Oceana Generic Channel

Configuration, API description

CC R+D

07/29/2022

This describes the configuration and API implementation for the Generic Channel feature.

AVAYA SOFTWARE DEVELOPMENT KIT LICENSE AGREEMENT

REVISED: January 14, 2022

READ THIS CAREFULLY BEFORE ELECTRONICALLY ACCESSING OR USING THIS PROPRIETARY PRODUCT!

THIS IS A LEGAL AGREEMENT ("AGREEMENT") BETWEEN YOU, INDIVIDUALLY, AND/OR THE LEGAL ENTITY FOR WHOM YOU ARE OPENING, INSTALLING, DOWNLOADING, COPYING OR OTHERWISE USING THE AVAYA SOFTWARE DEVELOPMENT KIT ("SDK") (COLLECTIVELY, AS REFERENCED HEREIN, "YOU", "YOUR", OR "LICENSEE") AND AVAYA INC. OR ANY AVAYA AFFILIATE (COLLECTIVELY, "AVAYA"). IF YOU ARE ACCEPTING THE TERMS AND CONDITIONS OF THIS AGREEMENT ON BEHALF OF A LEGAL ENTITY, YOU REPRESENT AND WARRANT THAT YOU HAVE FULL LEGAL AUTHORITY TO ACCEPT ON BEHALF OF AND BIND SUCH LEGAL ENTITY TO THIS AGREEMENT. BY OPENING THE MEDIA CONTAINER, BY INSTALLING, DOWNLOADING, COPYING OR OTHERWISE USING THE AVAYA SOFTWARE DEVELOPMENT KIT ("SDK") OR AUTHORIZING OTHERS TO DO SO, YOU SIGNIFY THAT YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS AGREEMENT. IF YOU DO NOT HAVE SUCH AUTHORITY OR DO NOT WISH TO BE BOUND BY THE TERMS OF THIS AGREEMENT, SELECT THE "DECLINE" BUTTON AT THE END OF THE TERMS OF THIS AGREEMENT OR THE EQUIVALENT OPTION AND YOU SHALL HAVE NO RIGHT TO USE THE SDK.

1.0 DEFINITIONS.

1.1 "Affiliates" means any entity that is directly or indirectly controlling, controlled by, or under common control with Avaya Inc. For purposes of this definition, "control" means the power to direct the management and policies of such party, directly or indirectly, whether through ownership of voting securities, by contract or otherwise; and the terms "controlling" and "controlled" have meanings correlative to the foregoing.

1.2 "Avaya Software Development Kit" or "SDK" means Avaya technology, which may include Software, Client Libraries, Specification Documents, Software libraries, application programming interfaces ("API"), Software tools, Sample Application Code and Documentation.

1.3 "Client Libraries" mean any enabler code specifically designated as such and included in a SDK. Client Libraries may also be referred to as "DLLs", and represent elements of the SDK required at runtime to communicate with Avaya products or other SDK elements.

1.4 "Change In Control" shall be deemed to have occurred if any person, entity or group comes to own or control, directly or indirectly, beneficially or of record, voting securities (or any other form of controlling interest) which represent more than fifty percent (50%) of the total voting power of the Licensee.

1.5 "Derivative Work(s)" means any translation (including translation into other computer languages), port, compiling of Source Code into object code, combination with a pre-existing work, modification, correction, addition, extension, upgrade, improvement, compilation, abridgment or other form in which an existing work may be recast, transformed or adapted or which would otherwise constitute a derivative work under the United States Copyright Act. Permitted Modifications will be considered Derivative Works.

1.6 "Documentation" includes programmer guides, CDs, manuals, materials, and information appropriate or necessary for use in connection with the SDK. Documentation may be provided in machine-readable, electronic or hard copy form.

1.7 "Intellectual Property" means any and all: (i) rights associated with works of authorship throughout the world, including copyrights, neighboring rights, moral rights, and mask works, (ii) trademark and trade name rights and similar rights, (iii) trade secret rights, (iv) patents, algorithms, designs and other industrial property rights, (v) all other intellectual and industrial property rights (of every kind and nature throughout the world and however designated) whether arising by operation of law, contract, license, or otherwise, and (vi) all registrations, initial applications, renewals, extensions, continuations, divisions or reissues thereof now or hereafter in force (including any rights in any of the foregoing).

1.8 "Permitted Modification(s)" means Licensee's modifications of the Sample Application Code as needed to create applications, interfaces, workflows or processes for use with Avaya products.

1.9 “Specification Document” means any notes or similar instructions in hard copy or machine readable form, including any technical, interface and/or interoperability specifications that define the requirements and conditions for connection to and/or interoperability with Avaya products, systems and solutions.

