



IP Office™ Platform

DevConnect support, Service Monitoring
Web Services API

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

"Documentation" means information published by Avaya in varying mediums which may include product information, operating instructions and performance specifications that Avaya may generally make available to users of its products and Cloud Services. Documentation does not include marketing materials. Avaya shall not be responsible for any modifications, additions, or deletions to the original published version of documentation unless such modifications, additions, or deletions were performed by Avaya. End User agrees 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 websites referenced within this site or documentation 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 Avaya hardware and software. 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 website: <https://support.avaya.com/helpcenter/getGenericDetails?detailId=C20091120112456651010> under the link "Warranty & Product Lifecycle" or such successor site as designated by Avaya. Please note that if You acquired the product(s) from an authorized Avaya Channel

Partner outside of the United States and Canada, the warranty is provided to You by said Avaya Channel Partner and not by Avaya.

"Cloud Cloud Service" means a cloud service subscription that You acquire from either Avaya or an authorized Avaya Channel Partner (as applicable) and which is described further in the applicable Service Description or other service description documentation regarding the applicable cloud service. If You purchase a Cloud Service subscription, the foregoing limited warranty may not apply but You may be entitled to support services in connection with the Cloud Service as described further in your service description documents for the applicable Cloud Service. Contact Avaya or Avaya Channel Partner (as applicable) for more information.

Cloud Service

THE FOLLOWING APPLIES IF YOU PURCHASE A CLOUD SERVICE SUBSCRIPTION FROM AVAYA OR AN AVAYA CHANNEL PARTNER (AS APPLICABLE), THE TERMS OF USE FOR CLOUD SERVICES ARE AVAILABLE ON THE AVAYA WEBSITE, <https://www.avaya.com/en/legal/license-terms/> UNDER THE LINK "Avaya Terms of Use for Cloud Services" OR SUCH SUCCESSOR SITE AS DESIGNATED BY AVAYA, AND ARE APPLICABLE TO ANYONE WHO ACCESSES OR USES THE CLOUD SERVICE. BY ACCESSING OR USING THE CLOUD SERVICE, OR AUTHORIZING OTHERS TO DO SO, YOU, ON BEHALF OF YOURSELF AND THE ENTITY FOR WHOM YOU ARE DOING SO (HEREINAFTER REFERRED TO INTERCHANGEABLY AS "YOU" AND "END USER"), AGREE TO THE TERMS OF USE. IF YOU ARE ACCEPTING THE TERMS OF USE ON BEHALF A COMPANY OR OTHER LEGAL ENTITY, YOU REPRESENT THAT YOU HAVE THE AUTHORITY TO BIND SUCH ENTITY TO THESE TERMS OF USE. IF YOU DO NOT HAVE SUCH AUTHORITY, OR IF YOU DO NOT WISH TO ACCEPT THESE TERMS OF USE, YOU MUST NOT ACCESS OR USE THE CLOUD SERVICE OR AUTHORIZE ANYONE TO ACCESS OR USE THE CLOUD SERVICE. YOUR USE OF THE CLOUD SERVICE SHALL BE LIMITED BY THE NUMBER AND TYPE OF LICENSES PURCHASED UNDER YOUR CONTRACT FOR THE CLOUD SERVICE, PROVIDED, HOWEVER, THAT FOR CERTAIN CLOUD SERVICES IF APPLICABLE, YOU MAY HAVE THE OPPORTUNITY TO USE FLEX LICENSES, WHICH WILL BE INVOICED ACCORDING TO ACTUAL USAGE ABOVE THE CONTRACT LICENSE LEVEL. CONTACT AVAYA OR AVAYA'S CHANNEL PARTNER FOR MORE INFORMATION ABOUT THE LICENSES FOR THE APPLICABLE CLOUD

SERVICE, THE AVAILABILITY OF ANY FLEX LICENSES (IF APPLICABLE), PRICING AND BILLING INFORMATION, AND OTHER IMPORTANT INFORMATION REGARDING THE CLOUD SERVICE.

Licenses

THE SOFTWARE LICENSE TERMS AVAILABLE ON THE AVAYA WEBSITE, <https://www.avaya.com/en/legal/license-terms/>, UNDER THE LINK "AVAYA SOFTWARE LICENSE TERMS (Avaya Products)" OR SUCH SUCCESSOR SITE AS DESIGNATED BY AVAYA, ARE APPLICABLE TO ANYONE WHO DOWNLOADS, USES AND/OR INSTALLS AVAYA SOFTWARE, PURCHASED FROM AVAYA LLC, ANY AVAYA AFFILIATE, OR AN AVAYA CHANNEL PARTNER (AS APPLICABLE) UNDER A COMMERCIAL AGREEMENT WITH AVAYA OR AN AVAYA CHANNEL PARTNER. 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 CHANNEL PARTNER; 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 LLC OR THE APPLICABLE AVAYA AFFILIATE ("AVAYA").

Copyright

Except where expressly stated otherwise, no use should be made of materials on this site, the Documentation, Software, Cloud Service, or hardware provided by Avaya. All content on this site, the documentation, Cloud Service, and the product 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 unless expressly authorized by Avaya. 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.

Virtualization

The following applies if the product is deployed on a virtual machine. Each product has its own ordering code and license types. Note that each Instance of a product must be separately licensed and ordered. For example, if the end user customer or Avaya Channel Partner would like to install two Instances of the same type of products, then two products of that type must be ordered.

Third Party Components

"Third Party Components" mean certain software programs or portions thereof included in the Software or Cloud Service may contain software (including open source software) distributed under third party agreements ("Third Party Components"), which contain terms regarding the rights to use certain portions of the Software ("Third Party Terms"). As required, information regarding distributed Linux OS source code (for those products that have distributed Linux OS source code) and identifying the copyright holders of the Third Party Components and the Third Party Terms that apply is available in the products, Documentation or on Avaya's website at: <https://www.avaya.com/en/legal/third-party-terms/> or such successor site as designated by Avaya.

The open source software license terms provided as Third Party Terms are consistent with the license rights granted in these Software License Terms, and may contain additional rights benefiting You, such as modification and distribution of the open source software. The Third Party Terms shall take precedence over these Software License Terms, solely with respect to the applicable Third Party Components to the extent that these Software License Terms impose greater restrictions on You than the applicable Third Party Terms.

The following applies if the H.264 (AVC) codec is distributed with the product. THIS PRODUCT IS LICENSED UNDER THE AVC PATENT PORTFOLIO LICENSE FOR THE PERSONAL USE OF A CONSUMER OR OTHER USES IN WHICH IT DOES NOT RECEIVE REMUNERATION TO (i) ENCODE VIDEO IN COMPLIANCE WITH THE AVC STANDARD ("AVC VIDEO") AND/OR (ii) DECODE AVC VIDEO THAT WAS ENCODED BY A CONSUMER ENGAGED IN A PERSONAL ACTIVITY AND/OR WAS OBTAINED FROM A VIDEO PROVIDER LICENSED TO PROVIDE AVC VIDEO. NO LICENSE IS GRANTED OR SHALL BE IMPLIED FOR ANY OTHER USE. ADDITIONAL INFORMATION MAY BE OBTAINED FROM VIA LICENSING ALLIANCE. SEE <https://www.via-la.com/>.

Service Provider

THE FOLLOWING APPLIES TO AVAYA CHANNEL PARTNER'S HOSTING OF AVAYA PRODUCTS OR SERVICES. THE PRODUCT OR CLOUD SERVICE MAY USE THIRD PARTY COMPONENTS SUBJECT TO THIRD PARTY TERMS AND REQUIRE A SERVICE PROVIDER TO BE INDEPENDENTLY LICENSED DIRECTLY FROM THE THIRD PARTY SUPPLIER. AN AVAYA CHANNEL PARTNER'S HOSTING OF AVAYA PRODUCTS MUST BE AUTHORIZED IN WRITING BY AVAYA AND IF THOSE HOSTED PRODUCTS USE OR EMBED CERTAIN THIRD PARTY SOFTWARE, INCLUDING BUT NOT LIMITED TO MICROSOFT SOFTWARE OR CODECS, THE AVAYA CHANNEL PARTNER IS REQUIRED TO INDEPENDENTLY OBTAIN ANY APPLICABLE LICENSE AGREEMENTS, AT THE AVAYA CHANNEL PARTNER'S EXPENSE, DIRECTLY FROM THE APPLICABLE THIRD PARTY SUPPLIER.

WITH RESPECT TO CODECS, IF THE AVAYA CHANNEL PARTNER IS HOSTING ANY PRODUCTS THAT USE OR EMBED THE G.729 CODEC, H.264 CODEC, OR H.265 CODEC, THE AVAYA CHANNEL PARTNER ACKNOWLEDGES AND AGREES THE AVAYA CHANNEL PARTNER IS RESPONSIBLE FOR ANY AND ALL RELATED FEES AND/OR ROYALTIES. THE G.729 CODEC IS LICENSED BY Sangoma Technologies Corporation SEE <https://www.asterisk.org/products/add-ons/g729-codec/>. THE H.264 (AVC) CODEC IS LICENSED UNDER THE AVC PATENT PORTFOLIO LICENSE FOR THE PERSONAL USE OF A CONSUMER OR OTHER USES IN WHICH IT DOES NOT RECEIVE REMUNERATION TO: (i) ENCODE VIDEO IN COMPLIANCE WITH THE AVC STANDARD ("AVC VIDEO") AND/OR (ii) DECODE AVC VIDEO THAT WAS ENCODED BY A CONSUMER ENGAGED IN A PERSONAL ACTIVITY AND/OR WAS OBTAINED FROM A VIDEO PROVIDER LICENSED TO PROVIDE AVC VIDEO. NO LICENSE IS GRANTED OR SHALL BE IMPLIED FOR ANY OTHER USE. ADDITIONAL INFORMATION FOR H.264 (AVC) AND H.265 (HEVC) CODECS MAY BE OBTAINED FROM VIA LICENSING ALLIANCE. SEE <https://www.via-la.com/>.

Compliance with Laws

Customer acknowledges and agrees that it is responsible for complying with any applicable laws and regulations, including, but not limited to laws and regulations related to call recording, data privacy, intellectual property, trade secret, fraud, and music performance rights, in the country or territory where the Avaya product is used.

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 Toll Fraud intervention

If You suspect that You are being victimized by Toll Fraud and You need technical assistance or support, call Technical Service Centre Toll Fraud Intervention Hotline at +1-800-643-2353 for the United States and Canada. For additional support telephone numbers, see

DevConnect support, Service Monitoring Web Services API

the Avaya Support website: <https://support.avaya.com> or such successor site as designated by Avaya.

Security Vulnerabilities

Information about Avaya's security support policies can be found in the Security Policies and Support section of <https://support.avaya.com/security>. Suspected Avaya product security vulnerabilities are handled per the Avaya Product Security Support Flow (<https://support.avaya.com/css/P8/documents/100161515>).

Downloading Documentation

For the most current versions of Documentation, see the Avaya Support website: <https://support.avaya.com>, or such successor site as designated by Avaya.

Contact Avaya Support

See the Avaya Support website: <https://support.avaya.com> for product or Cloud Service notices and articles, or to report a problem with your Avaya product or Cloud Service. For a list of support telephone numbers and contact addresses, go to the Avaya Support website: <https://support.avaya.com> (or such successor site as designated by Avaya), scroll to the bottom of the page, and select Contact Avaya Support

Contents

1	Introduction.....	7
1.1	Purpose.....	7
1.2	Intended Audience	7
1.3	Document Changes.....	7
1.4	Requirements	7
2	System Monitoring API.....	8
2.1	Security	8
2.2	Configuration	8
2.3	License	9
2.4	Interface	9
2.5	Data format	15
2.6	Versioning.....	15
2.7	Server Edition and SCN	15
3	Data by Named URL.....	16
3.1	System	16
3.2	Alarms.....	17
3.3	Directory	23
3.4	Expansion Modules.....	26
3.5	Extensions.....	27
3.6	Licenses.....	30
3.7	Locations.....	31
3.8	IP 500v2 Internal Modules.....	31
3.9	Switch Summary.....	32
3.10	Time	33
3.11	Trunks	33
3.12	Voicemail	39
4	By Numeric Component ID(CID) of the SSA internal interface.....	40
4.1	CID URLs.....	41
5	Test/Demo Application.....	43
6	Selected Enumerations with Notes	45
6.1	System	45
6.2	Alarm Information	45
6.3	Extension	46

6.4	Directory	46
6.5	Licenses.....	47
6.6	Time	49
6.7	SysResourceType	49
6.8	Memory Card Slot Data	49
6.9	CallerIdType.....	49
7	References	50

1 Introduction

1.1 Purpose

This document sets out to provide information about the operation of the Service Monitoring Web Services API, first delivered on IP Office in Release 10.1.

1.2 Intended Audience

This document is part of the SDK for DevConnect partners who wish to develop applications to monitor the high level status of IP Office systems, typically for Proactive Resource Monitoring.

As well as this document, the SDK includes the XSD and a sample application, with source, and all items should be used together.

At first release the data is based on the data that is used for the IP Office System Status Application (SSA). Having SSA and its manual to hand will be useful for developers.

1.3 Document Changes

First GA Release - minor updates from the Beta Trial version.

Issue	Date	Change
1.01	09-Jun-2017	Update following DevConnect team feedback
1.02	20-Jul-2017	No change to API, standardise the presentation of the URLs, add more detail in several sections.
1.03	1-Aug-2025	Updated for release 12.2

1.4 Requirements

A developer intending to use this interface must be comfortable developing RESTful interfaces and handling data presented in XML. The developer should either be an IP Office certified installer/maintainer or have local access to someone with those skills.

Access to an IP Office is assumed and there are special purchase arrangements through DevConnect.

IP Office documentation is available on the IP Office Knowledge Base (see references) - registration required, and <http://support.avaya.com>.

2 System Monitoring API

The System Monitoring API is a REST Web Service which in response to an HTTPS GET request returns data in an XML Format.

The data available on this interface is intended for low traffic Resource Monitoring type usage, for example a maintainer or cloud host wishing to monitor general health, inventory, and utilisation for a fleet of IP Office systems. Alternatively a Business Partner may want to periodically query the inventory of systems under maintenance to ensure that the contract correctly covers the equipment that the customer has.

It is based on the internal data structures used to drive SSA but not all data available in SSA is available on this interface. For example call by call detail is not provided as there are APIs more suited for that operation.

