type is administered with a threshold of >= (greater than or equal to) five minutes. Whenever that VuStats format is displayed, if the oldest call in queue has been waiting for five minutes or longer, the VuStats lamp flashes on the phone. Each time the display updates, the threshold is checked for each data type being displayed.

Format description

Use Format Description to create labels on the display to identify data. For example, in the example figure Callmaster with VuStats display on page 0 , *AUX*= identifies the data type "split-agents-in-aux-all" (that is, the number of agents currently in AUX work mode for a specified split or skill). Text appears on the display exactly as you enter it in the field. Text is optional.

Because of the 40-character limit, use abbreviations when possible. For example, use `S=` to indicate split number.

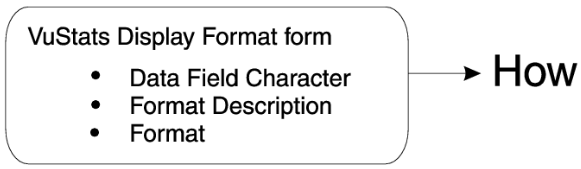
Display linking

Link display formats to increase the amount of information users can view. For example, link a display of information for an agent's first split or skill to a display of information for the agent's second split or skill. Or, link a display of information about the work states of all agents on a split or skill linked to another display of information about calls waiting, number of calls abandoned, or oldest call waiting for the split or skill.

If you use display linking, assign a Next button on agent telephones.

How the information looks

The following fields on the VuStats Display screen determine how information looks on the VuStats display.



VuStats statistics appear on the second line of 2-line DCP telephone displays or on the first line of 1-line DCP telephones and all BRI telephones. On telephones with 2 x 24 displays, the display automatically wraps to the second line of the display. When VuStats is activated, it overwrites and cancels any display feature on the second line of a 2-line display and on the first line of a 1-line display.

You define the following format information on the VuStats Display Format screen:

- Labels for data types and the amount of space reserved for data
- Order in which data types appear on the display
- Format for time-related data types
- Display links

When the information updates

The following forms and fields determine when VuStats displays update.

Screen	Field
Feature-Related Systems-Parameters	**BCMS/VuStats Measurement Interval**
VuStats Display Format	• **Update Interval** • **On Change** • **Display Interval** • **On Change** • **Display Interval**

Most display features that use the second line of a 2-line display or the first line of a 1-line display overwrite and cancel VuStats. Reason codes and Call Work codes only suspend VuStats; when the prompt is removed, the VuStats display reappears.

Agents press the normal (exit) button to clear the VuStats display.

Administer VuStats to display information until agents press the normal button or another operation overwrites the VuStats display, or administer VuStats to display for an interval of 5, 10, 15, or 30 seconds.

You can also administer VuStats to update displayed statistics every 10, 20, 30, 60 or 120 seconds or every time an agent changes work mode or a BCMS Measurement Interval is completed, or not update at all.

VuStats considerations

Some VuStats data is accumulated for an agent's login session. This shift data clears either at midnight or the next time the agent logs in depending upon how the system is administered. If the data clears at login and agents log out to go to lunch, the system clears their accumulated data when they log back in after lunch.

To accumulate a full day's statistics, you can require agents and supervisors to keep a running total of all their login sessions, or, to avoid this, use historical data, require agents to use AUX work mode when temporarily unavailable, or administer the system to clear shift data at midnight.

VuStats interactions

Interaction	Description
BCMS	You must have BCMS activated to receive BCMS reports. VuStats displays data collected by BCMS, but BCMS need not be enabled for you to use VuStats.
Call Prompting	When Call Prompting digits are displayed, VuStats is canceled. When an agent reactivates VuStats, the VuStats display overwrites the Call Prompting display.
Call Work Codes (CWC)	The CWC-display prompt suspends VuStats, so when the CWC prompt is removed, the VuStats display reappears. If VuStats is activated while a CWC is being entered (that is, the pound (#) sign is not yet dialed), the CWC display is overwritten. The CWC must be reentered.
Change skills	An agent changing skills automatically cancels VuStats. Display of the new skills overwrites the VuStats display. When the agent reactivates VuStats, the VuStats display overwrites the new skills display.

Interaction	Description
CMS:	Moving an agent from one split or skill to another does not affect the ID assigned to the vu-display button. If an agent is moved from one split or skill to another, the system does not associate VuStats buttons from the agent's previous split or skill to the new split or skill. Therefore if you must frequently move agents between splits/skills, do not associate agents' VuStats buttons with a specific split or skill. Instead, associate the VuStats button with the agent format (without an ID) on each agent's phone and use a split or skill reference to view the agent's split or skill.
EAS-PHD	When you have EAS-PHD enabled, VuStats can provide statistical data for all twenty skills. However, agent statistics by skill (agent or agent-extension object types) are available only for the current interval or for the shift-acd-calls and shift-average-acd-talk-time data types.
Integrated Directory	If an agent activates Integrated Directory, VuStats is automatically cancelled. The Integrated Directory display overwrites the VuStats display and the VuStats button extinguishes. When VuStats is reactivated, the VuStats display overwrites the Integrated Directory display.
Queue-Status Indications	The queue-status button display automatically cancels VuStats. When VuStats is reactivated, the VuStats display overwrites the queue-status display.
Reason Codes	Using certain VuStats data types, you can report real-time and historical AUX work mode time by reason code or AUX work mode time summed for each reason code. The reason codes display prompt suspends VuStats; when the reason codes prompt is removed, the VuStats display reappears.
Service Observing	On telephones with a 1-line display, the Service Observing button display automatically cancels VuStats. When VuStats is reactivated, the VuStats display overwrites the Service Observing display.

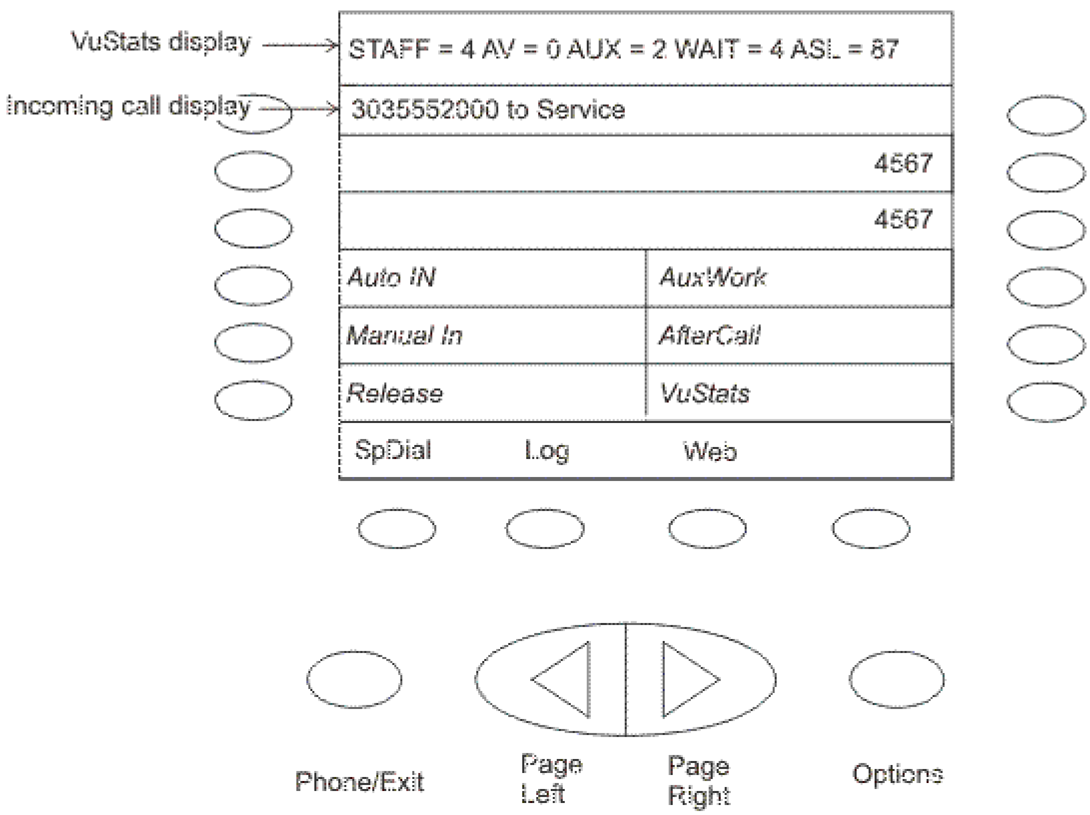
VuStats display interactions and modifications for IP telephones

If you have Communication Manager Release 3.1.2 (load 632) or later and are using a 4610, 4620, 4621, 4622, or 4625 type IP telephone:

- For incoming calls, the incoming call is seen automatically in the incoming call display line (second from top), and the top information display line continues to display the VuStats after a slight flicker.
- For outgoing calls, the top VuStats display line is suspended when the agent originates an outgoing call. The digits dialed are echoed on the associated incoming call display line. When the call is being made, the VuStats appear again on the top line after a slight flicker.

Other single-display-line sets (including the 2400 series) still function in the same way as they did prior to Release 3.1.2 (load 632).

The VuStats and incoming call information displays on these types of telephones are depicted in the following diagram of a 4622 telephone display.



Some situations could result in VuStats not being displayed, even though the**VuStats** button is highlighted for active VuStats mode. With certain VuStats formats, the VuStats display line does not display during agent login or logout, during any other Feature Access Code operation, or during an off- and on-hook sequence (such as a misdialed number). In these cases, the VuStats display line is restored on the next successful received or placed call or when the agent presses the**VuStats** button. As is normal with single-display-line sets, VuStats needs to be deactivated to see the Caller-Info (collected digits) display.

Feature-Related System-Parameters screen

Screen	Description
ACD Login Identification Length	If you are not using EAS, enter a number (1-9) that identifies the length of agent login IDs used by BCMS/VuStats. If you are not using BCMS/VuStats login IDs, accept the default 0. This field defines the ACD login ID length and the BCMS login ID length, so you must coordinate with the BCMS administrator before changing this field.
BCMS/VuStats Measurement Interval	This interval determines how frequently BCMS polls and records data for BCMS reports and VuStats displays. Set this field to half-hour or hour. If you specify hour, an entire day of traffic information is available for BCMS history reports. Otherwise, only half a day is available.There is a maximum of 25 measurement intervals, including the current interval.
BCMS/VuStats Abandon Call Timer	Set this field to 1-10, or leave blank. This value is the number of seconds a call can last and still be recorded as an abandoned call. For example, if you set this field to 5, a call could last up to 5 seconds and be recorded as abandoned. Thus, very short calls are not included as ACD calls in BCMS and VuStats statistics. Abandoned time is measured from the time the call is answered until the agent hangs up. Any time an agent is on a call that is within the abandon call timer value is recorded as total AUX time with the default reason code. Use this timer if your central office does not provide disconnect supervision.
Validate BCMS/VuStats login IDs	Set to n to allow entry of any ACD login of the proper length. Set to y to allow entry only of login IDs that have been entered on the BCMS/VuStats Login-ID screen.
Clear VuStats Shift Data	Set to on-login or at-midnight to specify when shift data for an agent is cleared.

Agent Login ID screen for VuStats

Administer agent login IDs for EAS. With EAS, VuStats accesses agent and agent-extension object type information based on agent login ID. Agents logging in agent IDs (administered on this screen or BCMS/VuStats Login ID screen) can view their own statistics on any VuStats phone they are using. If agent IDs are not administered, VuStats displays only statistics collected for the agent's extension.

Trunk Group screen

For each trunk group that will have VuStats display statistics, set **Measured** to `internal` or `both`. Specify internal to record statistics for BCMS/VuStats. Specify both to record statistics for BCMS/VuStats and CMS.

Attendant Console screen

Administer a VuStats feature button (vu-display) to allow an attendant to display VuStats statistics. There is no limit to the number of VuStats buttons that can be administered.

Field	Description
Fmt	When you assign VuStats feature buttons, an Fmt field appears. You can associate a VuStats feature button with a particular display format. The Fmt value identifies the VuStats format used to display the information. Specify 1 - 50 in the Fmt (1 is the default format).
ID number	Optionally administer an ID number for each vu-display button. Use the ID number to define the agent, split or skill, trunk group, or VDN that the VuStats display will describe. The ID can be an agent login ID or extension number, a split or skill or trunk group number, or a VDN extension. For example, a vu-display button administered with split or skill ID 6 is used to view statistics for split or skill number 6. Do not administer IDs for VuStats displays with the agent object type. Agent object type displays are limited to statistics for the logged-in agent. IDs allow supervisors and agents to bypass entering an agent extension, split or skill, or VDN number when viewing statistics. IDs can also be used to limit access to certain statistics to designated phones.

BCMS/VuStats Login ID screen

Administer agent login IDs if you do not have EAS. BCMS/VuStats login IDs can be used to track statistics by specific agent rather than extension number. Use any character, except a space, as a placeholder for data in Format Description text. The default is “$”. Each character holds a place for one character of data.

Hunt Group screen

Field	Description
ACD	Set this field to `y`.
Acceptable Service Level	Specify the number of seconds within which calls to this hunt group are answered. Calls answered within this time are considered acceptable. BCMS and VuStats use this value to determine the percentage of calls that meet the acceptable service level.
Measured	Set this field to `internal` or `both`. Specify internal to record statistics for BCMS/VuStats. Specify both to record statistics for BCMS/VuStats and CMS.
Objective	Specify an objective, or goal, for the split or skill. Examples include an agent objective of a number of ACD calls to be handled, an average talk time, or a percent of calls to be answered within the acceptable service level.

Station screen

Administer a VuStats feature button (vu-display) to allow agents to display VuStats statistics. For more information, see Attendant Console screen on page 419.

Vector Directory Number screen

For each VDN that has statistics displayed by VuStats, administer the following fields:

Field	Description
Acceptable Service Level	Specify the number of seconds within which calls to this VDN are answered. Calls answered within this time are considered acceptable. BCMS and VuStats use this value to determine the percentage of calls that meet the acceptable service level.
Measured	Set this field to internal or both. Specify internal to record statistics for BCMS/VuStats. Specify both to record statistics for BCMS/VuStats and CMS.

VuStats Display Format screen

For definitions related to completing this screen, see the VuStats Display Format screen.

Zip Tone Burst for Callmaster Endpoints

About Zip Tone Burst for Callmaster Endpoints

With Callmaster phones, this feature allows you to apply only one burst auto-answer zip tone for calls to ACD agents instead of the two bursts that are applied by default. In the **Zip Tone Burst for Callmaster Endpoints** field Call on page 13 of the Feature Related System Parameters screen, you can apply only one burst auto-answer zip tone for calls to ACD agents with Callmaster phones instead of the two bursts. This option eliminates the 2nd burst of zip tone to reduce the time it takes for the agent to start conversation with the caller and to reduce the possibility of the agent and the caller to hear “open mike” background noise between the first and second tones. This option applies to zip-tone applied for ACD calls with the station/ agent ID auto-answer option set to “acd” and for the ICI (“tweedle-dee”) tone applied for auto-answer non-ACD calls with the “all” setting.

Note:

Use this option only when the agent always hears enough of the single burst auto-answer to recognize that a call is being delivered. The default entry is “double” to retain existing two burst operation while the “single” entry reduces the zip tone application to a single burst.

Following are the two valid entries in this field:

Valid Entry	Usage
double	Default - retains existing operation which applies two bursts of zip tone for auto-answer ACD calls and two burst of ICI “tweedle-dee” tone for non-ACD auto-answer calls to ACD agents when the station set type is one of the Callmaster series.
single	Option to eliminate the 2nd burst of zip/ICI tone reducing time for agent to start conversation with the caller and possibility of the agent and the caller hearing “open mike” background noise between the first and second tones. Use with a Callmaster station type when the agent can always hear enough

Valid Entry	Usage
	of the single burst auto-answer to recognize that a call is being delivered.

Chapter 3: ACD and call management systems

About BCMS

The Basic Call Management System (BCMS) is a software package, residing on the PBX/ communication server, used to provide real-time and historical reports to assist in managing ACD splits/skills, agents (extensions), trunk groups and VDNs (G3 only). These reports, provided by the system, are a subset of those reports available with the CMS adjunct.

About CMS

The Avaya Call Management System (CMS) is an adjunct that collects specific ACD data on measured splits/skills, measured agents, measured extensions, measured trunks and measured trunk groups for reporting purposes. If Call Vectoring is purchased, ACD will report on measured VDNs and Vectors. CMS provides call management performance recording and reporting. It can also be used to perform some ACD administration. CMS is used by customers to determine how well their customers are being served (in other words., speed of call answers, number of calls) and how efficient their call management operation is (in other words, agents versus traffic requirements).

This section includes the following topics:

How CMS works with ACD

To collect information on ACD, CMS must be able to communicate with the ACD resident in the communication server. The communication server to CMS Platform communication consists of electronic messages sent back and forth between the ACD communication server and the CMS Platform using a data link. There are two types of messages:

- Translations - Tells CMS the configuration of the ACD. This includes what data is measured (to be collected) and the ACD assignments.
- Status Changes - Tells CMS when the states of agents or trunks change due to call activity. Occurrences are counted and durations are tracked.

Note:

CMS can also be used to change configurations within the ACD. Therefore, CMS can at times send translations back to the PBX.

Data measured by CMS

CMS measures data on ACD splits, agents, extensions, trunks and trunk groups. If Call Vectoring is purchased, ACD will report on VDNs and Vectors. The following table summarizes the types of data generated by the ACD and measured by the CMS.

Data type	Parameters
Agents	Agent States
Splits/Skills	Events Workload Distribution Split/Skill call totals

Data type	Parameters
Trunks	Trunk states
Trunk Groups	Events Workload Distribution Trunk call totals

Assigning CMS measurement of the ACD

CMS collects data on splits/skills, agents, extensions, trunks, trunk groups, VDNs, and vectors. However, for CMS to collect data, the appropriate items (splits/skills, extensions, and so on) must be identified as measured on the communication server.

Individual splits and trunk groups are assigned to CMS measurement through communication server administration. Extensions are measured by virtue of their assignment to measured splits. Trunks are measured by virtue of their assignment to measured trunk groups. The number of measured splits cannot be changed using the CMS ACD Status screen.

Measured splits need not be numbered sequentially. VDNs are measured individually. All vectors are measured.

Things to know about CMS

The CMS ACD Status window lists the total number of measured splits/skills, extensions (Agent Positions), trunks, and trunk groups established in an ACD. For more information, see Avaya CMS Administration.

You can add, delete, or change measured trunks, trunk groups, agent extensions, agent login IDs, VDN extensions, splits, and skills without busying out the link to CMS and losing CMS data.

Communication server features that affect CMS data

There are several communication server features that affect CMS data, such Conference, Transfer, Multiple Call Handling, Call Pickup, Intraflow, Interflow, Redirection on No Answer, Phantom Abandon Call Timer, Move Agent While Staffed, Expanded Agent Capabilities, Best Service Routing, and Universal Call ID.

Hold, Conference, and Transfer

CMS tracks any type of call an agent puts on hold by pressing the Hold button, dialing the hold access code, pressing the Conference or Transfer button, or flashing the switchhook. Information on all calls (split or skill ACD, direct agent ACD, and extension calls) and the time spent on hold is stored in agent database tables. Information on split or skill calls only and the time spent on hold is stored in split or skill tables.

About MCH

Avaya communication servers have options to the Multiple Call Handling (MCH) feature that can force agents to receive one or more ACD calls with other ACD calls or extension (non-ACD) calls on hold or active. For these forced options, talk time (and not ringing time) accumulates until the agent puts the current call on hold or releases it.

With Multiple Call Handling, an agent can put a call on hold and press the manual-in or auto-in button to receive another ACD call. When multiple calls are on hold at the same time, hold time accumulates for each call on hold, and the total hold time can exceed clock time. For example, if two calls are on hold for 5 minutes each, 10 minutes of hold time accumulates.

Call Pickup

When an agent uses the Call Pickup feature to pick up an ACD call that rings at another agent's extension, CMS tracks the call as an AUX-IN call for the agent picking up the call. The split or skill of the agent originally called is credited with an outflow call, even if the agent who picked up the call is in the same split or skill. If an agent is logged into more than one split or skill, the call is counted for the split or skill the agent has been logged into the longest. Thus, when Call Pickup is used, CMS does not count the call as an ACD call, even though the call queued to a split or skill and was answered. Various other types of data associated with ACD calls (for example, Percent Answered Within Service Level and Average Speed of Answer) will also not include data on calls answered using the Call Pickup feature. Because the split or skill of the agent originally called is credited with an outflow call, the call counts against the Percent Answered Within Service Level for that split or skill.

Intraflow/interflow with CMS

When a call is intraflowed or interflowed from a split or skill, CMS counts the call as an outflow call for the split or skill. If a call is intraflowed into a split or skill, CMS counts the call as an inflow call for the split or skill. CMS counts interflowed calls as ordinary incoming calls for the split or skill. However, because calls can be intraflowed/interflowed to destinations that are not splits/skills or are not measured by CMS, an outflow call from a split or skill will not always show a corresponding inflow call for another split or skill. Conversely, because calls can be intraflowed/interflowed into a split or skill from originating locations that are not measured by CMS, an inflow call to a split or skill may not show a corresponding outflow from another split or skill.

If an intraflowed/interflowed call connects to an agent in the destination split or skill, that call is counted as an ACD call for the split or skill.

A dummy split or skill may be established which intraflows calls to another split or skill. For CMS to count outflow calls for dummy splits/skills, intraflow should be established using the Call Forwarding feature. If Call Coverage is used to intraflow calls, at least one agent must log into the dummy split or skill and go into ACW, and the call must queue to the dummy split or skill for at least one ring cycle for an outflow call to be counted.

For communication servers with the Call Vectoring feature, intraflow and interflow work differently, and CMS data related to intraflow and interflow are recorded differently.

About RONA

When a ringing call times out and is requeued to the same split or skill by the Redirection On No Answer (RONA) feature, Avaya CMS counts an outflow and an inflow for the split or skill. That is, the redirected call appears as two offered calls to the split or skill. If the call redirects from ringing to a VDN, there is outflow from the initial VDN and from the split or skill. If the call was in another VDN prior to redirection to another VDN, then there is inflow to that VDN.

Also, NOANSREDIR is incremented for the split or skill and the VDN. For CMS R3V2 and newer, the database item NOANSREDIR is also incremented for split or skill and for VDN, if the call is in a VDN. If a split or skill is set up so that split or skill calls do not redirect back to the split or skill except by way of the Redirection On No Answer feature, the unique calls offered to the split or skill can be calculated by subtracting the value of NOANSREDIR from CALLSOFFERED.

If a call redirects from ringing to a VDN, there is outflow from the split or skill and, if the call was in another VDN, there also is inflow to the new VDN and outflow from the initial VDN. The NOANSREDIR is incremented for split or skill and VDN.

Phantom abandon call timer

CMS can collect information about phantom abandon calls. When this capability is enabled, calls with a talk time (duration) shorter than the administered value (1 - 10 seconds) are counted as phantom abandon calls. Setting the timer to zero disables it. CMS uses the PHANTOMABNS database item to store the number of phantom abandon calls.

This capability is important in areas where the public network switches do not provide disconnect supervision. Without this capability, short-duration calls that queue to a split or skill and are answered by an ACD agent or other answering position are counted as ACD calls, even if the calling party hangs up before the call is answered. This type of call is called a phantom or ghost call.

About moving an agent while staffed

A staffed agent can be moved between splits or the skill assignments for staffed agents can be changed. If the agent has any call on the telephone or is in ACW, then the move cannot take place immediately, but is pending the agent telephone going idle (all calls have been terminated), or the agent changing out of the ACW mode.

CMS provides two real-time database items in the agent data, MOVEPENDING and PENDINGSPLIT, that can be accessed by using custom reports to provide information about whether agents have moves pending and, if so, the split or skill to which they are being moved. Note that in the case that the agent's skills are being changed and the change adds more than one skill, the PENDINGSPLIT item will show the first skill that is being added. It is also possible for MOVEPENDING to be set, but for PENDINGSAPLIT to be blank (or 0). This can happen, for example, when the link to the communication server comes up and a move is pending for an agent. CMS will be notified by the communication server that the move is pending, but PENDINGSPLIT will not be set.

Expanded agent capabilities

Expanded Agent Capabilities allow EAS agents to have up to 20 or 120 skills (depending on platform) assigned. Each skill may be assigned a level from 1 to 16, where Reserve 1 and Reserve 2 are the highest levels and 16 is the lowest. (The numeric level replaces the skill type p or s used in earlier G3 EAS releases.) Agents may have a call handling preference based either on the skill level, meaning that the agent will serve calls waiting for his or her highest level skill before serving calls waiting for any lower level skills; or based on greatest need, meaning that the agent will serve the highest-priority, oldest call waiting for any of his or her skills, or percent allocation, based on the percent distribution of calls among the agent's skills.

The expanded agent capabilities feature also allows the specification of the skill to be used for the agent's direct calls. This also allows specification of the level for the direct agent skill, which, in conjunction with the agent's call handling preference, may affect the order in which a direct agent call is delivered to an agent. That is, direct agent calls need to be delivered for all skill ACD calls. A concept introduced in R3V5 CMS, that of the top skill, can be useful in EAS implementations that use skill level call handling preference for agents. An agent's first administered, highest level skill is the agent's top skill, since it is for this skill that the agent is most likely to handle calls. This skill can count on the agent.

Database items track the number of top agents in skills, as well as the time top agents spent available and in AUX.

The expanded agent capabilities on the communication server include an increased number of measured splits/skills to 600 and an increase in the number of measured agent/split or agent/skill pairs to 10,000 for the G3r processor, as well as new options for Most Idle Agent (MIA) call distribution. The new options allow selection of MIA distribution across skills, rather than for each skill, and selection of whether agents in ACW are or are not included in the agent free list. These options have no direct impact on CMS, since CMS does not keep track of the most idle agent.

About Best Service Routing (BSR)

Best Service Routing (BSR) allows calls to be balanced at a single site or between multiple sites. BSR is enhanced multi-site routing that provides new call vectoring functions that build upon the Look-Ahead Interflow feature to route a call to the best skill on a single ECS or to the best skill in a network of Avaya communication servers.

The best skill is defined as the local skill or remote ECS that offers the shortest waiting time for the call in a call surplus (calls queued) situation for the application. The waiting time is calculated using the Expected Wait Time (EWT) predictor, and can be adjusted by the user. In an agents available situation, the best skill is determined based on the assigned available agent strategy. BSR data is tracked in the vector, VDN, and call history tables.

BSR can be configured for either single-site or multi-site operation. Single-site BSR compares splits/skills on the Communication Manager where the BSR resides to find the best resource to service a call. Multi-site BSR extends this capability across a network of Communication Managers, comparing local splits/skills, remote splits/skills, or both, and routing calls to the resource that provides the best service.

About UCID

Universal Call ID (UCID) is a unique tag that is assigned to a call. The tag allows call-related data to be collected and aggregated from multiple sources and multiple sites. The UCID may then be used to group all the data from various sources about a particular call.

CMS will receive the UCID assigned to calls by a communication server when this feature is enabled. The UCID is then stored, along with data about the call itself, by the call history feature (which includes both internal and external call history). The data will be available to both Custom Reports and the Report Designer. UCID data is stored in the call history and agent trace tables.

Avaya Business Advocate with CMS

Avaya Business Advocate is a collection of features that provide flexibility in the way a call is selected for an agent in a call surplus situation and in the way that an agent is selected for a call in an agent surplus situation. Advocate also includes methods for automating staffing adjustments.

Chapter 4: Call Vectoring features

About Call Vectoring

Call Vectoring processes incoming and internal calls according to a programmed set of commands. These commands, called vector commands, determine the type of processing that calls receive. For example, vector commands can direct calls to on-premise or off-premise destinations, to any hunt group, split or skill, or to a specific call treatment such as an announcement, forced disconnect, forced busy, or delay. Vectors can queue or route calls based on a variety of different conditions, such as most important calls can be routed to better skilled agents. For more information about routing calls to agents by skill level, see *Administering Avaya Aura™ Call Center Features*.

There are many different applications for Call Vectoring. However, it primarily is used to handle the call activity of ACD splits/skills.

For more information about administering call vectoring, see Best Service Routing, Network Call Redirection (NCR) on page 293, and Interflow and Intraflow interactions on page 250.

For a description of routing calls a detailed description of Call Vectoring, see *Programming Call Vectors in Avaya Aura™ Call Center*.

Adjunct (ASAI) Routing

About Adjunct Routing

Adjunct Routing provides a means for an Adjunct Switch Application Interface (ASAI) processor to specify the destination of an arriving call when it encounters an **`adjunct routing link`** vector command during vector processing. An adjunct is any processor that is connected to an Avaya Communication Manager (sometimes called a "switch") that can use the ASAI protocol. The adjunct makes a routing decision according to caller information and/or agent availability, and returns a call route response to the switch.

The switch provides information in an ASAI route request message that the adjunct application uses to access a database and determine a route for the call. In a typical application, the ASAI adjunct might use the dialed number, the Calling Party Number/Billing Number (CPN/BN), or the digits that are collected by way of Call Prompting to access caller information and thereby determine an appropriate call route.

Adjunct Routing can be used in conjunction with the Call Prompting and Look-Ahead Interflow features. When combined with one of those features, the following rules apply:

- When combined with Call Prompting, Adjunct Routing can pass up to 16 digits that are collected from the last relevant **`collect digits`** vector command.
- When combined with Look-Ahead Interflow (LAI), Adjunct Routing can pass the LAI information element or other call center-related data (with enhanced Information Forwarding) that was passed from the originating switch in the message or associated with the call from the local switch.

Considerations for implementing adjunct routing

You should understand the following considerations before you implement a call center solution that uses the Adjunct Routing feature:

- An adjunct specified in an **`adjunct routing link`** command can route a call to an internal number, an external number, a split, a VDN, an announcement extension, or a particular agent. An adjunct can also provide priority ringing, priority queuing, and specify that a route-to an agent be done as a direct agent call.
- If your specific application permits you to do so, you can include two or more consecutive adjunct routing link steps in a vector. This approach provides the following advantages:
    - Redundancy in case of ASAI link/application failure.
    - Simultaneous processing of multiple route requests, which distributes incoming call load more efficiently and results in faster call processing times. For more information, see Multiple outstanding route requests on page 445.
- Vector processing continues to occur while an ASAI route request is being processed. For this reason, the first step to follow one or more adjunct routing link steps should be either an announcement, or a wait time step that adheres to the following rules:
    - If an announcement step follows immediately after an adjunct routing link step, the announcement should not contain any information that is essential to the caller (such as further instructions), since it will immediately terminate when the switch receives a destination from the ASAI adjunct.
    - If a wait-time step follows immediately after an adjunct routing link step, it should usually specify either ringback or music (but not silence) as the feedback option, so that the caller is less likely to abandon the call.

Important:

If an ASAI link/application specified in the adjunct routing link step is out of service, the step is skipped. If the next step is not a wait-time, announcement, or adjunct routing link step, as much as six minutes may elapse before the switch determines that the adjunct application is out of service.

- The second step after the adjunct routing link step can, and often should, be implemented as a default treatment in case the host application or ASAI link is down. Speed of execution for the default treatment step (for example, route-to number 0 if unconditionally) is controlled by the following factors:
  - If the ASAI link is down, and if the first non-adjunct routing link step is either a wait-time or an announcement treatment, then the treatment step is skipped and the default step that follows the skipped treatment executes immediately.
  - If the host application is not down, the default step executes only if the adjunct does not provide a route within the time defined by the first non-adjunct step. For example, if the first non-adjunct step is an announcement, the default step executes only after the time defined by the length of the announcement is exceeded.
- When a vector contains an **`adjunct routing link`** command, and an ASAI link/application failure event occurs, special rules apply to vector processing operations that result. Adjunct Routing vectors should be designed to take these special processing operations into account. For more information, see Special vector processing considerations associated with adjunct routing on page 436.
- Since vector processing continues to occur while an ASAI call route request is processed at an adjunct, succeeding vector steps can terminate an ASAI call route request if they execute before a call route can be provided by the adjunct. Alternately, the adjunct may reject the call route request, and subsequent vector processing proceeds in a normal manner. For more information, see Vector steps that terminate an ASAI call route request on page 440.
- The **`wait-time hearing i-silent`** command is used in cases where it is important to allow the adjunct to decide whether to accept an incoming ISDN-PRI call. When this step is encountered after an adjunct routing link step, the switch does not return an ISDN PROGress message to the originating switch. This is particularly important for Network ISDN features and the Look-Ahead Interflow feature.

Receiving and implementing an ASAI call route

A switch that receives an adjunct-supplied call route performs various checks to validate the call route before it is implemented. When the adjunct-supplied route is validated, the operations that result are similar to those in effect for a **`route-to xxxxx with coverage=y`** command. The caller hears normal call progress tones and feedback, and if the call routes to an extension with no available call appearances and no coverage path, the caller hears a busy signal.

Any other features that may be in effect at the adjunct-supplied destination, such as Send-All-Calls or Call Forwarding, interact with the routed call.

Also, Look-Ahead Interflow operations are not applied when calls are routed over trunks. Instead, ASAI-routed calls are directed to their adjunct-supplied destination without waiting for call acceptance.

The processes associated with receiving and implementing and ASAI call route are described in the related sections:

Related topics:

Validation for an adjunct-supplied call route on page 434
Switch response to validated adjunct-supplied call routes on page 434
Switch response to invalid adjunct-supplied call routes on page 435

Validation for an adjunct-supplied call route

When the switch receives adjunct-supplied call route instructions, the switch validates the route according to the following process:

1. The switch verifies that the COR rules specified for the target VDN permit the call to be terminated at the adjunct-supplied destination.
2. The switch validates the following information:
    - Destination number
    - ACD split
    - TAC/AAR/ARS access code
    - Dial plan compatibility
    - Other options specified by the adjunct
3. If the ASAI adjunct specifies the direct agent call option, the destination number (agent) must be logged into the adjunct-specified ACD split.
4. If the destination for the call is external, the switch verifies that a trunk is available for the call.