1.10 “Source Code” means human readable or high-level statement version of software written in the source language used by programmers and includes one or more programs. Source Code programs may include one or more files, such as user interface markup language (.mxml), action script (.as), precompiled Flash code (.swc), java script (.js), hypertext markup language (.html), active server pages (.asp), C# or C# .Net source code (.cs), java source code (.java), java server pages (.jsp), java archives (.jar), graphic interchange format (.gif), cascading style sheet (.css), audio files (.wav) and extensible markup language (.xml) files.

1.11 “Sample Application Code” means Software provided for the purposes of demonstrating functionality of an Avaya product through the Avaya Software Development Kit.

1.12 “Software” means data or information constituting one or more computer or apparatus programs, including Source Code or in machine-readable, compiled object code form.

2.0 LICENSE GRANT.

2.1 SDK License.

A. Provided Licensee pays to Avaya the applicable license fee (if any), Avaya hereby grants Licensee a limited, non-exclusive, non-transferable license (without the right to sublicense, except as set forth in 2.1B(iii)) under the Intellectual Property of Avaya and, if applicable, its licensors and suppliers to (i) use the SDK solely for the purpose of Licensee's internal development efforts to develop applications, interfaces, value-added services and/or solutions, workflows or processes to work in conjunction with Avaya products; (ii) to package Client Libraries for redistribution with Licensee’s complementary applications that have been developed using this SDK, subject to the terms and conditions set forth herein; (iii) use Specification Documents solely to enable Licensee’s products, services and application solutions to exchange messages and signals with Avaya products, systems and solutions to which the Specification Document(s) apply; (iv) modify and create Derivative Works of the Sample Application Code, Specification Documents and Documentation solely for internal development of applications, interfaces, workflows or processes for use with Avaya products, integration of such applications, interfaces, workflows and processes with Avaya products and interoperability testing of the foregoing with Avaya products; and (v) compile or otherwise prepare for distribution the Sample Application Code with Permitted Modifications, into an object code or other machine-readable program format for distribution and distribute the same subject to the conditions set forth in Section 2.1B.

B. The foregoing license to use Sample Application Code is contingent upon the following: (i) Licensee must ensure that the modifications made to the Sample Application Code as permitted in clause (iv) of Section 2.1A are compatible and/or interoperable with Avaya products and/or integrated therewith, (ii) Licensee may distribute Licensee’s application that has been created using this SDK, provided that such distribution is subject to an end user pursuant to Licensee’s current end user license agreement (“Licensee EULA”) that is consistent with the terms of this Agreement and, if applicable, any other agreement with Avaya (e.g., the Avaya DevConnect Program Agreement), and is equally as protective as Licensee’s standard software license terms, but in no event shall the standard of care be less than a reasonable degree of care, and (iii) Licensee ensures that each end user who receives Client Libraries or Sample Application Code with Permitted Modifications has all necessary licenses for all underlying Avaya products associated with such Client Libraries or Sample Application Code.

Your Licensee EULA must include terms concerning restrictions on use, protection of proprietary rights, disclaimer of warranties, and limitations of liability. You must ensure that Your End Users using applications, interfaces, value-added services and/or solutions, workflows or processes that incorporate the API, Client Libraries, Sample Code or Permitted Modifications adhere to these terms, and You agree to notify Avaya promptly if You become aware of any breach of the terms of Licensee EULA that may impact Avaya. You will take all reasonable precautions to prevent unauthorized access to or use of the SDK and notify Avaya promptly of any such unauthorized access or use.

C. Licensee acknowledges and agrees that it is licensed to use the SDK only in connection with Avaya products (and if applicable, in connection with services provided by or on behalf of Avaya).

D. With respect to Software that contains elements provided by third party suppliers, Licensee may install and use the Software in accordance with the terms and conditions of the applicable license agreements, such as “shrinkwrap” or “click-through” licenses, accompanying or applicable to the Software.

2.2 No Standalone Product. Nothing in this Agreement authorizes or grants Licensee any rights to distribute or otherwise make available to a third party the SDK, in whole or in part, or any Derivative Work in source or object code format on a standalone basis other than the modifications permitted in Section 2.1B of this Agreement.

2.3 Proprietary Notices. Licensee shall not remove any copyright, trade mark or other proprietary notices incorporated in the copies of the SDK, Sample Application Code and redistributable files in Licensee’s possession or control or any modifications thereto. Redistributions in binary form or other suitable program format for distribution, to the extent expressly permitted, must also reproduce Avaya’s copyright, trademarks or other proprietary notices as incorporated in the SDK in any associated Documentation or “splash screens” that display Licensee copyright notices.

2.4 Third-Party Components. You acknowledge certain software programs or portions thereof included in the SDK 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 SDK (“Third Party Terms”). Information identifying the copyright holders of the Third Party Components and the Third Party Terms that apply is available in the attached Schedule 1 (if any), SDK, Documentation, or on Avaya’s web site at: http://support.avaya.com/Copyright (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 this Agreement, 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 this Agreement, solely with respect to the applicable Third Party Components, to the extent that this Agreement imposes greater restrictions on You than the applicable Third Party Terms. Licensee is solely responsible for procuring any necessary licenses for Third Party Components, including payment of licensing royalties or other amounts to third parties, for the use thereof.