2.1 Security

Any developer or user of this interface should be aware of the IP Office Security Guidelines;

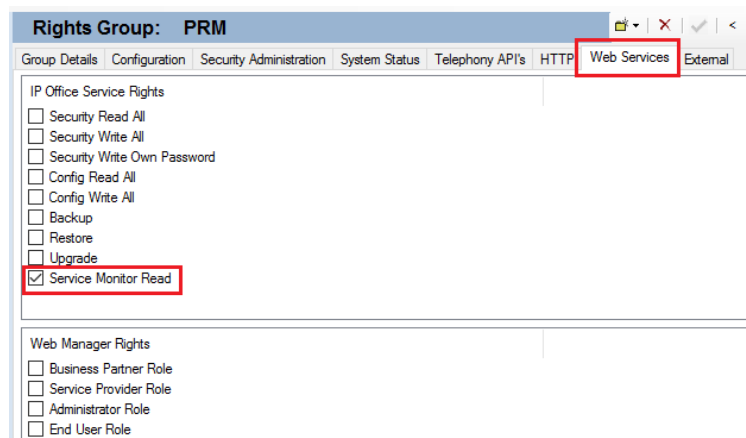
<https://ipofficekb.avaya.com/knowledgebase/businesspartner/ipoffice/mergedProjects/security/index.htm>

.

There is no default account for this interface and IP Office incorporates account lockout in default.

2.2 Configuration

A Service User must be configured to provide access to the API, following good security practices there is no default account;



Rights Group: PRM

Group Details Configuration Security Administration System Status Telephony API's HTTP Web Services External

IP Office Service Rights

- ☐ Security Read All
- ☐ Security Write All
- ☐ Security Write Own Password
- ☐ Config Read All
- ☐ Config Write All
- ☐ Backup
- ☐ Restore
- ☐ Upgrade
- ☒ Service Monitor Read

Web Manager Rights

- ☐ Business Partner Role
- ☐ Service Provider Role
- ☐ Administrator Role
- ☐ End User Role

Ideally a specific Rights Group and User with minimum permissions (just this API) should be created for use by any application using this interface. This is a standard IT security principle - Least Privilege.

IP Office Manual section on creating Service users -

<https://ipofficekb.avaya.com/knowledgebase/businesspartner/ipoffice/mergedProjects/manager/ServiceUsers.html>

At default IP Office includes account lockout on repeated failure with no notification to the interface, so care is required to ensure credentials are entered correctly - access will be rejected even if you use the right credentials in the lockout period. Note that User name is case sensitive. When an account is locked alarms are generated and can be viewed with standard monitoring tools (SSA for example).

IP Office can be configured to only respond to HTTP/HTTPS requests from recognised Avaya devices – this should be disabled to use this API

00E007052805*

System LAN1 LAN2 DNS Voicemail **Telephony** Directory Services System

Name 00E007052805

Contact Information

Set contact information to place System under special control

Device ID

TFTP Server IP Address 255 . 255 . 255 . 255

HTTP Server IP Address 0 . 0 . 0 . 0

Phone File Server Type Memory Card

Manager PC IP Address 255 . 255 . 255 . 255

Avaya HTTP Clients Only ☒

Enable Softphone HTTP Provisioning ☐

Automatic Backup ☒

There is an optional NoUser Source Number (NUSN) to control the interval when the Trunk Utilisation data is latched. In default this is every 15 minutes starting at midnight - this interval can be increased (not decreased) by entering a NoUser Source Number in the form;

NoUser:

User Voicemail DND ShortCodes **Source Numbers** Telephony

Source Number

SSI_UTIL_CACHE_INTERVAL=60

The Value is in Minutes and the Maximum is 1440 - 24 hours

2.3 License

This interface does not require a license on the IP Office to enable its operation.

2.4 Interface

Connection is per System in an SCN/Server Edition (Primary, Secondary, and Expansion systems) - currently there is no centralised API.

This is a REST API only supporting GETs, HTTPS only, with Basic Authentication (RFC7617).

2.4.1 URL Format

Example URL: <https://ipoffice.example.com:8443/ws/ssi/sw?depth=all&deltas=true>

- Address - IP Office, IP Address or by DNS
- Port - 8443 fixed
- url - see later, case insensitive.
- Depth (optional) - how deep to go down the tree
as demonstrated in the demo app the data is organised hierarchically in a tree structure, and depth determines how far down that structure IP Office will go when constructing the response. Specified as an integer depth or 'all', if not specified then all is assumed.
- deltas (optional) - when set to true only changed objects are returned, all fields for changed objects are returned. Etag must be specified in the "If-None-Match" header.

2.4.2 Conditional Gets

The interface supports conditional GETs by Modified Date/Time or by Etag.

Date and time uses the "If-Modified-Since" header and Etag uses the "If-None-Match" header in the GET. The Etag is provided in the response header, it is an arbitrary value, with greater granularity than the modified time. 0 can be used if there is no previous value.

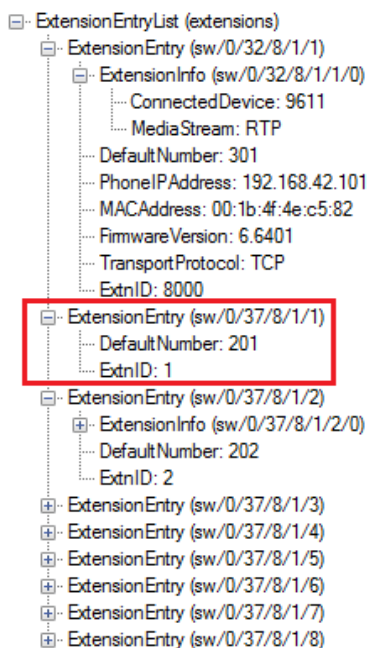
The GET Request

```
GET /ws/ssi/trunks?depth=all HTTP/1.1
Authorization: Basic QWRtaW5pc3RyYXRvcjpbBdmF5YTEyM0E=
User-Agent: Avaya-WebAdmin
If-Modified-Since: Mon, 30 Jan 2017 11:12:33 GMT
If-None-Match: 1833953
Host: 192.168.42.1:8443
```

Response from the IP Office (see below for mechanisms to see these)

```
HTTP/1.1 304 Not Modified
Date: Mon, 30 Jan 2017 11:14:49 GMT
Cache-Control: no-cache
Server: IPOffice/
Etag: 1833953
Content-Type: text/plain
Content-Length: 0
Last-Modified: Mon, 30 Jan 2017 11:12:33 GMT
```

For example, using the test application and getting the list of extensions whilst one is disconnected;



Then after connecting an extension and performing the request with Get Deltas checked;

	<pre><?xml version="1.0" encoding="UTF-8"?> <response status="1"> <data> <ExtensionEntryList> <ExtensionEntry CID="sw/0/37/8/1/1"> <ExtensionInfo CID="sw/0/37/8/1/1/0"> <ConnectedDevice>89</ConnectedDevice> </ExtensionInfo> <DefaultNumber>201</DefaultNumber> <ExtID>1</ExtID> </ExtensionEntry> </ExtensionEntryList> </data> </response></pre>
--	---

Note that all the fields for the changed Extension object are sent, including the un-changed ones (Extension ID and Default Number).

2.4.3 Limits

As a simple HTTPS GET based interface there is no active session, so no limit on the number of places that the API can be queried from. There is a limit that the API will only process one GET request at a time, additional requests received whilst processing the first one will be rejected with HTTP Error 503. The 503 response will include a retry time (currently 5 seconds), the client should re-try having waited for at least the stated timer. Any application using this interface should include handling of the 503 response.

There is no explicit rate limit on the interface, except that IP Office will not accept, nor queue a request if it is already processing one, a rejection will be returned.

2.4.4 Error Responses

Errors can occur at two levels in the API, at the HTML layer where the request is not suitable for processing by the API and at the API level where the request is received by the API but can't or isn't processed.

2.4.4.1 HTML

400 - Bad Request for example requesting an inappropriate object – external expansions on a Linux platform

401 - check permissions (See Section 2.2) - Note user name is case sensitive.

404 - bad request - for example requesting Delats but not including the required headers.

304 - Not Modified – conditional get with nothing changed.

503 - API is busy, retry after delay.

2.4.4.2 API

Where there is some form of valid response but the request has not been actioned the body will have an XML response with a `response status` of 0, and more detail of the error;
(indented for readability)

```
<?xml version="1.0" encoding="UTF-8"?>
  <response status="0">
    <data>
      <error>
        <error_code>6</error_code>
        <error_desc>Invalid use of wildcard in Component ID</error_desc>
      </error>
    </data>
  </response>
```

Possible API Errors are;

Code	Description	Recommended Action
2	No such field	Check the Query
3	Request failed due to other system activity	Retry the Query, or simplify if error persists
4		
5	Too many wildcards in request	Check wildcard Usage
6	Invalid use of wildcard in Component ID	

2.4.5 Successful Response

A successful GET will return a body formatted in XML with `response status` of 1;
(indented for readability)

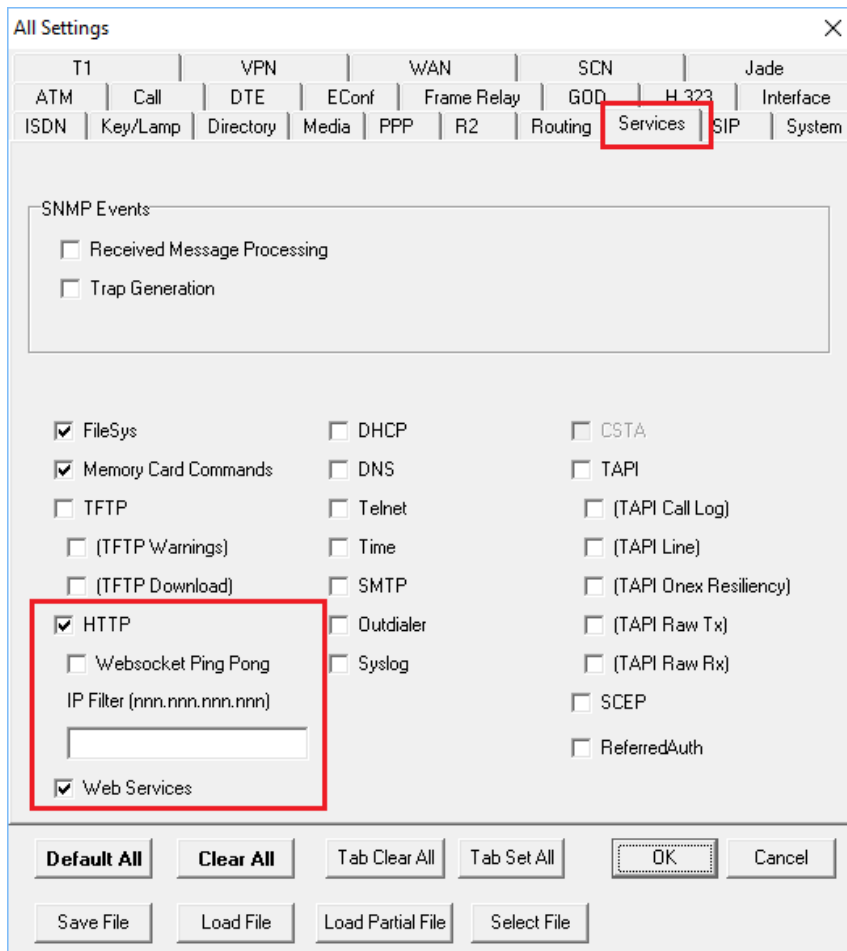
```
<?xml version="1.0" encoding="UTF-8"?>
  <response status="1">
    <data>
      <Switch CID="sw">
        <TotalExtensions>11</TotalExtensions>
        <TotalTrunks>8</TotalTrunks>
        <SwitchName>00E007052805</SwitchName>
        <SwitchAddress>192.168.42.1</SwitchAddress>
        <StartTime>05/31/2017 08:01:18 0</StartTime>
        <SwitchUpTime>23898942</SwitchUpTime>
        <LastRestartReason>4</LastRestartReason>
        <PlatformType>34</PlatformType>
        <FirmwareVersion>10.1.0.0.0 build 224</FirmwareVersion>
        <UniqueCallCount>0</UniqueCallCount>
      </Switch>
    </data>
  </response>
```

Where the request is for an empty table, there will be no error, and there will be no data between the `<data>` and `</data>` tags.

```
<?xml version="1.0" encoding="UTF-8"?>
  <response status="1">
    <data>
    </data>
  </response>
```

2.4.6 Debugging

As this is an HTTPS only interface it cannot be easily traced with tools like Wireshark. To inspect the headers the IP Office System Monitor application can be used - there are Filter options for HPPT and Web Services on the Services tab;



A successful request will look like this (192.168.42.234 is a PC running the sample application);

```
14:20:42 24696967mS PRN: Service Access Connection from 192.168.42.234(51499) to Port 8443
14:20:42 24696968mS HTTP: 192.168.42.234(51499)-(8443) HTTPSession(Secure) (Total = 3)
14:20:42 24696970mS HTTP: 192.168.42.234(51499)-(8443) HTTPSession: Operational
14:20:43 24697330mS HTTP: 192.168.42.234(51499)-(8443) HTTPSession: TLSOperational Resumed=false
14:20:43 24697573mS HTTP: Secure Rx Src: 192.168.42.234(51499)-(8443)
    GET /ws/ssi/alarms?depth=all HTTP/1.1
    Authorization: Basic QWRtaW5pc3RyYXRvcjpbDmF5YTEyM0E=
    User-Agent: ServiceMonitorTester
    Host: 192.168.42.1:8443
    Connection: Keep-Alive
14:20:43 24697573mS HTTP: 192.168.42.234(51499)-(8443) HTTPServerSessionIO: stCreationCallback(8)
14:20:43 24697574mS HTTP: 192.168.42.234(51499)-(8443) HTTPServerSessionIO: stCreationCallback
URI is authenticated
14:20:43 24697574mS HTTP: ClientSessionsMgr::PopulatePwd(): Enter
14:20:43 24697575mS WebServices: 192.168.42.234(51499) HTTPWebServiceServerSessionIO::
stCreationCallback
```