Switch response to validated adjunct-supplied call routes

If the switch validates an adjunct-supplied call route, the following operations occur:

1. Vector processing in the VDN that contains the initiating **`adjunct routing link`** command terminates immediately.
2. The switch signals the ASAI adjunct that the route is accepted.
3. The switch routes the call to the destination specified by the ASAI adjunct.

Switch response to invalid adjunct-supplied call routes

If any of requirements for call route validation listed in Validation for an adjunct-supplied call route on page 434 are not met, items the following operations occur:

1. The switch discards the route.
2. The switch signals the ASAI adjunct that the route is invalid.
3. Vector processing of any other default treatment steps in the VDN that contains the initiating adjunct routing link proceeds.

Data sent with an ASAI call route request

When a call encounters an **`adjunct routing link`** command and if the call is not queued to a split, the switch sends an ASAI message that requests a call route over the specified adjunct link. The following list identifies the contents of the message, along with a comment or a brief explanation for each item:

Calling number information

The calling party number or billing number (CPN/BN) that is provided by ISDN-PRI or R2-MFC signaling facilities. If the call originates from a local switch extension, this extension is the calling number.

Originating line information (II-digits)

A two-digit code that is provided by ISDN-PRI facilities that indicates the type of originating line. This information is not provided by SIP facilities.

Called number

The originally called extension if a call is forwarded to a VDN, or the first VDN through which the call was routed if the call was not forwarded to the VDN.

If the VDN Override for the Trunk ASAI Messages feature is in effect for an incoming call, the active VDN extension (instead of the Called Number received in the SETUP or INVITE message) is sent in the Called Number for the Call Offered, Alerting, Queued, Connect, and Adjunct Route-Request ASAI Event Reports.

Routing VDN

The last VDN that routed the call to the vector that contains the **`adjunct routing link`** command.

Call identifier

An ASAI identifier that permits the ASAI adjunct to track multiple calls by either Event Notification or 3rd Party Call Control. For more information on ASAI, see Communication Manager CallVisor ASAI Technical Reference.

Enhanced Information Forwarding (related data) and Look-Ahead Interflow information (if any)

Includes the original VDN display information, the priority level of the call at the originating switch, and the time that the call entered vector processing. For more information, see Look-Ahead Interflow (LAI), and Information Forwarding on page 225.

Digits collected by Call Prompting or Caller Information Forwarding (CINFO) (if any; maximum of 16 digits)

Digits that are collected by the most recent **`collect digits`** command. For more information, see Call Prompting, ANI /II-digits routing and Caller Information Forwarding (CINFO) in the *Programming Call Vectors in Avaya Aura™ Call Center* document, and Information Forwarding on page 225.

User-to-User Information (UUI)

User-provided data that is associated with the call. If provided by ASAI, this data was provided in a 3rd-Party-Make-Call, Auto-Dial, or Route-Select message. If provided over an ISDN or SIP trunk, the data was in the SETUP or INVITE message that delivered the call to this switch. Calls that contain UUI specifically used by ASAI allow ASAI UUI to be propagated to the new call during a manual transfer or conference operation. ASAI UUI is propagated to a new call during its establishment when the agent presses the transfer/conference button the first time. If the call is transferred to a remote switch, the ASAI UUI from the first call is copied into the SETUP or INVITE message sent for the second call, in which case, the alerting event message sent to an ASAI application contains the ASAI information.

Special vector processing considerations associated with adjunct routing

When you design call vectors that include one or more **`adjunct routing link`** commands, you must be aware of a number of special operational features.

Related topics:

[Effects of ASAI link/application failure on vector processing](#) on page 436
[Simultaneous processing of vector steps and ASAI call route requests](#) on page 440
[Adjunct routing-initiated path replacement](#) on page 441
[Phantom calls](#) on page 442
[Single-step conference](#) on page 444
[Multiple outstanding route requests](#) on page 445

Effects of ASAI link/application failure on vector processing

An ASAI link failure can change the manner in which subsequent announcement or wait-time treatment steps are processed.

In the following simplified vector example, the step that follows immediately after an **adjunct routing link** command is a **wait-time** command. If the adjunct routing link step fails at either the ASAI link or adjunct application, the wait-time step is skipped.

The second step after the adjunct routing link step is often implemented as a default treatment. In the example shown above, the default treatment in step 3 is a route to an attendant. If the switch recognizes that the ASAI link or adjunct application is out of service, this step executes immediately. Otherwise, the step executes only if the application does not respond with a route within 60 seconds (the wait-time assigned in the example).

Simplified example of vector processing in an ASAI link/application failure condition

```
1. adjunct routing link 11  [link/application is down]
2. wait-time 60 seconds hearing ringback  [step is skipped]
3. route-to number 0 with cov n if unconditionally [step is executed]
4. disconnect after announcement 2000
```

Related topics:

Vector processing with goto steps in an ASAI link/application failure condition on page 437

Vector processing with goto steps in an ASAI link/application failure condition

Processing rules for a vector that includes one or more **adjunct routing link** commands and has an ASAI link/application failure condition in effect are summarized as follows:

An announcement or wait time treatment is skipped whenever one of the following conditions is true:

- The treatment step follows immediately after a failed **adjunct routing link** command
- The treatment step is the first non-goto step that follows a goto step that succeeds. In this context, a goto step is considered to succeed when the specified goto condition is true, and the call branches from the goto step to the treatment step.
- The treatment step is the first non-goto step that follows a failed goto step. In this context, a goto step is considered to fail when the specified **goto** condition is true, the call fails to branch, and control proceeds to the treatment because it is the next step listed in the vector sequence.

Note:

The treatment step is skipped even when a failed goto step that precedes it is, in turn, preceded by one or more successful goto steps.

The rules listed above for vector processing under ASAI link/application failure conditions are further illustrated in the following examples.

Example 1 - Vector processing with goto steps in an ASAI link/application failure condition

```
VDN (extension=1040  name=''Ad Route''  vector=40)
Vector 40
```

```
 1. adjunct routing link 10  [link/application is down]
 2. wait-time 10 seconds hearing ringback [step is skipped]
 3. adjunct routing link 20 [link/application is down]
 4. goto step 7 if available-agents in split 20 < 1 [step executes and condition is
false]
 5. wait-time 10 seconds hearing ringback [step is skipped]
 6. goto vector 50 @step 1 if unconditionally  [step executes, go to vector 50]
 7. goto step 10 if calls-queued in split 20 pri l > 50
 8. announcement 4001
 9. goto vector 50 @step 1 if unconditionally
10. route-to number 6000 with cov n if unconditionally
   VDN (extension=6000  name=‘‘Message’’  vector=60)
```

Based on the scenario presented in the example shown above, the following vector processing events occur:

Step 1 fails: For purposes of this example, assume that the adjunct link or application is out of service. The **adjunct routing link** command in step 1 fails.

Step 2 is skipped: Because the **wait-time** command in step 2 immediately follows an **adjunct routing link** command whose adjunct link is out of service, the wait-time step is skipped.

Step 3 fails: For purposes of this example, step 3 contains another **adjunct routing link** command whose adjunct link is assumed to be out of service. The step fails, and control is passed to the **goto step** command in step 4.

Step 4 executes: A goto step that immediately follows a failed **adjunct routing link** command is always executed. In this example, the command fails to branch because there is at least one available agent in split 20.

Step 5 is skipped: The wait-time step that follows the unsuccessful goto step (step 4) is skipped, because in an ASAI link failure condition, the first non-goto step to be processed after the first successful first goto step is always skipped if it is either **announcement** or **wait-time**. Control is passed to the **goto vector** command in step 6.

Step 6 executes: Step 6 routes the call to vector 50 (not shown), which is designed to queue the call and provide standard call treatment.

In the next example, assume that the **goto step** command in step 4 succeeds. In this context, the goto step succeeds when the specified condition is true (no agents are available in Split 20), and control is passed to step 7, where another goto step determines whether there are more than 50 calls in split 20. If the condition is true, step 7 succeeds and control is sent to step 10, where the **route-to number** command sends the call to vector 60.

The example processing events are described in the following figure.

Example 2 - Vector processing with goto steps in an ASAI link/application failure condition

```
VDN (extension=1040  name=‘‘Ad Route’’  vector=40)
Vector 40
 1. adjunct routing link 10 [link/application is down]
 2. wait-time 10 seconds hearing ringback [step is skipped]
 3. adjunct routing link 20 [link/application is down]
 4. goto step 7 if available-agents in split 20 < 1 [step executes and condition is
true]
```

```
 5. wait-time 10 seconds hearing ringback
 6. goto vector 50 if unconditionally
 7. goto step 10 if calls-queued in split 20 pri l > 50 [step executes and condition
is true]
 8. announcement 4001
 9. goto vector 50 if unconditionally
10. route-to number 6000 with cov n if unconditionally [step executes
unconditionally]

VDN (extension=6000  name=‘‘Message’’  vector=60)
Vector 60

1. announcement 4000 [
We’re sorry.  We are still unable to connect you to an agent. If you’d
      like to leave a message, please do so after the tone.
]
2. wait-time 6 seconds hearing silence
3. messaging split 18 for extension 1500
4. announcement 4010 [
We’re sorry.  We were unable to connect you to our voice mail. If
      you’d like to try to leave a message again, please do so after the tone.
Otherwise,
      please call back weekdays between  8:00 A.M. and 5:00 P.M.
]
5. goto step 2 if unconditionally
```

Based on the scenario presented in the example shown above, the following vector processing events occur:

Step 1 fails: For purposes of this example, the adjunct link or application is out of service. The **adjunct routing link** command in step 1 fails.

Step 2 is skipped: Because the **wait-time** command in step 2 immediately follows an **adjunct routing link** command whose adjunct link is out of service, the **wait-time** step is skipped.

Step 3 fails: For purposes of this example, step 3 contains another **adjunct routing link** command whose adjunct link or application is also out of service. The step fails, and control is passed to the **goto step** command in step 4.

Step 4 executes: A **goto** step that follows a failed **adjunct routing link** command is always executed. In this example, the command succeeds and branches to step 7, because no agents are available in split 20.

Step 7 executes: Again, a goto step that follows a failed **adjunct routing link** command is always executed. In this example, the command branches unconditionally to Vector 60

Step 10 executes: In this example, step 10 (route-to number) is the first non-goto step immediately preceded by one or more goto steps in an ASAI link fail condition. The step executes, because it not an **announcement** or **wait time** command.

Vector 60: Step 1 executes: The first step in this vector is an **announcement** command. In this example, this is the first step in the processing sequence to be either an announcement or wait time step. However, this step is not skipped, since it is not the first non-go to step in the processing sequence. Instead, step 10 in Vector 40 (a route-to number step) is the first non-goto step.

Simultaneous processing of vector steps and ASAI call route requests

When the switch sends a route request to an ASAI adjunct, vector processing continues for any vector steps that follow the **`adjunct routing link`** command. Therefore, non-adjunct routing link step that follows immediately after an adjunct routing link step (or multiple adjunct routing link steps in uninterrupted succession) can determine:

- The maximum length of time that the switch waits for a call route reply from the ASAI adjunct
- In some cases, whether or not the ASAI call route request is allowed to finish processing

If the next step is not a **`wait-time`**, **`announcement`**, or another **`adjunct routing link`** command, as much as six minutes may elapse before the switch determines that the adjunct application is out of service. For this reason, the recommended practice is to design vectors so that the next step to follow an **`adjunct routing link`** command is either a **`wait-time`**, or **`announcement`** command.

Related topics:

Vector steps that terminate an ASAI call route request on page 440

Vector steps that terminate an ASAI call route request

If an adjunct routing link step is followed by a wait-time or announcement treatment, and the treatment completes before an ASAI call route request is returned by the adjunct, call processing continues for any vector steps that may follow the treatment. In this case, certain vector commands will terminate the ASAI call route request when they are executed. Vector commands that terminate an active ASAI call route request include:

- **`busy`**
- **`check split`**
- **`converse-on split`**
- **`queue-to split`**
- **`collect digits`**
- **`disconnect`**
- **`messaging split`**
- **`route-to`**

If a valid ASAI call route message is received by the switch before one of the vector commands listed above can execute, the system routes the call to the destination specified by the adjunct route. Otherwise, the ASAI route request is terminated.

Note:

The adjunct can also reject a call request by negatively acknowledging the route request that is sent by the switch. When the switch receives a a route request rejection message from the adjunct, any announcement or wait-time step that is being executed is immediately terminated. Call processing then continues with the next vector step.

Adjunct routing-initiated path replacement

Path replacement for calls in queue and vector processing, using QSIG or DCS with Reroute using SSE, is available for Avaya switch software R9.5 or later. For calls that are waiting in queue or in vector processing, even if the call is not connected to an answering user, path replacement can be attempted to find a more optimal path for this call. This results in more efficient use of the trunk facilities.

When adjunct routing is used with a call, path-replacement can be initiated when the following criteria are true:

- The inbound call is over a QSIG trunk or DCS SSE trunk
- A route-select response is received from the CTI application after the **`adjunct route`** vector command has been executed
- The routing destination that is contained in the route select ASAI message is to an outbound QSIG trunk or out bound DCS SSE trunk

When all three criteria are met, the trunk is then seized and used for the call.

The ability to track a measured ACD call after a path replacement has taken place is available for CMS versions r3v9ai.o or later. Starting with the r3v12ba.x release, CMS reports a path replacement as a**rename** operation rather than a path replacement. The**rename** operation properly reports scenarios where a path replacement takes place from a measured to an unmeasured trunk facility. Avaya recommends that you upgrade CMS to r3v12a.x or later and administer all trunks associated with path replacement as**measured** by CMS to ensure better CMS tracking of path-replaced calls.

Related topics:

Example vector for adjunct routing-implemented path replacement on page 441

Example vector for adjunct routing-implemented path replacement

The following Call Vector example shows how a vector for adjunct routing can be written to trigger path-replacement at the terminating switch.

Note:

In order for a path-replacement to be attempted, the incoming and outgoing trunks that are used for the call must be administered with the **Supplementary Service Protocol** field set to `b`.

Adjunct routing-initiated path-replacement vector

```
1. announcement 5996
2. adjunct routing link 11
```

```
3. wait 20 seconds hearing ringback
4. announcement 3111
```

At the terminating (receiving) switch, the vector that is executed by the incoming call must be programmed with an **announcement**, **wait hearing music**, or **wait hearing ringback** vector command. The use of one of these commands is what makes it possible for path-replacement to take place while the call is in vector processing.

Phantom calls

A phantom call is a call that originates from a nonphysical device by way of an ASAI application and may be placed anywhere. In general, phantom calls

- Use less resources
- Are treated like voice calls

Related topics:

How do phantom calls work? on page 442
How are phantom calls used? on page 442
How do phantom calls affect Call Vectoring? on page 443
Phantom call administration on page 444

How do phantom calls work?

First, an application requests a phantom call by sending an ASAI third_party_make_call or auto_dial capability message to the switch.

If the specific extension of a station Administration Without Hardware (AWOH) is specified as the originator, the switch places the call from that extension if the extension is available.

It is also possible to specify a hunt group extension with members that are AWOH extensions as the originator.

How are phantom calls used?

Applications use phantom calls when they need to originate a call without using a physical device and thus not use extra resources. For example, applications may need to:

Action	Description
Reserve a queue slot	Many call centers handle incoming requests as voice, video, data, voice messages, faxes, and e-mail. Agents who work in these call centers need to handle the mix of requests. However, a single queue needs to manage and distribute the work load for these agents. For each non-voice request, the application can place a phantom call into the queue. When the phantom call reaches the head of the queue, it is delivered to the agent. The agent is then given the corresponding work item on the desktop, for example, the fax.

Action	Description
Conference control	Multiple parties (both internal and external) can be conferenced into a call. The initial call is placed as a phantom call. When answered, the call is placed on hold by the application and another phantom call is made. The two calls are then conferenced together. This process is repeated until all parties are added to the call.
Help with trunk-to-trunk transfers	Working with the Single Step Conference feature, applications can use the phantom call feature to help with trunk-to-trunk transfers, that is, transferring a trunk-to-trunk call to another trunk. For information about single step conferences, see Communication Manager CallVisor ASAI Technical Reference.
Alerts (wake-up, maintenance, and security)	Applications can use phantom calls to alert users of various conditions such as wake-up, maintenance, or security.

How do phantom calls affect Call Vectoring?

Because phantom calls can be directed anywhere, you must properly configure the application and the switch to ensure that the vector commands that are executed for these calls make sense. For more information, see Communication Manager CallVisor ASAI Technical Reference.

The switch does not block phantom calls from executing any vector commands because phantom calls follow the same vector processing as regular voice calls. However, it might not make sense to have phantom calls enter certain vector steps such as:

Steps	Description
Announcements	Because there is nobody listening to an announcement that is made to a phantom call, there is no sense in playing one.
collect steps	In a phantom call, the collect step fails because it can not connect a tone receiver to a station Administration Without Hardware (AWOH); it times out because there is nobody to put in the expected digits. The busy step provides a busy signal to the caller. In a phantom call, the busy step disconnects the call because the switch clears a phantom call when the call cannot terminate at a specific local destination.

Phantom call administration

There are no administration screens that are specific to phantom calls, but the following criteria must be met in order for the feature to work:

- Some stations AWOH must be administered.
- If a hunt group is specified as originator, a non-ACD hunt group with AWOH members must also be administered.
- It is recommended that meaningful names are assigned for the stations AWOH that are used by phantom calls if the calling party name will appear on the agent's or Service Observer's display.

Single-step conference

The Single-Step Conference (SSC) feature is available for Avaya switch software R6.3 or later. SSC allows an application to:

- Add a device into an existing call, for example, to play announcements or make voice recordings
- Facilitate application-initiated transfers and conferences

Stations that are AWOH are eligible for single-step conference. The party may be added to a call in listen only mode (no visibility) or with listen and talk capability (visibility).

Single-step conference is only available through an ASAI link. For more information about single-step conference, see Communication Manager CallVisor ASAI Technical Reference.

Related topics:

How does SSC work with Call Vectoring? on page 444

How does SSC work with Call Vectoring?

The call to which an extension is to be single-step conferenced is not allowed to be in vector processing unless the visibility option with the single-step conference request indicates no visibility.

To be transferred to a VDN, a conference call must not have more than two parties.

Note:

Invisible (listen-only) single-step-conference parties are not counted in the two-party limit for a conference call transfer to a VDN.

Multiple outstanding route requests

This feature allows multiple ASAI route requests for the same call to be simultaneously active. The route requests can be over the same or over different ASAI links.

Route requests are all made from the same vector. They must be specified without intermediate (wait-time, announcement, goto, or stop) steps. If the adjunct routing link commands are not specified back-to-back, standard adjunct routing functionality applies and previous outstanding route requests are cancelled when an adjunct routing link vector step is executed.

The first route select response that is received by the switch is used as the route for the call and all other outstanding route requests for the call are canceled.

With multiple outstanding route requests, multiple adjuncts can process the route call request without waiting for the first route attempt to fail. An application can make use of this feature to distribute the incoming call load evenly across adjuncts based on the adjunct's current CPU load.

Note:

Each link has a unique extension number, even in a configuration where there might be multiple links to the same adjunct.

Related topics:

Multiple call route request example on page 445

Multiple call route request example

The following example shows a typical vector where multiple adjunct route requests to multiple links are active at the same time. The first adjunct to route the call is the active adjunct and it specifies which VDN the call should be routed to at that point.

Sample adjunct routing link vector with redundancy

```
1. wait-time 0 seconds hearing ringback
2. adjunct routing link 11
3. adjunct routing link 12
4. adjunct routing link 13
5. wait-time 6 seconds hearing ringback
6. route-to number 1847 with cov n if unconditionally
```

Advanced Vector Routing - EWT and ASA

About Advanced Vector Routing features

Several Advanced Vector Routing features can be used to enhance conditional routing capabilities of Basic Call Vectoring in order to achieve additional efficiencies in call center operations. These features include:

Rolling Average Speed of Answer (ASA)

Rolling ASA Routing allows routing decisions to be based on the current average time for a call to be answered in a skill or VDN, so that vectors route calls to the VDN or skill where it is likely to be answered most quickly.

Expected Wait Time (EWT)

EWT routing allows you to make routing decisions based on the wait time in queue for a call or split. The EWT can also be passed to a Integrated Voice Response (IVR) or Voice Response Unit (VRU) so that a caller can be notified of his or her expected time in queue.

VDN Calls

Vector Directory Number (VDN) Calls routing helps you to make routing decisions that are based on the quantity (number) of incoming trunk calls that are currently active in a VDN. With the VDN Calls conditional, a vector can be used to limit the number of simultaneous calls that are made to a particular VDN. For example, if a service agency is contracted to handle 100 simultaneous calls for a client, calls in excess of that number can be routed to a **busy** step.

Advanced Vector Routing command set

The commands used in Advanced Vector Routing are listed in the following table.

Command category	Action taken	Command
Routing		
	Queue the call to a backup ACD split.	**check split**
Branching/programming		
	Go to a vector step. Go to another vector.	**goto step** **goto vector**

When to use wait time predictions

A number of factors can affect the accuracy of wait time predictions. Wait time predictions are best suited for medium-volume or high-volume call scenarios. The potential accuracy of a wait time predictor increases as the rate of removal from queue increases.

Under all conditions, EWT is the most accurate wait time predictor, but EWT is most accurate when the rate of removal from queue at a given split priority level is at least one call every 30 seconds. For more information, see Expected Wait Time (EWT) on page 448.

Predictions can be made for a split with multiple priority levels as long as the majority of calls are delivered to lower priority levels. If the majority of calls are queued at the higher-priority levels, any predictions made for the lower-priority levels may not be accurate.

The following circumstances can limit the accuracy of the wait time predictions.

Factors limiting wait time prediction accuracy	Description
System restart or new split administration	The EWT algorithm uses a combination of historical and real-time information to make predictions. When no historical information exists, such as when a new split is added or a reset system 3 or 4 is completed, there is the potential for inaccuracies. To prevent inaccurate predictions when there is no historical information, administer the **Expected Call Handling Time** field on the Hunt Group screen. The value in this field is then used in place of the missing historical data. If the value of this field does not accurately reflect the call handling times of the split, EWT predictions may be inaccurate until some call history is generated. The algorithm normally requires about 30 queued calls to be answered from a split priority level before it reaches its maximum accuracy. You can change the value in the **Expected Call Handling Time** field by executing a **`change hunt group`** command. Changing the value does not disrupt EWT predictions by overwriting EWT history. The value is stored and used the next time a reset system 3 or 4 is executed.
Low call volume applications	Split priority levels where the rate of removal from the queue is very low can only be predicted with limited accuracy.
Sites with frequent staffing changes	Although EWT immediately adjusts for all types of staffing changes, since predictions may have already been made for calls that are waiting in queue, those past predictions were based on staffing information which is now out of date. Therefore, the EWT in scenarios where large staffing changes are continually happening can only be predicted with limited accuracy.

Factors limiting wait time prediction accuracy	Description
Staffed agents who rarely answer calls to a split	The EWT algorithm takes account of agents in multiple splits in its calculation. However, suppose there are many agents who are assigned to a split but spend most of their time answering calls in their other splits. If a large number of these agents are moved to or from the split, the EWT for this split may be temporarily inaccurate until it adjusts to those changes.
Applications with widely varying call handling times	If the majority of calls to a split are handled within a narrow range of times, the accuracy of any predictor will be much greater than that for a split where call handling times are widely different.

Expected Wait Time (EWT)

EWT routing allows you to make routing decisions based on the time that a caller can expect to wait in queue.

Related topics:

How EWT is calculated on page 448
EWT for a split on page 449
EWT for a call on page 450
Passing EWT to a VRU on page 450
Notifying callers of wait time without a VRU on page 451
Using EWT to route to the best split on page 452

How EWT is calculated

Depending on how the EWT condition is used in a vector step, the predicted wait time calculation is derived by the following rules:

- If the call is currently queued to a split, the EWT is based on the actual current position of the call in the queue at a particular priority level and the rate of service of calls from the queue at that priority level.
- If the call is not yet queued to a split, the EWT is based on the assumption that the call is placed at the end of the queue and then considers the factors listed above.

EWT also adjusts for many other factors such as multiple split queuing, call handling times, and the impact of direct agent calls on the wait time of other calls to the split. The algorithm adjusts EWT immediately for changes in staffing, such as agents logging in or taking breaks in AUX work mode.

The EWT can also be passed to a VRU so that a caller can be notified of his or her expected time in queue. The expected-wait condition can be used with either the **goto** or **check** commands.

Call vectoring offers several conditionals that can be used to estimate predicted wait time on a queue, including EWT, rolling ASA and Oldest Call Waiting (OCW), but EWT uses the most accurate method of prediction. EWT considers more real-time and historical information, such as priority level, position in queue, and number of working agents.

EWT is responsive to changing call center conditions. For example, EWT adjusts instantly to any staffing changes in the split, or if agents moves in or out of auxiliary work mode, the wait time predictions immediately adjust.

EWT does not include the time in a call vector before the call enters a queue. It also does not include the time that the call rings at a telephone after it is removed from the queue.

For more information about the use and accuracy of wait time predictors, see When to use wait time predictions on page 447.

EWT for a split

The EWT for a split is the time that an incoming call is expected to remain in queue if it is queued to the split at the specified priority level. It is generally used to determine if a call should be queued to the split.

The following vector example shows how to use EWT to determine if a call should be queued to a split.

```
1. goto step 3 if expected-wait for split 1 pri l < 600
2. busy
3. queue-to split 1 pri l
4. announcement 3001
5. wait-time 998 secs hearing music
```

In the example shown above, the following wait time conditions are possible:

- If there are agents available, EWT is zero.
- EWT is infinite if:
  - There are no logged-in agents.
  - All logged-in agents are in AUX work mode.
  - The split queue is full.
  - There is no split queue and all agents are busy.
  - The split queue is locked. This occurs when the last working agent in a non-vector-controlled split attempts to go into AUX work mode.

EWT for a call

EWT for a call is the remaining time that a caller can expect to wait before his or her call is serviced from the queue. If the call is queued to multiple splits, the remaining queue time for each of the splits is calculated, and the shortest calculation is used as the EWT.

For a call to have an expected wait time it must be queued to at least one split. If it is not queued, or if it is queued to splits that are not staffed, the EWT value is infinite.

The following vector example vector shows how EWT is used to determine call treatment.

```
1. queue-to split 1 pri m
2. check split 2 pri m if expected-wait < 30
3. goto step 5 if expected-wait for call < 9999
4. busy
5. announcement 3001
6. wait-time 998 secs hearing music
```

Passing EWT to a VRU

The EWT for a call can be passed to a VRU to inform callers about their expected time in queue. EWT is passed to the VRU with the **`converse-on`** command as wait data. The value that is outpulsed to the VRU is the expected wait time of the call in seconds. The VRU can then convert the seconds to a spoken message. The expected wait is calculated after the VRU port answers the call, so queuing to a converse split does not adversely impact the EWT value that is passed to the VRU.

No zero padding is added to the wait time that is passed to the VRU. If the EWT for the call is 128 seconds, the digits 1, 2, and 8 are outpulsed. If the EWT is 5 seconds, the digit 5 is outpulsed.

The wait time that is passed to the VRU is the most accurate prediction possible. On average, 50% of the time the actual wait time will be shorter and 50% of the time it will be longer. Avaya recommends that VRU applications be configured to make an upward adjustment of the prediction so that the majority of callers receive a predicted wait time that is either equal to, or greater than, the actual wait time.

The VRU can also announce EWT at set intervals while the call is in queue, but this strategy should be used with caution. Circumstances such as a reduction in the number of agents or a sudden influx of higher priority calls could cause the caller’s EWT to increase from one announcement to the next.

If the call is not queued, or if it is queued only to splits that are unstaffed or splits where all agents are in AUX work mode, the end-of-string character # is the only data item that is outpulsed to the VRU.

The following vector example illustrates routing based on the predicted split wait time and passing wait data to the VRU. Wait time is given to the caller only if the caller is expected to wait a total of more than 60 seconds in queue. Callers who would wait more than 10 minutes are told to call back later.

```
1. goto step 3 if expected-wait for split 32 pri l < 600
2. disconnect after announcement 13976
3. queue-to split 32 pri l
4. wait-time 20 secs hearing ringback
5. goto step 7 if expected-wait for call < 40
6. converse-on split 80 pri l passing wait and none
7. announcement 11000
8. wait-time 60 secs hearing music
9. goto step 7 if unconditionally
```

Calls that have predicted wait times greater than 10 minutes fail step 1 and are disconnected after an announcement. If the expected wait time is less than 10 minutes step 1 routes the call to step 3 where it is queued to split 32 and waits 20 seconds hearing ringback. After 20 seconds if the expected wait time for the call is less than 40 seconds, step 5 routes the call to an announcement followed by a wait with music. If the expected wait time for the call is equal to or greater than 40 seconds, step 6 informs the caller of the amount of time that he or she can expect to wait before the call is answered.

Notifying callers of wait time without a VRU

You can use EWT to notify callers of their expected wait time without a VRU. This can be done using recorded announcements and by associating each recorded announcement with a time band, as shown in the following example.

```
VECTOR 101
1. queue-to split 3 pri h
2. goto step 4 if expected-wait for call <= 600
3. busy
4. wait-time 12 seconds hearing ringback
5. announcement 3001 [
Thank you for calling ABC Inc. All agents
 are busy, please wait and we will get to your call as soon as
 possible
]
6. goto vector 202 if unconditionally
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

VECTOR 202
1. goto step 13 if expected-wait for call > 280
2. goto step 11 if expected-wait for call > 165
3. goto step 9 if expected-wait for call > 110
4. goto step 7 if expected-wait for call > 55
5. announcement 3501 [
Thank you for waiting.
  Your call should be answered within the next minute.
]
6. goto step 14 if unconditionally
7. announcement 3502 [
Thank you for waiting. Your call should be
  answered within approximately one to two minutes.
]
8. goto step 14 if unconditionally
9.  announcement 3503 [
```

```
Thank you for waiting. Your call should be
  answered within approximately two to three minutes.
]
10. goto step 14 if unconditionally
11. announcement 3504 [
Thank you for waiting. Your call should be
  answered within approximately three to five minutes.
]
12. goto step 14 if unconditionally
13. announcement 3505 [
We apologize for the delay. Due to heavy
  call volume, you may have to wait longer than five minutes
  to speak to a representative. If possible, we suggest that you
  call between the hours of 8am and 10am for the fastest service.
]
14. wait-time 120 secs hearing music
15. goto step 1 if unconditionally
```

In step 1 of the example shown above, the call is queued to split 3 at high priority. If the call fails to get a queue slot in split 3, if split 3 has no working agents, or if the wait time in split 3 at high priority exceeds 10 minutes, step 2 fails and the caller receives a busy signal. If step 2 succeeds, the caller hears ringback and an announcement and is then sent to vector 202.

Steps 1 through 4 of vector 202 determine tests to determine a predicted time band interval for the remaining queuing time for the call. One of five recorded announcements is then played to provide the caller with an expected wait time.

You may want to program your vectors so that few callers experience wait times that exceed the wait time of the announcement. In the example shown above, the EWT thresholds are set lower than the times that are quoted in the recorded announcements.

Using EWT to route to the best split

You may want to use EWT to change the normal queuing strategy for multiple splits to ensure that calls are answered in the shortest possible time. However, this strategy uses additional system resources and can make it more difficult to read and analyze split reports. Alternately, you can use EWT to identify the best for each call and avoid multiple split queuing.

The following vector example shows a scenario that includes a main split (1) and a backup split (2). In this example, the preference is for an agent from the main split service the call, but a 30-second maximum wait time is a competing preference.

The strategy in this vector is to use the backup split only if the backup split can answer the call within 30 seconds and the main split cannot.

```
1. goto step 5 if expected-wait for split 1 pri m <= 30
2. goto step 5 if expected-wait for split 2 pri m > 30
3. check split 2 pri m if unconditionally
4. goto step 6 if unconditionally
5. queue-to split 1 pri m
6. wait-time 12 secs hearing ringback
7. announcement 3501
8. converse-on split 18 pri m passing wait and none
```

```
9. wait-time 120 secs hearing music
10. goto step 8 if unconditionally
```

In the example shown above, step 1 branches to step 5 (queue to the main split) if the main split can answer the call within 30 seconds. If the main split cannot answer the call within 30 seconds, step 2 checks to see if the backup split can answer the call within 30 seconds. If the test fails, the call branches to step 5 and is queued to the main split. If possible, the call is queued to the backup split in step 3. At this point, the call is queued either to the main split or to the backup split, but not to both.

Steps 6 through 10 provide audible feedback to the caller while the call is in the queue. Note that in step 8, which is executed every 2 minutes, a VRU is used to provide the caller with his or her remaining wait time.

Factors that affect EWT values

Factors that increase EWT for a split priority level

The most common causes for an increase in EWT for a split priority level are:

- The number of calls that are in queue increases
- Agents log out
- Agents go on break or are otherwise in the AUX work mode
- Agents are moved to another split
- Agents with multiple splits answer an increasing number of calls in other splits

Other conditions that may also cause EWT for a split priority level to increase include:

- The average talk time increases
- The number of calls at a higher priority increases
- The number of direct agent calls increases
- The number of RONA calls increases
- The number of abandoned calls decreases
- The number of calls that are queued in this split but answered in another decreases.

Factors that decrease EWT for a split priority level

The most common causes for a decrease in EWT for a split priority level are:

- The number of calls in queue decreases
- Agents log in (and start answering calls)

- Agents return from break or otherwise are no longer in the AUX work mode
- Agents are moved from another split
- Agents with multiple splits answer fewer calls in other splits

The following conditions may also cause a decrease in EWT for a split priority level:

- The average talk time decreases
- The number of calls at higher priority decreases
- The number of direct agent calls decreases
- The number of RONA calls decreases
- The number of abandoned calls increases
- The number of calls queued in this split but answered in another increases

Troubleshooting EWT

To verify that your EWT is operating as intended, use the **`list trace ewt`** command to observe processing events of all calls.For more information, see Appendix D: Troubleshooting vectors.

Note:

The **`list trace ewt`** command is blocked when the Tenant Partitioning feature is enabled.