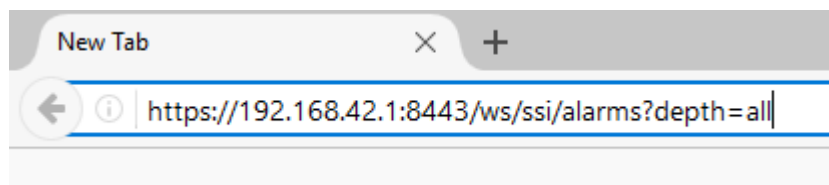
```

14:20:43 24697576mS WebServices: WS SSI: process_get
14:20:43 24697579mS HTTP: 192.168.42.234(51499)-(8443) HTTPServerSessionIO: SetState
GracefulClose
14:20:43 24697580mS HTTP: Secure Tx Dest: 192.168.42.234(51499)-(8443)
    HTTP/1.1 200 OK
    Date: Mon, 12 Jun 2017 14:20:43 GMT
    Expires: Mon, 12 Jun 2017 14:21:43 GMT
    Cache-Control: private,max-age=60
    Server: IPOffice/
    Etag: 396145
    Content-Type: text/xml; charset=utf-8
    Content-Length: 2099
    Last-Modified: Mon, 12 Jun 2017 07:35:41 GMT
14:20:43 24697582mS HTTP: 192.168.42.234(51499)-(8443) HTTPServerSessionIO: Percent(100)
14:20:43 24697585mS HTTP: 192.168.42.234(51499)-(8443) HTTPServerSessionIO: SetState
AllSentAndAcknowledged
14:20:43 24697585mS HTTP: 192.168.42.234(51499)-(8443) HTTPServerSessionIO: Persisting TCP
Session
14:20:43 24697585mS HTTP: 192.168.42.234(51499)-(8443) HTTPServerSessionIO: SetState Delete

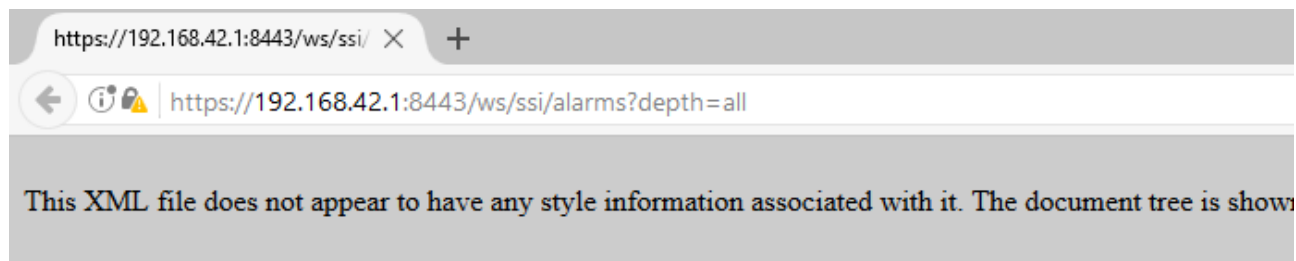
```

As part of the SDK a sample application is provided to exercise the API and show the results, both as XML and formatted in to a simple tree. See Section 5

As a simple REST interface this API can be exercised by simply entering the URL in the address bar of the browser;



The browser will prompt for credentials. And the response will be displayed;



```

- <response status="1">
  <script/>
  - <data>
    - <AlarmList>
      - <Alarm CID="sw/10/1">
        <TrunkAlarm>sw/0/62/6/17/6/0</TrunkAlarm>
        - <ReferencedAlarm>
          - <TrunkGroupAlarm CID="sw/0/62/6/17/6/0">
            - <GenericInfo CID="sw/0/62/6/17/6/0/0">
              <LastOccurrenceTime>06/12/2017 14:32:15 0 25388679</LastOccurrenceTime>
              <NumberOfOccurrences>1584</NumberOfOccurrences>
              </GenericInfo>
              <AlarmType>6</AlarmType>
            </TrunkGroupAlarm>
          </ReferencedAlarm>
        </Alarm>
      </AlarmList>
    </data>
  </response>

```

Browser Rest plugins may be available for your favourite browser or command line tools like cURL or Wget however these need careful consideration as entering the authentication credentials on the command line is a security risk.

2.5 Data format

An XSD file is provided in the SDK giving a definition of the XML data structure.

The following sections provide background to that structure; it does not give a complete description, consult the XSD for the schema. Note that many fields are optional and only delivered based on context – for example the Trunk information has IP data as well as Digital signalling type – only one set will be delivered for any one trunk.

Many responses are enumerations – the names are given in the XSD and if that is in the same directory as the test app they are displayed as names. Selected examples are presented in Chapter 7 for easy reading of this doc, with some additional information but the XSD is the main reference.

Due to the derived nature of this datastream, within the hardware descriptions and licenses there are enumerations that refer to hardware that can never be seen on the API due to them becoming obsolete before the API was added to IP Office.

2.5.1 Specific field formatting

Type	Format
Dates	UTC, US format- MM/dd/yyyy for example “01/19/2017” = 19 th January 2017
Times	24 hour - HH:mm:ss for example “13:22:14” = 1:22:14 in the afternoon
Switch Up Time	Milliseconds. for example “732598”= 12 minutes, 12.598 seconds
CPU Utilisation	Ten thousandths for example 665=6.65%
Memory	kilobytes (1024 bytes)
Alarm Last Occurrence	[Date] [Time] “0” [Switch Up Time]
Last Update	For example “01/19/2017 13:22:14 0 732598”
Utilisation (trunk)	Seconds
Strings	UTF8

2.6 Versioning

The version of this interface is the same as the version of the IP Office providing it. This can be checked using the Switch URL - section 3.9. The API is new in IP Office release 10.1 and is not available in prior releases.

The API will not change within a Major.Minor version except in exceptional circumstances. Future IP Office releases may add URLs or extra data to existing structures, any changes will be documented in updates to this document - available via DevConnect.

2.7 Server Edition and SCN

This API will only return responses relevant to the system/server that it is queried on, with the exception for example of the list of other systems in the Small Community Network. In an SCN or Server Edition deployment, an application should query each system/node in the SCN or Server Edition (Primary, Secondary, and Expansion). There is no Failover /Resilience operation for this API.

3 Data by Named URL

These have been created as a method of accessing areas of the Monitoring dataset without having to get too involved with the IP Office internal representation of the dataset. The response always includes the Component ID (CID) which can be queried instead.

3.1 System

Path=/ws/ssi/system

This is a summary of the IP Office platform;

IP500V2	Linux Host
SystemInfo (sw/6) MaxConfigSize: 2048 CurrentConfigSize: 31 CPUUsage: 665 MemoryFree: 59726 OperatingMode: IPOffice LoaderVersion: P14 Loader v1.40 CPUVersion: MPC8248 CPU Revision 0x0c10 PCBVersion: 0 PLDVersion: 32 PCBOptions: 34304 FPGAId: 1 FPGAIssue: 0 FPGABuild: 2087 NANDFlashSize: 64 NANDFlashInfo: Hynix LAN1MACAddress: 00:e0:07:05:28:05 LAN2MACAddress: 00:e0:07:85:28:05 MemoryUsed: 13144	SystemInfo (sw/6) MaxConfigSize: 49152 CurrentConfigSize: 17 CPUUsage: 18 MemoryFree: 2870056 OperatingMode: Primary CPUVersion: Intel(R) Core(TM)2 Duo CPU E6850 @ 3.00GHz LAN1MACAddress: 00:1e:c9:82:0d:d8 MemoryUsed: 52312

```
<xs:element type="xs:long" name="MaxConfigSize" default="-1"/>
<xs:element type="xs:long" name="CurrentConfigSize" default="-1"/>
<xs:element type="xs:long" name="CPUUsage" default="-1"/>
<xs:element type="xs:long" name="MemoryFree" default="-1"/>
<xs:element type="xs:long" name="OperatingModeEnumeration" name="OperatingMode"/>
<xs:element type="xs:string" name="LoaderVersion"/>
<xs:element type="xs:string" name="CPUVersion"/>
<xs:element type="xs:long" name="PCBVersion" default="-1"/>
<xs:element type="xs:long" name="PLDVersion" default="-1"/>
<xs:element type="xs:long" name="PCBOptions" default="-1"/>
<xs:element type="xs:long" name="FPGAId" default="-1"/>
<xs:element type="xs:long" name="FPGAIssue" default="-1"/>
<xs:element type="xs:long" name="FPGABuild" default="-1"/>
<xs:element type="xs:long" name="NANDFlashSize" default="-1"/>
<xs:element type="xs:string" name="NANDFlashInfo"/>
<xs:element type="xs:string" name="LAN1MACAddress"/>
<xs:element type="xs:string" name="LAN2MACAddress"/>
<xs:element type="xs:long" name="MemoryUsed" default="-1"/>
```

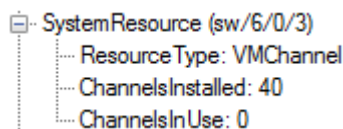
- Config Size Max/Current - in kilobytes (1024 bytes)
- CPU Utilisation - 665=6.65%, this is a spot measure, no averaging
- Memory Used/Available - in kilobytes (1024 bytes)
- Operating Mode (an enumeration 4=IP Office) - see section 7
- Then general Hardware platform information.
- Unlike most other Named URLs it doesn't include sub lists

This is the CID 6 but with the sub lists and alarms omitted (as noted in the XML and tree presentation).

3.1.1 System - Resources

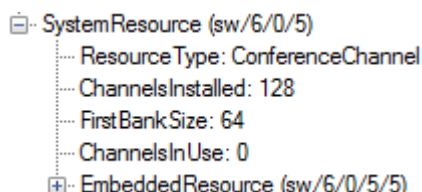
Path=/ws/ssi/system/resources

A list of resources offered by the platform. The total quantity of resources may not be available due to licensing – for example an IP500V2 will support 40 VocieMail Pro channels but the number available is down to licensing.



```
<xs:element type="SysResourceTypeEnumeration" name="ResourceType"/>
<xs:element type="xs:long" name="ChannelsInstalled" default="-1"/>
<xs:element type="xs:long" name="FirstBankSize" default="-1"/>
<xs:element type="xs:long" name="ChannelsInUse" default="-1"/>
<xs:element type="CongestionAlarm" name="CongestionAlarm"/>
<xs:element type="SystemResource" name="EmbeddedResource"/>
```

An embedded resource is a special type of the primary resource – for example high quality conference channels.

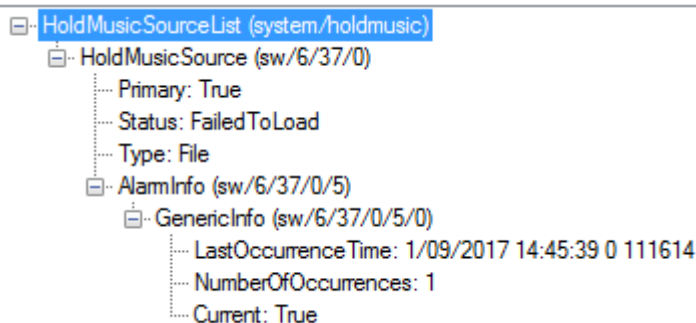


3.1.2 System - Holdmusic

Path=/ws/ssi/system/holdmusic

A list of configured MoH sources and their status

```
<xs:element type="xs:boolean" name="Primary"/>
<xs:element type="xs:string" name="Name"/>
<xs:element type="xs:string" name="Origin"/>
<xs:element type="MOHStatusEnumeration" name="Status"/>
<xs:element type="MOHSourceEnumeration" name="Type"/>
<xs:element type="MOHSourceAlarm" name="AlarmInfo"/>
```



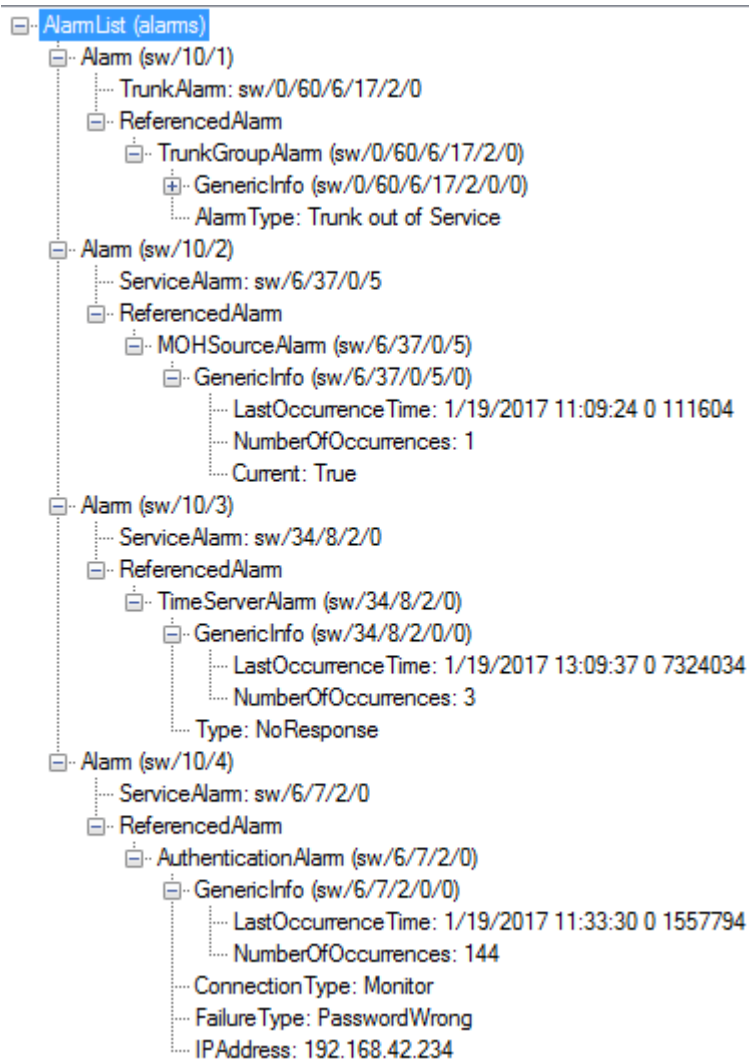
3.2 Alarms

Historical alarm data stored in the core is available through his API, though the usual mechanism for monitoring alarms would by SysLog or SSA.

The storage for alarms is limited and only the last instance of an alarm is kept, plus a repeat count, when the internal table is full the oldest, lowest priority alarm is discarded to make room for a new alarm.