Rolling Average Speed of Answer (ASA)

Rolling ASA Routing helps you to make routing decisions that are based on the current average time that it takes for a call to be answered in a split or VDN. In this way, a vector can route a call to the VDN or split where it is likely to be answered most quickly.

Related topics:

Rolling ASA versus interval ASA on page 454
When to use rolling ASA on page 455
Rolling ASA split calculation on page 455
Rolling ASA VDN Calculation on page 456
Combining VDN and ASA routing example on page 456

Rolling ASA versus interval ASA

The ASA calculation used for vector routing is called *rolling* ASA to differentiate it from the *interval* ASA that is recorded in Basic Call Management System (BCMS) and Avaya Call Management System (CMS) reports.

Rolling ASA is a running calculation that does not take into account the 15-minute, half-hour, or hour reporting intervals. It does not reflect interval boundaries.

The interval ASA used for BCMS and CMS reports is calculated on reporting interval boundaries and clears to zero at the start of each reporting interval.

The rolling ASA for a split or VDN is calculated based on the speed of answer for all calls recorded since system start-up, and is recalculated every time a call is answered. During each calculation, the speed of answer for the current call is given a weighted value that is greater than previous calls. Approximately 95% of the value of rolling ASA is obtained from the previous ten calls.

Note:

Calls that are not answered, such as calls that receive a forced busy, are not considered in the rolling ASA calculation.

The rolling ASA is calculated for an entire split or VDN. The calculation does not consider the priority levels of answered calls.

When to use rolling ASA

Rolling ASA is best used to test whether vector processing should queue the call to additional splits/skills when the main split/skill does not currently meet the targeted threshold.

Rolling ASA conditionals should *not* be used to prevent calls from queuing to the main split/skill or being answered in the principal VDN. If no calls are being answered in the main split/skill or VDN, the value of rolling ASA does not change. This could result in all future calls being locked out of the main split/skill or VDN unless there are other call vectors in the system that are directing calls to them.

Important:

To implement a call flow that tests whether or not to queue a call to a main split/skill, use the EWT feature.

Rolling ASA split calculation

The rolling ASA for a split is the average call answer time, as specified by the time interval that starts when call processing attempts termination to a split, and ends when the call is answered in that split. The measured interval includes both time in queue and ringing time at the agent station.

If the call is answered in another split or the call is abandoned by the caller, rolling ASA is not recorded for the call. If a call flows into a split from another split, the time queued and ring time for the previous split are not included. If a call is queued in multiple splits, only the rolling ASA for the split in which the call is answered is measured.

Rolling ASA VDN Calculation

The rolling ASA for a VDN is the average call answer time, as specified by the time interval that starts when call processing is initiated within the VDN until it is answered. The measured interval includes:

- Time elapsed in vector processing, including time in announcements.
- If the call is answered by an agent, time in queue and time ringing at the agent station.

Note:
If a call flows between VDNs, only the time elapsed in the answering VDN is used in the calculation.

Related topics:

Rules for specifying VDNs on page 456

Rules for specifying VDNs

Rolling ASAfollows the rules used for other Advanced Vector Routing conditionals to specify a VDN in a goto step:

- A VDN number.
- The value designated as *latest*. The latest VDN is the VDN currently processing the call. The latest VDN is not affected by VDN override settings.
- The value *active*. The active VDN is the VDN of record, which is the called VDN as modified by override rules. For example, if a call routes from a VDN with override set to *yes* then the new VDN is the active VDN. If a call routes from a VDN with override set to *no*, the previous VDN is the active VDN.

Combining VDN and ASA routing example

The following vector example combines VDN and split ASA routing.

```
1. queue-to split 10 pri h
2. goto step 6 if rolling-asa for split 10 <= 30
3. check split 11 pri h if rolling-asa <= 30
4. check split 12 pri h if rolling-asa <= 30
5. check split 13 pri h if rolling-asa <= 30
6. announcement 10000
7. wait-time 40 secs hearing music
8. goto step 3 if unconditionally
```

Step 1 queues the call to the main split. If the main split is currently answering calls within the target time of 30 seconds, step 2 bypasses all of the backup splits and goes directly to the announcement in step 6. The assumption is that the call will be handled by split 10 within the time constraints. However, if the call is not answered by the time that vector processing reaches step 8, the backup splits are checked.

If the rolling ASA for the main split is greater than 30 seconds, steps 3, 4, and 5 check the backup splits. The call is queued to any of these splits that have a rolling ASA of 30 seconds or less. If the call still is not answered by the time that vector processing reaches step 8, the backup splits are checked again.

VDN Calls

VDN Calls routing allows you use the counted-calls conditional to make routing decisions on the number of incoming trunk calls that are currently active in a VDN.

Related topics:

How VDN Call counts are calculated on page 457
Using the counted-calls conditional example on page 458

How VDN Call counts are calculated

The counted-calls conditional allows a vector to limit the number of simultaneous calls directed to a particular VDN. For example, if a service agency is contracted to handle 100 simultaneous calls for a client, calls in excess of that number can be routed to a busy step.

VDN Call counts follows the rules used for other Advanced Vector Routing conditionals to specify the VDN in a goto step:

- A VDN number.
- The value designated as latest. The latest VDN is the VDN currently processing the call. The latest VDN is not affected by VDN override settings.
- The value active. The active VDN is the VDN of record, which is the called VDN as modified by override rules. For example, if a call routes from a VDN with override set to **yes** then the new VDN is the active VDN. If a call routes from a VDN with override set to **no**, the previous VDN is the active VDN.

When Advanced Vector Routing is enabled, a count of active incoming trunk calls is kept for each VDN. The VDN counter increments each time that an incoming call is placed to the VDN and decremented each time that an incoming call is released. A call is considered active in a VDN from the time the call routes to the VDN until all parties on the call are dropped and the call is released.

Note:

The call is counted for the originally called VDN only. When a call is routed to another VDN, the call counter for the subsequent VDN does not increment, nor does the call counter for the original VDN decrement.

The VDN Call count includes the following types of calls:

- Incoming trunk calls routed directly to the VDN.
- Incoming trunk night service calls in which the VDN is the night service destination.

- Calls that cover or forward to the VDN if it is the first VDN routed to and the call is an incoming trunk call.
- Already counted calls that are conferenced with counted or not counted calls from the same VDN.

The VDN call count does not include:

- Internal calls to the VDN.
- Calls that are transferred to the VDN.
- Calls that are redirected to their VDN return destination.
- Conferenced calls that were previously counted on different VDNs.

Using the counted-calls conditional example

The following vector example shows how the counted-calls conditional can be used to route calls.

Using VDN call counting to route calls

```
1. goto step 3 if counted-calls to vdn 1234 <= 100
2. busy
3. queue-to split 60 pri l
4. wait-time 20 seconds hearing ringback
5. announcement 27000
6. wait-time 60 seconds hearing music
7. goto step 5 unconditionally
```

If more than 100 calls are active in VDN 1234, the caller hears a busy signal and vector processing is terminated. If 100 or fewer calls are active, the call queues to split 60.

Attendant Vectoring

About Attendant Vectoring

The Attendant Vectoring feature enables a set of commands that can be used to write call vectors for calls to be routed in non-call center environments. When Attendant Vectoring is enabled, all attendant-seeking or dial 0 calls are processed using the call vectors, not the normal attendant console call routing.

The main reason to use Attendant Vectoring is to allow flexible routing of attendant-seeking calls. If users are instructed to dial an attendant VDN, the call could be answered by an attendant, but it may also be covered to the voice mailbox of a night station. Training users to understand these different call routing options is something you should consider before using Attendant Vectoring.

If you use Attendant Vectoring and night service to route calls to a voice mail system, you can also use the Automatic Message Waiting (AMW) feature to notify after-hours personnel that there are messages in the night service station mailbox by assigning an AMW lamp on one or more backup telephones. When personnel see that there are new messages, they can check those messages after hours and act upon them as needed.

Attendant Vectoring command set

The following table lists the commands associated with Attendant Vectoring.

Table 41: Attendant vectoring command set

Command category	Action taken	Command
Treatment		
	Play an announcement.	**announcement**
	Play a busy tone and stop vector processing.	**busy**
	Disconnect the call.	**disconnect**
	Delay with audible feedback of silence, ringback, system music, or alternate audio/ music source.	**wait-time**
Routing		
	Queue the call to an attendant group.	**queue-to attd-group**
	Queue the call to an attendant extension.	**queue-to attendant**
	Queue the call to a hunt group.	**queue-to hunt-group**
	Route the call to a specific extension number.	**route-to number**
Branching/programming		
	Go to a vector step.	**goto step**
	Go to another vector.	**goto vector**
	Stop vector processing.	**stop**

Treatment commands

announcement command

Syntax

`announcement <extension>`

The usage for the **`announcement`** command is the same as in Basic Call Vectoring. For details on using this command, see the Basic Call Vectoring section.

busy command

Syntax

`busy`

The usage for the **`busy`** command is the same as in Basic Call Vectoring. For details on using this command, see the Basic Call Vectoring section.

disconnect command

Syntax

`disconnect after announcement <extension>`

The usage for the **`disconnect`** command is the same as in Basic Call Vectoring. For details on using this command, see the Basic Call Vectoring section.

wait-time command attendant vector usage

Syntax

`wait-time <time> secs hearing <silence, ringback, music>`

This use of the **`wait-time`** command was slightly modified for attendant vector usage. The i-silent treatment choice was removed because it does not pertain to attendant vectoring. The **`wait-time <seconds> secs hearing <extension> then <silence, ringback, music, continue>`** command was left unchanged. No other changes or attendant specific considerations apply, so these commands work as they do in Basic Call Vectoring.

Routing commands for Attendant Vectoring

Attendant Vectoring allows use of several ROUTING commands, including:

- queue-to attd-group command
- queue-to attendant command
- queue-to hunt-group command
- route-to number command

Note:

A “wait-time 0 secs hearing ringback” step should be used to give immediate feedback to the caller. The **`queue-to`** command does not provide ringback until the call is actually ringing the attendant. The wait-time step should be implemented as the first vector step or as the step immediately before the queue-to step.

The following sections detail the syntax that can be used for these commands and any information that is specific to their use in Attendant Vectoring.

Related topics:

queue-to attd-group command on page 461
queue-to attendant command on page 462
queue-to hunt-group command on page 462
route-to number command for Attendant Vectoring on page 463

queue-to attd-group command

Syntax

`queue-to attd-group`

The **`queue-to attd-group`** vectoring command is available only for attendant vectors. If an attendant group call is redirected to vector processing that queues the call to the attendant group, the group to which the call gets queued is determined by the TN assignment that is associated with the call. If an attendant in the group is available to take the call, it is terminated to the attendant, not queued, and vector processing terminates.

Attendant group based on tenant number

When attendant group calls are redirected to vector processing and are programmed to queue to the attendant group, the attendant group is the group that is designated for the call’s associated tenant number.

If an attendant group call is redirected to vector processing that queues the call to the attendant group, the call is placed in the queue using the priority that is assigned for the call. Attendant queue priorities are assigned on a system-wide basis, not on an individual partition basis.

Attendant group queue

Calls that are queued to the attendant group by way of attendant vector processing are queued with the system-administered priority for the call. If an attempt is made to queue the call and it fails, the vector event for queue failure is logged.

As with other vector queue commands, vector processing continues with the next step following the **queue-to attd-group** command regardless of success or failure. The **goto step if queue-fail** command is provided for handling failure conditions. Otherwise, on success, announcements or other feedback can be applied while the call is in queue. Other than the provision of caller feedback, attendant queue functionality is unchanged. If no commands follow a successful queue step, the call is left in the queue with no feedback. If no commands follow a failed queue step, the call is dropped. Anytime the end of vector processing is reached without the call being placed in queue, it is dropped and an event is logged.

queue-to attendant command

Syntax

`queue-to attendant <extension>`

The **queue-to attendant** vectoring command is available only for attendant vectors. If an attendant group call is redirected to vector processing that queues the call to an individual attendant, the attendant to whom the call gets queued must be a member of the attendant group that is indicated by the TN assignment associated with the call. If the attendant is available to take the call, the call is terminated to the attendant, not queued, and vector processing terminates.

The success of this command depends on having individual attendant access. These calls are queued based on the priority that is assigned to individual attendant access calls.

Individual attendant queue

Calls that are queued to the individual attendant using attendant vector processing are queued with the system-administered priority for individual attendant access calls. If the indicated attendant is not a member of the associated attendant group, the command is considered failed and vector processing continues with the next vector step. If an attempt is made to queue the call and it fails, a vector event is logged.

As with other vector queue commands, vector processing continues with the next step following the **queue-to attendant** command regardless of success or failure. The **goto step if queue-fail** command is provided for handling failure conditions. Otherwise, on success, announcements or other feedback can be applied while the call is in the queue. If no commands follow a successful queue step, the call is left in the queue with no feedback. If no commands follow a failed queue step, the call is dropped. Anytime the end of vector processing is reached without the call being placed in queue, the call is dropped and an event is logged.

queue-to hunt-group command

Syntax

`queue-to hunt-group <#> pri <l (low), m (medium), h (high), t (top)>`

This vectoring command is available only for attendant vectors. However, it is the functional equivalent of the split queueing command. As such, a call can be queued to up to three hunt groups. If an attendant group call is redirected to vector processing that queues the call to a hunt group, the call is queued with the indicated priority. If a hunt group member is available to take the call, it is terminated to the member, not queued, and vector processing terminates. In order to use a hunt group in vectoring, it must be administered as a vector controlled group. However, it can be any type of hunt group, including UCD, ACD, and so forth.

Hunt group queue

Calls that are queued to a hunt group by way of attendant vector processing are queued with the indicated priority for the call. If an attempt is made to queue the call and it fails, a vector event is logged.

As with other vector queue commands, vector processing continues with the next step following the **`queue-to hunt-group`** command regardless of success or failure. The **`goto step if queue-fail`** command is provided for handling failure conditions. Otherwise, on success, announcements or other feedback can be applied while the call is in the queue. Since these hunt groups are required to be vector-controlled, announcements are provided by way of vectoring commands and hunt group-specific forced announcements do not apply. If no commands follow a successful queue step, the call is left in the queue with no feedback and vector processing terminates. If no commands follow a failed queue step, the call is dropped. Anytime the end of vector processing is reached without the call being placed in the queue, it is dropped.

route-to number command for Attendant Vectoring

Syntax

`route-to <number> with cov <y, n> if <unconditionally>`

This command is slightly modified from standard usage when used for attendant vectoring and unconditionally is the only available option. Existing choices allow routing with if unconditionally, digit, name, or interflow-qpos. Since digit comparison and interflow do not pertain to attendant vectoring, the options are not available. No other changes or attendant specific considerations apply. This command works as it does in standard usage. This command is provided by administration that is defined on the Console Parameters screen. Therefore, call processing requirements are not needed.

Syntax

`route-to ~r<number>`

For incoming calls to the Communication Manager, NCR can be activated using the route-to number vector step, where the **number** field in the vector step has a ~r in the first digit position. This allows for the route-to number vector step to interflow an incoming attendant call to another Communication Manager over the PSTN since no trunks are tied up at the redirecting switch.

Branching/programming commands for Attendant Vectoring

Attendant Vectoring allows use of several branching/ programming commands, including:

- goto step command
- goto vector command
- stop command

The following sections detail the syntax that can be used for these commands and any information that is specific to their use in Attendant Vectoring.

Related topics:

goto step command for Attendant Vectoring on page 464
goto vector command for Attendant Routing on page 465
stop command for Attendant Routing on page 465

goto step command for Attendant Vectoring

Syntax 1

```
goto step <step #> if time-of-day is <day><hour>:<minute> to
<day><hour>:<minute>
```

This use of the **goto step** command is the same as in Basic Call Vectoring. For details on using this command, see the Basic Call Vectoring section.

Syntax 2

```
goto step <step #> if <unconditionally>
```

This use of the **goto step** command is the same as in Basic Call Vectoring. For details on using this command, see the Basic Call Vectoring section.

Syntax 3

```
goto step <step #> if queue-fail and goto vector <vector #> if queue-
fail
```

These vectoring conditionals are available only for attendant vectors. Any time an attempt is made to queue a call and it cannot be queued, these commands can be used to direct vector processing. For attendant vectoring, there is no attempt to determine whether a call can be queued before attempting to do so. Therefore, one of these commands can be used to provide alternate processing when calls cannot be queued. Some examples of why calls can fail to queue are as follows, but this is not a complete list of the causes of failure:

- The queue is full
- The attendant group is in night service and there is no night console
- The individual attendant is not a member of the associated attendant group
- There were invalid multiple queue attempts. For more information, see Attendant Vectoring and multiple queueing on page 472.

Failure to queue

The queue failure conditional is set following a queue command that fails to queue the call. It always indicates the result of the most recent queue command. If the failure conditional is set, vector processing is redirected as indicated.

goto vector command for Attendant Routing

Syntax 1

```
goto vector <vector #> if time-of-day is <day><hour>:<minute> to
<day><hour>:<minute>
```

The use of the **goto step** command is the same as in Basic Call Vectoring. For details on using this command, see the Basic Call Vectoring section.

Syntax 2

```
goto vector <vector #> if unconditionally
```

The use of the **goto step** command is the same as in Basic Call Vectoring. For details on using this command, see the Basic Call Vectoring section.

stop command for Attendant Routing

The use of the **stop** command is the same as in Basic Call Vectoring. For details on using this command, see the Basic Call Vectoring section.

Attendant Vectoring overview

The Attendant Vectoring capability enables you to use certain vector commands in a non-call center environment. For example applications of Attendant Vectoring see Call Vectoring applications.

Attendant Vectoring is available in non distributed attendant environments and distributed attendant environments for IAS and QSIG CAS.

Related topics:

Vector screen example on page 466
Console Parameters screen example on page 466
TN assignments on page 468
Restrictions for attendant and non-attentdant vectoring on page 468
Attendant queue on page 468
Hunt group queue for Attendant Vectoring on page 468
Redirecting calls to attendant VDNs on page 468
Night service on page 469
Attendant VDNs on page 469

Vector screen example

The following example shows the Call Vector screen with the Attendant Vectoring field enabled.

Call Vector screen

```
change vector xxx                                        page 1 of 3
              CALL VECTOR

    Number: xxx                  Name: ___________________________
Multimedia? n
Attendant Vectoring? y
     Meet-me Conf? y           Lock? y
     Basic? n    EAS? n   G3V4 Enhanced? n   ANI/II-Digits? n   ASAI Routing? n
 Prompting? n    LAI? n  G3V4 Adv Route? n   CINFO? n   BSR? n   Holidays? n

01 ____________
02 ____________
03 ____________
04 ____________
05 ____________
06 ____________
07 ____________
08 ____________
09 ____________
10 ____________
11 ____________
```

The **Attendant Vectoring** field appears only when Attendant Vectoring is enabled on the Customer Options screen. If either Basic Vectoring or Prompting are set to y, the Attendant Vectoring field defaults to n. If Basic Vectoring, Prompting, and Enhanced Conference are not enabled on the Customer Options screen, the Attendant Vectoring field defaults to y, and it cannot be changed to n. When the Attendant Vectoring field on the Call Vector screen is set to y, that vector is used as an attendant vector.

To associate VDNs and vectors for attendant vectoring, a field on the VDN and the call vectoring screens indicates attendant vectoring. When attendant vectoring is indicated for VDNs and vectors, all call center-associated fields (such as Skills and BSR) are not displayed.

Console Parameters screen example

When Attendant Vectoring is enabled, a field on the Console Parameters screen identifies the assigned Attendant Vectoring VDN. The following examples show the Console Parameters screens.

Console Parameters screen (Page 1)

```
change console-parameters                                       Page   1 of   4
                              CONSOLE PARAMETERS
         Attendant Group Name: OPERATOR
                          COS: 1                                  COR: 1
       Calls in Queue Warning: 1                    Attendant Lockout? y
        Ext Alert Port (TAAS): 01A1216
                          CAS: none
                                              Night Service Act. Ext.: 195
                 IAS (Branch)? n              IAS Tie Trunk Group No.:
         IAS Att. access Code:                  Alternate FRL Station:
              Backup Alerting? y        DID-LDN Only to LDN Night Ext? n
```

```
Attendant Vectoring VDN: 2000
```

Console Parameters screen (Page 2)

```
change console-parameters                                      Page   2 of   4
                              CONSOLE PARAMETERS

TIMING
  Time Reminder on Hold (sec): 30           Return Call Timeout (sec): 30
  Time in Queue Warning (sec): 15

  INCOMING CALL REMINDERS
      No Answer Timeout (sec): 10                      Alerting (sec): 10
                               Secondary Alert on Held Reminder Calls? y

ABBREVIATED DIALING
     List1:                  List2:                  List3: system

                           COMMON SHARED EXTENSIONS
            Starting Extension: 670            Count: 3
```

Console Parameters screen (Page 3)

```
change console-parameters                                      Page   3 of   4
                              CONSOLE PARAMETERS

QUEUE PRIORITIES

               Emergency Access: 1
                Assistance Call: 2
                        CO Call: 2
               DID to Attendant: 2
                       Tie Call: 2
            Redirected DID Call: 2
                Redirected Call: 2
                    Return Call: 2
                    Serial Call: 2
    Individual Attendant Access: 2
                Interpositional: 2
       VIP Wakeup Reminder Call: 2
             Miscellaneous Call: 2

       Call-Type Ordering Within Priority Levels? n
```

Console Parameters screen (Page 4)

```
change console-parameters                                      Page   4 of   4
                              CONSOLE PARAMETERS

ASSIGNED MEMBERS ( Installed attendant consoles )
              Type       Grp  TN            Type       Grp  TN
           1: principal   1   1          9:
           2:                           10:
           3:                           11:
           4:                           12:
           5:                           13:
           6:                           14:
           7:                           15:
           8:                           16:
```

TN assignments

Just as TN assignment determines the attendant group to which calls are terminated, the TN assignment also determines the VDN to which calls are redirected. If a VDN is administered, attendant group calls are redirected to the VDN rather than the attendant group. If a VDN is not assigned, calls terminate to the associated attendant group.

The selected TN for calls that are covered to an attendant group is the called user's TN, not the calling user's TN. When Tenant Partitioning is not administered, the system can have only one partition and attendant group. All attendant group calls are directed to attendant group 1. The screen to administer TN associations is not accessible, so system-wide console assignments apply. To follow the existing principals of this administration, the attendant vectoring VDN assignment appears on the Console Parameters screen when partitioning is turned off. When it is turned on, the field is removed from the console screen and the contents are automatically copied to TN 1.

Restrictions for attendant and non-attentdant vectoring

No restrictions apply to attendant and non attendant vectoring. For example, an attendant VDN can point to a non attendant vector and vice versa. The same is true for vector commands.

For example, an attendant VDN that points to an attendant vector can have a vector step that routes to another non attendant VDN. In this case, the call is removed from the queue and treated as though it just entered vector processing rather than as a continuation from one VDN to another. The reverse is also true if a non attendant VDN is routed to an attendant VDN.

Attendant queue

If attendant vectoring results in putting a call in the attendant queue, it is placed in queue with the priority as administered on the console parameter screen. There are no changes made to the attendant priority queue for attendant vectoring. Even when partitioning is turned on and multiple attendant groups exist, all queues have the same priority assignments. Priority queue administration also applies for calls to an individual attendant, by way of the assigned extension.

Hunt group queue for Attendant Vectoring

If attendant vectoring results in putting a call in the hunt group queue, it is placed in the queue with the indicated priority. To use this command, the hunt group must be vector controlled.

Redirecting calls to attendant VDNs

Because it is not possible to apply vector commands or specialized administration to specific types of attendant group calls, the following can not be redirected to the attendant VDN.

Type of calls	Description
Emergency Access	These calls are still sent directly to the attendant group. However, an attendant vectoring VDN can be assigned as the emergency access redirection extension.

Type of calls	Description
Attendant return calls	These calls are still sent to the original attendant if the original attendant is available or will be placed into the attendant group queue if no attendants are available.
Serial calls	As with return calls, serial calls are still returned to the original attendant if the original attendant is available and are placed into the attendant queue if no attendants are available.
VIP Wakeup calls	These reminder calls are still sent directly to the attendant group.
Call Park time-out	These calls result in a conference (caller, principal, and attendant) and call vectoring does not allow conferenced calls to be vectored.
Call Transfer time-out	These calls are controlled by the attendant return call timer and are processed as though they are attendant extended calls, in other words, actual attendant return calls.

Night service

There is no additional night service functionality provided for attendant vectoring. Night service routing can be provided using the existing night station service in conjunction with attendant vectoring. All existing night service rules remain in place (for example, night console service supersedes night station service, which supersedes TAAS). Attendant group calls are not redirected to attendant vectoring when the system is in night service unless a night console is available. Otherwise, they continue to be redirected to the applicable night service processing. To achieve attendant vectoring for calls when the system is in night service without a night console, the night station service extensions must be attendant vectoring VDN extensions.

Attendant VDNs

The fact that VDN extensions can be dialed directly or calls can be transferred to VDN extensions is unchanged for attendant VDNs.

Currently, VDN extensions can be assigned to:

Hunt group night destination

An attendant vectoring VDN can be assigned as a hunt group's night destination. Calls to that hunt group when it is in night service are redirected to the VDN and attendant vectoring applies.Hunt group night service does not apply if the hunt group is vector controlled. When **vector?** on the Hunt Group screen is `y` the**night service destination** field is removed from the screen. In order for a hunt group to be available in vectoring for the **`queue-to hunt-group`** command, the hunt group must be vector controlled. The hunt group in the **`route-to`** command could be in night service and the call would then terminate to the indicated night service destination. If the hunt group is accessed using the **`queue-to hunt-group`** command no night service applies.

LDN and trunk night destination

One or all trunk groups can be placed into night service and an attendant vectoring VDN can be assigned as the group's night service destination. If a night destination is assigned for LDN calls, it overrides (for LDN calls) the trunk group's night destination. Either of these destinations can be an attendant vectoring VDN. However, if Tenant Partitioning is administered and the trunk group night service destination is the attendant group, the call is redirected to the VDN that is associated with the trunk group's TN. If, instead, the night service destination is explicitly assigned to a particular attendant vectoring VDN, it may or may not be the VDN that would have resulted had the night destination been the attendant group.

Trunk group incoming destination

The incoming destination can be an attendant vectoring VDN except for RLT trunk groups. As in trunk group night service, an assigned incoming destination to an attendant vector could result in the call being sent to a different VDN than if the destination had been assigned to the attendant group.

Last coverage point in a coverage path

An attendant VDN can be assigned as a coverage point. If an Attendant VDN is assigned as a coverage point, it should be the last point in the coverage path.

Abbreviated dialing lists

Attendant VDNs can be assigned to abbreviated dialing lists.

Emergency access redirection

An attendant VDN can be assigned to emergency access redirection. When the attendant's emergency queue overflows or when the attendant group is in night service, all emergency calls are redirected to this VDN. Careful thought should be given to routing these calls off-switch.

QSIG CAS number for attendant group calls

An attendant VDN can be assigned to this number which determines where attendant group calls at a QSIG Branch are processed. This allows local vectoring at a Branch prior to routing the calls to the Main or elsewhere.

Auxiliary data for the following button assignments

In keeping with existing procedures, attendant VDNs will not be denied as auxiliary button data for:

- Facility busy indication. Visual indication of busy or idle status for the associated extension.
- Manual message waiting indication. Lights a message waiting lamp on the station that is associated with the button.
- Manual signaling. Rings the station that is associated with the button.
- Remote message waiting indicator. Message waiting status lamp automatically lights when a LWC message is stored in the system for the associated extension.

Attendant Vectoring and attendant VDNs

When Attendant Vectoring is administered and if an attendant VDN is assigned, attendant group calls are intercepted and sent through vector processing. The attendant VDN can be assigned on the Console Parameters screen if Tenant Partitioning is turned off or on the Tenant screen if partitioning is turned on. If an attendant VDN is assigned, the call is redirected to the

VDN for vector processing. If a VDN is not assigned, the call is directed to the attendant group. Attendant group calls can only be redirected to attendant VDNs.

Related topics:

[Intercept attendant group calls](#) on page 471
[Allow override](#) on page 472
[Interflow between vectors](#) on page 472
[Music source](#) on page 472

Intercept attendant group calls

When calls are placed to the attendant group or become attendant group calls for the reasons listed below, a check is made for an assigned attendant VDN. If an attendant VDN is assigned and either the system is not in night service or the system is in night service and a night console is available, the call is redirected to the VDN for subsequent vector processing. Otherwise, the call is treated with typical attendant group procedures.

The following occurrences can cause a call to become an attendant group call:

- Listed Directory Number (LDN)
- Attendant group in coverage path
- Attendant control of trunk group access
- Calls forwarded to attendant group
- Controlled Restriction
- Dialed attendant access code
- DID/Tie/ISDN intercept treatment
- DID time-out due to Unanswered DID Call Timer expiration
- DID busy treatment
- Security Violation Notification (SVN)
- Multi frequency signaling with attendant group as terminating destination
- CDR buffer full with attendant group as Call Record Handling Option
- Trunk incoming destination is attendant group
- Trunk group night service destination is attendant group
- Hunt group night service destination is attendant group
- Automatic Circuit Assurance (ACA) referral
- VDN routes to the attendant access code.

Vector override always applies to attendant VDNs. The **Allow VDN Override?** field will not be available so `yes` is assumed.

Allow override

VDN override always applies to attendant VDNs.

To provide the most flexibility possible, there are no restrictions placed on the vector that is assigned to a VDN. A non attendant vector can be assigned to an attendant VDN and an attendant vector can be assigned to a non attendant VDN. Obviously, doing so is not recommended. Assigning an attendant vector to a non attendant VDN severely restricts processing for basic call vectoring since only limited vectoring commands are available in attendant vectors. Assigning a non attendant vector to an attendant VDN also severely restricts attendant vectoring since the attendant-specific commands are not available in basic call vectoring. In addition, it removes basic call vectoring information from attendant VDNs. Also, there are no restrictions in vector chaining between attendant and non attendant vectors (for example, using the **`goto vector`** or **`route-to number`** commands).

Interflow between vectors

When calls interflow from one type of vector processing to another, they are removed from the queue (if applicable) and treated as new calls to vectoring, not continuations of vectoring.

Tenant Partitioning assignments apply to attendant VDNs the same as they do for non attendant VDNs. Therefore, care must be taken that a VDN assignment on the partitioning screen has a compatible TN number assigned to the VDN. For example, tenant partition 1 can be assigned a VDN which belongs to tenant partition 2 so long as partition 1's permissions allow access to partition 2. However, music source selection is based on the tenant partition where the VDN is assigned rather than the partition to which the VDN belongs.

Music source

When music is to be provided for attendant vectored calls, the source that is assigned to the tenant partition of the attendant seeking call is used rather than the source that is assigned to the partition of the VDN.

Attendant Vectoring and multiple queueing

Calls can exist in only one type of queue, which can be an attendant group, and individual attendant, or a hunt queue, and cannot be moved from one queue to another. For example, if a call is queued to the attendant group and a subsequent command attempts to queue the call to an individual attendant or hunt group, it is considered a failed queue attempt.

Related topics:

Restrict queueing to only one type of queue on page 473
Allow multiple priority queueing within hunt queues on page 473
Allow multiple hunt group queueing on page 473

Restrict queueing to only one type of queue

Once a call is queued to the attendant group, individual attendant, or hunt group, any attempt to queue the call to another type of queue is considered a failed queue attempt.

Multiple attempts to queue to attendant groups or individual attendants are also considered failed queue attempts. For example, if a call is queued to attendant X and a subsequent command attempts to queue the call to attendant Y, the second `queue` command fails.

Allow multiple priority queueing within hunt queues

Since hunt group queueing is based on the indicated priority, multiple queue attempts are valid. There is no limitation on the number of attempts to queue to a particular hunt group so long as the command changes the priority at which a call is to be queued. For example, a call can be queued at low priority and subsequently re-queued at medium and/or high priority. However, a second attempt to queue a call at the same priority for which it was previously queued is considered a failed queue attempt. Hunt group queueing is the functional equivalent to split queueing. As such, calls can be queued to a maximum of three different hunt groups at the same time.

Once a call is queued to a hunt group, any subsequent attempt to queue with a different priority results in the call being re-queued with the new priority. Any subsequent attempt to queue with the same priority at which the call is already queued is considered a failed queue attempt.

Allow multiple hunt group queueing

A call can be queued to a maximum of three different hunt groups. Once this maximum is reached, any subsequent attempt to queue a call to a different hunt group is considered a failed queue attempt.

Attendant Vectoring considerations

The main consideration with Attendant Vectoring is training users to understand that calls placed to an attendant console may not always be answered by a live operator. If users are instructed to dial an attendant VDN, the call could be answered by an attendant, but it may also be covered to the voice mailbox of a night station. Training users to understand these different call routing options is something you should consider before using Attendant Vectoring.

If you use Attendant Vectoring and night service to route calls to a voice mail system, you can also use the Automatic Message Waiting feature to notify after-hours personnel that there are messages in the night service station mailbox by assigning an AMW lamp on one or more backup telephones. When personnel see that there are new messages, they can check those messages after hours and act upon them as needed.

Holiday Vectoring

Holiday Vectoring

Holiday Vectoring enables a set of commands that can be used to write (create) call vectors for calls to be routed a non-standard way on holidays or any days when special processing is required.

This section gives you the information you need to use this vectoring option.

Related topics:

Holiday Vectoring command set on page 474

Holiday Vectoring command set

The following table shows the commands that are available for use in Holiday Vectoring.

Table 42: Holiday Vectoring command set

Command category	Action taken	Command
Branching/programming		
	Go to a vector step	**goto step**
	Go to a vector	**goto vector**

Branching/programming commands

goto step command for Holiday Vectoring

Syntax 1

```
goto step <step #> if holiday in table <table #>
```

This command directs the call to a specific vector step if the conditions of the call match a holiday that is in the specified Holiday Table.