3.2.1 Main URL

Path=/ws/ssi/alarms



All alarms have at least the Generic Alarm Information which comprises of the last occurrence, a count of occurrences and a flag to indicate if the thing generating the alarm is still in error.

```
<xs:element type="GenericAlarmInfo" name="GenericInfo"/>
  <xs:element type="xs:string" name="LastOccurrenceTime"/>
  <xs:element type="xs:long" name="NumberOfOccurrences" default="-1"/>
  <xs:element type="xs:boolean" name="Current"/>
```

Date Format for alarms is US style - month/day/year;

```
<LastOccurrenceTime>1/09/2017 15:04:34 0 1246908</LastOccurrenceTime>
```

3.2.2 Generic Alarms

The Following Alarms have just the generic Alarm data – more context is given by the alarm name or the higher level object that this alarm is raised against when viewing those components.

```
<xs:complexType name="DeviceChange">
<xs:complexType name="ValidLicenseAlarm">
<xs:complexType name="AlarmTestAlarm">
<xs:complexType name="MOHSourceAlarm">
<xs:complexType name="CongestionAlarm">
<xs:complexType name="LogStampedAlarm">
<xs:complexType name="GeneralAlarm">
```

```

<xs:complexType name="NonSelectAlarm">
<xs:complexType name="CCRUnsupportedAlarm">
<xs:complexType name="HardDriveAlarm">
<xs:complexType name="TCPCongestionAlarm">
<xs:complexType name="InsufficientRAMAlarm">
<xs:complexType name="PortRangeAlarm">

```

3.2.3 IP DECT

See Enumeration Description

```

<xs:complexType name="IPDECTAlarm">
  <xs:element type="GenericAlarmInfo" name="GenericInfo"/>
  <xs:element type="IPDECTAlarmTypeEnumeration" name="Type"/>
  <xs:element type="IPDECTFeatureEnumeration" name="Feature"/>

```

3.2.4 Memory Card

```

<xs:complexType name="OEMCardSlotAlarm">
  <xs:element type="GenericAlarmInfo" name="GenericInfo"/>
  <xs:element type="OEMCardSlotAlarmTypeEnumeration" name="AlarmType"/>
  <xs:element type="xs:long" name="Slot" default="-1"/>

```

3.2.5 Hardware

Errors regarding IP500 Device Hardware

```

<xs:complexType name="HardwareAlarm">
  <xs:element type="GenericAlarmInfo" name="GenericInfo"/>
  <xs:element type="HardwareAlarmTypeEnumeration" name="HardwareType"/>
  <xs:element type="HardwareFailureTypeEnumeration" name="FailureType"/>
  <xs:element type="xs:long" name="Slot" default="-1"/>

```

3.2.6 Empty Codec list or no matching codecs

```

<xs:complexType name="CodecAlarm">
  <xs:element type="GenericAlarmInfo" name="GenericInfo"/>
  <xs:element type="xs:long" name="LineNumber" default="-1"/>
  <xs:element type="xs:long" name="ExtnNumber" default="-1"/>

```

3.2.7 Emergency Call Attempt

Emergency Call alarm is usually consumed by an on-site notification application though a different API.

```

<xs:complexType name="DialEmergencyAlarm">
  <xs:element type="GenericAlarmInfo" name="GenericInfo"/>
  <xs:element type="CallerIdTypeEnumeration" name="CallerIDType"/>
  <xs:element type="FailureCauseEnumeration" name="FailureCause"/>
  <xs:element type="xs:string" name="DialledNumber"/>
  <xs:element type="xs:string" name="CalledNumber"/>
  <xs:element type="xs:string" name="CallerId"/>
  <xs:element type="xs:string" name="UserMakingCallNumber"/>
  <xs:element type="xs:string" name="UserMakingCallName"/>
  <xs:element type="xs:string" name="BaseExtnNumber"/>
  <xs:element type="xs:long" name="BaseExtnId" default="-1"/>
  <xs:element type="xs:long" name="CallerCardNumber" default="-1"/>
  <xs:element type="xs:long" name="CallerModuleNumber" default="-1"/>
  <xs:element type="xs:long" name="CallerPortNumber" default="-1"/>
  <xs:element type="xs:string" name="CallerIPAddress"/>
  <xs:element type="xs:string" name="CallerPhoneNumber"/>
  <xs:element type="xs:long" name="CallerLineNumber" default="-1"/>
  <xs:element type="xs:string" name="CallerLocation"/>
  <xs:element type="xs:string" name="CallerMACAddress"/>
  <xs:element type="xs:string" name="IncomingCallId"/>

```

3.2.8 Platform Resources

Server Edition host machine resources

```

<xs:complexType name="PlatformResourceStatusAlarm">
  <xs:element type="GenericAlarmInfo" name="GenericInfo"/>
  <xs:element type="PlatformResourceTypeEnumeration" name="Resource"/>
  <xs:element type="PlatformResourceStatusEnumeration" name="Status"/>

```

3.2.9 TLS certificate expiration

```

<xs:complexType name="ExpirationAlarm">
  <xs:element type="GenericAlarmInfo" name="GenericInfo"/>
  <xs:element type="ExpirationAlarmTypeEnumeration" name="Type"/>

```

```
<xs:element type="xs:long" name="Value" default="-1"/>
```

3.2.10 Authentication Alarm

```
<xs:complexType name="AuthenticationAlarm">
  <xs:element type="GenericAlarmInfo" name="GenericInfo"/>
  <xs:element type="ConnectionTypeEnumeration" name="ConnectionType"/>
  <xs:element type="FailureTypeEnumeration" name="FailureType"/>
  <xs:element type="xs:string" name="UserName"/>
  <xs:element type="xs:string" name="UserNumber"/>
  <xs:element type="xs:string" name="IPAddress"/>
  <xs:element type="xs:string" name="Service"/>
  <xs:element type="xs:string" name="WebService"/>
  <xs:element type="xs:long" name="BlockInterval" default="-1"/>
```

3.2.11 Outgoing Congestion

```
<xs:complexType name="OGFullAlarm">
  <xs:element type="GenericAlarmInfo" name="GenericInfo"/>
  <xs:element type="xs:long" name="OGGNumber" default="-1"/>
```

3.2.12 Change in Digital Trunk Clock Source

```
<xs:complexType name="ClockSourceAlarm">
  <xs:element type="GenericAlarmInfo" name="GenericInfo"/>
  <xs:element type="xs:string" name="PreviousSource"/>
```

3.2.13 Discarded Alarms

The internal buffer for alarms has overflowed

```
<xs:complexType name="LostAlarm">
  <xs:element type="GenericAlarmInfo" name="GenericInfo"/>
  <xs:element type="xs:long" name="DiscardedEvents" default="-1"/>
```

3.2.14 Call Failure

Reason is in the Enumeration

```
<xs:complexType name="CallAlarm">
  <xs:element type="GenericAlarmInfo" name="GenericInfo"/>
  <xs:element type="CallAlarmTypeEnumeration" name="Type"/>
```

3.2.15 QoS Alarm

Generated at the end of a call segment where configured QoS parameters have been exceeded

```
<xs:complexType name="CallQoSAlarm">
  <xs:element type="GenericAlarmInfo" name="GenericInfo"/>
  <xs:element type="xs:long" name="CallId" default="-1"/>
  <xs:element type="xs:string" name="Address"/>
  <xs:element type="xs:string" name="PeerAddress"/>
  <xs:element type="xs:string" name="ExtensionNumber"/>
  <xs:element type="xs:string" name="LineNumber"/>
  <xs:element type="xs:long" name="Jitter" default="-1"/>
  <xs:element type="xs:long" name="RoundTripDelay" default="-1"/>
  <xs:element type="xs:long" name="PacketLoss" default="-1"/>
```

3.2.16 Trunk Group Alarm

Detail in the Enum – typically related to the circuit

```
<xs:complexType name="TrunkGroupAlarm">
  <xs:element type="GenericAlarmInfo" name="GenericInfo"/>
  <xs:element type="TrunkGroupAlarmTypeEnumeration" name="AlarmType"/>
```

3.2.17 Trunk (TDM Circuit) Loopback change

```
<xs:complexType name="LoopbackAlarm">
  <xs:element type="GenericAlarmInfo" name="GenericInfo"/>
  <xs:element type="LoopbackPresentList" name="LoopbacksPresent"/>
```

3.2.18 Trunk Channel

Detail in the Enum – typically related to a specific channel/timeslot in the circuit

```
<xs:complexType name="TrunkChannelAlarm">
  <xs:element type="GenericAlarmInfo" name="GenericInfo"/>
  <xs:element type="TrunkChannelAlarmTypeEnumeration" name="AlarmType"/>
  <xs:element type="xs:long" name="DDIDigitsRequired" default="-1"/>
  <xs:element type="xs:string" name="DDI"/>
```

3.2.19 Non-Unique entries in SCN

```
<xs:complexType name="DuplicateEntries">
  <xs:element type="GenericAlarmInfo" name="GenericInfo"/>
  <xs:element type="ConflictingEntryList" name="ConflictingEntries"/>
  <xs:element type="ConflictingSourceList" name="ConflictingSources"/>
</xs:complexType>
```

3.2.20 License Alarm

Attempt to use a feature where there's no License available – license is in the enum

```
<xs:complexType name="NoLicenseAlarm">
  <xs:element type="GenericAlarmInfo" name="GenericInfo"/>
  <xs:element type="LicenseTypeEnum" name="LicenseType"/>
</xs:complexType>
```

3.2.21 PLDS/Leb LM License Alarms

```
<xs:complexType name="PLDSFileAlarm">
  <xs:element type="GenericAlarmInfo" name="GenericInfo"/>
  <xs:element type="PLDSFileAlarmTypeEnum" name="AlarmType"/>
</xs:complexType>

<xs:complexType name="LicenseErrorAlarm">
  <xs:element type="GenericAlarmInfo" name="GenericInfo"/>
  <xs:element type="LicenseErrorTypeEnum" name="ErrorType"/>
  <xs:element type="xs:long" name="GraceDays" default="-1"/>
  <xs:element type="LicenseTypeEnum" name="LicenseType"/>
</xs:complexType>

<xs:complexType name="WebLicensingAlarm">
  <xs:element type="GenericAlarmInfo" name="GenericInfo"/>
  <xs:element type="LicenseServerAlarmTypeEnum" name="AlarmType"/>
  <xs:element type="LicenseTypeEnum" name="LicenseType"/>
  <xs:element type="xs:long" name="InstancesRequested" default="-1"/>
  <xs:element type="xs:long" name="InstancesAvailable" default="-1"/>
</xs:complexType>
```

3.2.22 VoiceMail Storage Alarms

Embedded VM or relayed from VMPro

```
<xs:complexType name="StorageAlarm">
  <xs:element type="GenericAlarmInfo" name="GenericInfo"/>
  <xs:element type="StorageStatusEnum" name="CurrentStatus"/>
</xs:complexType>
```

3.2.23 SSL VPN

```
<xs:complexType name="SSLVPNAlarm">
  <xs:element type="GenericAlarmInfo" name="GenericInfo"/>
  <xs:element type="SSLVPNAlarmTypeEnum" name="AlarmType"/>
  <xs:element type="xs:string" name="ServiceName"/>
</xs:complexType>
```

3.2.24 Time Server

```
<xs:complexType name="TimeServerAlarm">
  <xs:element type="GenericAlarmInfo" name="GenericInfo"/>
  <xs:element type="TimeServerAlarmTypeEnum" name="Type"/>
  <xs:element type="xs:string" name="ServerAddress"/>
  <xs:element type="xs:string" name="HostName"/>
</xs:complexType>
```

3.2.25 TLS Alarm

```
<xs:complexType name="TLSErrorAlarm">
  <xs:element type="GenericAlarmInfo" name="GenericInfo"/>
  <xs:element type="xs:string" name="IPAddress"/>
  <xs:element type="xs:long" name="Port" default="-1"/>
  <xs:element type="xs:string" name="PeerIPAddress"/>
  <xs:element type="xs:long" name="PeerPort" default="-1"/>
  <xs:element type="TLSErrorAlarmTypeEnum" name="Type"/>
</xs:complexType>
```

3.2.26 SRTP Alarm

```
<xs:complexType name="SRTPAlarm">
  <xs:element type="GenericAlarmInfo" name="GenericInfo"/>
  <xs:element type="xs:string" name="IPAddress"/>
  <xs:element type="xs:long" name="Port" default="-1"/>
  <xs:element type="xs:string" name="PeerIPAddress"/>
  <xs:element type="xs:long" name="PeerPort" default="-1"/>
  <xs:element type="SRTPAlarmTypeEnum" name="Type"/>
  <xs:element type="xs:string" name="SourceEnd"/>
  <xs:element type="xs:string" name="DestinationEnd"/>
</xs:complexType>
```

3.2.27 WS API

```
<xs:complexType name="WebServiceAlarm">
  <xs:element type="GenericAlarmInfo" name="GenericInfo"/>
  <xs:element type="WSAlarmTypeEnumeration" name="Type"/>
  <xs:element type="WSAlarmCodeEnumeration" name="Code"/>
</xs:complexType>
```

3.2.28 Call Admission Control call limit reached

```
<xs:complexType name="LocationAlarm">
  <xs:element type="GenericAlarmInfo" name="GenericInfo"/>
  <xs:element type="LimitTypeEnumeration" name="LimitType"/>
</xs:complexType>
```

3.2.29 ACCS Integration

```
<xs:complexType name="ACCSAlarm">
  <xs:element type="GenericAlarmInfo" name="GenericInfo"/>
  <xs:element type="ACCSAlarmTypeEnumeration" name="Type"/>
  <xs:element type="xs:string" name="HostName"/>
</xs:complexType>
```

3.2.30 Down

```
<xs:complexType name="DownAlarm">
  <xs:element type="GenericAlarmInfo" name="GenericInfo"/>
  <xs:element type="xs:string" name="Description"/>
</xs:complexType>
```

3.2.31 Filtered lists

The Alarm list can be filtered to only return a subset of Alarms;

Area	Path	Alarms included
Config	/ws/ssi/alarms/config	ProcessorAlarm (various) DuplicateEntries LostAlarm HardwareAlarm CodecAlarm InsufficientRAMAlarm TrunkChannelAlarm (NoRoute)
Link	/ws/ssi/alarms/link	DeviceChange SCEPLinkAlarm TCPCongestionAlarm UpdownHistory

Area	Path	Alarms included
Service	/ws/ssi/alarms/service	ACCSAlarm ProcessorAlarm (various) NoLicenseAlarm PLDSFileAlarm ValidLicenseAlarm LicenseErrorAlarm WebLicensingAlarm LocationAlarm IPDECTAlarm ClockSourceAlarm AuthenticationAlarm DialEmergencyAlarm OGFULLAlarm PortRangeAlarm OEMCardSlotAlarm PlatformResourceStatusAlarm HardDriveAlarm LogStampedAlarm GeneralAlarm NonSelectAlarm CCRUnsupportedAlarm ExpirationAlarm MOHSourceAlarm CongestionAlarm AlarmTestAlarm TimeServerAlarm SSLVPNAlarm CallAlarm StorageAlarm WebServiceAlarm
Trunk	/ws/ssi/alarms/trunk	TrunkGroupAlarm TrunkChannelAlarm
QoS	/ws/ssi/alarms/qos	CallQoSAlarm
TLS	/ws/ssi/alarms/tls	TLSAlarm
SRTP	/ws/ssi/alarms/srtp	SRTPAlarm

3.3 Directory

List of all local Directory Entries, SCN systems, and other sources.

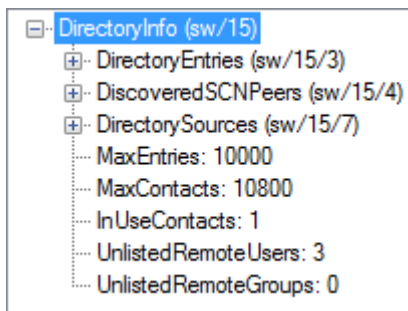
3.3.1 Main URL

Path=/ws/ssi/directory

```

<xs:element type="DirectoryListEntryList" name="DirectoryEntries"/>
<xs:element type="DiscoveredSCNPeerList" name="DiscoveredSCNPeers"/>
<xs:element type="DuplicateEntries" name="ConflictingDialplan"/>
<xs:element type="DirectorySourceEntryList" name="DirectorySources"/>
<xs:element type="xs:long" name="MaxEntries" default="-1"/>
<xs:element type="xs:long" name="MaxContacts" default="-1"/>
<xs:element type="xs:long" name="InUseContacts" default="-1"/>
<xs:element type="xs:long" name="UnlistedRemoteUsers" default="-1"/>
<xs:element type="xs:long" name="UnlistedRemoteGroups" default="-1"/>

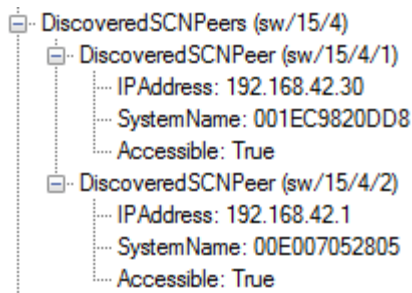
```



Contacts are the internal Personal directory/speed dial entries.