Syntax 2

```
goto step <step #> if holiday not-in table <table #>
```

This command directs the call to a specific vector step if the conditions of the call do not match any of the holidays that are in the specified Holiday Table.

goto vector command for Holiday Vectoring

Syntax 1

```
goto vector <vector #> if holiday in table <table #>
```

This command directs the call to a specific vector if the conditions of the call match a holiday that is in the specified Holiday Table.

Syntax 2

```
goto vector <vector #> if holiday not-in table <table #>
```

This command directs the call to a specific vector if the conditions of the call do not match any of the holidays that are in the specified Holiday Table.

Holiday Vectoring overview

Holiday Vectoring is an enhancement that simplifies vector writing for holidays. It is designed for customers who need to reroute or provide special handling for date-related calls on a regular basis.

This feature provides the user with the capability to administer as many as 999 different Holiday Tables, then use those tables to make vectoring decisions. Each table can contain up to 15 dates or date ranges. All of this can be done in advance to ensure seamless call routing over holidays when staffing is reduced or call centers are closed.

When vector processing encounters a goto xxx if holiday in table # step, it determines if the current date and time qualifies as a holiday according to the given table. That information is then used to decide whether the goto condition is true or false, and therefore, whether to goto the given step or vector or not. The date and time match is done at the time that the call is in vector processing. It is done just like time-of-day routing. This means that it is checking the system date and time on the Processor Port Network (PPN), rather than the local port network time on the Expansion Port Network (EPN).

The Holiday Vectoring feature is not limited to holiday use, but can also be applied to any date-related special processing. For example, vectors can be modified or created to perform special processing during a two-week television promotion or a semiannual sale.

This feature was developed in response to customer needs, especially for some customers who may have as many as 30 bank holidays to administer throughout the year. Holiday Vectoring streamlines vectoring tasks and ensures seamless operation over holiday (or special-event) periods.

Without this feature, call center administrators had to write special vectors for each holiday or other special date-related circumstances, and make sure that these vectors were administered at the appropriate times. In some cases, administrators were required to go to work on holidays just to administer vectors. This feature was developed in response to customer needs, especially for some customers who may have as many as 30 bank holidays to administer throughout the year.

Administering Holiday Vectoring

Enabling Holiday Vectoring

The Holiday Vectoring customer option can be enabled if either Vectoring (Basic) or Attendant Vectoring is enabled.

You can have up to 999 different Holiday Tables if you have the Communication Manager 3.0 Enhanced Vectoring option enabled. Otherwise, you can have up to 10 Holiday Tables.

On the Customer Options Screen, the Vectoring (Holidays) field should be set to **y**. If the feature is not enabled, contact your Avaya customer support or authorized representative to have the feature enabled.

Setting up a Holiday Table

Holiday Table command syntax

This section describes the syntax of each Holiday Vectoring command.

Syntax 1

```
change holiday-table x
```

This command allows you to change the entries in a Holiday Table.

To create a new Holiday Table, you must use the change command and give the number of a blank table. For example, change holiday-table 9, where table 9 has not been used to define holidays.

Syntax 2

```
display holiday-table x
```

This command allows you to display the entries in a Holiday Table.

Syntax 3

```
list holiday-table
```

This command lists all of the Holiday Tables.

Syntax 4

```
list usage holiday-table x
```

This command lists all vector steps that refer to the selected Holiday Table.

Using the Holiday Table commands

After ensuring that Holiday Vectoring is enabled on the Customer Options screen, enter:

`change holiday-table 1`

On the Holiday Table screen, which is shown in the following example, enter the holiday information.

Setting up a Holiday Table

```
change holiday-table 1                                          page 1 of 1
                                HOLIDAY TABLE
    Number: 1              Name: Bank Holidays
          START                   END
    Month Day Hour Min     Month Day Hour Min      Description
    12    24               12    31                Christmas
    01    01  00   00      01    01  10   00       New Year's Day
```

Note:

When using a range of dates, the end date must be greater than the start date. Ranges must be within one calendar year. In the example above, two entries were made, one for each calendar year.

The Holiday Table screen can be used for entering individual holidays or holiday ranges. The following rules apply to entering dates on this screen:

1. If a day is entered, the corresponding month must be entered.
2. If a month is entered, the corresponding day must be entered.
3. If an hour is entered, the corresponding minute must be entered.
4. If a minute is entered, the corresponding hour must be entered.
5. If an hour and minute is entered, the corresponding month and day must be entered.
6. If a month and day is entered, the corresponding hour and minute is not required.
7. If an end month and day is entered, the corresponding start month and day must be entered.
8. If a start month and day is entered, the corresponding end month and day is not required.
9. To enter an individual holiday, enter a start month and day, but do not enter an end month and day.
10. To enter a holiday range, enter both a start month and day and an end month and day.

11. The start month, day, hour, and minute must be less than or equal to the end month, day, hour, minute.
12. The description field is an alpha-numeric field that is used for identification.

Result

After creating a holiday table, use the **`display holiday-table`** command to view the entries. To list all of the holiday tables, use the **`list holiday-table`** command, as shown in the following example.

Listing the Holiday Tables

```
list holiday-table
                                HOLIDAY TABLES

                          Table Number      Name
                              01            Business Holidays
                              02            Annual Promotion Dates
                              03            Summer Special
                              04
                              05
                              06
                              07
                              08
                              09
                              10
```

Changing vector processing for holidays

After administering the holiday tables, add or change vector processing for those holidays.

On the command line, enter `change vector` x (where x is the vector number). The Call Vector screen contains a display-only field that indicates that Holiday Vectoring is enabled. On the Call Vector screen, customers can enter a new goto conditional for the holidays.

When Holiday Vectoring is optioned, a field on the Vector screen identifies if the vector on which you are currently working is a Holiday Vectoring vector, as shown in the following example.

Call Vector screen

```
change vector x                                                 page 1 of 3
              CALL VECTOR

    Number: xxx      Name: ___________________________
Multimedia? n      Attendant Vectoring? n    Meet-me Conf? n          Lock? y
     Basic? y   EAS? n   G3V4 Enhanced? n   ANI/II-Digits? n   ASAI Routing? n
 Prompting? y   LAI? n  G3V4 Adv Route? n   CINFO? n    BSR? n   Holidays? y

01 ____________
02 ____________
03 ____________
04 ____________
05 ____________
06 ____________
```

```
07 ____________
08 ____________
09 ____________
10 ____________
11 ____________
```

The Holiday Vectoring field is a display-only field and appears only when Holiday Vectoring is enabled on the Customer Options screen. If either Basic Vectoring or Attendant Vectoring are set to `y`, then the Holiday Vectoring field can be set to `y`.

The following examples use goto commands to route calls for holidays.

Holiday Vectoring example 1

```
 change vector 1                                             Page 1 of 3
      CALL VECTOR

    Number: 1        Name: In Germany
Multimedia? n      Attendant Vectoring? n    Meet-me Conf? n           Lock? y
     Basic? y   EAS? n   G3V4 Enhanced? n   ANI/II-Digits? n   ASAI Routing? n
 Prompting? y   LAI? n  G3V4 Adv Route? n   CINFO? n   BSR? n   Holidays? y

01 goto         vector 2  if holiday         in    table 1
02 route-to     number 123456789       with cov n if unconditionally
03
04
05
06
07
08
09
10
11
```

Holiday Vectoring example 2

```
change vector 3                                              Page 1 of 3
      CALL VECTOR

    Number: 3        Name: In Ireland
Multimedia? n      Attendant Vectoring? n    Meet-me Conf? n           Lock? y
     Basic? y   EAS? n   G3V4 Enhanced? n   ANI/II-Digits? n   ASAI Routing? n
 Prompting? y   LAI? n  G3V4 Adv Route? n   CINFO? n   BSR? n   Holidays? y

01 goto         step   2  if holiday         in    table 2
02 route-to     number 45678          with cov n if unconditionally
03 stop
04 announcement 2721
05
06
07
08
09
10
11
```

After you have assigned Holiday Tables to several vectors, you can use the **`list usage holiday-table`** command, as shown in the following example, to display which vectors and vector steps are using the selected Holiday Table.

List of Holiday Table use in vectors

```
list usage holiday-table
                              LIST USAGE REPORT

     Used By
     Vector                 Vector Number 1                      Step 1
     Vector                 Vector Number 3                      Step 1
```

Holiday Vectoring considerations

Consider the following when administering Holiday Vectoring:

- Administration of Holiday Tables is supported only on the Communication Manager and cannot be changed using adjunct vectoring tools.
- Holiday Vectoring is only available when Vectoring (Basic) or Attendant Vectoring is enabled.
- There is no validation that verifies the consistency among the 15 holidays in any table. If the same holiday is entered twice, the system stops checking with the first entry that is found.
- With holidays that are ranges of dates, the ranges could overlap. When a call is in vector processing, the holidays are checked from top to bottom on the table and the check stops if a match is found. Even though there might be multiple entries that would match, the check stops at the first match.
- There is a validation that the day of the month that is entered is valid with the given month. Specifically, if the month is April, June, September, or November, then the date must be a number between 1 and 30. If the month is January, March, May, July, August, October, or December, then the date can be a number between 1 and 31. If the month is February, then a the date can be a number between 1 and 29.

  Note:

  The year is not checked in holiday vector processing. This allows the same holidays to be used year-to-year when the holiday is on a fixed date. For holidays where the date changes from year-to-year, the holiday tables must be readministered.
- When disabling the Holiday Vectoring feature (changing the value of the Vectoring (Holidays) field from `y` to `n` on the Customer Options screen, the vectors are checked for any goto if holiday steps. If any of these steps are found, an error message is displayed, and the change is not allowed. The customer must remove those vector steps first before the feature can be disabled.

Meet-me Conference

About Meet-me Conference

The Meet-me Conference feature allows you to set up a dial-in conference of up to six parties. The Meet-me Conference feature uses Call Vectoring to process the setup of the conference call.

Meet-me Conference can be optionally assigned to require an access code. If an access code is assigned, and if the vector is programmed to expect an access code, each user dialing in to the conference call must enter the correct access code to be added to the call.

The Meet-me Conference extension can be dialed by any internal or remote access users, and by external parties if the extension number is part of the customer's DID block.

Meet-me Conference command set

The following table lists the commands associated with Meet-me Conference.

Table 43: Meet-me Conference command set

Command category	Action taken	Command
Information collection		
	Collect information from the calling party.	**collect digits**
Treatment		
	Play an announcement.	**announcement**
	Play a busy tone and stop vector processing.	**busy**
	Disconnect the call.	**disconnect**
	Delay with audible feedback of silence, ringback, system music, or alternate audio or music source.	**wait-time**
Routing		
	Route to the appropriate meet-me conference and stop vector processing.	**route-to**

Command category	Action taken	Command
Branching/Programming		
	Go to a vector step.	**goto step**
	Go to another vector.	**goto vector**
	Stop vector processing.	**stop**

Information collection commands

collect command

Syntax

`collect 6 digits after announcement <extension>`

When the **Meet-me Conf** field is enabled, the collect vector step has been modified to collect the next six digits and use those digits as the access code for a Meet-me Conference call. Though not required, the digits can be collected after a recorded announcement.

Treatment commands

announcement command usage for Meet-me Conference

Syntax

`announcement <extension>`

The usage for the **announcement** command is the same as in Basic Call Vectoring. For details on using this command, see the Basic Call Vectoring section.

busy command usage for Meet-me Conference

Syntax

`busy`

The usage for the **busy** command is the same as in Basic Call Vectoring. For details on using this command, see the Basic Call Vectoring section.

disconnect command usage for Meet-me Conference

Syntax

```
disconnect after announcement <extension>
```

The usage for the **disconnect** command is the same as in Basic Call Vectoring. For details on using this command, see the Basic Call Vectoring section.

wait-time command usage for Meet-me Conference

Syntax

```
wait-time <time> secs hearing <silence, ringback, music>
```

The usage for the **wait-time** command is the same as in Basic Call Vectoring. For details on using this command, see the Basic Call Vectoring section.

Routing commands for Meet-me Conference

The following section details the syntax that can be used for this command and any information that is specific to the Meet-me Conference feature.

Related topics:

route-to meetme command on page 483

route-to meetme command

Syntax

```
route-to meetme
```

The route-to vector step uses the condition meetme only for the Meet-me Conference feature. When successful, this condition adds the caller to the Meet-me Conference call and all parties on the call hear an entry tone to signify that another caller has joined the conference. This condition is valid when the caller has entered the correct access code and there are not already six parties on the call.

If the route to meetme step ever fails, vector processing stops and the caller hears busy tone.

Branching/programming commands for Meet-me Conference

Meet-me Conference uses several branching/ programming commands, including:

- goto step command
- stop command

The following sections detail the syntax that can be used for these commands and any information that is specific to their use in Attendant Vectoring.

Related topics:

[goto step command for Meet-me Conference](#) on page 484
[stop command for Meet-me Conference](#) on page 484

goto step command for Meet-me Conference

Syntax 1

```
goto step <step #> if meet-me-idle
```

Syntax 2

```
goto step <step #> if meet-me-full
```

The goto step vector step has two conditions used for the Meet-me Conference feature:

- meet-me-idle
- meet-me-full

The meet-me-idle condition routes the first caller accessing a Meet-me Conference to the conference call. An announcement step saying they are the first party to access the call can be given to the caller.

The meet-me-full condition is used when the Meet-me Conference already has the maximum of six parties on the call.

Syntax 3

```
goto step <step #> if digits = meet-me-access
```

The goto step vector step supports the option, meet-me access, for the digits condition to verify that the access code is valid. If the access code entered by the caller equals the access code administered for the VDN, vector processing continues.

stop command for Meet-me Conference

The use of the **stop** command is the same as in Basic Call Vectoring. For details on using this command, see the Basic Call Vectoring section.

Administering Meet-me Conference

Activating the Meet-me Conference feature

Meet-me Conference is available for all switch models that support the R11 call processing software.

To enable the Meet-me Conference feature:

1. The G3 Version field of the Customer Options screen must be set to V11 or later.
2. The Enhanced Conferencing field of the Customer Options screen must be enabled.

   This feature has an RTU cost and must be enabled through the License File process.

Creating a Meet-me Conference VDN

To create a Meet-me Conference VDN (using example VDN 36090):

1. Enter:

   `add vdn 36090`

   The system displays the VDN screen:

```
add vdn 36090                                                    Page   1 of   3   SPE A
                                VECTOR DIRECTORY NUMBER

                                 Extension: 36090
                                      Name: Enhanced Conf. Meet-me
VDN                             Vector Number: 90

                       Meet-me Conferencing? y
```

2. Enter a name, a vector number, and enter `y` in the **Meet-me Conferencing** field.
3. Press **NEXTPAGE** to display page 2.

   The system displays page 2 of the VDN screen:

```
add vdn 36090                                                   Page   2 of   3   SPE A
                                VECTOR DIRECTORY NUMBER

                             MEET-ME CONFERENCE PARAMETERS
       Conference Access Code: 937821
        Conference Controller: 80378
              Conference Type: 6-party
              Route-to Number:
```

4. Enter a conference access code.

   If you do not want an access code, leave the field blank. Once an access code is assigned, an asterisk displays in this field for subsequent change, display, or remove operations by all users except the init super user login.

   Security alert:

   You should always assign an access code to a Meet-me Conference VDN.

5. Enter a conference controller extension.

   If an extension number is entered, a user at that extension can change the access code for the Meet-me Conference VDN using a feature access code. If this field is blank, only a station user that is assigned with console permissions can change the access code for the Meet-me Conference VDN using a feature access code. In addition, remote access users can change a Meet-me Conference access code using the feature access code.

6. Enter the conference type.

   This field can have the following values:

   - 6-party - Enter this value to administer a regular 6-party conference. This value is the default.
   - expanded - Enter this value if you want to administer up to a 300-party conference.

7. If you set the **Conference Type** field to expanded, use the **Route-to Number** field to administer the ARS/AAR Feature Access Code, the routing digits, and the conference ID digits for the VDN.
8. Press `ENTER` to submit the VDN.

Creating a Meet-me Conference vector

To create a Meet-me Conference vector (using example vector number 90):

1. Enter:

   `change vector 90`

   The system displays the CALL VECTOR screen.

2. Enter **y** in the **Meet-me Conf** field.

   This designates the vector as a Meet-me Conference vector.

3. Create a vector as shown in the following example:

```
change vector 90                                    Page   1 of   3   SPE A
                                  CALL VECTOR

    Number: 90               Name: Meet-me Vec
Multimedia? n     Attendant Vectoring? n    Meet-me Conf? y          Lock? y
    Basic? y   EAS? n   G3V4 Enhanced? n   ANI/II-Digits? n   ASAI Routing?
n
 Prompting? y   LAI? n  G3V4 Adv Route? n   CINFO? n   BSR? n   Holidays? n

01 collect      6    digits after announcement 12340
02 goto         step   6   if digits          =      meet-me-access
03 collect      6    digits after announcement 12341
04 goto         step   6   if digits          =      meet-me-access
05 disconnect   after announcement 12342
06 goto         step   11  if meet-me-idle
07 goto         step   14  if meet-me-full
08 announcement 12343
09 route-to     meetme
```

```
10 stop
11 announcement 12344
```

```
change vector 90                                   Page   2 of   3   SPE A
                                    CALL VECTOR

12 route-to     meetme
13 stop
14 disconnect   after announcement 12345
15 stop
16
17
18
19
20
21
22
```

4. Press `ENTER` to submit the vector.

Interactions

General Meet-me Conference interactions

Both Attendant Vectoring and Meet-me Conference cannot be enabled at the same time.

If Enhanced Conferencing is enabled, but no other vectoring customer options are enabled, only Meet-me Conference vectors can be assigned.

A non Meet-me Conference vector cannot be assigned to a Meet-me Conference VDN and a Meet-me Conference vector cannot be assigned to a non Meet-me Conference VDN.

There will be no restrictions in vector chaining between Meet-me Conference and non Meet-me Conference vectors (for example, using the **`goto vector`** or **`route-to number`** commands. When calls interflow from one type of vector processing to another, they will be removed from any queue (if applicable) and treated as new calls to vectoring, not a continuation of vectoring.

Call Detail Recording

As parties join a Meet-me Conference, a call record is created if required by system administration. If a record is required, the called party will be the Meet-me Conference VDN number and the duration will be the length of time that the party was included in the call. There will be an individual record for each party that will be output when the party drops from the call. One option that will record all calls to Meet-me Conference VDNs is to activate the Intra-switch CDR feature and populate all the Meet-me Conference VDN numbers in the system.

If the Intra-switch CDR feature is used with the Meet-me Conference VDNs, the condition code should be set to C for all call records as is done with traditional conference calls when Intra-switch CDR is active.

If Intra-switch CDR feature is not active for Meet-me Conference VDNs, the creation and contents of call records will depend on the trunk group translations for external callers to the Meet-me Conference. Internal callers to the Meet-me Conference will not generate any records if the Intra-switch CDR feature is not active for either the Meet-me Conference VDN or the calling extension.

Changing vector types

To change a Meet-me Conference vector to a non Meet-me Conference vector, the administrator must first remove all vector steps. To change a non Meet-me Conference vector to a Meet-me Conference vector, the administrator must first remove all vector steps. If either of these conditions exist, a warning message displays that states VDNs currently assigned to this vector may not operate as expected. The next time the administrator tries to submit a change to the Meet-me Conference VDN, they would be forced to assign the VDN to a Meet-me Conference vector.

Direct Inward Dialing (DID)

If the VDN extension is part of the customer's DID block, external users will be able to access the conference VDN. If the VDN extension is not part of the customer's DID block, only internal callers on the customer's network (including DCS or QSIG) or remote access callers can access the conference VDN.

Disabling Enhanced Conferencing

If Meet-me Conference VDNs are assigned when disabling the Enhanced Conferencing option, the change is not allowed and the message, `Must first remove all Meet-me Conf VDNs and vectors, is displayed`. The administrator must remove those VDNs and vectors before the option can be disabled.

Removing stations

A station that is administered as a controlling station for a Meet-me Conference VDN cannot be removed without first removing the assignment on the VDN. The following message displays:

```
Must first remove as conference controller on VDN form.
```

Meet-me Conference security issues

The Meet-me Conference feature is a potential security problem. If Meet-me Conference VDNs are assigned without access codes, hackers could tie up Meet-me Conference facilities, keeping others from conducting legitimate business, and could potentially access the switch and use the switch to make unauthorized calls. Therefore, we should recommend that all Meet-me Conference VDNs have access codes that are known only to administrators and users on a need to know basis. We should also recommend that access codes be changed on a regular basis to reduce the risk of unauthorized access to the switch.

If a user tries to change the access code of a Meet-me Conference and is unsuccessful, or if a user tries to access a Meet-me Conference and uses an invalid access code, a meet-me event is logged. For more information, see Tracking unexpected events in *Programming Call Vectors in Avaya Aura™ Call Center*.

Meet-me Conference capacity issues

Meet-me Conference calls count towards the maximum number of 3-way and 6-way conference calls.

Users cannot add more parties to a conference call once the system maximum is reached.

For Category A, the number of Meet-me Conference VDNs is a subset of the total number of VDNs allowed in the system.

For Category B, the total number of VDNs and vectors is doubled from the normal limit if both Call Vectoring and Enhanced Conferencing are enabled. However, the maximum number of VDNs and vectors available for call center applications is unchanged.

Meet-me Conference call processing scenario

Joe Davis has a sales review scheduled with four associates located in different cities. He has reserved Meet-me Conference telephone number 865-253-6090. In switch administration, this number has been assigned to vector 90. See the following screen.

```
add vdn 36090                                            Page   1 of  3   SPE A
                                VECTOR DIRECTORY NUMBER

                                 Extension: 36090
                                 Name: Meet-me VDN                   Vector Number: 90

                      Meet-me Conference? y
```

VDN 36090 is administered with an access code of 835944.

When each associate calls the Meet-me Conference telephone number, the following vector processing occurs:

```
change vector 90                                         Page   1 of   3   SPE A
                                  CALL VECTOR

    Number: 90               Name: Meet-me Vec
                   Attendant Vectoring? n    Meet-me Conf? y           Lock? y
     Basic? y   EAS? n   G3V4 Enhanced? n   ANI/II-Digits? n   ASAI Routing? n
 Prompting? y   LAI? n  G3V4 Adv Route? n   CINFO? n   BSR? n   Holidays? n

01 collect      6    digits after announcement 12340
02 goto         step   6   if digits          =      meet-me-access
03 collect      6    digits after announcement 12341
04 goto         step   6   if digits          =      meet-me-access
05 disconnect   after announcement 12342
06 goto         step   11  if meet-me-idle
07 goto         step   14  if meet-me-full
08 announcement 12343
09 route-to     meetme
10 stop
11 announcement 12344
```

```
change vector 90                                         Page   2 of   3   SPE A
                                  CALL VECTOR
12 route-to     meetme
13 stop
14 disconnect   after announcement 12345
15 stop
16
17
18
19
20
21
22
```

Each caller hears announcement 12340, which says something similar to *Welcome to the Meet-me Conferencing service. Enter your conference access code.* Each caller enters the access code 835944.

The **`collect`** vector step 1 collects the access code digits. If the access code is valid, the vector processing continues with vector step 6. If the access code is invalid, the vector processing continues with vector step 3, which plays announcement 12341. Announcement 12341 says something similar to *This access code is invalid. Please enter the access code again.* If the caller enters the wrong access code again, the vector processing continues with vector step 5, which plays announcement 12342. Announcement 12342 says something similar to *This access code is invalid. Please contact the conference call coordinator to make sure you have the correct conference telephone number and access code. Good-bye.*

Vector step 6 is only valid for the first caller into the Meet-me Conference. The meet-me-idle condition routes the first caller to announcement 12344 (vector step 11). The recorded announcement says something similar to, *You are the first party to join the call.* The caller is then routed to the Meet-me Conference call by vector step 12 and vector processing stops.

Vector step 7 is used when the Meet-me Conference already has the maximum of six parties on the call. The meet-me-full condition disconnects the caller after playing announcement

12345 (vector step 14). The recorded announcement says something similar to, *This Meet-me Conference is filled to capacity. Please contact the conference call coordinator for assistance. Good-bye.*

If a caller enters the correct access code, is not the first caller, and the conference call is not full, vector processing continues with vector step 8, which plays announcement 12343. The announcement says something similar to *Your conference call is already in progress.* The caller is then routed to the Meet-me Conference call by vector step 9 and vector processing stops. As each caller enters the conference call, all parties on the call will hear an entry tone.

When the conference call is over and callers drop out of the conference call, any remaining parties on the call will hear an exit tone.

Troubleshooting

Conference call drops

The conference call drops abruptly for no apparent reason.

Possible reason:

The Vector Disconnect Timer on the System-Parameters Features screen is set to a value that does is shorter than the duration of the Meet-Me Conference session.

Solution:

Increase the Vector Disconnect Timer value.

Sound volume is too low

Voice volume levels for some conference participants is too low.

Possible reason	The affected conference participants connect through international trunks in which Central Office (CO) loss plans are set for too much loss.
Solution	In the System-Parameters Country Options screen, go to Tone & Country Loss Plans (page 3) and change the values specified in the **End-to-End total loss (dB) in a n-party conference** field.

Percentage allocation routing

About percentage allocation routing

This feature allows you to distribute calls among a set of call centers or Vector Directory Numbers (VDNs) based on specified percent allocation. Various types of incoming calls that arrive at a particular VDN can now be initially directed to a Policy Routing Table (PRT) for Percent Allocation instead of to a vector. The PRT then distributes the calls to the administered Route-to VDNs based on your specified (administered) percent allocation targets.

This feature is useful for segmented call-handling, outsourcing, and optimizing call handling in a multiple-location enterprise. Using percentage allocation routing, you can allocate target percentages and, for example, do the following:

- Allocate certain call types among multiple answering groups with similar skills
- Allocate maximum calls to a more economical calling group
- Ensure that the organization meets the terms of a service level agreement

Screens and fields used to administer percentage allocation

The following screens and fields are required to administer the percentage allocation routing options:

Screen	Field
Vector Directory Number (VDN)	**Destination** **Number** (select Policy Routing Table and enter the PRT number)
Policy Routing Table (PRT)	**Number** **Name** **Type** **Period** **Route-to VDN** **Target %**

For a detailed description of the VDN and PRT screens, see the Screens reference chapter in *Administering Avaya Aura™ Call Center Features*.

Percentage allocation routing example

The Destination and Number fields in the Vector Directory Number screen allow you to specify the routing destination as either a vector or a PRT table. To implement percentage allocation routing, you need to specify routing destination as a PRT table (Destination: Policy Routing Table and Number: PRT table number).

Using the Policy Routing Table screen, you can specify the various routing destinations and the percentages allocated for each of the destinations. The following example Policy Routing Table screen includes the Target and Actual traffic routed to each of the VDNs.

```
display policy-routing-table 1957
                             POLICY ROUTING TABLE
Number: 1957  Name: % distribution      Type: percentage   Period: max count

                                                 Target Actual Call
Index Route-to VDN  VDN NAME                      %      %      Counts
  1   2220071       Gizmo support                25     27.2      3
  2   2221501       Ultra support                 5      9.0      1
  3   2220601       Customer Service South       35     27.2*     3
  4   2220511       Outsourcer Charlie           10      9.0      1
  5   2220501       Survey after service         10      9.0      1
  6   2220072       Outsourcer International     15     18.1      2
  7
  8
  9
 10
 11
 12
                                          Totals 100              11

Command:
```

Considerations for implementing percentage allocation routing

Consider the following when implementing percentage allocation routing:

- Modification of the table while calls are being routed to the VDN(s) the PRT is assigned to can result in calls being miss-routed, miss-appropriated or other indeterminate actions. To avoid this, first create a new PRT table with the required changes and then replace the old version.
- The following are the various valid entries in the period field:
  - 100_count (default): Call counts (and displayed %) are reset after total calls for the PRT reach 100 which is when the total calls match the target routing pattern percentages. This ensures that the routing points have equal distribution of calls all the time.
  - max_count: Call counts are maintained until calls delivered to at least one of the VDNs exceed 65,400. At that point calls are continued to be distributed over the

VDNs but the call counts are reset when the actual percentages equal the targets for all of the VDNs at the same time.

- Half-hour: Resets the call counts at the top of the hour and at the 30 minute point.
- hour: Resets the call counts at the top of the hour.
- daily: Resets the call counts at midnight, every night.
- weekly: Resets the call counts at midnight on Saturday.

Note:
For a detailed description of Policy Routing Table screen and its fields, see *Administering Avaya Aura™ Call Center Features*.

- Actual % and Call Counts are reset whenever the PRT form is changed.
- Calls are routed to the VDN destination that is farthest from meeting its target allocation.
- Calls are routed based on actual call counts, not actual %. When there are no actual call counts (at startup or after reset), the route point with the highest target is selected.
- For VDN domain monitoring with CTI/ASAI, a call to a VDN that is routed via a PRT appears as if the call was routed to the destination VDN by a route-to number command in a vector assigned to the original VDN
- Calls routed through a PRT are reported to BCMS, CMS and IQ as though they were routed to a VDN assigned to Vector 0 (vector number set to 0). VDN reports can be created using existing CMS custom reporting capabilities to show the percentages and the number of calls that have distributed among the destination VDNs. Avaya IQ will provide additional reporting including what VDN the call was routed to, the PRT table number and type used and the matched percentage.
- A Policy Routing Table can be assigned to multiple VDNs. There are no restrictions as to the VDNs that can be entered as route points.
- There are limits for the number of PRT tables that can defined and there is a system routing point (PRTs x VDN destinations) limit. Both of these limits can be found in the Communication Manager System Capacity tables and displayed on the display capacity form.
- In addition to add, change, display, list, and remove, the following commands also support PRT:
  - list vdn: Displays destination type V(ector) or P(RT)
  - list usage policy-routing-table: Lists the VDNs that use the specified PRT
  - list history: displays history of add and change policy-routing-table commands
  - list trace vdn: displays calls to a PRT
- When Destination is a PRT, the Attendant Vectoring and Meet-Me Conferencing fields do not appear.

- For VDN domain monitoring with CTI/ASAI, a call to a VDN that is routed via a PRT appears as if the call was routed to the destination VDN by a route-to number command in a vector assigned to the original VDN
- To evenly distribute calls across 3 routing points, administer the 3 routing points with a 33% target, then add a 4th routing point that routes back to the PRT with a target of 1%. Set the Period to “100_count” and calls will evenly distribute.

Rules for percentage allocation routing

Consider the following rules when implementing percentage allocation routing:

- The target percentages must be whole numbers (integers). The form will not let you use decimals or fractions.
- The target percentage of all the VDNs must add up to 100% before form submittal.

Service Hours Table Routing

Service Hours Table Routing overview

Use Service Hours Table Routing to simplify the vectors you use for handling calls based on office hours. Vectors use the Service Hours Routing tables to determine how to handle calls that are received during working hours versus calls that are received out of hours. Customers can use this feature as an alternative to tod (time of day) routing and can specify working hours on a daily or hourly basis. This feature allows you to administer as many as 999 different tables, then use those tables to make vectoring decisions.

Before this feature, customers added multiple time-of-day steps to their vectors in order to define the hours of operation for a specific business application (VDN or vector). This feature allows customers to define service hours clearly, simply, and in one place. One simple vector command can check to see if the call meets the administered service hours.

Related topics:

goto processing for Service Hours Table Routing on page 495

goto processing for Service Hours Table Routing

When vector processing encounters a goto if service-hours step, it determines if the current day of week and time is within the service hours listed in the corresponding table. This information is used to decide if the **goto** condition is true or false, and therefore, whether or not to go to the given step or vector. The day of week and time match is based on the system time on the Communication Manager that receives the call. The time used in the calculations is the time the call reaches the goto step.

Related topics:

Time adjustments on the Service Hours Table screen on page 496

Time adjustments on the Service Hours Table screen

The time used in the calculations can be adjusted using the **Use time adjustments from location** field on the Service Hours Table screen. This field indicates the location number on the Locations screen that specifies how the adjustments are performed.

You can make the following time adjustments using the **Use time adjustments from location** field:

- Adjust the daylight savings time from the system time
- Apply the time zone for a specific location
- Apply the daylight savings time for a specific location

If this field is blank, no adjustments are made.

Administering Service Hours Table Routing

Administering the Service Hours Table screen

To administer a Service Hours Table:

1. Enter **`add service-hours-table x`**
   x = 1-999
2. Enter values in the following fields:
   - **Description**
   - **Use time adjustments from location**
   - **Start**
   - **End**

   For a description of these fields, see *Administering Avaya Aura™ Call Center Features*.

Administering the goto conditional

goto step command for Service Hours Table Routing

Syntax 1

`goto step x if service-hours in table y`

This command directs the call to a specific vector step if the conditions of the call match the service hours specified in the Service Hours Table.

Syntax 2

`goto step x if service-hours not-in table y`

This command directs the call to a specific vector step if the conditions of the call do not match any of the service hours that are in the specified Service Hours Table.

goto vector command for Service Hours Table Routing

Syntax 1

`goto vector x @step z if service-hours in table y`

This command directs the call to a specific vector if the conditions of the call match service hours that are in the specified Service Hours Table.

Syntax 2

`goto vector x @step z if service-hours not-in table y`

This command directs the call to a specific vector if the conditions of the call do not match any of the service hours that are in the specified Service Hours Table.

Service Hours Table Routing considerations

Consider the following when administering Service Hours Table Routing:

- Service Hours Table Routing is not available when upgrading from a previous release.
- Vectoring (Basic) must be enabled.
- The Call Center Release field must be set to 4.0 or later.

Service Hours Table Routing scenario

The following is a very basic scenario (not considering time adjustments):

Basic Service Hours Table scenario

```
                              SERVICE HOURS TABLE

                                Number: 99
                       Description: Call-ahead Reservations_____
          Use time adjustments from location: 2__

     MON           TUE           WED           THU           FRI
 Start  End    Start  End    Start  End    Start  End    Start  End
 08:00 12:30   10:30 02:00   __:__ __:__   __:__ __:__   __:__ __:__
 13:00 16:30   15:00 20:30   __:__ __:__   __:__ __:__   __:__ __:__
 __:__ __:__   __:__ __:__   __:__ __:__   __:__ __:__   __:__ __:__
 __:__ __:__   __:__ __:__   __:__ __:__   __:__ __:__   __:__ __:__
 __:__ __:__   __:__ __:__   __:__ __:__   __:__ __:__   __:__ __:__

                          SAT           SUN
                      Start  End    Start  End
                      __:__ __:__   __:__ __:__
                      __:__ __:__   __:__ __:__
                      __:__ __:__   __:__ __:__
                      __:__ __:__   __:__ __:__
                      __:__ __:__   __:__ __:__
```

VECTOR 1:

goto vector 2 @step 1 if service-hours not-in table 99

<Service hours – Call-ahead Reservation processing>

VECTOR 2:

<After hours processing>

The following table shows how calls at different times will be processed:

Day of week / Time of call	Which processing?
Sunday / any time	After hours
Tuesday / 14:59	After hours
Tuesday / 15:00	Service-hours
Monday / 12:30	Service-hours
Monday / 12:31	After hours

The same scenario (considering time adjustments):

```
                                DAYLIGHT SAVINGS RULES
Rule          Change Day                  Month    Date     Time    Increment
 0: No Daylight Savings
 1: Start: first Sunday       on or after April     1     at 02:00   01:00
     Stop: first Sunday       on or after October   25    at 02:00
```

```
 2: Start: first Sunday       on or after April     1    at 02:00   02:00
     Stop: first Sunday       on or after October   25   at 02:00
```

```
LOCATIONS
ARS Prefix 1 Required For 10-Digit NANP Calls? y
 Loc.    Name          Timezone  Rule    NPA                           Proxy Sel.
 No.                   Offset                                          Rte. Pat.
 1:     Main          + 00:00     1
 2:     Branch        + 02:00     2
 3: :
 4: :
 5: :
 6: :
 7: :
 8: :
 9: :
10: :
11: :
12: :
13: :
14: :
```

The following table shows how calls at different times will be processed:

System time	Daylight Savings?	Adjustments	Adjusted time	Which processing?
Tuesday / 07:30	yes	- subtract 1 hour (put system into standard time) - add 2 hours (location time zone) - add 2 hours (location DST)	Tuesday / 10:30	Service-hours
Tuesday / 11:01	yes	- subtract 1 hour (put system into standard time) - add 2 hours (location time zone) - add 2 hours (location DST)	Tuesday / 2:01	After hours
Monday / 06:00	no	- add 2 hours (location time zone)	Monday / 08:00	Service-hours
Monday / 10:31	no	- add 2 hours (location	Monday / 12:31	After hours

VDN in a Coverage Path

About VICP

VDN in a Coverage Path (VICP) enhances Call Coverage and Call Vectoring. If Basic Call Vectoring or Call Prompting is enabled on your communication server, you can assign a Vector Directory Number (VDN) as the last point in a coverage path. Calls that reach this coverage point can be processed by a vector or by Call Prompting.

VICP considerations

Once a call has covered to a VDN, it cannot be further redirected by features such as Call Coverage, Call Forwarding, or Night Service.

A VDN is not allowed to be a member of a coverage answer group. A vector cannot route a covered call to a coverage answer group - a coverage answer group can only be a point in a coverage path.

Removing a VDN from the system with the **`remove vdn <extension>`** command automatically removes the VDN from any coverage paths.

VICP interactions

Interaction	Description
AAR/ARS Partitioning	The class of restriction assigned to the VDN determines the partition group number (PGN). The PGN in turn determines the AAR or ARS routing tables used by **`route-to`** commands.
ASAI	For direct calls to a VDN, the **`adjunct routing`** command operates like the command **`route to digits with coverage=y`**. For calls that cover to a VDN, however, the **`adjunct routing`** command operates the same as a **`route to digits with coverage=n`** command. Since calls redirected once to coverage should not be redirected again, the coverage option is disabled for the **`adjunct routing`** command in this situation.
Attendant	A call covering to a VDN can be connected to an attendant queue or hunt group by a vector. Internal calls that route to an attendant display the class

Interaction	Description
	of restriction of the originating station if the attendant presses the **display COR** button. An attendant cannot establish a conference with a call covering to a VDN if the call is in vector processing. If a call placed to a local destination has covered to a VDN and the attendant attempts to add this call to a conference, the conference will be denied until the call has completed vector processing. An attendant-extended call that covers to a VDN will not return. If the attendant extends a call to a local destination that covers the call to a VDN, the attendant's return call timer is canceled when vector processing begins and the Return Call button will not affect the call. If a call covers to a VDN and is then routed to an attendant, the attendant can transfer the call to another VDN.
AUDIX	Calls that cover to a VDN can be routed to an AUDIX by the **`route-to`** or **`messaging`** vector commands. Calls that cover to a VDN may be subsequently transferred to AUDIX. Calls may also be transferred out of AUDIX to a VDN.
Automatic Call Distribution (ACD)	A VDN can be the last point in an agent's coverage path for direct agent calls.
Call Coverage	A VDN cannot be a member of a coverage answer group. A vector cannot route a covered call to a coverage answer group. Calls that have covered to a VDN cannot be redirected again by Call Coverage. Coverage Callback and Leave Word Calling work normally when a vector delivers a call to a covering user.
Call Forwarding	Calls that have covered to a VDN cannot be redirected by Call Forwarding.
Call Park	A parked call will not cover to a VDN. When a call is parked at an extension with a VDN in its coverage path, the call will continue ringing the extension. If the call is parked to a hunt group extension and the call is in queue, the call will remain in the queue until it is retrieved, or answered by an agent, or abandoned by the caller. A vector event is generated for these calls when the administered coverage criteria are met. Once a call covers to a VDN, Call Park cannot be established until the call is delivered to an extension and vector processing ends.
Call Vectoring	The class of restriction assigned to a VDN determines the partition group number (PGN). The PGN in turn determines the AAR or ARS routing tables used by **`route-to`** commands. When a call covers to a VDN, VDN override has no effect on the display shown on an answering display. This station will show the normal display for a covered call. adjunct routing: For direct calls to a VDN, the **`adjunct routing`** command operates like the **`route to digits with coverage=y`**

Interaction	Description
	command. For calls that cover to a VDN, however, the **adjunct routing** command operates the same as a **route to digits with coverage=n** command. Calls redirected once to coverage should not be redirected again, however, so in this situation the coverage option is disabled for the **adjunct routing** command. converse: Covered calls to a VDN work with the **converse** command. If a call in vector processing is connected to an agent in a converse split, the agent cannot activate Consult, Coverage Callback, or Coverage Leave Word Calling. messaging: The **messaging** command handles covered calls differently depending on whether an extension is specified in the command. If the command **messaging split xxxx extension none** is used, the mailbox of the principal extension is used for the call. The number of the principal extension and the reason for redirection are passed to the messaging adjunct in the CONNECT message. When an extension is specified in the **messaging** command, no information about the principal extension is passed to the adjunct. Instead, the number of the extension specified in the command is passed to the adjunct in the CONNECT message along with the reason for redirection. The mailbox for the specified extension is used. route-to: A call covering to a VDN can be routed to any valid destination by the call vectoring command **route-to**. The coverage option for the **route-to digits** command is disabled for covered calls. In other words, the **route-to digits with coverage=y** functions like the **route-to digits with coverage=n** command when processing covered calls. When the **route-to** command terminates a covered call locally, information identifying the principal and the reason for redirection are retained with the call. This information can be displayed on display phones or passed to an AUDIX or Message Center system.
Class of Restriction (COR)	The COR assigned to the covering VDN governs the vector routing of the call.
Conference	Calls in an established conference will not cover to a VDN. Once a call covers to a VDN, a conference cannot be established until the call is delivered to an extension and vector processing ends.
Consult	The Consult feature normally uses a Temporary Bridged Appearance on the principal's set. Call coverage to a VDN removes the Temporary Bridged Appearance from the principal's set, but the Consult feature still works.
Hunt Groups	A VDN can be the last point in a hunt group's coverage path. If the coverage vector for a split or hunt group routes calls to another using a **route-to** or **messaging** command, calls will queue at the second resource with the queue priority assigned for the first split or hunt group. If a **queue-to**, **check**, or **converse** command is used, calls will queue at the second split or hunt group with the priority specified in the command. If an inflow threshold has been assigned to a hunt group, the group will not allow new calls to queue when the oldest call in queue has exceeded the

Interaction	Description
	threshold. Therefore, covered calls are not connected to a hunt group when the group's inflow threshold has been exceeded. Note that this interaction can also occur when a **`messaging split`** or **`route-to`** command routes a covered call to a split that isn't vector-controlled.
Look-Ahead Interflow	For calls that have covered to a VDN, LAI works like a **`route-to digits/number with cov=n`** vector command. Any Dialed Number Identification Service (DNIS) digits sent with the interflowed call will indicate the VDN to which the call covered, not any VDN the call encountered before it went to coverage.
Night Service	Calls that have covered to a VDN cannot be redirected by Night Service.
Personal CO lines (PCOL)	A VDN may be assigned as the last point in a PCOL coverage path.
Phone Display	Calls covering to a VDN and then directed to an agent in a split or hunt group by a **`queue-to`**, **`check`**, **`converse`**, or **`route-to`** command display the following information to the agent. `a=EXT 3174   to   EXT 3077     b` In this example, station A called station B. Station B was busy, and the call covered to a VDN.
Redirection on No Answer (RONA)	RONA applies to calls that cover to a VDN. If the vector associated with the VDN queues the call to a resource (for example, a split or agent) that uses RONA, the call can be requeued for the same resource. The call cannot be redirected, however, since it has already covered to the VDN.
Terminating Extension Groups	A VDN may be assigned as the last point in the coverage path for a Terminating Extension Group.irected, however, since it has already covered to the VDN.
Transfer	Calls may be transferred to extensions that cover to a VDN. Users who receive a covered call may transfer it to a VDN. If a transfer attempt goes to coverage and covers to a VDN, the user at the answering station can complete the transfer by pushing the Transfer button (or by flashing the switchhook on an analog station). Calls that cover to a VDN may be subsequently transferred to AUDIX. Calls may also be transferred out of AUDIX to a VDN.

VDN Time Zone Offset

Reason to use VDN Time Zone Offset

If you have identical opening and closing times in different locations, you can use a single vector to handle the opening and closing time checks using VDN Time-Zone Offset in a manner similar to skill preferences. This simplifies programming and allows sharing of vectors. For example, you can use 9 to 5 as a time in all vectors without converting to the local time reflected on each switch clock.

Operation

Call Vectoring time-of-day (TOD) goto step vector conditionals are calculated based on the main server system clock local time. The main server system clock uses the local server rules for the date, day, year, time-zone, and Daylight Savings Time (DST). The default setting for DST is for the main location (location 1) with the Multiple Locations feature active.

Using VDN Time-Zone Offset, you can modify the time used for the TOD conditional calculation based on the active VDN for the call. This way you can base the TOD values on the local time relative to the VDN where the calls are directed. In addition, if you apply the offset on a VDN basis, you can apply common call flows using the same vector for calls to different VDNs whose application requires the TOD conditional calculations based on different time zones.

Interactions with other features

VDN Time Zone Offset interacts with other features as follows:

- VDN Time Zone Offset does not apply to time calculations associated with the doy, dow and tod Variables in Vectors variable types. Those variable types use the server local time.
- The VDN Time Zone Offset used is the one assigned to the active VDN for the call. The active VDN follows the VDN Override rules. For more information, see *Programming Call Vectors in Avaya Aura™ Call Center*.
- The LSP and ESS servers should be synchronized with the main server system time. This ensures that if a switchover occurs to the survivability server, the VDN offset is applied consistently when the vector TOD conditional steps are being processed by the survivability server.

Example of VDN Time Zone Offset

In this example, a call center company has locations in London, New York, and Denver. The server is located in London and the gateways are located in London, New York, and Denver. All of the locations share the same opening and closing times. Opening time is 9:00 a.m. and closing time is 5:00 p.m. Calls routed to each of the locations are given a separate VDN, each dedicated to routing calls to that location. The company wants to program one vector to handle each of the locations, including the opening and closing time checks. They can do this by using the VDN Time Zone Offset feature along with skill preferences. This company uses Expert Agent Selection (EAS) and the system switch clock is set to GMT.

Assuming Daylight Savings Time is not active, the tod conditional check done in step 2 for calls to VDN1 is based on the server local time in London England (Greenwich Mean Time). For calls to VDN2, the time used is the server local time GMT-5 hours or Eastern Standard Time. For calls to VDN3 the time used is GMT-7 hours or Mountain Standard Time.

The VDN assignments are described in the following table.

VDN	Location	Extension number	Time Zone Offset	Skill preference
VDN1	London	10001	+00:00	lst = 51
VDN2	New York	10002	-5:00	lst = 60
VDN3	Denver	10003	-7:00	lst = 75

Each of the following VDNs are assigned to Vector 201:

1. wait-time 0 secs hearing ringback
2. goto step 7 if time-of-day is all 17:00 to all 09:00
3. queue-to skill 1st pri 1
4. announcement 30002 [*All our agents are busy. Please wait.*]
5. wait-time 60 secs hearing music
6. goto step 4 unconditionally
7. disconnect after announcement 30003 [*Our hours are between 9 a.m. and 5 p.m. Please call back.*]

VDN Variables

VDN Variables provide more opportunities for VDNs to use a smaller set of vectors.

You can:

- Assign up to five variable fields, V1 through V9, on theVDN screen
- Use the VDN Variables in all vector commands that support vector variables except as afor parameter with the**`collect-digits`** command
- Use up to 16-digits to assign a number to the VDN variable and use up to 15 characters to describe the VDN variable
- Use VDN Variables as indirect references to announcement extensions and other numerical values in vector commands

Related topics:
Reasons to use VDN Variables on page 506

Reasons to use VDN Variables

You can create general-purpose vectors that support multiple applications with call-wait treatments that are tailored to the application. For example, you can create a single vector that can be used by multiple applications that are the same except for the announcement. Even when using only one vector, callers can still hear an announcement that is appropriate for their call. This can reduce the need for more vector capacity.

For more information about VDN variables, see *Programming Call Vectors in Avaya Aura™ Call Center*.

Chapter 5: Call Vectoring and BCMS/CMS interactions

About Call Vectoring and BCMS/CMS interactions

Call Vectoring interacts with a management information system that helps to monitor and report on the activity within Call Vectoring. In most cases, the management system is either the Call Management System (CMS) or the Basic Call Management System (BCMS).

The CMS, which resides on an adjunct processor, collects and processes ACD information to generate reports. BCMS, which resides on the switch, also collects ACD information and generates a limited number of reports. The CMS reporting and data storage capabilities are much more extensive than those of the BCMS.

BCMS collects and processes ACD information to generate various reports.

This section is intended to illustrate how this system interprets these management systems interpret and reports report on activity within Call Vectoring. Special emphasis is placed on interpreting and reporting on this activity as it occurs within splits during a series of Call Vectoring events.

Note:
`Call Vectoring commands` provides a summary of the CMS/BCMS interactions with each Call Vectoring command (where applicable).

CMS/BCMS tracking in a Call Vectoring environment

About CMS/BCMS tracking

Tracking is the identifying of call flows and other actions relevant to call handling. There are three classes of call flows: split flows, VDN flows, and vector flows. We are most concerned with tracking in the Call Vectoring environment. The specific types of call flows and actions in this environment that are tracked by the CMS/BCMS include the following:

- Inflows (flow ins)
- Outflows (flow outs)
- Dequeues
- Abandons
- Answers
- Busies
- Disconnects

The split supervisor can use VDN and vector flows to evaluate how effective vector programming is at the site in question. The supervisor can use split flows to determine the manner in which the splits at the site are handling incoming telephone calls.

Defining and interpreting call flows

The manner in which specific call flows are defined and interpreted depends upon the call flow class in question, the management system in effect, and the version of the switch being used. Management systems include CMS and BCMS.

The following sections define and interpret specific call flows according to these parameters.

Related topics:

Answered and abandons on page 508
Busies and disconnects on page 509
VDN inflows and outflows on page 509
Vector inflows and outflows on page 510
Split inflows, outflows, and dequeues on page 510

Answered and abandons

The most important tracking items for most VDNs and vectors are the number of calls answered and the number of calls abandoned. The CMS provides VDN profiles that show when calls are answered and abandoned. Ten service level intervals are administered for these profiles. These intervals can have smaller time intervals around the time most calls are answered and when most calls abandon to get more detailed information.

This data can be used to determine what an acceptable service level is for most callers. The percentage answered within the administered acceptable service level is also shown on the Call Profile reports. For VDNs, the calculation is ACD calls answered and non ACD calls connected within the service level divided by calls offered to the VDN (including calls that inflow to the VDN).

For split/skill statistics, the calculation is ACD calls answered within the service level divided by calls queued to the split/skill (answered calls, abandoned calls, calls that flow out, calls that dequeue). In most cases the VDN percentage will be higher then the split percentage since calls dequeued from a split/skill are counted as answered, abandoned, or outflows for the VDN.

Changes made to a vector or to staffing will typically affect the VDN call profile. Even the wording of an announcement can affect the abandon profile. It is worthwhile to review the VDN's call profile before and after any change to determine if the change had a positive impact.

Busies and disconnects

Busy calls and forced disconnects reported on the CMS indicate how many calls this VDN/ vector turned away. If forced disconnect is used out of business hours, this item would indicate how many customers expected you to be operating during a specific time interval. If busies are given when the queues are full or waiting times are long, the number of busies in an interval might suggest a staffing change is needed. If disconnect is used to deny a look-ahead interflow attempt, a large number of denials would indicate a busy time at multiple sites.

VDN inflows and outflows

The following section discusses the specific VDN flows for CMS and BCMS.

Related topics:

CMS and BCMS standards for interpreting VDN inflows and outflows on page 509

CMS and BCMS standards for interpreting VDN inflows and outflows

The following table illustrates how CMS and BCMS interprets specific VDN flows from the switch:

CMS and BCMS standards for interpreting VDN flows		
Flow type	**Management system**	**Interpretation**
VDN flow in	CMS	Calls that flow into the VDN using a **`route-to VDN`** command or by Redirection on No Answer to a VDN.
	BCMS	(Not tracked.)
VDN flow out	CMS	Calls that successfully flow out of a VDN to another VDN or to an external location using a **`route-to`** command.
	BCMS	Same as for CMS.

Vector inflows and outflows

The following section discusses the specific vector flows as recorded by CMS.

Related topics:

[CMS standards for vector inflows and outflows](#) on page 510

CMS standards for vector inflows and outflows

Vector flow in pertains to calls that flow into a vector from another vector using a route to or a goto vector command. Vector flow out pertains to calls that successfully flow out of a vector using a route to or a goto vector command.

Split inflows, outflows, and dequeues

The following sections discuss the various split flow types for CMS and the BCMS.

Related topics:

[CMS and BCMS standards for interpreting split flows](#) on page 510
[Examples of split flow tracking](#) on page 511
[Evaluating split performance](#) on page 515

CMS and BCMS standards for interpreting split flows

The CMS and the BCMS are grouped together because both of these systems interpret two split flow types identically. These flows include inflow and outflow. The CMS interprets another split flow type, dequeue. The BCMS does not interpret this split flow type because it does not have a dequeue tracking item. This means that in a situation where the CMS tracks a dequeue, BCMS does not because it is unable to do so.

Before we detail how the CMS and the BCMS interpret split flows, we should discuss the term primary split, since this concept plays a significant role in tracking. Primary split is defined as the first split in a VDN to which a call actually queues. Therefore, this split is not necessarily the first split referenced in the vector.

Another split becomes the primary split if either of the following events occurs:

- Call cannot queue to the originally-targeted split because the split has no queue slots available.
- Call leaves the VDN (using a **`route-to`** VDN command, for example) and is queued to another split as a result.

If the call leaves vector processing and does not queue to another split (as a result of a **`route-to extension`** command, for example), there is no new primary split.

With this discussion in mind, let's take a look at the following table to see how CMS and BCMS interpret split flows for the switch.

Table 44: CMS and BCMS standards for interpreting split flows

Flow type	Management system	Interpretation
Inflow	CMS	Calls that ring at an agent in a split other than the primary.
	BCMS	Same as for CMS.
Outflow	CMS	Calls that are dequeued from a primary split using a **`route-to`** or **`messaging split`** command, or by ringing at or being answered by an agent in another split to which the call is also queued.
	BCMS	Same as for CMS.
Dequeue	CMS	Calls that are dequeued from and not answered by any split other than the primary split in a VDN.
	BCMS	Not tracked.

When a call is not answered (due to an outflow, abandon, busy, or disconnect), the call's disposition is tracked for the primary split as long as the call is still queued when the call abandons, outflows, etc. However, if the call abandons or outflows from ringing, the disposition is recorded for the split for which it was ringing. On the CMS, the other splits to which the call is queued track a dequeue when the call outflows, abandons, is given busy treatment, or is disconnected.

If the primary split in a VDN is unmeasured, an outflow, abandon, busy, or disconnect is not tracked for the call. Also, an answer is not tracked if the call is answered by an agent in the primary split.

Examples of split flow tracking

The following sections provide some examples of tracking in CMS and BCMS. Each section first presents a scenario of Call Vectoring events. The scenario is then followed by a table in which the tracking for the various splits involved is recorded. Following each tracking table, an explanation of the tracking procedure is provided.

The scenarios presented include the following:

- Call answered by a primary split.
- Call answered by a non primary split.
- Call abandoned from queue.
- Call answered by a primary split after a route to VDN.
- Call answered by a non primary split after a route to VDN.
- Call answered after a route to split.

Note:

Inflows, outflows, and dequeues are not tracked for splits administered by the **`converse-on split`** command. However, if a call is answered both by a converse split and (subsequently) by a non converse split, an answer is tracked for each split. However, a call is really considered answered only when it is answered by a non converse split. Therefore, traffic measurements for converse splits should be used only to measure converse split traffic and not to calculate the total number of calls.

Call answered by a primary split:

The following scenario involves a call answered by the primary split. The scenario is as follows:

1. Call comes into a VDN whose vector queues the call to splits 1, 2 and 3.
2. Call is answered in split 1.

The following table shows the tracking table for this scenario:

Table 45: Tracking for call answered by primary split

Split tracking			
	1	2	3
CMS	answer	dequeue	dequeue
BCMS	answer		

Comments:

- CMS: Dequeue is tracked in split 2 as well as in split 3 because the call is answered by the primary split (split 1) and is thus dequeued from splits 2 and 3 without being answered in these splits.
- BCMS: No dequeue tracking item is available.

Call Answered by a non-primary split: The following scenario involves a call answered by a non primary split. The scenario is as follows:

1. Call comes into a VDN whose vector queues the call to splits 1, 2 and 3.
2. Call is answered in split 2.

The following table shows the tracking table for this scenario:

Table 46: Tracking for call answered by non-primary split

	Split tracking		
	1	2	3
CMS	outflow	inflow answer	dequeue
BCMS	outflow	inflow answer	

Comments:

- CMS: Outflow is tracked in split 1 because the call is answered by an agent in another split to which the call is queued (that is, split 2). Although the call is obviously removed from split 1 after it is answered in split 2, dequeue is not tracked in split 1 because split 1 is the primary split. Inflow is tracked in split 2 because the call is answered in this split and the split is not the primary split. Dequeue is tracked in split 3 because the call is removed from the split without being answered there. When the call is removed from split 3, outflow is not tracked in split 3 because this split is not the primary split.

- BCMS: Outflow is tracked in split 1 because the call is answered by an agent in another split to which the call is queued (that is, split 2). Inflow is tracked in split 2 because the call is answered in this split and the split is not the primary split. When the call is removed from split 3, outflow is not tracked in split 3 because this split is not the primary split.

Call Abandoned: The following scenario involves a call abandoned by the caller. The scenario is as follows:

1. Call comes into a VDN whose vector queues the call to splits 1, 2, 2 and 3.
2. Call is abandoned.

The following table shows the tracking table for this scenario:

Table 47: Tracking for Abandoned Calls

	Split Tracking		
	1	**2**	**3**
CMS	abandon	dequeue	dequeue
BCMS	abandon		

Comments:

- CMS: Abandon is tracked in split 1 because this split is the primary split. Dequeue is tracked in splits 2 and 3 because the call is dequeued from these splits without being answered in either split.
- BCMS: Abandon is tracked in split 1 because this split is the primary split. Tracking is not recorded in splits 2 and 3 because no dequeue tracking item is available.

Call answered by a primary split after a route to VDN: The following scenario involves a call answered by the primary split after a **`route-to`** VDN command is executed. The scenario is as follows:

1. Call comes into a VDN whose vector queues the call to splits 1, 2 and 3.
2. Vector executes a **`route-to VDN`** step.
3. Call is then queued to splits 4, 5 and 6.
4. Call is answered in split 4.

The following table shows the tracking table for this scenario.

Table 48: Tracking for call answered by primary split after route to VDN

	Split tracking					
	1	**2**	**3**	**4**	**5**	**6**
CMS	outflow	dequeue	dequeue	answer	dequeue	dequeue
BCMS	outflow			answer		

Comments:

Split 1 is the original primary split, because this is the first split to which the call actually queues. However, split 4 becomes the new primary split because:

- Call leaves the original VDN upon execution of the **`route-to VDN`** step.
- Split 4 is the first split to which the call queues upon execution of this step.
- CMS: Outflow is tracked in split 1 because this split is the original primary split, and the call is dequeued from this split using a **`route-to VDN`** step. Dequeue is tracked in splits 2, 3, 5, and 6 because the call is dequeued from each of these splits without being answered in any one of them.
- BCMS: Outflow is tracked in split 1 because this split is the original primary split.

Call answered by the non-primary split after a route to VDN: The following scenario involves a call answered by the non primary split after a **`route-to VDN`** command is executed. The scenario is as follows:

1. Call comes into a VDN whose vector queues the call to splits 1, 2 and 3.
2. Vector executes a **`route-to VDN`** step.
3. Call is then queued to splits 4, 5 and 6.
4. Call is answered in split 5.

The following table shows the tracking table for this scenario:

Table 49: Tracking for call answered by non-primary split after route to VDN

	Split tracking					
	1	**2**	**3**	**4**	**5**	**6**
CMS	outflow	dequeue	dequeue	outflow	inflow answer	dequeue
BCMS	outflow			outflow	inflow answer	

Comments:

- CMS: Outflow is tracked in split 1 because this split is the original primary split, and the call is dequeued from this split using a **`route-to`** VDN step. Dequeue is tracked in splits 2, 3, and 6 because the call is dequeued from each of these splits without being answered in any one of them. Outflow is tracked in split 4 because this split becomes the new primary split after the **`route-to VDN`** step is executed and the call is subsequently dequeued from this split by being answered in another split (split 5) to which the call is also queued. Finally, inflow is tracked in split 5 because the call is answered in this split, and the split is not the primary split.
- BCMS: Outflow is tracked in split 1 because this split is the original primary split. Outflow is tracked in split 4 because this split becomes the new primary split after the **`route-to VDN`** step is executed. Finally, inflow is tracked in split 5 because the call is answered in this split, and the split is not the primary split.

Call answered after a route to split: The following scenario involves a call answered after it is routed to a split using a **`route-to digits`** or **`messaging split`** command. The scenario is as follows:

1. Call comes into a VDN whose vector queues the call to splits 1, 2 and 3.
2. Vector executes a **`route-to digits`** (or **`messaging split`**) step.
3. Call is queued to split 4 and answered by an agent in split 4.

The following table shows the tracking table for this scenario:

Table 50: Tracking for call answered after route to split

	Split tracking			
	1	2	3	4
CMS	outflow	dequeue	dequeue	answer
BCMS	outflow			answer

Comments:

- CMS: Outflow is tracked in split 1 because this split is the original primary split, the call is dequeued from this split using a **`route-to digits`** (or **`messaging split`**) step, and the call is answered in split 4, which becomes the new primary split. Dequeue is tracked in splits 2 and 3 because the call is dequeued from each of these splits without being answered in any one of them.
- BCMS: Outflow is tracked in split 1 because this split is the original primary split, and the call is answered in split 4, which becomes the new primary split.

Evaluating split performance

By using the information presented to this point, along with the information from various reports (as discussed in the next section), the split supervisor can answer one or more questions concerning split performance and then make adjustments, if necessary. Here are some of the questions the supervisor can answer:

1. How many ACD calls offered to my split were mine (that is, were offered to this split as the primary split)?

   Note:
   Split ACD calls include direct agent calls for BCMS, but not for CMS, which tracks direct agent calls separately.

2. How many of my ACD calls did my split not answer?
3. How many ACD calls that I didn't answer weren't mine?

The following sections present the answers to these questions from the perspective of the CMS and BCMS.

CMS

The following answers reflect the use of the CMS:

- The number of calls offered to my (primary) split that were mine can be determined by an examination of the CMS Split Summary Report. The algorithm is as follows: CALLSOFFERRED - INFLOWCALLS - DEQUECALLS (that is, the total number of calls offered minus the number of calls not mine that I answered minus the number of calls not mine that I didn't answer.)
- The number of my calls that my split didn't answer can be determined by an examination of the CMS VDN Report. The algorithm is as follows: ABNCALLS + BUSYCALLS + DISCCALLS + OUTFLOWCALLS (that is, the number of abandoned calls plus the number of busy calls plus the number of disconnected calls plus the number of calls outflowed from my split tagged as a primary split).
- The number of calls not mine that my split didn't answer is DEQUECALLS, which is indicated in the CMS Split Summary Report.

BCMS

The number of calls offered to my split that were mine can be determined by an examination of the BCMS Split Report. The algorithm is as follows: ACDCALLS + ABNCALLS + OUTFLOWCALLS - INFLOWCALLS (that is, the total number of calls answered plus the total number of calls abandoned from my split tagged as a primary split plus the number of calls that outflowed my split tagged as a primary split minus the number of calls answered that were not directed to my split tagged as a primary split).

Using CMS and BCMS reports to evaluate Call Vectoring activity

There are a number of CMS and BCMS reports that allow you to evaluate Call Vectoring activity. Some of these facets include the call flows present within Call Vectoring as well as the speeds at which calls are answered. The sections that follow identify and discuss the CMS and BCMS reports that indicate this activity.

This section includes the following topics:

- CMS reports on page 516
- BCMS reports on page 517

Related topics:

CMS reports on page 516
BCMS reports on page 517

CMS reports

CMS has real-time, historical, and integrated reports. Most of the CMS historical reports are available in four versions: intra-hour, daily, weekday, and monthly. The following list identifies and describes several CMS reports that summarize Call Vectoring activity. For further details on these and other related reports, see Avaya CMS Supervisor Reports.

Note:
The reports described in this section are generated in CMS R3 and newer releases of the CMS. Corresponding CMS R2 reports do not provide information that reflects capabilities that are new to the switch (for example, internal/external call tracking).

- Split Summary Report summarizes the call activity for an entire split. Among other information, the report provides the number of calls answered, the total number of flow ins (inflows), flow outs (outflows), dequeues, and abandoned calls.

  The report also indicates the average speed of answer (interval ASA) for calls. This refers to the sum of the queue time and ring time for a call within the answering split only. Finally, the report indicates the dequeued average queue time, which is the average time a call waits until it is answered by another split to which the call is also queued.
- VDN Report summarizes VDN activity for specific vectors. Among other information, the report provides calls answered, connected, abandoned, the number of VDN Flow Ins/Outs, calls forced busy, and calls forced disconnect. VDN Flow In pertains to calls that flow into a VDN from another VDN using a **`route-to`** command. VDN Flow Out pertains to calls that successfully flow out of VDN to another VDN or external location using a **`route-to`** command.
- Vector Report summarizes vector activities. Among other information, the report provides the number of calls offered, calls answered, calls abandoned, Vector Flow Ins/Outs, calls forced busy, and calls forced disconnect. Vector Flow In pertains to calls that flow into a vector from another vector using a **`route-to`** or **`goto vector`** command. Vector Flow Out pertains to calls that successfully flow out of a vector using a **`route-to`** or **`goto vector`** command.

BCMS reports

BCMS has a real-time split report, split historical reports, real-time VDN reports, and VDN historical reports. The following list identifies and describes several BCMS reports that summarize Call Vectoring activity. For more information on these and other related reports, refer to Avaya Aura™Communication Manager 5.2 Software - Basic Call Management System (BCMS) Operations.

BCMS Split Report

Summarizes the call activity for an entire split. The information can be requested either daily or by the administered time period. Among other information, the report provides the total number of flow ins (inflows) and flow outs (outflows), the calls answered and calls abandoned. The report also provides the average speed of answer time for calls handled by the split during the indicated time period.

VDN Summary Report

Summarizes statistical information for all internally-measured VDNs. The information can be requested by the administered time interval or daily. The list bcms vdn report gives multiple time periods or days for a single VDN. The list bcms summary vdn report gives a one-line summary per vdn (with data from the specified times or days), but can give the data for numerous vdns.

The report also indicates the total number of flow outs, specifically, the number of calls that route to another VDN or to a destination external to the switch. However, calls that encounter a `goto vector` command are not shown as outflows. No further measurements are taken on the calls once the calls have outflowed. If an outflowed call later abandons, this is not indicated in the report.

Among other information, the VDN report provides a total for offered calls, answered calls, abandoned calls, and also one for calls that were either forced busy or forced disconnect.

VDN Real-Time Report

Provides statistical information including the number of calls currently waiting and the oldest call waiting. The VDN real-time report has the same characteristics as other real-time BCMS reports.

Using CMS in an EAS environment

The same tracking and database items used within a traditional Call Vectoring environment are used within an EAS environment but there are also new items that are specific to EAS. All existing custom reports should work when you are upgrading to EAS.

Related topics explain how the following entities are tracked in an environment with EAS optioned.

Related topics:

Agents and their skills on page 518
DAC calls on page 519
Non-ACD calls on page 519
VDN skill preferences on page 519
EAS administration from CMS on page 520

Agents and their skills

The fields under the Extn column in the CMS Real-Time Agent Report show the extension that the agent is logged into. These fields can be used to locate the agent or to service observe the agent.

With EAS optioned, the Skill Status Report replaces the Split Status Report. This report indicates the skills logged into and the skill level of each skill. If too many calls are waiting, or if calls are waiting too long (also shown on the Skill Status report), it is possible that not enough agents have the skill administered at a high enough skill level.