Unlisted Remote Users/Groups refer to Users/Groups on other nodes in the SCN

Discovered SCN Peers is a list of the Systems in the SCN/Server Edition including this node



```
<xs:element type="xs:string" name="IPAddress"/>
<xs:element type="xs:string" name="SystemName"/>
<xs:element type="xs:boolean" name="Accessible"/>
```

SCN is a self-discovering internal peer-to-peer network – the only configuration item is one or more trunks to other nodes in the network, all other data is discovered and maintained at run time and no data is persisted. Accessible False indicates a node that was connected to the SCN but is not currently visible, it will eventually get dropped.

Duplicate Entries List is an Alarm which flags where Names are duplicated in the SCN.

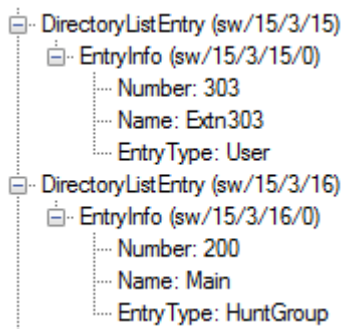
```
<xs:element type="GenericAlarmInfo" name="GenericInfo"/>
<xs:element type="ConflictingEntryList" name="ConflictingEntries"/>
<xs:element type="ConflictingSourceList" name="ConflictingSources"/>
```

3.3.2 Entries

Path=/ws/ssi/directory/entries

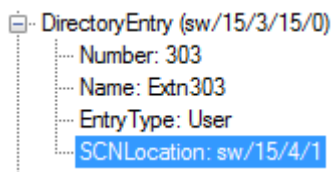
A List of DirectoryEntry types

```
<xs:element type="xs:string" name="Number"/>
<xs:element type="xs:string" name="Name"/>
<xs:element type="DirectoryEntryTypeEnumeration" name="EntryType"/>
<xs:element type="xs:string" name="SCNOwner"/>
<xs:element type="xs:string" name="SCNLocation"/>
```

User and Groups – we don't list all the System or External entries.

When a User from this system is Hot Desked to another system in the SCN we give a reference to the CID for that system – see above. SCNOwner may be show for certain Contact Centre Agents.



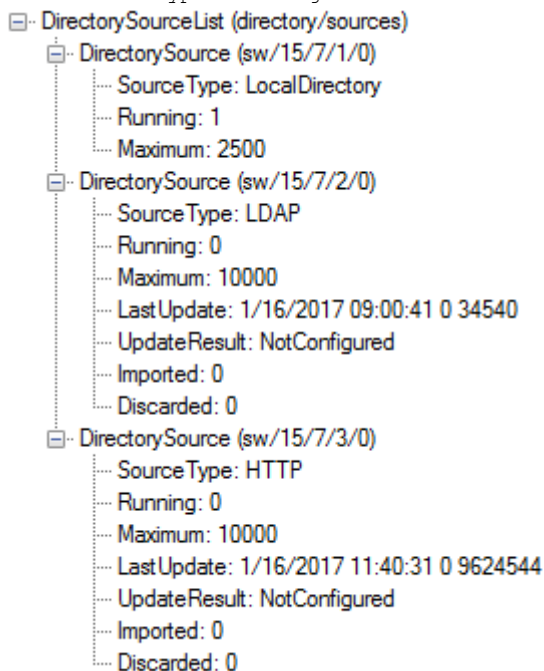
3.3.3 Sources

Path=/ws/ssi/directory/entries

A List of DirectorySource types

```

<xs:element type="DirectorySourceTypeEnumeration" name="SourceType"/>
<xs:element type="xs:long" name="Running" default="-1"/>
<xs:element type="xs:long" name="Maximum" default="-1"/>
<xs:element type="xs:string" name="LastUpdate"/>
<xs:element type="UpdateResultEnumeration" name="UpdateResult"/>
<xs:element type="xs:long" name="Imported" default="-1"/>
<xs:element type="xs:long" name="Discarded" default="-1"/>
  
```



Directory can be Configured in the IP Office, copied from another IP Office, or read in through LDAP. This lists the sources and the status of them.

3.4 Expansion Modules

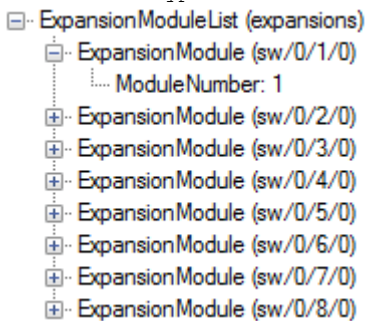
Detail of the IP500V2 external Expansion Modules, if any. IP500v2 only, not available on Linux servers.

List of type `ExpansionModule`.

3.4.1 Main URL

Path=/ws/ssi/expansions

```
<xs:element type="xs:long" name="ModuleNumber" default="-1"/>
<xs:element type="LoadableModule" name="LoadableModuleInfo"/>
<xs:element type="SSIGenericObject" name="GroupInfo"/>
```



`LoadableModule`

```
<xs:element type="ModuleTypeEnumeration" name="ModuleType"/>
<xs:element type="xs:long" name="HardwareType" default="-1"/>
<xs:element type="xs:long" name="FlashType" default="-1"/>
<xs:element type="xs:string" name="PCBId"/>
<xs:element type="xs:string" name="ChassisId"/>
<xs:element type="xs:string" name="FirmwareVersion"/>
```

`SSIGenericObject`

Details of the Group (Trunks or Extensions) that this module provides.

3.4.2 Specific Module

Path=/ws/ssi/expansions/x

Where X is the Expansion module number - based on the socket number, for example;

`https://ipoffice.example.com:8443/ws/ssi/expansion/3`

Detail of just the Expansion Module connected to socket 3 on the back of an IP500v2

`https://ipoffice.example.com:8443/ws/ssi/expansion/10`

Detail of just the Expansion Module connected to socket 10 on a 4 port Expansion card plugged in to the front of an IP500v2.

3.5 Extensions

3.5.1 Main URL

Path=/ws/ssi/extensions

Set of lists of all extensions by type.

The Presented data depends on the Extension type, details below. Default Number is the number of the User who will be assigned to this phone, if they are not assigned elsewhere and don't have Force Login set. This is also used for Registration of IP phones.

ExtnID is automatically assigned by IP Office numbers below 8000 are Digital/Analogue, and above are IP.

ExtensionEntry

```
<xs:element type="Extension" name="ExtensionInfo"/>
<xs:element type="xs:string" name="DefaultNumber"/>
<xs:element type="xs:string" name="PhoneIPAddress"/>
<xs:element type="xs:string" name="MACAddress"/>
<xs:element type="DeviceChange" name="DeviceChange"/>
<xs:element type="VirtualExtensionEnumeration" name="VirtualExtension"/>
<xs:element type="xs:string" name="FirmwareVersion"/>
<xs:element type="xs:boolean" name="SecondaryGatekeeper"/>
<xs:element type="xs:string" name="DeviceFeatures"/>
<xs:element type="DECTSubscriptionStateEnumeration" name="SubscriptionState"/>
<xs:element type="xs:string" name="HardwareRelease"/>
<xs:element type="xs:boolean" name="Simultaneous"/>
<xs:element type="xs:string" name="Location"/>
<xs:element type="TransportProtocolEnumeration" name="TransportProtocol"/>
<xs:element type="xs:long" name="SimultaneousID" default="-1"/>
<xs:element type="SIPApplicationTypeEnumeration" name="SIPApplication"/>
<xs:element type="xs:long" name="ExtnID" default="-1"/>
<xs:element type="xs:boolean" name="IPOBehindNAT"/>
<xs:element type="xs:boolean" name="PhoneBehindNAT"/>
<xs:element type="xs:boolean" name="PhoneMediaBehindNAT"/>
<xs:element type="xs:string" name="PrivateIPAddress"/>
<xs:element type="SetTypeEnumeration" name="LastDeviceType"/>
```

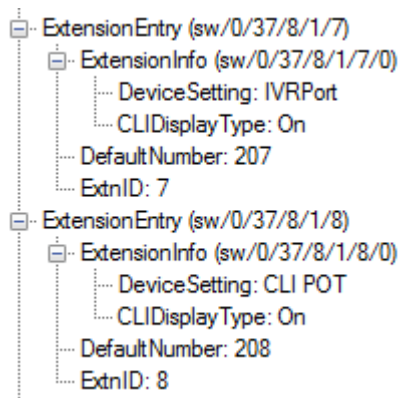
Extension

```
<xs:element type="POTSPhoneSettingEnumeration" name="DeviceSetting"/>
<xs:element type="CLIDisplayTypeEnumeration" name="CLIDisplayType"/>
<xs:element type="SetTypeEnumeration" name="ConnectedDevice"/>
<xs:element type="MediaStreamTypeEnumeration" name="MediaStream"/>
<xs:element type="xs:string" name="BaseStationName"/>
<xs:element type="DECTConnectionTypeEnumeration" name="DECTConnectionType"/>
<xs:element type="xs:string" name="PrimaryPBX"/>
<xs:element type="xs:string" name="BackupPBX"/>
```

Optionally specific types of extension can be queried.

3.5.2 Extensions - Analogue

Path=/ws/ssi/extensions/analogue

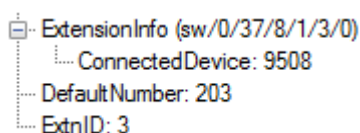


Analogue port settings are in the Manager Help –

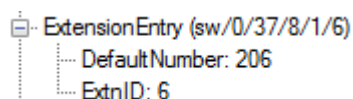
https://ipofficekb.avaya.com/knowledgebase/businesspartner/ipoffice/mergedProjects/manager/Html.Extension_Analog.html

3.5.3 Extensions - Digital

Path=/ws/ssi/extensions/digital



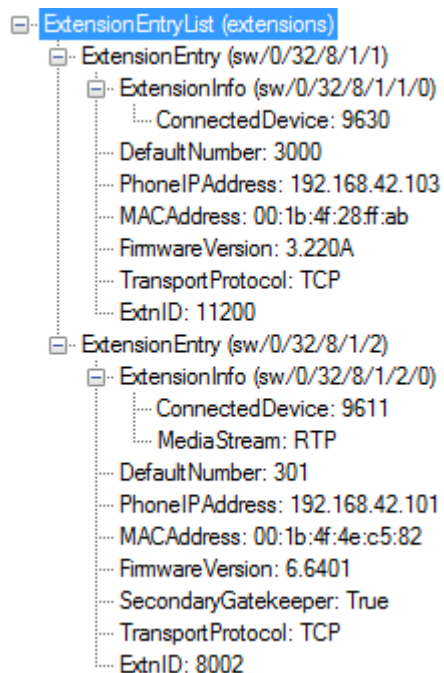
Connected Device is the currently connected Phone, for an empty port there is a record without ExtensionInfo



3.5.4 Extensions - H323

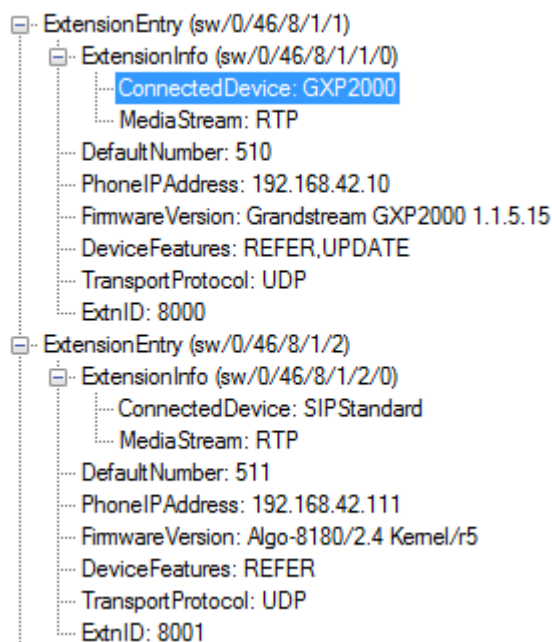
Path=/ws/ssi/extensions/h323

This example is two Avaya Feature Phones. `SecondaryGatekeeper` being true on 301 indicates a phone that has failed over to this system. Media Stream is only shown for devices that can support SRTP.



3.5.5 Extensions - SIP

Path=/ws/ssi/extensions/sip



Some devices are recognised and have explicit Enum values – others are just generic “SIP Standard”.

3.5.6 Extensions - ADMM

Path=/ws/ssi/extensions/admm

IP DECT, or DECT R4 Extensions

3.5.7 Extensions - VoIP

Path=/ws/ssi/extensions/voip

3.5.8 Extensions - Virtual

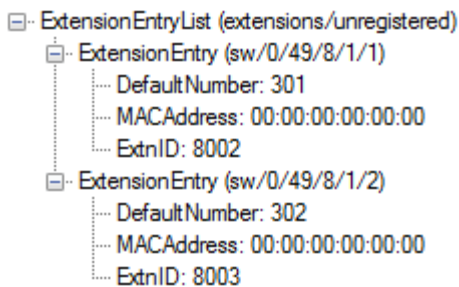
Path=/ws/ssi/extensions/virtual

Dynamically created internal extension records used for features like Mobile Twinning.

3.5.9 Extensions - Unregistered

Path=/ws/ssi/extensions/unregistered

VoIP Extensions that have Extension records but are not currently registered. Can be used for example to detect extensions that have gone off-line due to network issues.



3.5.10 Extensions - Simultaneous

Path=/ws/ssi/extensions/simultaneous

Extensions that are dynamically created by IP Office for Simultaneous operation – for example a User with a desk phone and one-X Mobile Preferred.