An agent may be denied login to some skills if the maximum agents/skill number is met or if the CMS limit on agent/skill pairs logged in has been reached.

CMS reports show only the first 15 skills that an agent is logged into.

DAC calls

Waiting direct agent calls are not included in the Calls Waiting and Oldest Call Waiting report fields for skills because such calls are not skill calls. However, direct agent calls are included in these two report fields for VDNs.

The Queue/Agent Summary Real-Time Report lists separately the direct agent calls waiting in a skill queue. direct agent calls are queued to the skill that is administered as the direct agent skill. To manage the skill's queue slots effectively, it is recommended that a skill be dedicated for direct agent calls.

Since direct agent calls are not skill calls, the skill tables do not track direct agent calls; however, the tables do monitor skill queue slots. The agent's time is tracked as OTHER in the skill tables. In the agent tables, there are separate direct agent call items. The standard CMS agent reports add the direct agent calls and the skill ACD calls and report these calls as ACD calls. The VDN tables track direct agent calls as ACD calls.

Non-ACD calls

The first measured skill that an EAS agent is logged into is used by CMS to track non-ACD calls unless the agent has an ACD call on hold. If an ACD call is on hold, outgoing non-ACD calls are counted for the skill of the held ACD call.

VDN skill preferences

VDN skill preference data is collected to provide information on what groups of agents (skills) are handling calls and on how effectively each skill group handles a particular VDN.

Real-time and historical VDN Skill Preference reports can be used to compare the percentage of calls being answered by the 1st, 2nd, and 3rd VDN preferences against an objective. If too few calls are being answered by the 1st skill preference, the vector can be adjusted to allow more time for the 1st skill preference group to answer calls; another alternative is to train or hire more agents with the 1st skill preference.

You can use VDN skill preference data to compare the average talk time and average ACW time for agents in the 1st, 2nd, and 3rd skill groups. If these times vary too much across groups, more training may be needed for the backup groups (that is, the 2nd and 3rd skill groups).

VDN skill preference data is tracked according to the skill preferences (1st, 2nd, 3rd) assigned to the VDN. Whenever a vector step either references a 1st, 2nd, or 3rd skill or specifies a skill number that matches the 1st, 2nd, or 3rd skill administered, the new database items are tracked. For example, if VDN 1000 has Skills 21, 22, and 23 administered as the 1st, 2nd, and 3rd skills, respectively, and if the vector associated with VDN 1000 has a queue to main skill 22 step, tracking occurs for the 2nd VDN skill preference if the call is answered by an agent in Skill 22. Skill preference tracking also occurs for Skills 21 and 23. This allows users who prefer to specify the actual skill number in the vector to take advantage of the tracking for VDN skill preferences.

EAS administration from CMS

CMS can be used to administer vectors as well as skills for agents and VDNs. The ACD Administration: Change Agent Skills CMS screen is used to display and modify the skills and levels assigned to an agent, as well as the assigned direct agent skill and call handling preference.

The ACD Administration: Change VDN Skill Preferences screen is used to request a VDN's skill preferences and to modify the VDN's skills.

The CMS Vector Contents screen is used to create and modify vectors. CMS supports the Call Vectoring commands that queue calls to the 1st, 2nd, or 3rd VDN skill.

Chapter 6: Call Vectoring Job aids

Vector commands job aid

The vector command job aid shown in this section lists the Call Vectoring commands, together with the various conditions, and parameter options and values that are available for use with each command.

Most vector commands require one or more input values for the command, as well as for various parameters, such as an announcement extension number, a time interval, a maximum queue size, and so forth. When the minimum and maximum ranges for command parameter values are identical for all Avaya switch platforms, the limiting ranges are specified in the job aid. Alternately, when the minimum and maximum ranges for a parameter value are not the same among Avaya switch platforms, the upper limit of a value range is indicated by the term *switch max*.

To determine the maximum values you can use in Call Vectoring commands, see System Capacities Table for Communication Manager on Avaya Media Servers. You can find the latest capacity tables from the Avaya support Website at:

http://www.avaya.com/support

For detailed information about these commands, see Call Vectoring commands.

#	[A comment command that adds a note with up to 71 characters.]
	[A comment out command that tells a vector step to ignore processing. Use the edit function, <esc> f6, to insert this command.]

`adjunct routing link`	*1-64* - CTI Link ID[23] *[A-Z, AA-ZZ] V1-V9*

`announcement`	*extension no. [A-Z , AA-ZZ] V1-V9*

busy

[23] Link capacity depends on your release and configuration. For more information, see System Capacities Table for Communication Manager on Avaya Media Servers .

<table>
<tr><td rowspan="9">Check</td><td rowspan="3" colspan="2">best</td><td rowspan="3">if</td><td colspan="2">expected wait</td><td colspan="3">< 1-9999 seconds,
> 0-9999 seconds</td></tr>
<tr><td colspan="5">unconditionally</td></tr>
<tr><td colspan="2">wait improved</td><td colspan="3">< 1-9999 seconds,
> 0-9999 seconds</td></tr>
<tr><td rowspan="3">skill</td><td rowspan="3">hunt group[24], skills for VDN: 1st, 2nd, 3rd</td><td rowspan="3">pri</td><td rowspan="3">priorities:
l = low
m = medium
h = high
t = top</td><td rowspan="3">if</td><td rowspan="3">available-agents
> 0-1499[25]</td><td>all-levels</td></tr>
<tr><td>pref-level
skill level 1[26]</td></tr>
<tr><td>pref-range
skill level 1[26]
to skill level 2[26]</td></tr>
<tr><td>skill</td><td>hunt group[24], skills for VDN: 1st, 2nd, 3rd</td><td rowspan="2">pri</td><td rowspan="2">priorities:
l = low
m = medium
h = high
t = top</td><td rowspan="2">if</td><td rowspan="2" colspan="2">calls-queued < 1-999[25]
expected-wait < 1-9999 seconds
oldest-call-wait > 1-999 seconds
rolling-asa < 1-999 seconds
staffed-agents > 0-1499[25]
wait-improved > 0-9999 seconds
unconditionally</td></tr>
<tr><td>split</td><td>hunt group[24]</td></tr>
<tr><td>split</td><td>hunt group[24]</td><td>pri</td><td>l = low
m = medium
h = high
t = top</td><td>if</td><td colspan="2">available-agents > 0-1499[25]</td></tr>
</table>

<table>
<tr><td rowspan="3">collect</td><td>ced</td><td rowspan="2">for</td><td colspan="3" rowspan="2">none
A-Z, AA-ZZ</td></tr>
<tr><td>cdpd</td></tr>
<tr><td>1-16</td><td>digits after announcement</td><td>extension no.
none A-Z, AA-ZZ V1-V9</td><td>for</td><td>none A-Z, AA-ZZ</td></tr>
</table>

<table>
<tr><td>consider</td><td>location[27] (multi-</td><td>1-255 A-Z, AA-ZZ V1-V9</td><td>adjust by</td><td>0-100 percent A-Z, AA-ZZ V1-V9</td></tr>
</table>

24 A valid hunt group is a vector-controlled ACD split or skill assigned on a hunt group form.

25 The maximum limit is less on some platforms. Use the help key for your switch administration software to determine the applicable limit for your system.

26 Skill levels are 1-16 (1 is best, 16 is lowest). Skill Level 2 must be greater than or equal to Skill Level 1.

27 This item is available only with the Virtual Routing feature.

	site BSR only)					
	skill	*hunt group*[28] , skills for VDN: **1st** , **2nd** , **3rd**	**pri**	priorities: **l = l** ow **m = m** edium **h = h** igh **t = t** op		
	split	hunt group[28]				

converse-on	**skill**	*hunt group*[29] , skills for VDN: **1st** , **2nd** , **3rd**	**pri**	priorities: **l = low** **m = medium** **h = high** **t = top**	**passing**	*6-digit string* *** # ani vdn digits qpos wait** *A-Z, AA-ZZ V1-V9*	**and**	*6-digit string* *** # ani vdn none digits qpos wait** *A-Z, AA-ZZ V1-V9*
	split	hunt group[29]				**none**	**and**	**none**

disconnect	after announcement	*extension no.* **none** *A-Z, AA-ZZ V1-V9*

goto step and goto vector		
goto step ***1-99*** **if** **or** **goto vector** ***1-8000***[30] **@step** ***1-99*** **if**		
A-Z, AA-ZZ	>,<,=,<>, >=,<=	threshold value or string of digits: *1-16, wildcards* (*?* , +)[31] , *[A-Z, AA-ZZ], V1-V9*
	=,<>	none[32], #[33]
	in table	1-100[30], [A-Z, AA-ZZ], V1-V9
	not-in table	

28 A valid hunt group is a vector-controlled ACD split or skill assigned on a hunt group form.
29 A valid hunt group is a vector-controlled ACD split or skill assigned on a hunt group form.
30 The maximum limit is less on some platforms. Use the help key for your switch administration software to determine the applicable limit for your system.
31 The question mark (?) is a wild card that matches any digit (0-9) at the specified position. The plus sign (+) matches any or no characters at the specified position.
32 Use the word "none" in the threshold field to test for an empty digits string. Only the = or the <> comparators are valid in this case.
33 The # character is used in the threshold field to match a single # digit entered by the caller or an ASAI adjunct in the dial-ahead buffer. In this case, only the = or <> comparators are valid.

goto step and goto vector				
`goto step` ***1-99*** `if` or `goto vector` ***1-8000***[30] `@step` ***1-99*** `if`				
ani	>,>=,<>,=,<,<=	1-16, wildcards (?, +)[32], [A-Z, AA-ZZ], V1-V9		
	=,<>	none[32], #[33]		
	in **table**	1-100[30], [A-Z, AA-ZZ], V1-V9		
	not-in **table**			
available - agents	**in skill**	*hunt group*[34], skills for VDN: 1st, 2nd, 3rd	>,>=,<>,=, <,<=	*0-1499*[30] *1-1500*[30] *A-Z, AA-ZZ V1-V9*
	in split	hunt group[34]		

goto step and goto vector						
`goto step` ***1-99*** `if` or `goto vector` ***1-8000***[35] `@step` ***1-99*** `if`						
calls-queued	**in skill**	hunt group[36], skills for VDN: 1st, 2nd, 3rd	**pri**	priorities: **l** = low **m** = medium **h** = high **t** = top	> >= <> = < <=	*0-098*[35] *1-999*[35] *A-Z, AA-ZZ V1-V9*
	in split	hunt group[36]				
counted-calls	**to vdn**	vdn extension, latest, active[37]	>,>=,<>,=,<, <=		*0-998*[35] *1-999*[35] *A-Z, AA-ZZ V1-V9*	
digits	>,>=,<>,=,<,<=	threshold value or string: *1-16, wildcards* (*?, +*)[38] , *[A-Z, AA-ZZ], V1-V9*				

34 A valid hunt group is a vector-controlled ACD split or skill assigned on a hunt group form.

35 The maximum limit is less on some platforms. Use the help key for your switch administration software to determine the applicable limit for your system.

36 A valid hunt group is a vector-controlled ACD split or skill assigned on a hunt group form.

37 *Active* refers to the VDN specified by VDN Override settings. *Latest* refers to the VDN specified for the current vector.

38 The question mark (?) is a wild card that matches any digit (0-9) at the specified position. The plus sign (**+**) matches any or no characters at the specified position.

goto step and goto vector						
goto step ***1-99*** **if** **or** **goto vector** ***1-8000***[35] **@step** ***1-99*** **if**						
	<>,=	none[39]				
	=	meet-me-access[40]				
	in table	1-100[35], [A-Z, AA-ZZ], V1-V9				
	not-in table					
expected-wait	**for** best	>,>=,<>,=,<,<=	0-9999 seconds, [A-Z, AA-ZZ], V1-V9			
	for call					
	for split	hunt group[36]	**pri**	priorities: **l = low** **m = medium** **h = high** **t = top**	> >= <> = < <=	*0-9998* sec *1-9999* sec *A-Z, AA-ZZ V1-V9*
	for skill	*hunt group*[36], skills for VDN: 1st, 2nd, 3rd				

goto step and goto vector		
goto step ***1-99*if** **or** **goto vector** ***1-8000***[41] **@step** ***1-99*if**		
holiday	in table	1-99, [A-Z, AA-ZZ], V1-V9
	not-in table	
ii-digits	>, >=, <>, =, <, <=	2-digit string, wildcards (?, +)[42], [A-Z, AA-ZZ], V1-V9
	<>, =	none[43]

39 Use the word none in the threshold field to test for an empty digits string. Only the = or the <> comparators are valid in this case.

40 This item is available only with meet-me conference vectors.

41 The maximum limit is less on some platforms. Use the help key for your switch administration software to determine the applicable limit for your system.

42 The question mark (?) is a wild card that matches any digit (0-9) at the specified position. The plus sign (**+**) matches any or no characters at the specified position.

43 Use the word none in the threshold field to test for an empty digits string. Only the = or the <> comparators are valid in this case.

<table>
<tr><th colspan="4">goto step and goto vector</th></tr>
<tr><th colspan="4">goto step 1-99if
or
goto vector 1-8000[41] @step 1-99if</th></tr>
<tr><td rowspan="2"></td><td>in table</td><td colspan="2" rowspan="2">1-100[41], [A-Z, AA-ZZ], V1-V9</td></tr>
<tr><td>not-in table</td></tr>
<tr><td>interflow-gpos</td><td>>,>=,<>,=,<,<=</td><td colspan="2">1-9, [A-Z, AA-ZZ], V1-V9</td></tr>
<tr><td rowspan="3">media-gateway</td><td>H.248 gateway ID[44] 1-999</td><td rowspan="3">=, <></td><td rowspan="3">registered</td></tr>
<tr><td>all</td></tr>
<tr><td>any</td></tr>
<tr><td colspan="4">meet-me-full[45] (goto step only)</td></tr>
<tr><td colspan="4">meet-me-idle[45] (goto step only)</td></tr>
<tr><td colspan="4">no match[46]</td></tr>
</table>

<table>
<tr><th colspan="7">goto step and goto vector</th></tr>
<tr><th colspan="7">goto step 1-99if
or
goto vector 1-8000[47] @step 1-99if</th></tr>
<tr><td rowspan="2">oldest-call-wait</td><td>in skill</td><td>hunt group[48], skills for VDN: 1st , 2nd , 3rd</td><td rowspan="2">pri</td><td rowspan="2">priorities: l = low m = medium h = high t = top</td><td rowspan="2">>,>=,<>,=,<=</td><td rowspan="2">0-998 sec 1-999 sec A-Z, AA-ZZ V1-V9</td></tr>
<tr><td>in split</td><td>hunt group[48]</td></tr>
</table>

44 The maximum number of port networks and media-gateways supported varies with the server platform. For example, the S8710 server supports up to 64 port networks and 250 media gateways. Check capacity tables for supported limits.

45 This item is available only with meet-me conference vectors.

46 This item is available only with the Dial by Name feature.

47 The maximum limit is less on some platforms. Use the help key for your switch administration software to determine the applicable limit for your system.

48 A valid hunt group is a vector-controlled ACD split or skill assigned on a hunt group form.

goto step and goto vector				
goto step ***1-99*****if** or **goto vector** ***1-8000*****[47] @step** ***1-99*****if**				
port-network	Port network ID[49] *1-999*		=, <>	registered
	all			
	any			
queue-fail[50]				
rolling-asa	**for skill**	*hunt group*[48], skills for VDN: 1st, 2nd, 3rd	>, >=, <>, =, <, <=	*0-998 sec, 1-999 sec A-Z, AA-ZZ V1-V9*
	for split	hunt group[48]		
	for vdn	vdn extension, latest, active[51]		
server	=, <>	main, ess, lsp		
service-hours	in table		1-99, [A-Z, AA-ZZ], V1-V9	
	not-in table			
staffed-agents	**in skill**	*hunt group*[48], skills for VDN: 1st , 2nd , 3rd	>, >=, <>, =, <, <=	*0-1499*[47], *1-1500*[47] *A-Z, AA-ZZ V1-V9*
	in split	hunt group[48]		

49 The maximum number of port networks and media-gateways supported varies with the server platform. For example, the S8710 server supports up to 64 port networks and 250 media gateways. Check capacity tables for supported limits.

50 This item is available only with the Attendant Vectoring feature.

51 *Active* refers to the VDN specified by VDN Override settings. *Latest* refers to the VDN specified for the current vector.

goto step and goto vector								
goto step ***1-99*** **if** **or** **goto vector** ***1-8000***[52] **@step** ***1-99*** **if**								
time-of-day	**is**	mon, tue, wed, thu, fri, sat, sun, all	hour: 00-23	minute: 00-59	**to**	mon, tue, wed, thu, fri, sat, sun, all	hour: 00-23	minute: 00-59
V1-V9	>, <, =,<>, >=, <, <=	threshold value or string of digits: 1-16, wildcards (?, +), [A-Z, AA-ZZ], V1-V9						
	=,<>	none[53], #[54]						
	in table	1-100[51], [A-Z, AA-ZZ], V1-V9						
	not-in table							
wait-improved for	best	>, >=, <>, =, <, <=				*0-9998* sec *1-9999* sec *A-Z, AA-ZZ* *V1-V9*		
	skill	*hunt group*[55], skills for VDN: 1st , 2nd , 3rd	**pri**	priorities: **l = low** **m = medium** **h = high** **t = top**	>,>=, <>,=,<=			
	split	hunt group[5]						
unconditionally								

messaging	**skill**	*hunt group*[56] **1st** (VDN skill) **2nd** (VDN skill) **3rd** (VDN skill)	**for extension**	*extension no.* **latestactive**[57] *A-Z, AA-ZZ V1-V9*

52 The maximum limit is less on some platforms. Use the help key for your switch administration software to determine the applicable limit for your system.

53 Use the word “none” in the threshold field to test for an empty digits string. Only the = or the <> comparators are valid in this case.

54 The # character is used in the threshold field to match a single # digit entered by the caller or an ASAI adjunct in the dial-ahead buffer. In this case, only the = or <> comparators are valid.

55 A valid hunt group is a vector-controlled ACD split or skill assigned on a hunt group form.

56 A valid hunt group is an ACD split or skill or a non-ACD hunt group assigned for AUDIX, remote AUDIX, MSA, or QSIG MWI on the hunt group.

57 *Active* refers to the VDN specified by VDN Override settings. *Latest* refers to the VDN specified for the current vector.

split	hunt group[56]			

queue-to	**attd-group**[58]			
	attendant[58]	extension no.		
	best			
	hunt-group[58]	group number[59]	**pri**	priorities: **l** = low **m** = medium **h** = high **t** = top
	skill	hunt group[60], VDN skills (**1st**, **2nd**, **3rd**)		
	split	hunt group[60]		

reply-best

return

route-to[61]	digits	with coverage	y, n				
	meetme[62]						
	number[63]	up to 16 digits (0-9) <digits>[A-Z, AA-ZZ, V1-V9]<digits>*<digits>A[64] [65]<digits>#<digits>A <digits>~p<digits>A	with cov	y, n	if	digit	> >= <> = <=	*0-9#* [67]

58 This item is available with only the Attendant Vectoring feature.

59 A valid group number is a vector-controlled hunt group of any type (ACD, UCD, and so on).

60 A valid hunt group is a vector-controlled ACD split or skill assigned on a hunt group form.

61 The route-to digits and route-to number commands support the Service Observing FACs, remote logout of agent FAC, remote access extension, attendant access number, and other dialable destination numbers.

62 This item is available only with meet-me conference vectors.

63 A destination for the route-to is entered in the number field. This field can contain an administration limit of a maximum of 16 decimal digits or combination of characters and numbers that total 16. Special notations (for example, ~p) with a ~ followed by a character are counted as two digits towards the 16. The number field supports some feature activations using Feature Access Codes (FACs) alone or followed by digits including Service Observing, Remote Logout of Agent, remote access extension, attendant access number, Forced Logout/Aux and other destination numbers that can be dialed with a phone. The number field also supports vector variables (A-Z, AA-ZZ) and VDN variables (V1-V9) whose value in decimal digits is defined elsewhere before the route-to number command is to be executed.

64 The notation means that 1 or more digits in the range of 0 to 9 can be inserted when necessary for the application. The total of digits and characters must be within the 16 digit positions total.

65 Either a vector variable (A-Z, AA-ZZ) or VDN variable (V1-V9) can be entered at the end of any entry (digits or special character) or entered in place of ; this is shown with an “A” in the other examples. Each variable whether a single character or double character counts as two digits towards the maximum of digits in the number field. The variable can be preceded by digits as long as the total is within the 16 digit/character position limit. The variable must always be the last entry and can not be followed by a digit. Use of a variable allows having a route-to number destination address of more than 16 digits since a variable can be assigned up 16 digits during processing and will be combined with the entry in the number field.

		<digits>~mA<digits>A <digits>~s<digits>A <digits>~w<digits>A <digits>~W<digits>A ~r~r+[66]				interflow-qpos	< = <=	1-9
						unconditionally		
	name1 [68]	with coverage	y, n					
	name2 [68]							
	name3 [68][67]							

set [*vector variable*, Digits] = [*operand1*] [*operator*] [*operand2*]

Command	Variables or digits		Operand1	Operator	Operand2
set	user-assigned[69] type A-Z or AA-ZZ vector variable	=	user-assigned[69]type A-Z or AA-ZZ vector variable	**ADD** **SUB** **MUL** **DIV** **CATL** **CATR** **MOD10** **SEL**	user-assigned[69] A-Z or AA-ZZ vector variable
	asaiuui A-Z or AA-ZZ vector variable				
	digits[70]		system-assigned[71] A-Z or AA-ZZ vector variable		system-assigned[71] A-Z or AA-ZZ vector variable
			V1-V9 VDN variable		directly-entered numeric string[72]
			digits		V1-V9 VDN variable
			none		**digits**

66 When the specified number is preceded by ~r, Network Call Redirection (NCR) invocation is attempted back over the trunk group to the network Service Provider. The ~r sequence is counted as two digit positions toward the 16 total. The + character is a special indication for E.164 numbering required by some network Service Providers for NCR invocation over SIP trunking. The "+" character is counted as two digit positions towards the 16 total. The ~r or ~r+ entries must be in the initial digit/character positions of the number field.

67 The # character is used in the threshold field to match a single # digit entered by the caller or an ASAI adjunct in the dial-ahead buffer. In this case, only the = or <> comparators are valid.

68 This item is available only with the Dial by Name feature.

69 Only global or local collect type vector variables can be assigned using the set command.

70 The collected digits buffer holds up to 16 digits.

71 For example, ani, asaiuui, doy, and so on.

72 Limited to 4294967295 with ADD, SUB, MUL, or DIV. For all other operators, the limit is 16 digits.

Command	Variables or digits		Operand1	Operator	Operand2
					none

```
stop
```

wait-time	0-999	**secs**	**hearing**	**music, ringback, silence, i-silent**		
	0-480[73]	**mins**		*audio source ext.*[74] *A-Z, AA-ZZ V1-V9*	**then**	**music ringback silence continue**[75]
	0-8[73]	**hrs**				

Vector variables job aid

For detailed information about variable types, see Variables in Vectors.

Items in bold are default values that cannot be changed.

Variable type	Description	Scope	Specification	Max digit length	Assigns
ani	Tests the caller's phone number	*L*	Start digit position and Length	16	Incoming call data
asaiuui	Processes call-specific user data associated with the call	*L*	Start digit position and Length	16 out of a total of 96	Incoming call or ASAI application data
collect	Processes collected digits for user-defined control, routing, or treatment	L, P, or G	Start digit position and Length	16	The for parameter of the collected digits command or assignment in the variables table
tod	Holds the current time of day in 24-	*G*	None	Always 4	The main server system clock - for

[73] This option is not available for vector administration done through Avaya Call Management System or Visual Vectors.
[74] This consists of a valid announcement or music source extension that is defined on the announcement audio sources form.
[75] The continue treatment is valid only with Multiple Audio/Music Sources. It indicates that the caller continues to hear the alternate audio or music source using an announcement until another vector command takes effect.

Variabl e type	Description	Scope	Specification	Max digit length	Assigns
	hour time for processing				example, 0219 = 2:19 am
dow	Holds the current day of week for processing	*G*	None	1	The main server system clock (1-7) - for example, 1 = Sunday
doy	Holds the current day of year for processing	*G*	None	Alway s 3	The main server system clock (1-365) - or 1 -366 in a leap year
stepcnt	Counts the number of vector steps executed for the call, including the current step	*L*	None	4	The vector processing step counter
value	Holds a single numerical digit (0-9) for user-defined processing	*G*	None	1	A user-defined value entered using the VAC FAC procedure or assignment in the variables table
vdn	Holds the VDN extension number of the call for processing	*L*	Active or Latest	7	Routing for a call
vdntim e	Provides the time in seconds that a call has been in vector processing by the call center	*L*	None	Alway s 4	Time in vector processing including prior processing for a call routed by BSR/LAI

Chapter 7: Advanced multi-site routing

Advanced multi-site routing

This section supplements the Look-Ahead Interflow (LAI) and Best Service Routing (BSR) sections covered in the chapter ACD Call Center Features.

This section is intended for users whose call center networks meet either or both of the following criteria:

- Five or more switches in the network
- Combination of low- and high-volume locations

Related topics:

Application architecture in multi-site BSR on page 533
User adjustment considerations and recommendations on page 533

Application architecture in multi-site BSR

Multi-site applications may be structured in a variety of ways. In general, however, most applications will fit one of two models: distributed or centralized. When each switch in a network may interflow calls to other switches and receive interflows, this is called a distributed system. A centralized system, by contrast, is one in which all calls are initially delivered to a single call center (the hub) and distributed from this site to queues at remote switches. A centralized system requires greater inter-switch trunking, since a greater percentage of calls need to be redirected. However, it may be an appropriate configuration if your organization has a significant investment in VRU and CTI technology at the hub.

Which architecture you choose for an application has direct implications for your choice of user adjustments and polling patterns.

User adjustment considerations and recommendations

User adjustments in consider split and consider skill steps may be set at the user's discretion. In distributed multi-site applications, however, adjustments must be carefully considered because of their potential affect on costs and inter-switch trunk capacity. In centralized applications all calls are redirected anyway so it's OK to use adjustments of 0. In distributed applications, though, a user adjustment of 0 for a “consider location” step is almost never practical or efficient.

In distributed applications, the smaller the adjustment the closer the load balance across the network, but the greater the percentage of calls redirected between switches (and thus the greater the demands on inter-switch trunking). Higher adjustments reduce interflows, but at the cost of allowing greater imbalance in the load between switches. It will take some time and effort to find the best combination of user adjustments in any particular network, but Table 51: Recommended initial user adjustments on page 534 contains recommended ranges for initial user adjustments under different conditions. Adjustments may vary between different call center applications so apply these guidelines for each of your applications separately.

Table 51: Recommended initial user adjustments

Recommended adjustments…	If the following criteria apply…
10-15	• You want to balance wait times across the network as much as possible. • Trunk facilities between switches are plentiful. • Each switch receives more than 1 call every 10-15 seconds (more than 240-360 calls/hour) for this application.
20	• Balancing wait times across the network is important to you. • Adequate trunk facilities are available to support the desired balance. • Each switch receives more than 1 call every 20 seconds (more than180 calls/hour) for this application.
30 or higher	• Gains in agent efficiency are more important to you than balancing wait times across the network. • Trunk facilities are scarce. • Call interflow is costly. • Each switch receives no more than 1 call every 30 seconds (around 120 calls/hour or lower) for this application.

In your first multi-site application, it is recommended that you begin with a remote adjustment of 30. This can easily be reduced later if inter-switch trunking is under-utilized. On the other hand, if trunk exhaustion is a common occurrence then user adjustments are probably set too low. Care should be taken not to lower remote user adjustments to such an extent that all trunk resources are regularly exhausted. When trunks are exhausted, no further load balancing can take place and the overall balance may deteriorate.

User adjustments should also be set high enough that calls are not interflowed to gain the equivalent of a fraction of a queue position. The following equation will give you the minimum recommended user adjustment for each remote switch:

$$\frac{\mathrm{AverageCallHandlingTime}}{\mathrm{NumberOfFullTimeEquivalentAgents}} \leq \mathrm{UserAdjustment}$$

Adjustments for remote locations will probably be in the range of 10–30 in most distributed applications.

Related topics:

User adjustments and the balance in wait times on page 535

User adjustments and the balance in wait times

Changing conditions can produce significant variations between user adjustments and the balance in wait times across a network, but on average you can predict the balance in wait times for a given user adjustment.

Let's say a user adjustment of 20 is chosen for all remote resources in a network and all the remote sites are polled. When waiting times are short (< 100 secs), the highest and lowest EWTs for this application on the network should stay within a range of approximately 20 seconds (30-50 seconds, for example). When waiting times are long (> 100 secs), the highest and lowest EWTs for the application should stay within a range of approximately 20% (5 to 6 minutes, for example).

Status polling

About status polling

Status polls are the key element in multi-site BSR applications. Status polls provides the communication links between a switch that wants to interflow a call and the switches that might service that call.

The vectors you write in multi-site applications must balance the costs of time and trunk usage with the benefit of better customer service. BSR is designed to help you achieve this balance, incorporating mechanisms to maximize improvements in customer service while minimizing inter-switch communications with its attendant delays and trunk usage. This section explains those mechanisms and the benefits they provide as you write vectors.

How long do status polls take?

One consider location step polls one remote location. Does this mean that an optimal multi-site BSR application polls every switch in a network? No.

Let's look at an example of a moderately large network, containing 16 switches. The primary vector on switch #1 could be written as shown in the following vector example. Polling response times are variable. Let's assume that this is a slow response network and that each status poll takes 1 second. The consider series in this vector could add as much as 15 seconds to a call's time in vector processing! In fact, the vector shown below is provided as an example of what NOT to do. The benefits of BSR can be obtained much more efficiently.

Intelligent polling for multi-switch networks

```
1. wait time 0 secs hearing ringback
2. consider skill    1 pri m   adjust-by  0
3. consider skill     2 pri m  adjust-by 20
4. goto step 20 if expected-wait for best = 0
5. consider location 1     adjust-by 30
6. consider location 2     adjust-by 30
7. consider location 3     adjust-by 30
8. consider location 4     adjust-by 30
9. consider location 5     adjust-by 30
10. consider location 6     adjust-by 30
11. consider location 7     adjust-by 30
12. consider location 8     adjust-by 30
13. consider location 9     adjust-by 30
14. consider location 10     adjust-by 30
15. consider location 11     adjust-by 30
16. consider location 12     adjust-by 30
17. consider location 13     adjust-by 30
18. consider location 14     adjust-by 30
19. consider location 15     adjust-by 30
20. queue-to best
21. announcement 1001
22. wait time 60 secs hearing music
23. goto step 21 if unconditionally
```

First, even in very large networks you can obtain nearly all of the possible benefits in agent utilization with very few polling connections. In a network of 16 switches, 99% of the total benefits possible with BSR can be obtained if each switch polls just 4 others. For more information, see How many switches should one switch poll? on page 538.

Now our vector looks like the following. Is polling time now cut from 15 seconds to 4 seconds, proportional to the reduction in consider steps?

```
1. wait time 0 secs hearing ringback
2. consider skill    1 pri m   adjust-by  0
3. consider skill     2 pri m  adjust-by  0
4. goto step 9 if expected-wait for call = 0
5. consider location 5     adjust-by 30
6. consider location 10     adjust-by 30
7. consider location 13     adjust-by 30
8. consider location 15     adjust-by 30
9. queue-to best
10. announcement 1001
11. wait time 60 secs hearing music
12. goto step 10 if unconditionally
```

In fact, polling time in this vector may be around 0.4 seconds per call because of mechanisms in BSR that constantly react to network conditions and resource usage to minimize the number of status polls. These mechanisms, whose combined operation is called intelligent polling, also function to make each status poll as productive as possible.

Intelligent polling

A BSR application will only poll the switches that are likely to provide the best service at any given time. If a remote switch is polled and returns an adjusted EWT greater than that of the current best resource, polling of the remote switch will be suppressed for a period of time proportional to the difference between the two adjusted EWT values. (In other words, polling of a given location is suppressed whenever the adjusted EWT returned by that location is subsequently replaced by a better adjusted EWT from another resource.) The consider step for this location will be skipped during this period and vector processing will continue at the next step. When the suppression period is over, the consider step will once again poll this location. If the location returns the best adjusted EWT, the next call processed by the vector will also cause this location to be polled. If it is not the best, polling will again be temporarily suppressed, and so on.

If no calls are in queue at the remote location an agent might become available at any moment, and thus BSR will never suppress polling for longer than 5 seconds in such situations. BSR will never suppress polling of any remote location for more than 60 seconds, regardless of the differences between adjusted EWT returned by different switches.

Other conditions can also suppress status polls to a location:

- resource exhaustion (no trunks available, queue full)
- administration errors (badly written vectors, or no application plan)

This feature significantly reduces the average number of status polls placed per call. The greater the call volume, the greater the percentage reduction. Let's take another look at the vector in Screen 2.

Let's assume that the network is operating in a balanced state. EWTs are 30 seconds at all locations, and a call arrives every 3 seconds at each site. Adjusted EWTs are 30 seconds at the origin switch and 60 seconds for each remote switch. After each status poll under these conditions, polling will be suppressed for 30 seconds. Each remote location is polled therefore, by every 10th call. On average, this means that each call polls any one location 0.1 times. Since there are four consider steps, each call makes 0.4 polls. Remembering the 1-second polling response time given at the beginning of the example, the average time added to call processing for each call is 0.4 seconds.

The 1st-found available agent strategy, discussed in Best Service Routing (BSR), can cut average polling times further. With the 1st-found strategy, BSR will skip all subsequent consider steps in a series if a resource with an available agent is found and deliver the call to that resource.

Efficient polling patterns in large networks

Unless you have a small network, you won't benefit by having every switch poll every other switch. This section explains how many remote locations each switch needs to poll, and it provides guidelines for selecting which locations any given switch should poll.