3.6 Licenses

Path=/ws/ssi/licenses

Several features on IP Office are enabled through licenses, this gives a mechanism to check the licenses on a specific system. Details of Feature Key/PLDS ID are under CID: sw/14

InstalledLicenseTypeList, list of InstalledLicenseType

```

<xs:element type="LicenseTypeEnumeration" name="Type"/>
<xs:element type="InstanceList" name="Instances"/>
<xs:element type="xs:long" name="InstancesInUse" default="-1"/>
<xs:element type="ValidLicenseAlarm" name="LicenseOversubscribed"/>
<xs:element type="xs:long" name="InstancesAvailable" default="-1"/>
<xs:element type="xs:long" name="ReservedInstancesAvailable" default="-1"/>
<xs:element type="xs:long" name="ReservedInstancesInUse" default="-1"/>
  
```

InstanceList, List of Instance

ADI was a cumulative license mechanism and there could be several licenses of the same type – for example to provide 25 IP Phones you may have one license of 20 and one of 5 – Instance List breaks down this detail. Normally PLDS will only have a single instance but if there's a mix of WebLM and a local license file for example there may still be multiple entries.

```

<xs:element type="xs:string" name="LicenseString"/>
<xs:element type="LicenseStatusEnumeration" name="Status"/>
<xs:element type="xs:string" name="Expiry"/>
  
```

```

<xs:element type="xs:long" name="InstancesLicensed" default="-1"/>
<xs:element type="LicenseSourceEnumeration" name="LicenseSource"/>
  InstalledLicenseTypeList (licenses)
    InstalledLicenseType (sw/14/5/2)
      Type: CTI Link Pro
      Instances (sw/14/5/2/1)
        Instance (sw/14/5/2/1/18)
          LicenseString: CTI Link Pro
          Status: Valid
          Expiry: 4/08/2017 00:00:00 0
          InstancesLicensed: 1
          LicenseSource: File
        InstancesInUse: 1
        InstancesAvailable: 1
    InstalledLicenseType (sw/14/5/3)
      Type: Wave User
      Instances (sw/14/5/3/1)
        Instance (sw/14/5/3/1/19)
          LicenseString: Wave User
          Status: Valid
          Expiry: 4/08/2017 00:00:00 0
          InstancesLicensed: 16
          LicenseSource: File
        InstancesInUse: 0
        InstancesAvailable: 16

```

3.7 Locations

Path=/ws/ssi/locations

LocationList List of Location

```

<xs:element type="xs:string" name="Name"/>
<xs:element type="xs:string" name="SubnetAddress"/>
<xs:element type="xs:string" name="SubnetMask"/>
<xs:element type="xs:string" name="ParentLocation"/>
<xs:element type="xs:long" name="MaxCalls" default="-1"/>
<xs:element type="xs:long" name="CurrentCalls" default="-1"/>
<xs:element type="LocationCongestionList" name="LocationAlarms"/>
<xs:element type="xs:long" name="MaxExternalCalls" default="-1"/>
<xs:element type="xs:long" name="ExternalCalls" default="-1"/>
<xs:element type="xs:long" name="MaxInternalCalls" default="-1"/>
<xs:element type="xs:long" name="InternalCalls" default="-1"/>

```

Locations are used for Emergency Calls and controlling the number of calls to devices (CAC). Extensions can be explicitly assigned (where they have an extension record), or automatically assigned based on IP address. The address part of the Location data is not available in this API, there is a specific API for monitoring Location information for Emergency Calls, and it offers the ability to set a location for an extension - for example by interacting with an intelligent switch.

<https://ipofficekb.avaya.com/knowledgebase/businesspartner/ipoffice/mergedProjects/manager/Location2.html>

3.8 IP 500v2 Internal Modules

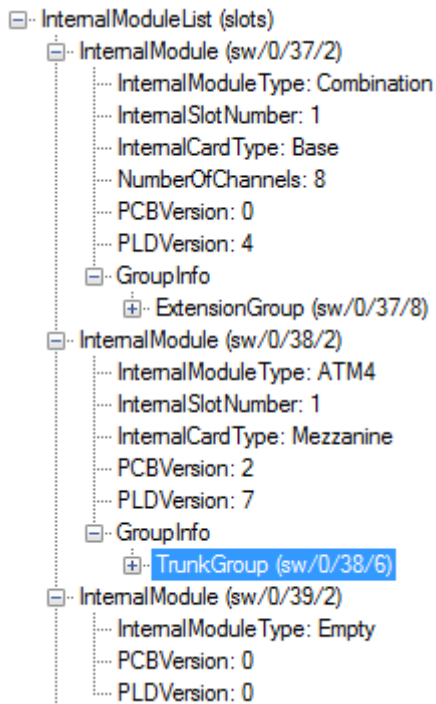
Detail of the Hardware in an IP500V2's slots. Each slot can take a Base card (typically an Extension or VCM) with a Mezzanine Card mounted on it (typically a Trunk card). A Combo card is a Base card with multiple

types of hardware (Digital and Analogue Extensions plus VCM), with a pre-installed trunk card (Analogue or Basic Rate).

3.8.1 Main URL

Path=/ws/ssi/slots

```
<xs:element type="InternalModEnumeration" name="InternalModuleType"/>
<xs:element type="xs:long" name="InternalSlotNumber" default="-1"/>
<xs:element type="CardTypeEnumeration" name="InternalCardType"/>
<xs:element type="xs:long" name="NumberOfChannels" default="-1"/>
<xs:element type="xs:long" name="PCBVersion" default="-1"/>
<xs:element type="xs:long" name="PLDVersion" default="-1"/>
<xs:element type="AppsProcessor" name="AppsProcessorInfo"/>
<xs:element type="SSIGenericObject" name="GroupInfo"/>
```



Extension and Trunk group information is described elsewhere.

3.8.2 Specific Slot URL

Path=/ws/ssi/slots/X

Where X is the slot number 1 to 4 from the left.

3.9 Switch Summary

Path=/ws/ssi/switch

```
<xs:element type="xs:long" name="TotalExtensions" default="-1"/>
<xs:element type="xs:long" name="TotalTrunks" default="-1"/>
<xs:element type="xs:string" name="SwitchName"/>
<xs:element type="xs:string" name="SwitchAddress"/>
<xs:element type="xs:string" name="StartTime"/>
<xs:element type="xs:long" name="SwitchUpTime" default="-1"/>
<xs:element type="RestartReasonEnumeration" name="LastRestartReason"/>
<xs:element type="xs:string" name="RestartedBy"/>
<xs:element type="ModuleTypeEnumeration" name="PlatformType"/>
<xs:element type="xs:string" name="FirmwareVersion"/>
<xs:element type="xs:long" name="UniqueCallCount" default="-1"/>
```



```

Switch (sw)
  TotalExtensions: 11
  TotalTrunks: 5
  SwitchName: 00E007052805
  SwitchAddress: 192.168.42.1
  StartTime: 1/16/2017 14:51:31 0
  SwitchUp Time: 7979687
  Last Restart Reason: UserInitiated
  RestartedBy: Administrator
  Platform Type: IP500V2
  Firmware Version: 10.1.0.0.0 build 33
  UniqueCallCount: 0

```

Unique call count is the count of current calls. For details of time format see Section 2.5.1

3.10 Time

Path=/ws/ssi/time

Details of how the IP Office is getting its time data. For details of time format see Section 2.5.1

```

<xs:element type="xs:string" name="DateTime"/>
<xs:element type="TimeSourceEnumeration" name="Source"/>
<xs:element type="PollResultEnumeration" name="LastPollResult"/>
<xs:element type="xs:string" name="LastPollTime"/>
<xs:element type="xs:string" name="ServerAddress"/>
<xs:element type="xs:long" name="ServerIndex" default="-1"/>
<xs:element type="xs:long" name="CurrentOffset" default="-1"/>
<xs:element type="DaylightSavingsEnumeration" name="DaylightSavings"/>
<xs:element type="TimeServerAlarmEntryList" name="TimeServerAlarms"/>

```

```

TimeServerInfo (sw/34)
  DateTime: 1/16/2017 16:56:31 0
  Source: SNTP
  LastPollResult: Failed
  LastPollTime: 1/16/2017 16:53:05 0 7294518
  ServerIndex: 0
  CurrentOffset: 0
  DaylightSavings: Disabled

```

This example failed as the server address is not configured – so not presented.

3.11 Trunks

3.11.1 Main URL

Path=/ws/ssi/trunks

Lists all trunks on the system. Or they can be queried by trunk number or grouped by type (below).

```

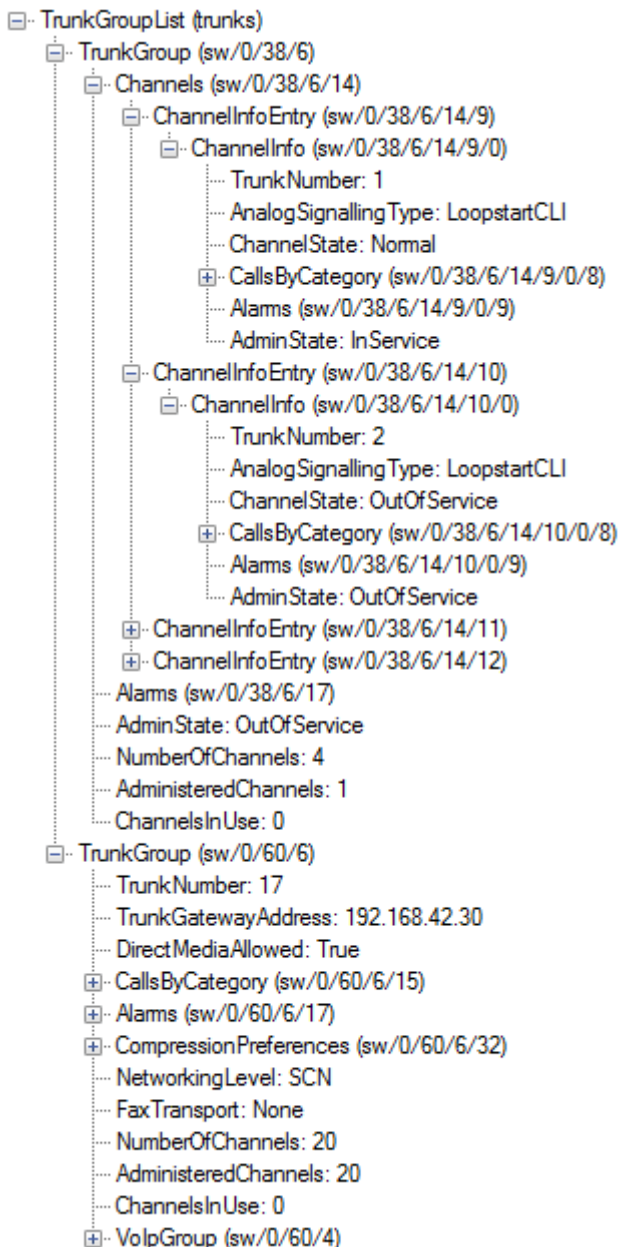
<xs:element type="xs:long" name="TrunkNumber" default="-1"/>
<xs:element type="DigitalTrunkTypeEnum" name="CircuitType"/>
<xs:element type="CCSTypeEnumeration" name="CCSProtocolType"/>
<xs:element type="xs:string" name="TrunkGatewayAddress"/>
<xs:element type="xs:boolean" name="DirectMediaAllowed"/>
<xs:element type="xs:boolean" name="EnableFastStart"/>
<xs:element type="xs:boolean" name="SilenceSuppression"/>
<xs:element type="SCNStatusEnumeration" name="SCNStatus"/>
<xs:element type="ChannelInfoEntryList" name="Channels"/>
<xs:element type="CallsByCategory" name="CallsByCategory"/>
<xs:element type="TrunkStatsTable" name="PerformanceHistory"/>

```

```

<xs:element type="TrunkGroupAlarmEntryList" name="Alarms"/>
<xs:element type="LoopbackAlarm" name="LoopbackAlarm"/>
<xs:element type="DisabledReasonEnumeration" name="TrunkGroupDisabled"/>
<xs:element type="TrunkPrecedenceEnumeration" name="Precedence"/>
<xs:element type="AdminStateEnumeration" name="AdminState"/>
<xs:element type="TrunkCodecInfoList" name="CompressionPreferences"/>
<xs:element type="NetworkingLevelEnumeration" name="NetworkingLevel"/>
<xs:element type="FaxTransportTypeEnumeration" name="FaxTransport"/>
<xs:element type="xs:boolean" name="RemoteClientIPOBehindNAT"/>
<xs:element type="xs:boolean" name="LocalServerIPOBehindNAT"/>
<xs:element type="xs:boolean" name="LocalClientIPOBehindNAT"/>
<xs:element type="xs:string" name="PublicIPAddress"/>
<xs:element type="xs:long" name="NumberOfChannels" default="-1"/>
<xs:element type="xs:long" name="AdministeredChannels" default="-1"/>
<xs:element type="xs:long" name="ChannelsInUse" default="-1"/>
<xs:element type="VoIpGroup" name="VoIpGroup"/>

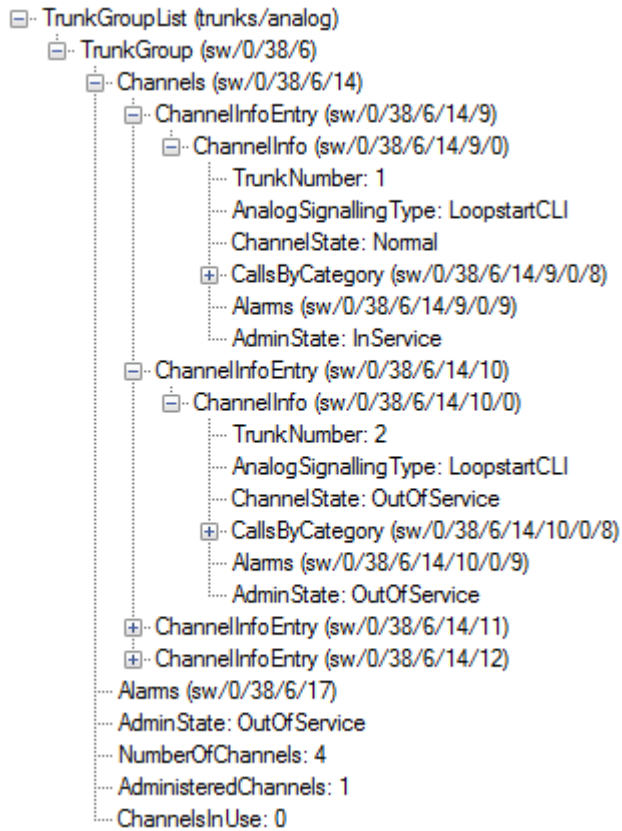
```



3.11.2 Analogue Trunks

Path=/ws/ssi/trunks/analog

Just the Analogue trunks – individual Analogue Trunks are presented as channels, grouped as part of the card (ATM4) or module (ATM16) that they are on.



3.11.3 Digital Trunks

Path=/ws/ssi/trunks/digital

An ETSI E1 trunk for example. Performance History presents the 24 hour error history (in 15 minute blocks).