Related topics:

How many switches should one switch poll? on page 538
Which remote switches should each switch poll? on page 539
Minimizing variations in wait time on page 541

How many switches should one switch poll?

It's not necessary to poll every switch in larger networks. Because of BSR's intelligent polling capabilities, you can obtain 99% of the possible benefits in agent utilization with very few polling connections.

For an example, let's look at a laboratory network of 16 switches that is used for simulations of BSR multi-site applications. As shown in the following table, approximately 99% of the possible benefits were obtained when any one switch polled 4 others.

Table 52: Effectiveness of status polls in a 16-switch network

Number of remote sites polled by each switch	ASA across the network (seconds)	Approximate percentage of total benefits obtained
0	192.8	0%
1	26.2	89%
2	10.6	95%
3	7.6	98%
4	6.5	99%
15	4.7	100%

For each switch to poll the other 11 switches in the network would only produce an additional 1% gain in ASA and agent utilization—an improvement which would be more than offset by the cost of additional messaging and trunking.

In most situations, you'll obtain the optimal results with your multi-site BSR applications if you follow the polling guidelines shown in the following table.

Table 53: Recommended number of locations to poll

If there are this many switches in the network…	Each switch should poll…
2-4	all the other switches
5-10	3 other switches

If there are this many switches in the network…	Each switch should poll…
11-20	4 other switches
21-40	5 other switches
41 or more	6 other switches

Which remote switches should each switch poll?

In networks with fewer than 5 switches, each switch can productively poll all the other switches in the network. In larger networks, each switch need not poll every other switch. But which switches should each switch poll? We'll use the term polling patterns to describe the relationships between switches in multi-site BSR applications.

Here are two patterns to avoid. They're simple and seem intuitively obvious, but they don't usually yield the best possible results:

- Mutual polling: As much as possible, 2 switches shouldn't poll each other. This is unavoidable in small networks, but in large networks it can and should be minimized.
- Polling chains: For example, if switch A polls B & C, B polls C & D, and so on, this is a polling chain.

You may want to experiment with polling patterns appropriate to your own network and applications (if you're not constrained by the physical structure of your network). The following table provides a template for creating polling patterns for applications of up to 12 switches. In the majority of situations, these patterns will produce results that are close to optimal. To use this table, first assign a number from 1 to x to each switch in your application. Next, find the column that matches the number of switches in your application. As you read down that column, you'll see which switches each particular switch in the application should poll.

Table 54: Polling patterns for networks of 5-12 switches

This switch …	Should poll the specific switches shown in the column for your network size							
	5	6	7	8	9	10	11	12
1	2,4,5	2,4,5	2,4,6	2,4,7	2,4,6	2,4,7	2,4,8,10	2,4,8,9
2	3,5,1	3,5,6	3,5,7	3,5,8	3,5,7	3,5,8	3,5,9,11	3,5,9,10
3	4,1,2	4,6,1	4,6,1	4,6,1	4,6,8	4,6,9	4,6,10,1	4,6,10,11
4	5,2,3	5,1,2	5,7,2	5,7,2	5,7,9	5,7,10	5,7,11,2	5,7,11,12
5	1,3,4	6,2,3	6,1,3	6,8,3	6,8,1	6,8,1	6,8,1,3	6,8,12,1
6		1,3,4	7,2,4	7,1,4	7,9,2	7,9,2	7,9,2,4	7,9,1,2
7			1,3,5	8,2,5	8,1,3	8,10,3	8,10,3,5	8,10,2,3
8				1,3,6	9,2,4	9,1,4	9,11,4,6	9,11,3,4
9					1,3,5	10,2,5	10,1,5,7	10,12,4,5

This switch …	Should poll the specific switches shown in the column for your network size							
	5	6	7	8	9	10	11	12
10						1,3,6	11,2,6,8	11,1,5,6
11							1,3,7,9	12,2,6,7
12								1,3,7,8

In applications of more than 12 switches, the following table provides the formulae you need to figure out the optimal polling pattern.

Table 55: Polling pattern formula for large networks

Number of switches in application	Switch i should poll…
13 or 16	i + 1, i + 3, i + 7, i +11
14 or 19	i + 1, i + 3, i + 7, i + 9
15	i + 1, i + 3, i + 7, i + 10
17 or 20	i + 1, i + 3, i + 7, i + 12
18	i + 1, i + 3, i + 7, i + 13
21-23	i + 1, i + 3, i + 7, i + 15, i + 17
24	i + 1, i + 3, i + 7, i + 15, i + 19
25	i + 1, i + 3, i + 7, i + 15, i + 20

To use one of these formulae, first assign a number from 1 to x to each switch in your application. Then, in the left-hand column of the table, find the number of switches in your application. The corresponding formula in the right-hand column is the one you should use.

In the formulae, i is the number of the switch for which you're calculating a polling pattern. For example, let's say you want to calculate the polling patterns in an application with 16 switches. The formula to use is

i + 1, i + 3, i + 7, i +11

as shown in the first row of the table. Here are the actual results of this formulae for the first 5 switches in this 16-switch application.Notice that the numbers wrap (start over at 1) after you've polled the last switch in the network: switch 5 polls switch 16 as its fourth poll, and then the polling pattern for switch 6 has switch 1 in the fourth position.

Switch number…	Should poll switches…
1	2 ,4, 8, 12
2	3, 5, 9, 13
3	4, 6 ,10, 14

Switch number…	Should poll switches…
4	5, 7, 11, 15
5	6, 8, 12, 16
6	7, 9, 13, 1
7	8, 10, 14, 2

Minimizing variations in wait time

When a network contains (or when a call center application combines) large resources and very small resources, BSR and LAI can be effectively combined. This section presents two sample vectors. The first example shows a primary vector intended for the smaller resources in a network when you want to avoid having a call in queue at one call center while an agent is available at another. This design will reduce wait time variation as well. The second example illustrates a primary vector for larger locations: this example shows you the best way to minimize wait times across a network.

Related topics:

[LAI as a backup](#) on page 541
[Single-queue FIFO hybrid configuration](#) on page 542

LAI as a backup

As noted above, if your principal concern is that a call not wait in queue while an agent is available elsewhere, use BSR at all locations in the network. At smaller locations, write primary vectors that will perform rapid LAI attempts to other (preferably larger) resources once the call has been queued.

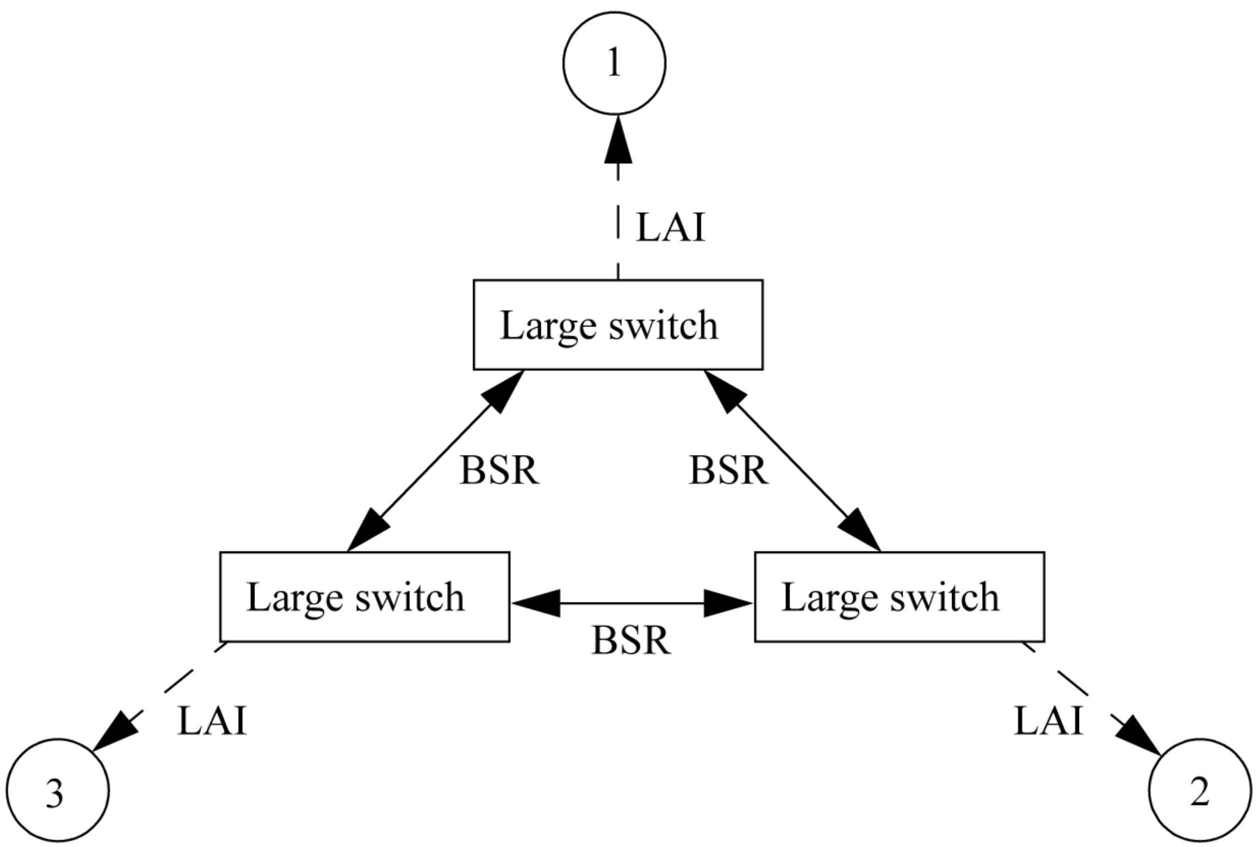
```
1.  wait time 0  secs hearing ringback
2.  consider skill 1st pri m adjust-by  0
3.  consider location 12     adjust-by 30
4.  consider location 22     adjust-by 30
5.  goto step 7 if expected-wait for call < 600
6.  disconnect after announcement 3501 "
Due to heavy call volume..."
7.  queue-to skill best
8.  announcement 3500 "
Thanks for calling...."
9.  goto step 13 if expected-wait for call < 90
10. wait time 45 secs hearing music
11. announcement 3502 "
Still busy..."
12. goto step 9 if unconditionally
13. route-to-number 913031234567 with cov n if interflow-qpos = 1
14. wait time 5 secs hearing music
15. goto step 13 if unconditionally
```

Steps 1 to 4 comprise a typical BSR vector. The origin switch considers a local resource and 2 remote resources. Before queuing or routing the call, however, the vector checks the expected wait time for the best resource. If this is 10 minutes or more, the caller receives a busy announcement. Otherwise, the queue-to best step sends the call to the best resource. Two vector loops follow: one 45-second loop with music and a delay announcement, and one 5-second loop that uses LAI. If the call is queued successfully in step 7 the first announcement loop (steps 9-12) executes until the call gets within a certain range of the head of the queue (at which point EWT is less than 90 seconds). At this time, step 9 sends the call to the second loop, where LAI attempts are placed every 5 seconds for the call at the head of the interflow eligible queue (**`interflow-qpos=1`**). If an agent becomes available at the larger remote resource, any call at the head of the eligible queue at the smaller location is outflowed to the larger resource, normally within a period of 5 seconds.

Single-queue FIFO hybrid configuration

To minimize variations in wait time across a network, the best strategy may be to let only the call centers with the larger resources receive calls. The following figure shows a network of 3 large and 3 small resources (call centers with large splits/skills and call centers very small splits/skills in the same application).

The large locations use BSR and all poll each other, while each location with a small resource (numbered 1, 2, 3) is treated as a satellite of one of the larger locations and only receives calls interflowed from that location. (Mutual polling is not optimal in larger networks, but it's OK for switches in such a small network to poll each other.) So BSR is used to balance the load between the locations with the larger resources. Then, each large switch executes a rapid LAI vector loop to one small switch to look for available agents. Since calls never queue at the small switches, the problem of highly variable wait times at the small resources is eliminated. This strategy will also give the best balance in wait times across resources.



The following vector example shows the primary vector that would be used at the large locations with this strategy. This vector is almost identical to the vector shown in LAI as a backup on page 541 above. The differences are at the application level. In contrast to the previous example:

- Only the locations with the larger resources receive calls.
- The primary vector shown here resides on the larger switches.

Steps 1 to 4 comprise a typical BSR vector. The origin switch considers a local resource and 2 remote resources. Before queuing or routing the call, however, the vector checks the expected wait time for the best resource. If this is 10 minutes or more, the caller receives a busy announcement. Otherwise, the queue-to best step sends the call to the best resource. Two vector loops follow: one 45-second loop with music and a delay announcement, and one 5-second loop that uses LAI. If the call is queued successfully in step 7, the first announcement loop (steps 9-12) executes until the call gets within a certain range of the head of the queue. At this time, step 9 sends the call to the second loop, where LAI attempts are placed every 5 seconds (only for the call at the head of the interflow eligible queue). If an agent becomes available at the smaller resource, any call at the head of the eligible queue at the larger location is outflowed to the smaller resource, normally within a period of 5 seconds.

Vector combining BSR and LAI

```
1.  wait time 0  secs hearing ringback
2.  consider skill 1st pri m adjust-by  0
3.  consider location 120    adjust-by 30
4.  consider location 220    adjust-by 30
5.  goto step 7 if expected-wait for best < 600
6.  disconnect after announcement 3501 "
Due to heavy call volume..."
7.  queue-to skill best
8.  announcement 3500 "
Thanks for calling...."
9.  goto step 13 if expected-wait for call < 90
10. wait time 45 secs hearing music
11. announcement 3502 "
Still busy..."
12. goto step 9 if unconditionally
13. route-to-number 913031234567 with cov n if interflow-qpos = 1
14. wait time 5 secs hearing music
15. goto step 13 if unconditionally
```

Similar vector loops can be added to the interflow vectors at each of the large switches. In other words, each vector that processes calls at the larger locations can use rapid LAI loops to interflow calls to its satellite resource. This system maximizes agent utilization and the distribution of call load while evening out wait times across the network.

Considerations for low volume splits/skills

About low volume splits/skills

Very small resources (for example, 2-3 agents) have special needs. With BSR, it is easy to obtain a very close balance of wait times across a network of call centers. however, for very small splits/skills, wait times for each call can vary significantly.

To see why this is, let's take an extreme example of a split with a single agent logged in with one call active and none in queue. Average call handling time is 3 minutes. Now, if a new call arrives in queue, that call could be answered almost immediately—or it might wait for 3 minutes or more. The variation in wait times is perhaps 5-180 seconds.

In general, the fewer agents logged into a split/skill, the greater the variability in wait times because agents become available less often. BSR will naturally favor large resources, steering calls away from smaller resources when there are no available agents or wait times are not the best in the application. This tendency helps reduce the possibility that an individual caller might have a disproportionately long wait at a small resource.

If your network includes very small splits/skills, you have three options:

- If your operation is not badly affected by a small percentage of calls having variable wait times, simply use BSR normally across the network.
- If your principal concern is that a call does not wait in queue while an agent is available elsewhere, use BSR normally but write primary vectors at smaller locations to perform rapid look-ahead attempts to other resources once the call has been queued. (Rapid LAI vector loops use the **`interflow-qpos`** conditional, which is an enhancement to LAI. For more information on LAI and the **`interflow-qpos`** conditional, see Look-Ahead Interflow (LAI).) For an example of this type of vector, LAI as a backup on page 541.
- If you want to answer every caller quickly, then the following configuration is recommended. Do not deliver or queue calls directly to the very small resources. Deliver or queue all incoming calls to larger resources, and use BSR to balance the load across these larger locations. Some or all of the larger locations should then perform rapid look-ahead attempts to one or more of the smaller resources. In this way, the members of the very small resource become an extension of the agent pool at one of the larger call centers. For an example of this design, see Single-queue FIFO hybrid configuration on page 542.

In any network, avoid having several large resources poll or make look-ahead attempts to a very small resource. Since the status at the very small resource changes infrequently, frequent polls to that resource are wasteful. A very small resource should receive look-ahead attempts or be polled only by other small resources or by one large resource.

Chapter 8: Call Flow and specifications for converse-on command

Call flow and specifications for converse - VRI calls

This section details call flow for calls involving a **`converse-on`** vector step and Voice Response Integration (VRI). This call flow is segmented into the following phases:

- Converse call placement
- Data passing
- VRU data collection
- Script execution
- Data return
- Script completion
- Switch data collection

Note:

If, during any phase of this call flow, a **`converse-on`** step is executed while the caller is in the split queue and an agent becomes available to service the caller, the VRU port is dropped, vector processing is terminated, and the calling party is immediately connected to the available agent.

Related topics:

Converse call placement on page 545
Data passing on page 547
VRU data collection on page 551
Script execution on page 552
Data return on page 552
Script completion on page 554
Switch data collection on page 555

Converse call placement

The first action taken by the converse-on step is to deliver the call to the converse split. Ringback tone is not heard by the caller. Any audible feedback supplied by vector processing remains until the VRU answers the call and all digits (if administered) have been outpulsed to the VRU. Vector processing is suspended. Callers remain in any non converse split queues, and they retain their position in queue while the converse session is active.

If a Call Prompting TTR is allocated to the call, the TTR is released. Any dial-ahead digits are discarded. However, any digits collected prior to the converse-on step are kept.

Calls to busy converse splits are allowed to queue. The priority of the call in queue is administrable within the converse-on step. Again, any audible feedback supplied by vector processing continues until the call is answered by the VRU and any data is outpulsed. Calls to busy converse splits have either no queue or a full queue fail. For this scenario, a vector event is logged, and vector processing continues at the next vector step.

Whenever a converse-on step places a call to an auto-available split whose agents are all logged out, the call is not queued. Instead, the converse-on step fails, a vector event is logged, and vector processing continues at the next vector step.

Note:

Usually, this scenario occurs whenever the Voice Response Unit (VRU) goes down, the ports are members of an Auto-Available Split (AAS) and the Redirection on No Answer (RONA) feature has taken all the ports out of service.

The originator’s display is not changed by the terminating or answering of a converse call. Also, whenever a call is delivered to a display station using a converse-on step, the station displays the following information: Originator Name to VDN Name. Conventional Call Vectoring rules for Override are in effect.

Valid destinations for converse calls must be vector-controlled and include the following:

- Hunt groups
- ACD (including Auto-Available) splits
- Agent (including Auto-Available) skill groups
- AUDIX hunt groups

Note:

Even though AUDIX hunt groups are valid destinations for converse calls, they do not need to be vector-controlled.

Undefined and non vector-controlled hunt group, split or skill numbers are rejected at administration time.

Any attempt to remove a hunt group, split or skill administered within a converse-on vector step is denied until the vector has been changed. Also, any attempt to make a hunt group, split, or skill non vector-controlled is denied if the hunt group, split, or skill is called by a converse-on step.

Data passing

The data passing phase is optional and is in effect only if the application calls for the switch to pass information in-band to the VRU.

The converse-on step may outpulse up to two groups of digits to the VRU. The digits can serve two major purposes:

- Notify the VRU of the application to be executed
- Share call-related data collected by the switch. This includes ANI, CINFO, or caller digits.

In many applications both application selection and data sharing are required.

Related topics:

Using the pound sign on page 547
How the outpulse sequence works on page 547
Values administered for <data_1> and <data_2> on page 548
Time delay administration on page 549
When the VRU hangs up during data passing on page 550
Ensuring robust operation of VRU data passing on page 550

Using the pound sign

Since in many cases the digit strings are of variable length, the switch always appends a pound sign (#) to the end of each digit string. Prompt and collect steps in the VRU script must always be administered to expect the pound sign as the end-of-string symbol and to include the pound sign in the digit count.

Sending the pound sign prevents excessive delays and other problems caused by digit timeouts.

How the outpulse sequence works

The complete outpulse sequence is summarized as follows:

1. The VRU answers the call.
2. Delay for the time administered in the Converse first data delay field in the System Parameters-Features screen occurs.
3. The <data_1> is outpulsed.
4. The pound sign is outpulsed.

5. Delay for the time administered in the Converse second data delay field in the System Parameters-Features screen occurs.
6. The <data_2> is outpulsed.
7. The pound sign is outpulsed.

Note:
The length of DTMF tones and the inter-digit pause between tones is administrable on the Feature-Related System Parameters screen. The optimum settings for Conversant/IR are 60 msec tones and 60 msec pauses that provide an 8.3 digits-per-second rate. These changes differ from the administration default.

Any audible feedback supplied by the switch is disconnected only after the outpulse sequence is completed. Also, any touch-tone dialing by the calling party during the data passing phase does not result in data corruption.

Values administered for <data_1> and <data_2>

You can administer the following values for <data_1> and <data_2> within the converse-on command:

Value	Description
Administered digit string	This string can contain up to six characters consisting of one or more digits (0 through 9) or asterisks (*). The pound sign (#) may not be included in a digit string because it is reserved as the end-of-string character. However, you can administer a single pound sign.
ani	If the call is a local call or an incoming DCS call, this data type causes the extension of the calling party to be outpulsed. If the call is an incoming ISDN PRI call with ANI (BN) provided to the switch, the calling party number/billing number (CPN/BN) of the calling party is outpulsed to the voice information system. If there is no ANI (BN) to send, the end-of-string pound sign is the only character outpulsed. Any other type of incoming call results in the pound sign being outpulsed.
vdn	This data type causes the VDN extension to be outpulsed. In cases where multiple VDNs are accessed, normal VDN override rules determine which VDN extension is outpulsed.
digits	This data type can be used only if Call Prompting is optioned, and it causes the most recent set of digits collected in vector processing to be outpulsed. If no digits are available, the end-of-string pound sign is the only character outpulsed.
qpos	This data type causes the value of the queue position of a call in a non converse split to be outpulsed. This value is a variable length data item from which between one and three digits can be

Value	Description
	outpulsed. If the call is not queued, the end-of-string pound sign is the only character outpulsed. **Note:** The use of this keyword is not recommended with multiple split queuing because any queue position value sent may not be meaningful. However, if the call is queued to multiple non converse splits, the value of the caller's queue position in the first non converse split is sent. This data may be used by the voice information system to inform callers of their position in queue or to decide whether to execute a long or short version of a voice response script.
wait	This data type sends the expected wait time for a call in vector processing that is queued to at least one split. It is a value from 0 to 9999 seconds (variable length that is not padded with zeros) always followed by a pound sign. If the call is not queued, or is queued only to splits with no working agents, only the pound sign is outpulsed.
A to Z and AA to ZZ	This data type causes the current numeric value of the vector variable to be outpulsed. If the value is undefined, a single # is outpulsed. The vector variable is defined by letters between A to Z and AA to ZZ.
V1 to V9	This data type causes the value of the VDN variable assigned to the active VDN for the call to be outpulsed. If the value is undefined, a single # is outpulsed. The VDN variables V1 through V9 are defined on the VDN screen for each VDN extension.
#	This is the only character outpulsed. Outpulsing this character causes the corresponding prompt and collect command in the voice response script to be skipped.
none	This data type causes no characters to be outpulsed. Also, no end-of-string pound sign is outpulsed, and no time delays are invoked. If <data_1> is administered as none, <data_2> must also be none. The switch always outpulses a pound sign at the end of each digit string. Where the pound sign is administered, or where the digits keyword is administered and the last digit collected from the caller is the pound sign, only one pound sign is outpulsed. No pound sign is outpulsed when the keyword none is administered.

Time delay administration

Any data passed to the VRU from the switch is outpulsed in-band. Customers can administer the converse first data delay and the converse second data delay time delays on the System Parameter-Features screen. These delays may range from 0 through 9 seconds, with a default of zero seconds for the converse first data delay and a default of two seconds for the converse second data delay. The delays may be needed to give the VRU time to invoke an application and allocate a touch-tone receiver to receive the passed digits.

- If <data_1> is not none, the converse first data delay timer starts when the call is answered by the VRU. Once the timer expires, the data_1 digits are outpulsed in-band to the VRU, followed by the end-of-string pound sign (#).
- If <data_2> is not none, the converse second data delay timer starts when the end-of-string pound sign from the first digit string is outpulsed. Once the timer expires, the data_2 digits are outpulsed in-band to the VRU, followed by the end-of-string pound sign.

No time delays are invoked when the keyword none is administered.

Note:
The outpulsing of digits is not heard by the caller.

When the VRU hangs up during data passing

If the VRU hangs up during the data passing phase, the switch will log a vector event, reactivate vector processing at the next vector step, and ensure the VRU port is accessible for future calls.

Once all digits have been passed to the VRU, any audible feedback is disconnected.

Note:
At this point, control has effectively been passed to the VRU.

Ensuring robust operation of VRU data passing

To ensure the robust operation of the VRU data passing operation, be sure to implement the following recommendations:

- Include the prompt and collect command in the VRU script for each data field passed in the converse-on step.
- Administer each prompt and collect to recognize the pound sign (#) as the end-of-string character.
- Ensure the number of digits expected is one greater than the number of digits passed to allow for the pound sign, which terminates every converse data field.

  Also, ensure no announcement is played in these prompt and collect steps.
- Ensure the first digit timeout in the prompt and collect steps is five seconds greater than the corresponding converse data delay. For example, if the converse-on step passes two data fields, and if the converse first data delay is 0 secs and the converse second data

delay is 4 secs, the first digit timeouts for the two prompt and collect commands should be at least 5 and 9 seconds, respectively.

- Ensure the inter-digit timeout in the prompt and collect steps is at least five seconds.
- Administer the converse first data delay to give a VRU under a heavy load sufficient time to allocate a DTMF touch-tone receiver after answering the call.
- Administer the converse second data delay to give a VRU under a heavy load sufficient time to complete any tasks between the first and second prompt and collect command. For example, the VRU can invoke a new application if the first data field passed is used to identify the application script to be executed.
- In general, for converse-on steps to pass data to the VRU, ensure the VRU script does not execute any commands between the time the call is answered and the time when the first prompt and collect command is executed.

VRU data collection

When digits are passed from the switch to the VRU, the first VRU script commands executed are answer phone and prompt and collect. No announcement is programmed for the prompt and collect command, and the pound sign (#) is programmed as the end-of-string sign. If two sets of digits (that is, <data_1> and <data_2>) are passed by the switch, there will be two prompt and collect commands on the VRU to receive them.

If the first digit string (<data_1>) passed to the VRU is for application selection, the Avaya Interactive Response Script Builder exec command invokes the appropriate script. If a second digit string (<data_2>) is also used to pass an argument to this selected application, the first command in the executed script is a prompt and collect command with no announcement prompt programmed and with the pound sign (#) programmed as the end-of-string character.

The “Converse second data delay” is used to give the VRU time to invoke the selected application before the <data_2> digit string is outpulsed.

The application developer should ensure the administered converse first data delay and converse second data delay timers allow sufficient time for the VRU to successfully collect all outpulsed digits, even during periods of heavy call volume. Loss of digits from <data_2> is an indication the converse second data delay timer needs to be increased or the VRU timing values may need tuning as appropriate to resolve issues.

Related topics:

Default and IVR converse settings on page 552

Default and IVR converse settings

The default for the converse signaling tone and pause on the Feature-related System Parameters screen is a 100 msec tone and 70 msec pause. This results in a 5.5 digits per second rate that provides a more conservative sending rate to support most VRUs.

For operation with Avaya Conversant or Avaya IR, change the default settings to a 60 msec tone and a 60 msec pause. This results in a more optimum rate of 8.3 digits per second supported by these products.

Script execution

During script execution, digits input by the calling party in response to prompt and collect commands are collected by the VRU but are not collected by the switch as dial-ahead digits. Also, audible feedback is determined by the VRU.

If an agent from a non converse split becomes available to service the call while the VRU script is being executed, the VRU port is dropped from the call, and the caller is immediately connected to the agent. Any digits collected prior to executing the converse-on step are still available and may be displayed using the CALLR-INFO button.

The entire call is dropped if the caller abandons during the execution of a converse-on step.

Data return

This phase is optional and is in effect only if the application calls for the VRU to return information to the switch before returning control to vector processing.

Digits returned by the VRU are treated as dial-ahead digits. The rules for collecting and processing VRU-returned digits are identical to those for collecting and processing Call Prompting digits (see Call Prompting).

VRU data return is done in a manner similar to an analog transfer. Specifically, the VRU does an analog switchhook flash, outpulses DTMF digits, and then hangs up. If converse data is returned, the DTMF digits comprise two parts. The first sequence of digits is the converse data return feature access code administered on the Feature-Access-Codes screen. The second sequence of digits is the sequence to be passed by the VRU. These digits are collected later during vector processing.

The Avaya Interactive Response VRU offers a built-in external function called converse_data. This function allows applications developers to perform this operation in a convenient and robust fashion.

To ensure the robust operation of the VRU data return operation, be sure to follow these recommendations:

- Set the analog flash timing to 600 msecs.
- Ensure DTMF tones last at least 70 msec and interdigit pauses last at least 50 msec. This results in an outpulsing rate up to 8.33 digits per second.
- (Avaya Interactive Response only) Use the converse_data external function to return data to the switch.
- Hang up line to switch after outpulsing digits. Assume that switch will wait between 1.2 and 1.5 secs to determine that the hang-up is a disconnect.

For applications involving VRUs other than Avaya Interactive Response VRUs, be sure to follow these recommendations:

- After the flash, ensure the VRU performs dial tone detection (stutter dial tone) for a sufficient period of time to ensure accurate detection (typically 0.6 to 1.0 secs) before outpulsing the converse data return feature access code.
- If no dial tone is received before the timeout, ensure the VRU does two more retries of the analog flash. Also, if no dial tone is detected after two retries, ensure the VRU logs an error.
- Whenever dial tone is detected, ensure the digits of the converse data return feature access code are outpulsed.
- After the converse data return feature access code is outpulsed, the returned digits can be outpulsed without waiting for the second dial tone.
- After the VRU digits are outpulsed, the line to the switch is dropped.

Assuming an outpulse rate of 8 digits per sec (0.125 secs per digit), a 3-digit feature access code and stutter dial tone detection time of 0.6 secs, the maximum of 24 digits passed to switch should take about 6 secs (1.2 secs disconnect plus 8 secs plus 0.125 secs per digit).

The Call Classifiers required by the Call Prompting feature are not required for returning digits in-band from the VRU to the switch. Instead, general purpose TTR boards are used. As long as dial-ahead digits are available, any collect digits steps following a converse-on step do not require a Call Classifier to be allocated to the call.

If no general purpose TTRs are immediately available, and if the call queues for a TTR, no dial tone is provided. For this scenario, the VRU does not outpulse any digits until a TTR is available and dial tone is provided.

If there are no general purpose TTRs available on the switch, and if there is no space in the TTR queue, the operation fails. Usually, the VRU logs an error and then quits, and vector processing continues at the next vector step. Existing system measurements reports indicate when the system is configured with an insufficient number of TTRs.

The Converse Data Return Code can be followed by a maximum of 24 digits. The VRU touch-tones the code and the digits in-band. However, the code and the digits are not heard by the caller. The digits are stored in the switch as Call Prompting dial-ahead digits. If x digits are collected by vector processing before the converse-on step is executed, the maximum number of digits that can be returned is reduced to 24-x. Any additional digits returned by the VRU are discarded. The data return is completed once the VRU hangs up.

The digit string returned by the VRU can consist of the digits (0 through 9) and pound signs (#). The pound sign (#) is interpreted by the collect digits step as an end-of-string character. If the digit string being returned is of variable length, the VRU can terminate the string with a pound sign (#) to avoid the ten second timeout delay that occurs when the digits are collected. If the digit string being returned is multi-part (that is, to be collected by multiple collect digits steps), and if some of the parts are of variable length, the pound sign (#) can be used to terminate each of the variable length parts.

Note:

An asterisk (*) may be included as part of the converse data return code. However, since the asterisk is interpreted as a delete character by the switch, it makes little sense to use it as a returned digit. If it is used as such, all characters returned prior to the asterisk are discarded.

During the data return phase, the caller is temporarily put on hold. Music-on-hold, if administered, is suppressed. Since the caller hears silence during this phase, feedback should be provided to the caller as soon as possible after the converse-on step is executed.

Any touch-tone digits dialed by the calling party during the data return phase are discarded. These digits do not cause data corruption, and they are not collected as dial-ahead digits by the switch.

If an interdigit timeout occurs during the data return phase, the switch logs a vector event, keeps the digits already returned, drops the VRU, and reactivates vector processing at the next vector step.

If the timeout occurs before the converse data return code is returned, the operation is the same except that no discarded digits will be available.

Script completion

The VRU script returns control to vector processing on the switch by simply hanging up the line. In cases where no data is returned to the switch, this is done usually by executing the `quit` command. In cases where data is returned, this occurs whenever the VRU hangs up on completion of the VRU data return operation.

The last set of digits collected before the converse-on split step is executed is still available and may be displayed by an answering agent on the non converse split by using the CALLR-INFO button.

A VRU script can be programmed to continue running after hanging up the voice line. This after-call work is usually very short, and it may involve either a final message to a host or a final update to a local database. For this scenario, the VRU port (channel) is still associated with the running script even though there is no longer a voice connection.

From the switch point of view, the agent (port) is available for the next call. If a call is delivered to this port, the VRU does not answer the call until the previous script has completed. As long as the VRU script's after call work is short in duration, this poses no significant problem for the VRI feature. However, high volume VRI applications with lengthy after call work periods should be avoided, especially if such periods are so lengthy they approach the administered timeout period on the switch for the Redirection on No Answer (RONA) feature. In such a case, RONA might think the VRU ports are faulty and might therefore start to take these ports out of service.

Switch data collection

This phase is in effect only if the VRU returns information to the switch.

Once the VRU script has completed and vector processing is reactivated, the returned digits are collected and processed by vector commands in the usual manner. Since the digits must be collected by a **`collect digits`** command, data may be returned and processed only if the Call Prompting option is enabled.

The data returned can consist of multiple parts. For example, the VRU could return a stream of seven digits in which a single digit success/fail code is followed by a six-digit account code. For this scenario, the converse-on step would be followed by a sequence of vector steps including two collect digits steps. The first collect digits step would collect one digit and then check the result code; the second collect digits step would collect the six-digit account code.

Any touch-tone digits dialed by the calling party during the data collection phase are discarded, do not cause data corruption, and are not collected as dial-ahead digits by the switch.

If VRU data is returned, the calling party is able to touch-tone a response to a switch prompt only after the data collection phase is completed and another collect digits step is executed. This is true because each executed collect digits step does not allocate a TTR when dial-ahead digits are present. Since VRU-returned digits are treated as dial-ahead digits, a TTR is attached to the call only after all returned digits are collected and another collect digits step is encountered. Only at this point can the caller hear an announcement for the **`collect digits`** command and successfully enter digits.

Chapter 9: Security issues

Security issues

Call Vectoring can be integrated into the security of your switch. For example, Call Vectoring and Call Prompting can be used to help prevent unauthorized users from gaining access to the switch using the Remote Access feature. This section explains how this is done.

Related topics:

Remote access on page 557
EAS on page 559
Limiting outside access using VDN COR restrictions on page 559
Vector-initiated service observing on page 560
Voice response integration on page 560
Attendant Vectoring security issues on page 561
Remote logout of agent on page 561

Remote access

Abuse of remote access on the switch is one of the main methods by which unauthorized users obtain telephone services illegally. This section explains how a number of Call Vectoring features can be used to prevent unauthorized use of the remote access feature. No new development is required for any of these services.

Two methods are available:

- Front-ending remote access
- Replacing remote access

Related topics:

Front-ending remote access on page 557
Replacing remote access on page 558

Front-ending remote access

This method gives authorized external callers a VDN extension to call instead of the remote access extension, which is kept private. The corresponding call vector can then implement a number of security checks before routing callers to the remote access extension. Routing can be done using a route-to number or route-to digits step.

The following advantages are possible using this method:

- Call Vectoring can introduce a delay before the dial-tone is provided to the caller. Immediate dial-tone is often one criterion searched for by a hacker's programs when the hacker is trying to break into a system.
- A recorded announcement declaring that the use of the switch services by unauthorized callers is illegal and that the call is subject to monitoring and/or recording can be played for the caller.
- Call Prompting can be used to prompt for a password. In such a case, the call is routed only if there is a match on the password.
- Use of the remote access extension can be limited to certain times of the day or certain days of the week.
- Real-time and historical reports on the use of the remote access feature can be accessed from CMS or from BCMS.
- Different passwords can be used on different days of the week or at different times during the day.
- Many VDNs that call the remote access extension can be identified. Accordingly, individuals or groups can be given their own VDN with unique passwords, permissions and reports. Any abuse of the system or security leak can then be attributed to an individual or a group.
- The caller can be routed to a VRU using the converse-on step where more sophisticated security checking, such as speaker recognition, can take place.
- Anyone failing any of the security checks can be routed to a security VDN that routes the caller to security personnel with a display set or to a VRU. Such a call would show security and possibly also the attempted password on the display. If the call is passed to a VRU, the VDN, the ANI and/or the prompted digits can be captured. CMS and BCMS reports on this security violation VDN will give information on how often and when security violations occur.

Replacing remote access

For this method, the remote access extension is not used. One or more VDNs are designed to access call vectors that can employ all of the security checks described in the previous section. The same reports and monitoring/recording capabilities described in the previous section can also be used. Instead of routing to the remote access extension, the vector collects digits from the caller and then routes to the given destination if there is a match on the password.

Again, multiple VDNs can be created for individuals or groups with different security checks and different permissions and/or restrictions. Destination numbers provided by callers can be screened by the vectors and denied if the user does not have permission to access that destination. For example, an individual user could be restricted to placing calls to numbers beginning with area codes 303 and 908.

EAS

With EAS, agent stations can be locked when they are not staffed. This is accomplished by assigning the station a Class of Restriction that does not allow outbound calls or it could be restricted from toll calls.

EAS agents have an optional password of up to nine digits to log in. This password is not displayed on DCP terminals when the agent is entering the password on the dial pad.

Limiting outside access using VDN COR restrictions

Routing calls through the Communication Manager with Call Vectoring can raise some security issues. A VDN has a Class Of Restriction (COR). Calls processed by the vector carry the permissions and restrictions associated with the COR of the VDN.

For example, if a vector in the switch is written to collect digits, and then to route to the digits dialed, the restrictions on what calls can be placed are determined by the COR of the latest VDN. Also, checks can be made on the digits that are dialed, using **`goto if digits`** vector commands (for example, **`goto if digits = 123`**) to disallow routing to undesired destinations. The collect digits step can also be limited to collect only the number of digits required (for example, only collecting five digits for internal dialing).

An incoming caller can access Trunk Access Codes, some Feature Access Codes, or most other sets of dialed digits. To deny incoming callers access to outgoing facility paths, the COR of the Vector Directory Number must be configured to disallow outgoing access. This should include the following: lowering the Facility Restriction Level (FRL) in the COR to the lowest acceptable value (FRL=0 provides the most restricted access to network routing preferences), assigning a Calling Party Restriction of Toll or Outward denying Facility Test Call capability, and blocking access to specific CORs assigned to outgoing Trunk Groups using the Calling Permissions section of the Class Of Restriction screen.

Review the Classes of Restriction assigned to your VDNs. If they are not restricted, consider assigning restrictions on the VDN and/or using goto tests on those digits to prevent callers from exiting the system using the vector.

Vector-initiated service observing

The following restrictions can be used with vector-initiated service observing to guard against unauthorized use.

- Call prompting commands can be used in service observing vectors to provide passcode protection, and to limit access to observing specific destinations or verified caller entered digits.
- Time of Day/Day of Week checks can be incorporated in service observing vectors.
- A vector can be created to be used exclusively for service observing.
- For a VDN to be observed as the result of a route-to command, the VDN must have a COR that allows it to be observed.
- The calling permissions of the COR assigned to the service observing VDN in conjunction with the can be observed settings of the COR assigned to the destination determine what agents, stations, or VDNS can be observed.

Voice response integration

When a converse step is used to access a VRU application that returns data for a collect digits step, the opportunity for toll fraud exists when the VRU application fails to return any data. To avoid this type of toll fraud be certain that one of the following is true:

- If the collected digits are used to route calls internally, be certain that the Class of Restriction (COR) for the Vector Directory Number (VDN) does not allow calls to route externally.
- If it is necessary to use the collected digits to route calls externally, use a password to verify that the collected digits have been passed by the VRU application. In the following vector example the VRU application returns a three-digit password followed by the eight-digit external number. The vector routes calls without the correct password to a different vector and routes calls with the correct password to the collected digits.

Voice Response Integration Security Example

```
converse-on split 10 pri m passing none and none
collect 3 digits after announcement none
goto vector 23 if digits <> 234
collect 8 digits after announcement none
route-to digits with coverage n
```

Attendant Vectoring security issues

Security Violation Notification (SVN) referral calls can be directed to an attendant group. These are priority calls and, as such, cannot terminate to a VDN. However, when these calls are sent to the attendant group, they are treated as ordinary calls - priority does not apply to attendant group processing. So, these will be treated as normal attendant group calls and will be sent through vector processing.

Remote logout of agent

See Remote access on page 557 for issues associated with accessing the switch from a remote location.

Chapter 10: Related documents

Other Call Center documents

These additional documents are issued for Avaya Call Center applications:

- *Avaya Aura™ Call Center Overview*: Provides an introduction to all the Call Center features and a high-level overview of the new features available for the most-current release.
- *Administering Avaya Aura™ Call Center Features*: Provides information on how to administer Call Center using the various screens and how to work with Time of Day Clock Synchronization, Recorded Announcements, and VRUs/IVRs as Station Ports.
- *Avaya Aura™ Call Center Feature Reference*: Provides detailed information on the various ACD and Call Vectoring features, including the relevant command and screens for each of the features.
- *Programming Call Vectors in Avaya Aura™ Call Center*: Provides information on how to write, use, and troubleshoot vectors.
- *Avaya Aura™ Communication Manager System Capacities Table*: Provides Communication Manager offer-defined capacities for various Avaya server platforms.
- *Communication Manager Call Center Software - Basic Call Management System (BCMS) Operations*: Provides information on the use of the BCMS feature for ACD reporting.
- *Avaya Business Advocate User Guide*: Provides a general understanding of how Avaya Business Advocate can be used for call and agent selection.
- *Avaya IQ Documentation DVD*: Provides information about Avaya's software-only reporting solution for its contact center portfolio.

Associated application documentation

The most recent application documentation for Communication Manager and Avaya Call Management System is available on the Avaya Support web site: http://www.avaya.com/support.

Chapter 11: Glossary

AAR	See Automatic Alternate Routing (AAR).
abandoned call	An incoming call in which the caller hangs up before the call is answered.
Abbreviated Dialing	A feature that allows callers to place calls by dialing just one or two digits.
ACA	See Automatic Circuit Assurance (ACA).
access code	A 1-, 2-, or 3-digit dial code used to activate or cancel a feature, or access an outgoing trunk.
access trunk	A trunk that connects a main communications system with a tandem communications system in an Electronic Tandem Network (ETN). An access trunk can also be used to connect a system or tandem to a serving office or service node. Also called an access tie trunk.
ACCUNET	A trademarked name for a family of digital services offered by AT&T in the United States.
ACD	See Automatic Call Distribution (ACD).
ACD agent	See agent.
ACD work mode	See work mode.
ACW	See After Call Work (ACW) mode.
adjunct	A processor that does one or more tasks for another processor and is optional in the configuration of the other processor. See also application.
Adjunct Routing	A means of evaluating calls before the calls are processed by requesting information from an adjunct. The communication server requests instructions from an associated adjunct and makes a routing decision based on agent availability or the caller information.
adjunct-controlled split	An Automatic Call Distribution (ACD) split that is administered to be controlled by another application. Agents logged into such splits must do all telephony work, ACD login/ logout, and changes of work mode through the adjunct (except for auto-available adjunct-controlled splits, whose agents may not log in/out or change work mode).

adjunct-monitored call	An adjunct-controlled call, active-notification call, or call that provides event reporting over a domain-control association.
Adjunct-Switch Application Interface (ASAI)	A recommendation for interfacing adjuncts and communications systems, based on the CCITT Q.932 specification for layer 3.
adjusted EWT	A Best Service Routing (BSR) term for Expected Wait Time (EWT) plus a user adjustment set by a `consider` command.
administration terminal	A terminal that is used to administer and maintain a system.
Administration Without Hardware (AWOH)	A feature that allows administration of ports without associated terminals or other hardware.
Advocate	See Avaya Business Advocate.
After Call Work (ACW) mode	A mode in which agents are unavailable to receive ACD calls. Agents enter the ACW mode to perform ACD-related activities such as filling out a form after an ACD call. Also see, auto-in work mode, manual-in work mode, and aux-work mode.
agent	A member of an ACD hunt group, ACD split, or skill. Depending on the ACD software, an agent can be a member of multiple splits/skills.
agent report	A report that provides historical traffic information for internally measured agents.
ANI	See Automatic Number Identification (ANI).
appearance	A software process that is associated with an extension and whose purpose is to supervise a call. An extension can have multiple appearances. Also called call appearance, line appearance, and occurrence. See also call appearance.
application	An adjunct that requests and receives ASAI services or capabilities. One or more applications can reside on a single adjunct. However, the communication server cannot distinguish among several applications residing on the same adjunct and treats the adjunct, and all resident applications, as a single application. The terms application and adjunct are used interchangeably throughout this document.
application plan	A plan used only in multi-site Best Service Routing (BSR) applications. The application plan identifies the remote switches that may be compared in a consider series. The plan also specifies the information used to contact each communication server and to interflow calls to the communication server.

applications processor A micro-computer based, program controlled computer providing application services for the switch. The processor is used with several user-controlled applications such as traffic analysis and electronic documentation.

ARS See Automatic Route Selection (ARS).

ASAI See Adjunct-Switch Application Interface (ASAI).

association A communication channel between adjunct and switch for messaging purposes. An active association is one that applies to an existing call on the switch or to an extension on the call.

attendant A person at a console who provides personalized service for incoming callers and voice-services users by performing switching and signaling operations. Also see attendant console.

attendant console The workstation used by an attendant. The attendant console allows the attendant to originate a call, answer an incoming call, transfer a call to another extension or trunk, put a call on hold, and remove a call from hold. Attendants using the console can also manage and monitor some system operations. Also called console. Also see attendant.

Audio Information Exchange (AUDIX) An Avaya messaging system. AUDIX has been replaced by Message Manager.

AUDIX See Audio Information Exchange (AUDIX).

auto-in work mode A mode in which an agent is ready to process another call as soon as the current call is completed. Auto-in work mode is one of four agent work modes. Also see, aux-work mode, manual-in work mode, and After Call Work (ACW) mode.

Automatic Alternate Routing (AAR) A feature that routes calls to a different route than the first-choice route when facilities are unavailable.

Automatic Call Distribution (ACD) A feature that answers calls, and then depending on administered instructions, delivers messages appropriate for the caller and routes the call to an agent when one becomes available.

Automatic Call Distribution (ACD) split A method of routing calls of a similar type among agents in a call center. Also, a group of extensions that are staffed by agents trained to handle a certain type of incoming call.

Automatic Callback A feature that enables internal callers, upon reaching a busy extension, to have the system automatically connect and ring both originating and receiving parties when the receiving party becomes available.

Automatic Circuit Assurance (ACA)	A feature that tracks calls of unusual duration to facilitate troubleshooting. A high number of very short calls or a low number of very long calls may signify a faulty trunk.
Automatic Number Identification (ANI)	A display of the calling number so that agents can access information about the caller.
Automatic Route Selection (ARS)	A feature that allows the system to automatically choose the least-expensive way to send a toll call.
automatic trunk	A trunk that does not require addressing information because the destination is predetermined. A request for service on the trunk, called a seizure, is sufficient to route the call. The normal destination of an automatic trunk is the communications-system attendant group. Also called automatic incoming trunk and automatic tie trunk.
auxiliary trunk	A trunk used to connect auxiliary equipment, such as radio-paging equipment, to a communications system.
aux-work mode	A mode in which agents are unavailable to receive Automatic Call Distribution (ACD) calls. Agents enter aux-work mode when involved in non-ACD activities such as taking a break, going to lunch, or placing an outgoing call. Also see, auto-in work mode, manual-in work mode, and After Call Work (ACW) mode.
available agent strategy	A strategy that determines how Best Service Routing (BSR) commands in a vector identify the best split or skill when several have available agents.
Avaya Business Advocate	A product that establishes different levels of service for different types of calls. For example, a company may decide that a premium customer gets faster service than other types of customers.
Avaya IQ	Avaya IQ is Avaya's next generation reporting platform.
AWOH	See Administration Without Hardware (AWOH).
barrier code	A security code used with remote access to prevent unauthorized access to the system.
Basic Call Management System (BCMS)	An application on the communication server that monitors the operations of an Automatic Call Distribution (ACD) application. BCMS collects data related to the calls on the communication server and organizes the data into reports that help manage ACD facilities and personnel.
BCC	See Bearer Capability Class (BCC).
BCMS	See Basic Call Management System (BCMS).

Bearer Capability Class (BCC) A code that identifies the type of a call (for example, voice and different types of data).

best The split, skill, or location that provides the most advantageous service for a caller as determined by Best Service Routing (BSR).

Best Service Routing (BSR) An Avaya communication server feature based on call vectoring that routes Automatic Call Distribution (ACD) calls to the split, skill, or contact center best able to service each call. BSR can be used on a single communication server, or it can be used to integrate resources across a network of communication servers.

bridge (bridging) The appearance of a telephone extension at one or more other telephones.

bridged appearance A call appearance on a telephone that matches a call appearance on another telephone for the duration of a call.

Business Advocate See Avaya Business Advocate.

call appearance 1. For the attendant console, the six buttons labeled a-f used to originate, receive, and hold calls. Two lights next to the button show the status of the call appearance. 2. For the telephone, a button labeled with an extension and used to place outgoing calls, receive incoming calls, or hold calls. Two lights next to the button show the status of the call appearance.

Call Detail Recording (CDR) A feature that uses software and hardware to record call data.

Call Management System (CMS) An application that enables customers to monitor and manage telemarketing centers by generating reports on the status of agents, splits, trunks, trunk groups, vectors, and VDNs. CMS enables customers to partially administer the Automatic Call Distribution (ACD) feature for a communications system.

call vector A set of vector commands used to process an incoming or internal call.

call work code A number entered by ACD agents to record the occurrence of customer-defined events (such as account codes, social security numbers, or phone numbers) on ACD calls.

callback call A call that automatically returns to a voice-terminal user who activated the Automatic Callback feature.

cause value A value that is returned in response to requests or in event reports when a denial or unexpected condition occurs.

CCS or hundred call seconds A unit of call traffic. Call traffic for a facility is scanned every 100 seconds. If the facility is busy, it is assumed to have been busy for the entire scan interval. There are 3600 seconds per hour. The Roman numeral for 100 is the capital letter C. The abbreviation for call seconds is CS. Therefore, 100 call seconds is abbreviated CCS. If a facility is busy for an entire hour, it is said to have been busy for 36 CCS.

CDR See Call Detail Recording (CDR).

Central Office (CO) A switch owned by a local telephone company that provides local telephone service (dial-tone) and access to toll facilities for long-distance calling.

Central Office (CO) trunk A telecommunications channel that provides access from the system to the public network through the local CO.

channel 1. A circuit-switched call. 2. A communications path for transmitting voice and data. 3. In wideband, all of the time slots (contiguous or noncontiguous) necessary to support a call. Example: an H0-channel uses six 64-kbps time slots. 4. A DS0 on a T1 or E1 facility not specifically associated with a logical circuit-switched call; analogous to a single trunk.

circuit 1. An arrangement of electrical elements through which electric current flows. 2. A channel or transmission path between two or more points.

circuit pack A card with microprocessors, transistors, and other electrical circuits. A circuit pack is installed in a switch carrier or bay. Also called a circuit board or circuit card.

Class of Restriction (COR) A feature that allows classes of call-origination and call-termination restrictions for telephones, telephone groups, data modules, and trunk groups. See also Class of Service (COS).

Class of Service (COS) A feature that uses a number to specify if telephone users can activate the Automatic Callback, Call Forwarding All Calls, Data Privacy, or Priority Calling features. See also Class of Restriction (COR).

CO See Central Office (CO).

communications server A software-controlled processor complex that interprets dialing pulses, tones, and keyboard characters and makes the proper connections both within the system and external to the system. The communications system itself consists of a digital computer, software, storage device, and carriers with special hardware to perform the connections. A communications system provides voice and data communications services, including access to public and private networks, for telephones and data terminals on a customer’s premises. Previously called a switch or a Private Branch eXchange (PBX).

confirmation tone A telephone tone confirming that feature activation, deactivation, or cancellation has been accepted.

connectivity A connection of disparate devices within a single system.

consider sequence	A consider series plus a `queue-to best`, `check-best`, or `reply-best` step is called a consider sequence.
consider series	A series of `consider` commands typically written in a set of two or more. A set of `consider` commands is called a consider series.
console	See attendant console.
COR	See Class of Restriction (COR).
COS	See Class of Service (COS).
coverage answer group	A group of up to eight telephones that ring simultaneously when a call is redirected to it by Call Coverage. Any one of the group can answer the call.
coverage call	A call that is automatically redirected from the called party's extension to an alternate answering position when certain coverage criteria are met.
coverage path	An order in which calls are redirected to alternate answering positions.
coverage point	An extension or attendant group, VDN, or ACD split designated as an alternate answering position in a coverage path.
covering user	A person at a coverage point who answers a redirected call.
CWC	See call work code.
data link	A configuration of physical facilities enabling end terminals to communicate directly with each other.
data terminal	An input/output (I/O) device that has either switched or direct access to a host computer or to a processor interface.
dial-repeating tie trunk	A tie trunk that transmits called-party addressing information between two communications systems.
dial-repeating trunks	A PBX tie trunk that is capable of handling PBX station-signaling information without attendant assistance.
direct agent	A feature, accessed only through ASAI that allows a call to be placed in a split queue but routed only to a specific agent in that split. The call receives normal ACD call treatment (for example, announcements) and is measured as an ACD call while ensuring that a particular agent answers.
Direct Inward Dialing (DID) trunk	An incoming trunk used for dialing directly from the public network into a communications system without help from the attendant.
domain	A group of VDNs, ACD splits, and stations.

Dynamic Percentage Adjustment	An Avaya Business Advocate feature that makes automatic adjustments to agents' target allocations as needed to help meet the administered service level targets.
Dynamic Queue Position	An Avaya Business Advocate feature that gives you the ability to queue calls from multiple VDNs to a single skill, while maintaining different service objectives for those VDNs.
Dynamic Threshold Adjustment	An Avaya Business Advocate Service Level Supervisor feature that automatically adjusts overload thresholds to engage reserve agents a bit sooner or a bit later to meet the administered service levels.
EAD-LOA	See Expert Agent Distribution-Least Occupied Agent (EAD-LOA).
EAD-MIA	See Expert Agent Distribution-Most Idle Agent (EAD-MIA).
Electronic Tandem Network (ETN)	A large private network that has automatic call-routing capabilities based on the number dialed and the most preferred route available. Each switch in the network is assigned a unique private network office code (RNX), and each telephone is assigned a unique extension.
EPN	See Expansion Port Network (EPN).
ETN	See Electronic Tandem Network (ETN),
EWT	See Expected Wait Time (EWT).
Exclusion	A feature that allows multi-appearance telephone users to keep other users with the same extension from bridging onto an existing call.
Expansion Port Network (EPN)	A port network that is connected to the Time Division Multiplex (TDM) bus and packet bus of a processor port network. Control is achieved by indirect connection of the EPN to the processor port network using a port-network link.
Expected Wait Time (EWT)	A prediction of how long a call waits in queue before the call is answered.
Expert Agent Distribution-Least Occupied Agent (EAD-LOA)	An agent selection method for delivery of calls. With EAD-LOA implemented, calls are delivered to the available agent with the highest skill level and the lowest percentage of work time since login (compared to other available agents with the same skill level). See also Expert Agent Distribution-Most Idle Agent (EAD-MIA), Uniform Call Distribution-Least Occupied Agent (UCD-LOA), and Uniform Call Distribution-Most Idle Agent (UCD-MIA).
Expert Agent Distribution-Most Idle Agent (EAD-MIA)	An agent selection method for delivery of calls. With EAD-MIA implemented, calls are delivered to the available agent with the highest skill level who has been idle the longest since their last ACD call (compared to other available agents with the same skill level). See also Expert Agent Distribution-Least

	Occupied Agent (EAD-LOA), Uniform Call Distribution-Least Occupied Agent (UCD-LOA), and Uniform Call Distribution-Most Idle Agent (UCD-MIA).
extension-in (EXT-IN)	A work state agents go into when they answer a non ACD call. If the agent is in Manual-In or Auto-In and receives an EXT-IN call, the call is recorded by the reporting adjunct as an AUX-IN call.
extension-out (EXT-OUT)	A work state that agents go into when they place a non-ACD call.
external call	A connection between a communications system user and a party on the public network, or on another communications system in a private network.
facility	A telecommunications transmission pathway and the associated equipment.
Forced Agent Logout from ACW mode	A feature used to automatically log out an Expert Agent Selection (EAS) agent who spends too much time in After Call Work (ACW) mode.
Forced Agent Logout by Clock Time	A feature used to automatically log out an Expert Agent Selection (EAS) agent at a pre-determined time. This feature is primarily used to automatically log off agents at the end of their shifts.
Forced Agent Logout \Aux Work by Location/Skill	The Forced Agent Logout/Aux Work by Location/Skill feature is used to force all agents in a location to logout, location into Aux Work, skill to logout, or skill into Aux Work. Using this feature, you can force all the agents in a given skill or location into the Aux work mode or to log out.
glare	A simultaneous seizure of a 2-way trunk by two communications systems resulting in a standoff.
ground-start trunk	A trunk on which, for outgoing calls, the system transmits a request for services to a distant switching system by grounding the trunk ring lead. To receive the digits of the called number, that system grounds the trunk tip lead. When the system detects this ground, the digits are sent.
holding time	A total length of time in minutes and seconds that a facility is used during a call.
intelligent polling	An automatic feature of Best Service Routing (BSR) that significantly reduces the number of status polls executed. When a remote location cannot be the best resource at a given moment in time, the intelligent polling feature temporarily suppresses polls to that location. Also see status poll.
intercept tone	A tone that indicates a dialing error or denial of the service requested.
interflow	An Automatic Call Distribution (ACD) term that refers to the ability to establish a connection to a second ACD and overflow a call from one ACD to the other.
internal call	A connection between two users within a system.

internal measurement A Basic Call Management System (BCMS) measurement that is made by the system.

intraflow An Automatic Call Distribution (ACD) term that refers to the ability for calls to redirect to other splits on the same communication server to backup the primary split.

in-use lamp A red light on a multiappearance telephone that lights to show which call appearance will be selected when the handset is lifted or which call appearance is active when a user is off-hook.

ISDN Gateway (IG) A feature allowing integration of the switch and a host-based telemarketing application using a link to a gateway adjunct. The gateway adjunct is a 3B-based product that notifies the host-based telemarketing application of call events.

ISDN trunk A trunk administered for use with ISDN-PRI. Also called ISDN facility.

line A transmission path between a communications system or Central Office (CO) switching system and a telephone.

line appearance See appearance.

line port A piece of hardware that provides the access point to a communications system for each circuit associated with a telephone or data terminal.

link A transmitter-receiver channel that connects two systems.

Location Preference Distribution A feature used to route incoming Automatic Call Distribution (ACD) calls to agents located at the same location where the trunk is located whenever possible.

maintenance Activities involved in keeping a telecommunications system in proper working condition: the detection and isolation of software and hardware faults, and automatic and manual recovery from these faults.

major alarm An indication of a failure that has caused critical degradation of service and requires immediate attention. Major alarms are automatically displayed on LEDs on the attendant console and maintenance or alarming circuit pack, logged to the alarm log, and reported to a remote maintenance facility, if applicable.

management terminal The terminal that is used by the system administrator to administer the switch. The terminal may also be used to access the Basic Call Management System (BCMS) feature.

manual-in work mode A mode in which an agent is ready to process another call manually. Also see, auto-in work mode, aux-work mode, and After Call Work (ACW) mode.

Maximum Agent Occupancy (MAO)	A feature used to set thresholds on the amount of time an agent spends on a call. MAO is used to prevent agent burnout. The MAO threshold is a system-administered value that places an agent in AUX mode when the agent exceeds the MAO threshold for calls.
message center	An answering service that supplies agents and stores messages for later retrieval.
message-center agent	A member of a message-center hunt group who takes and retrieves messages for telephone users.
messaging system	A generic name for a system that records, stores, plays, and distributes phone messages. Message Manager is the latest messaging system provided by Avaya.
minor alarm	An indication of a failure that could affect customer service. Minor alarms are automatically displayed on LEDs on the attendant console and maintenance or alarming circuit pack, sent to the alarm log, and reported to a remote maintenance facility, if applicable.
modular processor data module (MPDM)	A Processor Data Module (PDM) that can be configured to provide several kinds of interfaces (RS-232C, RS-449, and V.35) to customer-provided data terminal equipment (DTE).
Modular Trunk Data Module (MTDM)	A trunk-data module that can be configured to provide several kinds of interfaces (RS-232, RS-449, and V.35) to customer-provided data terminal equipment.
multiappearance telephone	A telephone equipped with several call-appearance buttons for the same extension, allowing the user to handle more than one call on that same extension at the same time.
Network Specific Facility (NSF)	An information element in an ISDN-PRI message that specifies which public-network service is used. NSF applies only when Call-by-Call Service Selection is used to access a public-network service.
NFAS	See Non-Facility Associated Signaling (NFAS).
Non-Facility Associated Signaling (NFAS)	A method that allows multiple T1 or E1 facilities to share a single D-channel to form an ISDN-PRI. If D-channel backup is not used, one facility is configured with a D-channel, and the other facilities that share the D-channel are configured without D-channels. If D-channel backup is used, two facilities are configured to have D-channels (one D-channel on each facility), and the other facilities that share the D-channels are configured without D-channels.
non switch-classified outbound calls	Proactive Contact outbound calls that are automatically launched by Communication Manager. See also, switch-classified outbound calls.

NSF	See Network Specific Facility (NSF).
occurrence	See appearance.
pickup group	A group of individuals authorized to answer any call directed to an extension within the group.
PMS	See Property Management System (PMS).
poll	See status poll.
poll suppression	An automatic feature of Best Service Routing (BSR) that significantly reduces the number of status polls executed. When a remote location cannot be the best resource at a given moment in time, the intelligent polling feature temporarily suppresses polls to that location. Also see status poll.
polling, intelligent	See intelligent polling.
PPN	See Processor Port Network (PPN).
primary extension	A main extension associated with the physical telephone or data terminal.
principal	A terminal that has its primary extension bridged on one or more other terminals.
principal (user)	A person to whom a telephone is assigned and who has message-center coverage.
private network	A network used exclusively for the telecommunications needs of a particular customer.
Processor Port Network (PPN)	A port network (PN) controlled by a switch-processing element that is directly connected to that Port Network's TDM bus and LAN bus.
Property Management System (PMS)	A stand-alone computer used by lodging and health-services organizations for services such as reservations, housekeeping, and billing.
public network	A network that can be openly accessed by all customers for local and long-distance calling.
queue	An ordered sequence of calls waiting to be processed.
queuing	A process of holding calls in order of their arrival to await connection to an attendant, to an answering group, or to an idle trunk. Calls are automatically connected in first-in, first-out sequence.
R2-MFC signaling	A signal consisting of two frequency components, such that when a signal is transmitted from a switch, another signal acknowledging the transmitted signal is received by the switch.

recall dial tone A tone signaling that the system has completed a function (such as holding a call) and is ready to accept dialing.

redirection criteria Information administered for each telephone's coverage path that determines when an incoming call is redirected to coverage.

Redirection on No Answer An optional feature that redirects an unanswered ringing ACD call after an administered number of rings. The call is then redirected back to the agent.

reorder tone A tone to signal that at least one of the facilities, such as a trunk or a digit transmitter, needed for the call was not available.

Service Level Maximizer (SLM) An agent selection strategy that ensures that a defined service level of X% of calls are answered in Y seconds. When SLM is active, the software verifies that inbound calls are matched with agents in a way that makes sure that the administered service level is met. SLM is an optional Call Vectoring feature that is used with Expert Agent Selection (EAS), and without Business Advocate.

simulated bridged appearance A feature that allows the terminal user (usually the principal) to bridge onto a call that had been answered by another party on his or her behalf. Also called a temporary bridged appearance.

SLM See Service Level Maximizer (SLM).

split (agent) status report A report that provides real-time status and measurement data for internally-measured agents and the split to which they are assigned.

split condition A condition whereby a caller is temporarily separated from a connection with an attendant. A split condition automatically occurs when the attendant, active on a call, presses the start button.

split number An identification of the split to the communication server and the Basic Call Management System (BCMS).

split report A report that provides historical traffic information for internally measured splits.

staffed An indication that an agent position is logged in. A staffed agent functions in one of four work modes: auto-in work mode, manual-in work mode, After Call Work (ACW) mode, or aux-work mode.

Station Message Detail Recording (SMDR) An obsolete term now called Call Detail Recording (CDR).

status lamp A green light that shows the status of a call appearance or a feature button by the state of the light (lit, flashing, fluttering, broken flutter, or unlit).

status poll A call placed by a consider location vector command to obtain status data from a remote location in a multi-site Best Service Routing (BSR) application.

stroke counts A method used by ACD agents to record up to nine customer-defined events per call when a reporting adjunct is active.

switch-classified outbound calls Outbound calls placed by the Proactive Contact dialer and connected to the agents. See also, non switch-classified outbound calls.

system printer An optional printer that may be used to print scheduled reports using the report scheduler.

system report A report that provides historical traffic information for internally-measured splits.

system-status report A report that provides real-time status information for internally-measured splits.

trunk A dedicated telecommunications channel between two communications systems or Central Offices (COs).

trunk allocation The manner in which trunks are selected to form wideband channels.

trunk group Telecommunications channels assigned as a group for certain functions that can be used interchangeably between two communications systems or Central Offices (COs).

UDP See Uniform Dial Plan (UDP).

Uniform Call Distribution-Least Occupied Agent (UCD-LOA) An agent selection method for delivery of calls. With UCD-LOA implemented, calls are delivered to the available agent with the lowest percentage of work time since login. Also see Expert Agent Distribution-Least Occupied Agent (EAD-LOA), Expert Agent Distribution-Most Idle Agent (EAD-MIA), and Uniform Call Distribution-Most Idle Agent (UCD-MIA).

Uniform Call Distribution-Most Idle Agent (UCD-MIA) An agent selection method for delivery of calls. With UCD-MIA implemented, calls are delivered to the available agent who has been idle the longest since their last ACD call. See also EAD-LOA, EAD-MIA, and UCD-LOA.

Uniform Dial Plan (UDP) A feature that allows a unique number assignment for each terminal in a multiswitch configuration such as a Distributed Communications System (DCS) or main-satellite-tributary system.

VDN See Vector Directory Number (VDN).

Vector Directory Number (VDN) An extension that provides access to the vectoring feature on the switch. Vectoring allows a customer to specify the treatment of incoming calls based on the dialed number.

vector-controlled split A hunt group or ACD split administered with the vector field enabled. Access to such a split is possible only by dialing a VDN extension.

work mode

A mode that an ACD agent can be in. Upon logging in, an agent enters aux-work mode. To become available to receive ACD calls, the agent enters auto-in work mode or manual-in work mode. To do work associated with a completed ACD call, an agent enters After Call Work (ACW) mode.

work state

An ACD agent may be a member of up to three different splits. Each ACD agent continuously exhibits a work state for every split of which it is a member. Valid work states are Avail, Unstaffed, AUX-Work, ACW, ACD (answering an ACD call), ExtIn, ExtOut, and OtherSpl. An agent's work state for a particular split may change for a variety of reasons. For example, an agent's work state changes when a call is answered or abandoned, or the agent changes work modes. The Basic Call Management System (BCMS) feature monitors work states and uses this information to provide BCMS reports.

Index

Special Characters

Numerics

A

B

C

D

E

L

M

N

O

P

Q

R

S

T

W

Z