- [-] TrunkGroup (sw/0/40/7/1/0)
 - TrunkNumber: 5
 - CircuitType: E1
 - CCSPProtocolType: ETSI
 - [-] CallsByCategory (sw/0/40/7/1/0/15)
 - CountStartTime: 2/01/2017 10:12:23 0 36267
 - TrunkNumber: 5
 - [+] UtilizationCache (sw/0/40/7/1/0/15/4)
 - [-] PerformanceHistory (sw/0/40/7/1/0/16)
 - RowCreationInterval: 900
 - OldestRow: 1
 - LastRowCreationTime: 2/01/2017 16:15:02 0 21795323
 - [+] ErrorRows (sw/0/40/7/1/0/16/3)
 - ValidRows: 26
 - [+] Alarms (sw/0/40/7/1/0/17)
 - AdminState: InService
 - NumberOfChannels: 30
 - AdministeredChannels: 30
 - ChannelsInUse: 0

3.11.4 IP Office Trunks

Path=/ws/ssi/trunks/ipoffice

IP Office trunks, typically used for SCN, but including inter IP Office trunks where the Networking level is None. For legacy reasons this will include H323 trunks.

```

TrunkGroup (sw/0/61/6)
  TrunkNumber: 17
  TrunkGatewayAddress: 192.168.42.30
  DirectMediaAllowed: True
  CallsByCategory (sw/0/61/6/15)
    CountStartTime: 2/01/2017 10:12:23 0 36656
    TrunkNumber: 17
    UtilizationCache (sw/0/61/6/15/4)
  Alarms (sw/0/61/6/17)
    TrunkGroupAlarmEntry (sw/0/61/6/17/2)
      AlarmInfo (sw/0/61/6/17/2/0)
        GenericInfo (sw/0/61/6/17/2/0/0)
          LastOccurrenceTime: 2/01/2017 10:12:49 0 62121
          NumberOfOccurrences: 1
          Current: True
          Alarm Type: Trunk out of Service
    CompressionPreferences (sw/0/61/6/32)
      TrunkCodecInfo (sw/0/61/6/32/0)
        Codec: G711A
      TrunkCodecInfo (sw/0/61/6/32/1)
        Codec: G711Mu
      TrunkCodecInfo (sw/0/61/6/32/2)
        Codec: G729A
      TrunkCodecInfo (sw/0/61/6/32/3)
        Codec: G7231
    NetworkingLevel: SCN
    FaxTransport: None
    NumberOfChannels: 20
    AdministeredChannels: 20
    ChannelsInUse: 0
  VolpGroup (sw/0/61/4)
    IpGroup Type: IPOfficeTrunk
    TrunkGatewayAddress: 192.168.42.30
    MediaStream: RTP
    TransportProtocol: WebSocketSecureClient

```

3.11.5 SIP Trunks

Path=/ws/ssi/trunks/sip

```

TrunkGroup (sw/0/63/6)
  TrunkNumber: 20
  CallsByCategory (sw/0/63/6/15)
  Alarms (sw/0/63/6/17)
    AdminState: InService
  CompressionPreferences (sw/0/63/6/32)
    NumberOfChannels: 10
    AdministeredChannels: 10
    ChannelsInUse: 0
  VolpGroup (sw/0/63/4)
    IpGroup Type: SIPTrunk
    TrunkGatewayAddress: 0.0.0.0
    DomainName: sip://0.0.0.0
    DeviceFeatures:
    ResolvedPeerAddress: 0.0.0.0
    MediaStream: RTP
    TransportProtocol: UDP

```

3.11.6 VoIP Trunks

Path=/ws/ssi/trunks/voip

IP Office, H323, and SIP trunks.

3.11.7 Specific trunk

Path=/ws/ssi/trunks/X

Where X is the trunk Line Number as displayed in Manager for example.

3.11.8 Trunk Utilisation Data

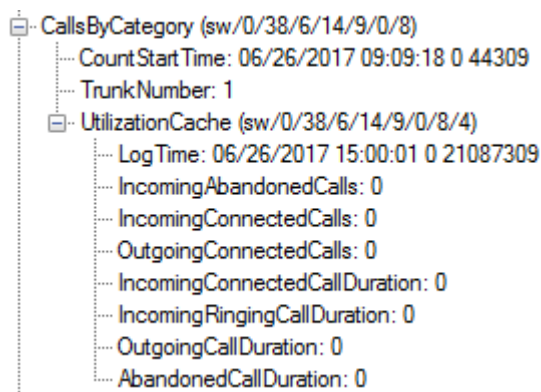
Path=/ws/ssi/trunks/[X or class]/utilization

X=is trunk number or class (analog/digital/etc. as above)

IP Office maintains Utilisation statistics for trunks as can be seen in SSA – this is cumulative since the system was restarted or the values reset on command (SSA only – not through API). Every 15 minutes the current utilisation data is copied to a cache. This request gets the value from the 15 minute updated cache, not the current value queried for SSA, this permits monitoring of a large number of systems and allowing for the time to poll them whilst still maintaining accurate 15 minute utilisation intervals. The 15 minute can be extended up to 24 hours, see Configuration section 2.2, and starts at midnight.

Durations are in seconds and are cumulative; subtract this set from the last set to get the 15 minute utilisation.

Count Start time is the time the/date when the current running totals were reset. Log time is the time when the cache copy is made. The cache copy is overwritten every period and if it hadn't been read the previous value is lost.



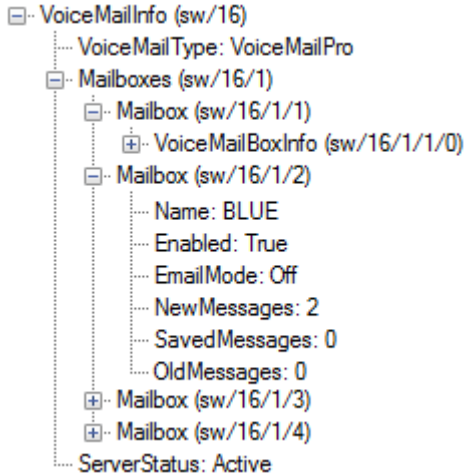
```
<xs:element type="xs:string" name="CountStartTime"/>
<xs:element type="xs:long" name="TrunkNumber" default="-1"/>
<xs:element type="TrunkUtilizationCache" name="UtilizationCache"/>
<xs:element type="xs:string" name="LogTime"/>
<xs:element type="xs:long" name="IncomingAbandonedCalls" default="-1"/>
<xs:element type="xs:long" name="IncomingConnectedCalls" default="-1"/>
<xs:element type="xs:long" name="OutgoingConnectedCalls" default="-1"/>
<xs:element type="xs:long" name="IncomingConnectedCallDuration" default="-1"/>
<xs:element type="xs:long" name="IncomingRingingCallDuration" default="-1"/>
<xs:element type="xs:long" name="OutgoingCallDuration" default="-1"/>
<xs:element type="xs:long" name="AbandonedCallDuration" default="-1"/>
```

3.12 Voicemail

Details of the VoiceMail for this system, Mailbox information is only available for VMPro, Centralised VMPro, and EVM.

3.12.1 Main URL

Path=/ws/ssi/voicemail



```
<xs:element type="VMTypeEnumeration" name="VoiceMailType"/>
<xs:element type="MailboxList" name="Mailboxes"/>
<xs:element type="StorageAlarm" name="StorageSpaceAlarm"/>
<xs:element type="UpdownHistory" name="VMLinkInfo"/>
<xs:element type="UpdownHistory" name="SMTPLinkInfo"/>
<xs:element type="EmbeddedService" name="EmbeddedServiceInfo"/>
<xs:element type="VMStatusEnumeration" name="ServerStatus"/>
<xs:element type="xs:long" name="ActivePorts" default="-1"/>
```

3.12.2 Mailboxes

Path=/ws/ssi/voicemail/mailboxes

MailboxList is a list of VoiceMailBox

```
<xs:element type="xs:string" name="Name"/>
<xs:element type="xs:boolean" name="Enabled"/>
<xs:element type="OffOnEnumeration" name="Broadcast"/>
<xs:element type="VMEEmailModeEnumeration" name="EmailMode"/>
<xs:element type="xs:string" name="EmailAddress"/>
<xs:element type="xs:boolean" name="EmailReading"/>
<xs:element type="xs:long" name="NewMessages" default="-1"/>
<xs:element type="xs:long" name="SavedMessages" default="-1"/>
<xs:element type="xs:long" name="OldMessages" default="-1"/>
```

4 By Numeric Component ID(CID) of the SSA internal interface

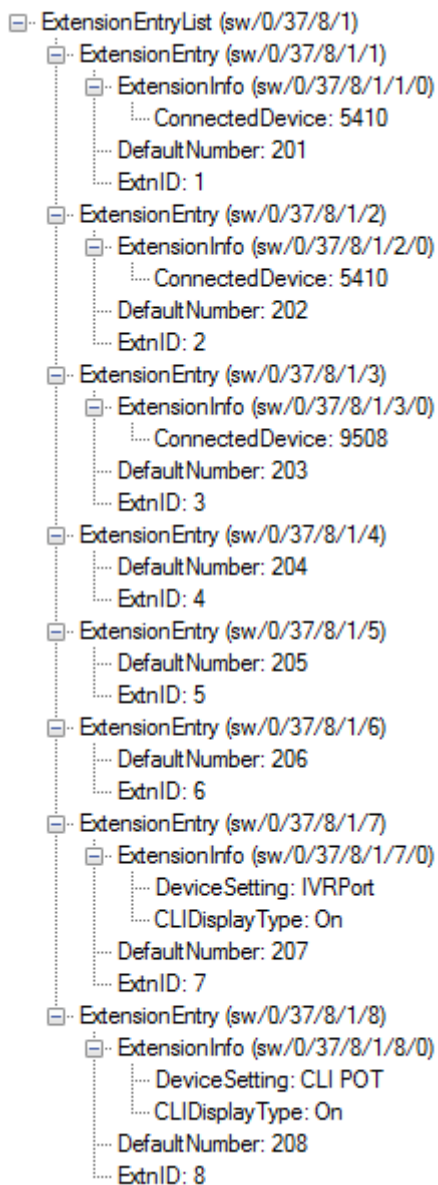
The format of the data is vaguely SNMP like (even though the interface doesn't use SNMP and this data is not available through SNMP). In this format the '/' is used as a component separator no the '.' Of SNMP.

Like SNMP all references are numeric, by Component ID.

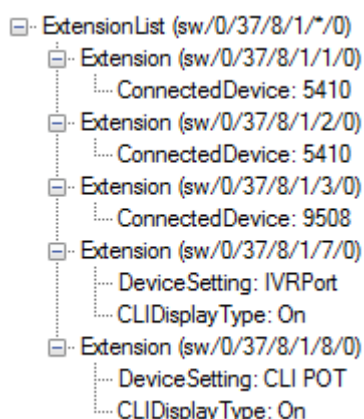
Individual fields cannot be queried by this method, only compound objects or lists. For example sw/1 is visible as an element when GETing "sw" but you cannot query the simple value on its own.

CID access supports wild cards for list index values. For example;

URL `sw/0/37/8/1` will return the detail of the extension ports on Base card 1 – which individually have URLs `sw/0/37/8/1/1` to `sw/0/37/8/1/8`. That could be performed as `sw/0/37/8/1/*`. A maximum of two wild cards may be used in a URL.



Requesting URL `sw/0/37/8/1/*0` will get the Extension Info element which has CID ...0 and only return a row for the objects that have a value



Up to two wildcards can be included in a URL.

This format may be deprecated in the future with preference being given to the named URLs.

4.1 CID URLs

All component ID URLs start “`sw`” and getting just that will obtain the whole tree down to the depth specified.

The components under `sw` are;

CID	Description
<code>sw/0</code>	Port Groups – see later
	Total Extensions
	Total Groups
<code>sw/3</code>	Base Module (Platform) Information
	Switch name
	Switch (LAN1) IP Address
<code>sw/6</code>	System Information – mostly the same as the named URL “system” and “system/resources”, “system/holdmusic”
	Start Time, IP Office not platform for Linux based systems
	Switch Up Time in milliseconds (as above)
<code>sw/10</code>	Alarms – same as named URL “alarms”
<code>sw/14</code>	License Information – same as named URL “licenses”
<code>sw/15</code>	Directory Information – same as named URL “directory”
<code>sw/16</code>	VoiceMail Information – same as the named URL “voicemail/mailboxes”
<code>sw/19</code>	All Calls – Alarms about Calls, and count of active calls – no information on individual calls
<code>sw/25</code>	Last Restart – reason for the last IP Office restart
<code>sw/26</code>	Restarted By
<code>sw/34</code>	Time Server Information – same as named URL “time”
<code>sw/35</code>	Security Information – Alarms and Blacklists
<code>sw/36</code>	Test Alarm
<code>sw/37</code>	Web Services Information – Alarms
<code>sw/39</code>	Location Information -
<code>sw/40</code>	ACCS Information
<code>sw/41</code>	IPDECT Alarms

sw/42	Platform Type (Enumeration)
sw/43	Firmware version
sw/44	Current Call Count

Port Groups is a representation of ‘ports’ on the IP Office – these may be physical TDM (or analogue) ports or IP ‘ports’ representing groups of IP Extensions or Trunks. In the data it is a list of objects indexed from one so the 1st Port Group is sw/0/1.

Under sw/0/i there are the following component IDs – presence of certain components describes the object in detail. A summary is presented below – full detail is available in the XSD.

CID	Description
sw/0/i/0	Expansion Module for IP500V2 Port for Expansion Modules on back panel
sw/0/i/0/0	Module Number – 1-8, or 1-12 if 4 port expansion is fitted.
Sw/0/i/0/1	LoadableModule information – a data structure giving detail of a module plugged in to this port
	No longer used
sw/0/i/2	Internal Modules the slots on the front of IP500V2
sw/0/i/2/2	Slot Number 1-4
sw/0/i/2/3	Card type 2=Base, 3=Mezzanine
sw/0/i/2/4	Number of channels
sw/0/i/2/5	PCB Version
sw/0/i/2/6	PLDVersion
sw/0/i/2/7	UCM card (Apps Processor) Information
sw/0/i/4	VoIP Group
sw/0/i/4/0	VoIP group type – see enumerations
sw/0/i/4/1	Trunk Group Gateway Address
sw/0/i/4/2	Domain Name (SIP or SES trunk)
sw/0/i/4/5	Device Features (SIP Extension)
sw/0/i/4/6	Registration List (SIP Trunk)
sw/0/i/4/7	Resolved Peer Address
sw/0/i/4/8	DNS Resolution Status
sw/0/i/4/9	Media Stream Encapsulation 1=RTP, 2=SRTP
sw/0/i/5	Analogue Trunk Group
sw/0/i/6	Multiple Trunk Group (Analogue or VoIP trunks)
sw/0/i/7	Digital Trunk Module
sw/0/i/7/i	Digital Trunk
sw/0/i/8	Extension Group
sw/0/i/9	Virtual Extension Group
sw/0/i/10	Module Failure Info – Alarms for External Expansion Modules

5 Test/Demo Application

As a simple REST API this can be exercised using widely available tools but we have built a Windows test/demo application to aid with developing against this API.

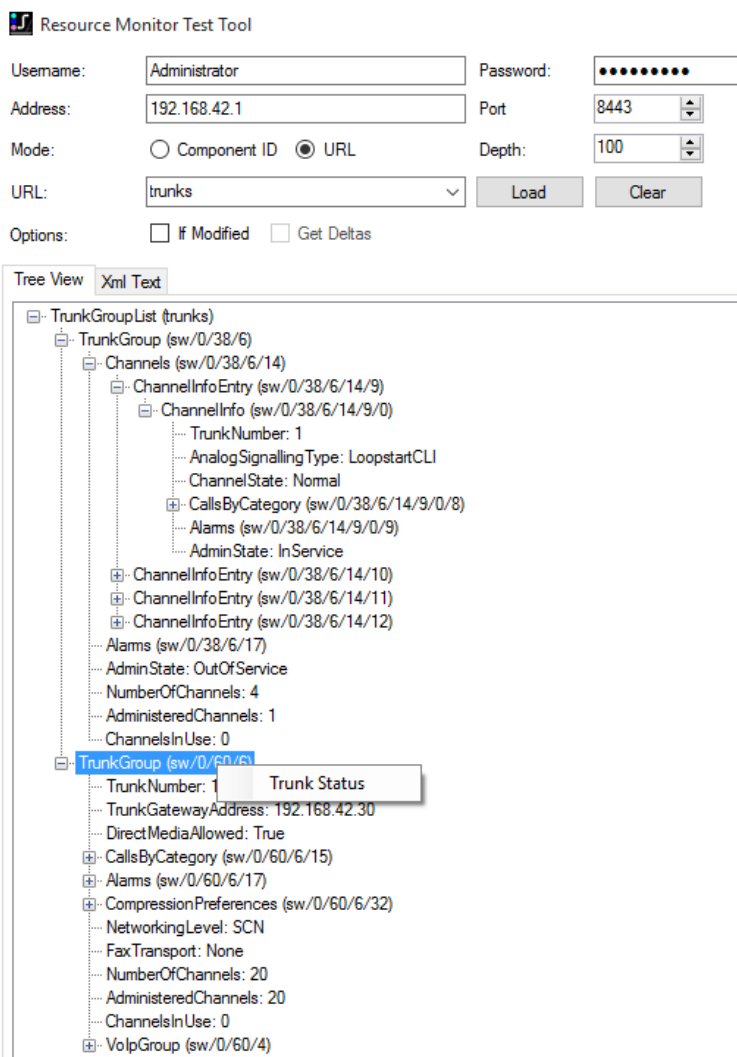
Built in C# (VS2013) with source provided to provide a method to experiment and exercise the API whilst partners are developing their own application. There is no Support for this application and it is not intended for end user use.

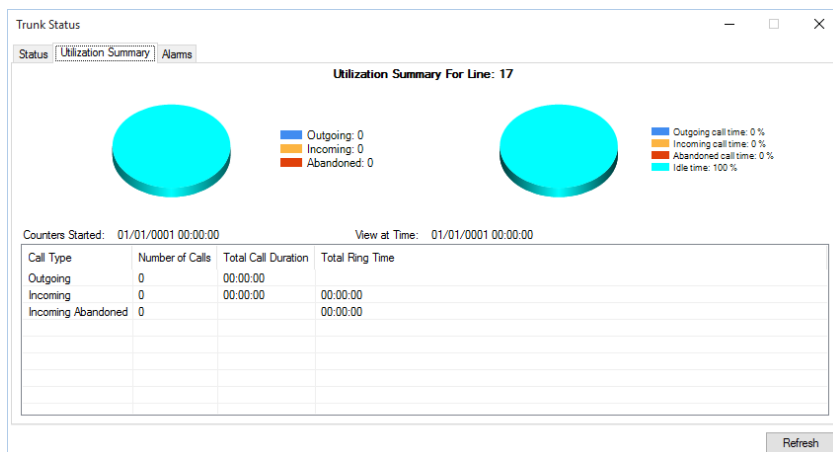
To use the sample application extract the executable, and optionally the XSD, from the SDK ZIP file and drop them in to a temporary folder and execute it from there. The application is dependent on .NET 4.5

The demo application shows the output in XML and as a formatted Tree structure and with some specific visualisations of data as examples. If the XSD is in the same directory as the test app for decoding the Enumerations are rendered as text

RtClick on Trunk Group for demo presentation of status/utilisation – not for analogue trunks.

When If Modified and Get Deltas are both checked the changed fields are highlighted - that is a function of the demo app, the Delta functionality of the API operates at the object level not the individual field level.





6 Selected Enumerations with Notes

Note that as this API is built off an existing internal interface, some values may not actually be possible on this interface – for example the DT type Expansion Modules were discontinued and not supported long before this API was created.

The reference location for all enumerations is the XSD file and that is the one that will be maintained.

For ease of lookup some of the enumerations are reproduced here with added notes.

6.1 System

6.1.1 Internal Card Type

1	Carrier	<i>Obsolete – Legacy Carrier Card</i>
2	Base	
3	Mezzanine	<i>Trunk Card attached to the base card</i>
4	Service	<i>Linux Platform</i>

6.2 Alarm Information

Some alarms are under the more general subject area – Licenses for example.

6.2.1 IP DECT

6.2.1.1 IPDECTAlarmType

1	SecurityViolation	IP DECT System early encryption security violation detected
2	FeatureUnsupported	IP DECT System feature not supported by current firmware – <i>see below</i>
3	ConfigurationError	IP DECT System inconsistent configuration received from remote node
4	LocalOveruse	IP DECT System <u>resiliency</u> not accepted - current node is <u>overused</u>
5	RemoteOveruse	IP DECT System <u>resiliency</u> not accepted - remote node is <u>overused</u>

6.2.1.2 IPDECTFeature

1	Line2Display	Line 2 Display
2	Provisioning	Provisioning
3	EarlyEncryption	Early Encryption
4	CallingPartyName	Calling Party Name
5	PBXInstructedOffHook	Off Hook
6	PBXInstructedOffHookLoudspeaker	Off Hook Loudspeaker
7	StandbyMaster	Standby Master
8	RedundantTrunk	Redundant Trunk

6.2.2 ProcessorAlarmType

Alarm Related to the UCM card

1	NoLicense	Installed UC Module is not licensed on this platform:
2	SoftwareMode	Installed UC Module is invalid on this platform:
3	ExcessCard	Installed UC Module exceeds the limit for this platform:
4	HeartbeatFailure	UC Module <u>heartbeat</u> failure:
5	PowerFailure	UC Module power failure:
6	InternalError	UC Module internal error:

7	RTCFailed	UC Module RTC failure:
8	BIOSError	UC Module BIOS detected an error:
9	FanFailure	UC Module fan failure:
10	ChargeFailure	UC Module failed to charge battery:
11	Storage	UC Module disk usage exceeds 90% of capacity:
12	BootloaderFailed	UC Module <u>bootloader</u> failed:
13	SelfTestFailure	UC Module self-test failure:
14	SSDFull	UC Module SSD card is full:
15	SSDWriteProtected	UC Module SSD card is write-protected:
16	SSDFault	UC Module SSD card fault detected:
17	OSBootFailure	UC Module OS boot failure:
18	USBBootFailure	UC Module USB boot failure:
19	USBUpgradeFailure	UC Module USB upgrade failure:
20	AppsStartFailure	UC Module application failed to start:
21	IPOCommsFailure	UC Module IP Office communication failure:
22	ApplicationFailure	UC Module application failure
23	NoBootDevice	
24	NoUSBBootDevice	
25	CompInstallFailure	
26	FirmwareMismatch	UC Module release version is not aligned with IP Office release version:

6.2.3 ACCSAlarmType

1	AuthenticationFailure	Avaya Contact Center Select data synchronization failed: authentication failure
2	SyncRequestTimeout	Avaya Contact Center Select data synchronization failed: no response from CCMA server
3	DNSResolutionFailure	Avaya Contact Center Select data synchronization failed: could not resolve CCMA server address
4	SIPRegistrationExpiry	Avaya Contact Center Select SIP registration expired

6.2.4 ExpirationAlarm

TLS Certificate Alarm detail

1	IdentityCertificate	Identity certificate
2	TCSCertificate	Trusted store certificate
3	TelephonyCertificate	Telephony certificate

6.3 Extension

6.4 Directory

6.4.1 UpdateResult

1	Failure	
2	SuccessWithOverflow	<i>Too many records received, excess discarded</i>
3	Success	
4	NotConfigured	
5	InvalidConfiguration	
6	InvalidResponse	

7	MemoryLow	
8	InProgress	
9	Pending	

6.5 Licenses

6.5.1 LicenseStatus

1	Valid	
2	Malformed	
3	FeatureKeyMismatch	
4	MACAddressMismatch	<i>Internal Lab licenses only</i>
5	Expired	
6	IncompatibleWithIP500Standard	<i>Select license</i>
7	Dormant	<i>ADI – has dependency on a License that is not present/valid</i>
8	Obsolete	<i>No longer used in this Firmware</i>

6.5.2 LicenseServerAlarmType

1	AllocateLicenseFailed	The license server cannot allocate the requested license instances
2	FreeLicenseFailed	The license server cannot free the requested license instances
3	LicenseExpired	The license server cannot fulfil the request as the license has expired
4	NoFeatureLicense	The feature license is not available on the license server
5	InvalidResponse	Invalid response received from the license server

6.5.3 LicenseErrorType

1	NativeLicensesExceeded	The number of Native users (configured or without extension) is higher than the number of Native Station licenses
2	SurvivableLicensesExceeded	The number of <u>Survivable</u> configured users is higher than the number of <u>Survivable</u> Station licenses
3	NativeMaxExceeded	The maximum number of Native Station licenses is lower than the number in use
4	SurvivableMaxExceeded	The maximum number of <u>Survivable</u> Station licenses is lower than the number in use
5	EVMMMaxExceede	The maximum number of Embedded <u>Voicemail</u> Channels licenses is lower than the number configured
6	VMPProMaxExceeded	The maximum number of <u>Voicemail</u> Pro Channels licenses is lower than the number configured
7	SIPMaxExceeded	The maximum number of Sip Trunk licenses is lower than the number configured
8	NoSMLine	There are <u>Survivable</u> Extensions configured but there is no SM line on the system
9	NoABGLicense	There is no valid <u>Avaya</u> Branch Gateway System Software license
10	NoLicenseFile	There is no license file on system
11	RestrictedMode	License Server Error: System is in License Restricted mode, services are affected
12	T1LicenseExceede	The number of in service T1 channels is higher than Additional T1 licenses
13	E1LicenseExceeded	The number of in service E1 channels is higher than Additional E1 licenses
14	LinuxGraceExpired	IP Office Grace period expired

15	GraceReminder	The <u>entitlement</u> period and virtual trial licenses will expire at the end of the grace period
16	FingerprintChanged	The System ID has changed: You must acquire licenses based on the new System ID
17	PowerUserExceeded	Power User License Exceeded: There are insufficient Power User licenses
18	AvayaIPEndpointExceeded	<u>Avaya IP Endpoint</u> License Exceeded: There are insufficient <u>Avaya IP Endpoint</u> licenses
19	IPEndpointExceeded	3 rd Party IP <u>Endpoint</u> License Exceeded: There are insufficient 3 rd Party IP <u>Endpoint</u> licenses
20	MidsizeEditionExceeded	Server Edition License Exceeded: There are insufficient Server Edition licenses
21	NoSoftwareLicense	No Valid Server Edition License: There is no valid Server Edition license present
22	ServerTimeout	License Server Error: License Server Request Timeout, Invalid/Error Response
23	OfficeWorkerExceeded	Office Worker User License Exceeded: There are insufficient Office Worker User licenses
24	SIPChannelsExceeded	SIP Channel License Exceeded: There are insufficient SIP Channel licenses
25	NoEssentialEdition	No valid Essential Edition license present
26	NoSoftwareUpgrade	No valid Software Upgrade license present
27	LinuxGraceReminder	The old licenses will still be valid for a limited period of time
28	LinuxGraceExpired	The old licenses validity has expired: You must acquire licenses based on the new System ID
29	FirstCallOnSystem	First call made on the system was in this software release
30	GenericExceed	There are insufficient licenses for this configuration.
31	WebLMGraceReminder	Cannot obtain necessary licenses. System is in WebLM Error Mode. The problem must be resolved within the remaining days or else service will be disrupted.
32	NoWebLMModel	License Server Error: The WebLM Model license is not available on the server
33	BusinessEditionUpgradeExceeded	Server Edition License Upgrade Exceeded: Insufficient Server Edition Upgrade licenses to run whole solution at same level as Primary
34	LicConfigurationMode	License Configuration Error: There are insufficient licenses for the current configuration
35	NoWebLMID	Licenses cannot be acquired from the WebLM server: WebLM ID not set
36	LinuxFingerprintChangedWebLM	The System ID has changed
37	CallAnalytics	Call Analytics request license is enabled but no license is present in WebLM

6.5.4 PLDSFileAlarmType

1	NoFile	The license file is not present
2	NoTags	The license file is corrupt and cannot be processed
3	InvalidSignature	The license file has an invalid product name and cannot be processed
4	InvalidName	The license file has an invalid product version and cannot be processed
5	InvalidVersion	The license file does not correspond to the feature key serial number on the SD card

6	InvalidHostId	The license server is not responding
7	NotResponding	The product license is not available on the license server
8	NoProductLicense	The product license is not at the correct release
9	InvalidVersion	System not licensed for Select
10	NoSelectLicense	The license file is not present

6.6 Time

6.6.1 TimeSource

1	Local	<i>IP500V2 internal RTC</i>
2	Voicemail Pro/Manager	<i>Legacy RFC??? TIME protocol</i>
3	SNTP	<i>Use NTP server</i>
4	NTPD	<i>Linux hosts</i>

6.6.2 TimeServerAlarmType

1	ResoulutionFailure	No response from Time Server
2	NoResponse	Failed to resolve Time Server host name

6.7 SysResourceType

Detail of resources – usually hardware based

1	DataChannel	
2	VCMChannel	<i>IP500V2- DSP Hardware based Codec</i>
3	VMChannel	<i>VoiceMail Pro channels</i>
4	ModemChannel	
5	ConferenceChannel	
6	EmbeddedChannel	<i>Embedded VoiceMail</i>
7	HighQualityConferenceChannel	
8	OutdialerVMChannel	<i>Obsolete</i>
9	RTPMediaChannel	<i>Software Based Codecs on Linux platforms</i>

6.8 Memory Card Slot Data

6.8.1 OEMCardSlotAlarmType

1	BootLocation	System not running System SD primary software
2	InvalidSD	Incompatible or invalid SD card fitted:
3	InvalidFeatureKey	No License Feature Key <u>dongle</u>

6.9 CallerIdType

Identifies the source of an Emergency Call

1	TDM
2	IP
3	TelecommuterOrMobility
4	SCNOrTrunk
5	Unknown
6	POTS
7	SIPDECT
8	H323DECT
9	SIP
10	H323

7 References

IP Office Knowledge Base	https://ipofficekb.avaya.com/knowledgebase/businesspartner/index.html
IP Office Manager Manual	https://ipofficekb.avaya.com/knowledgebase/businesspartner/ipoffice/mergedProjects/manager/_frame2.html
System Status Application Manual	https://ipofficekb.avaya.com/knowledgebase/businesspartner/ipoffice/mergedProjects/ssa/index.htm
IP Office Security Guidelines	https://ipofficekb.avaya.com/knowledgebase/businesspartner/ipoffice/mergedProjects/security/index.htm
The 'Basic' HTTP Authentication Scheme	https://tools.ietf.org/html/rfc7617

© 2025 Avaya LLC. All rights reserved.