2.5 Copies of SDK. Licensee may copy the SDK only as necessary to exercise its rights hereunder.

2.6a No Reverse Engineering. Licensee shall have no rights to any Source Code for any of the software in the SDK, except for the explicit rights to use the Source Code as provided to Licensee hereunder. Licensee agrees that it shall not cause or permit the disassembly, decompilation or reverse engineering of the Software. Notwithstanding the foregoing, if the SDK is rightfully located in a member state of the European Union and Licensee needs information about the Software in the SDK in order to achieve interoperability of an independently created software program with the Software in the SDK, Licensee will first request such information from Avaya. Avaya may charge Licensee a reasonable fee for the provision of such information. If Avaya refuses to make such information available, then Licensee may take steps, such as reverse assembly or reverse compilation, to the extent necessary solely in order to achieve interoperability of the Software in the SDK with an independently created software program. To the extent that the Licensee is expressly permitted by applicable mandatory law to undertake any of the activities listed in this section, Licensee will not exercise those rights until Licensee has given Avaya twenty (20) days written notice of its intent to exercise any such rights.

2.6.b License Restrictions. To the extent permissible under applicable law, Licensee agrees not to: (i) publish, sell, sublicense, lease, rent, loan, assign, convey or otherwise transfer the SDK; (ii) distribute, disclose or allow use the SDK, in any format, through any timesharing service, service bureau, network or by any other means; (iii) distribute or otherwise use the Software in the SDK in any manner that causes any portion of the Software that is not already subject to an OSS License to become subject to the terms of any OSS License; (iv) link the Source Code for any of the software in the SDK with any software licensed under the Affero General Public License (Affero GPL) v.3 or similar licenses; (v) access information that is solely available to root administrators of the Avaya products, systems, and solutions; (vi) develop applications, interfaces, value-added services and/or solutions, workflows or processes that causes adverse effects to Avaya and third-party products, services, solutions, such as, but not limited to, poor performance, software crashes and cessation of their proper functions; and (vii) develop applications, interfaces, value-added services and/or solutions, workflows or processes that blocks or delays emergency calls; (viii) emulate an Avaya SIP endpoint by form or user interface design confusingly similar as an Avaya product ; (ix) reverse engineer Avaya SIP protocol messages; or (x) permit or encourage any third party to do any of (i) through (x), inclusive, above.

2.7 Responsibility for Development Tools. Licensee acknowledges that effective utilization of the SDK may require the use of a development tool, compiler and other software and technology of third parties, which may be incorporated in the SDK pursuant to Section 2.4. Licensee is solely responsible for procuring such third party software and technology and the necessary licenses, including payment of licensing royalties or other amounts to third parties, for the use thereof.

2.8 U.S. Government End Users. The SDK shall be classified as "commercial computer software" and the Documentation is classified as "commercial computer software documentation" or "commercial items," pursuant to FAR 12.212 or DFAR 227.7202, as applicable. Any use, modification, reproduction, release, performance, display or disclosure of the SDK or Documentation by the Government of the United States shall be governed solely by the terms of the Agreement and shall be prohibited except to the extent expressly permitted by the terms of the Agreement.

2.9 Limitation of Rights. No right is granted to Licensee to sublicense its rights hereunder. All rights not expressly granted are reserved by Avaya or its licensors or suppliers and, except as expressly set forth herein, no license is granted by Avaya or its licensors or suppliers under this Agreement directly, by implication, estoppel or otherwise, under any Intellectual Property right of Avaya or its licensors or suppliers. Nothing herein shall be deemed to authorize Licensee to use Avaya's trademarks or trade names in Licensee's advertising, marketing, promotional, sales or related materials.

2.10 Independent Development.

2.10.1 Licensee understands and agrees that Avaya, Affiliates, or Avaya's licensees or suppliers may acquire, license, develop for itself or have others develop for it, and market and/or distribute applications, interfaces, value-added services and/or solutions, workflows or processes similar to that which Licensee may develop. Nothing in this Agreement shall restrict or limit the rights of Avaya, Affiliates, or Avaya's licensees or suppliers to commence or continue with the development or distribution of such applications, interfaces, value-added services and/or solutions, workflows or processes.

2.10.2 Nonassertion by Licensee. Licensee agrees not to assert any Intellectual Property related to the SDK or applications, interfaces, value-added services and/or solutions, workflows or processes developed using the SDK against Avaya, Affiliates, Avaya's licensors or suppliers, distributors, customers, or other licensees of the SDK.

2.11 Feedback and Support. Licensee agrees to provide any information, comments, problem reports, enhancement requests and suggestions regarding the performance of the SDK (collectively, "Feedback") via any public or private support mechanism, forum or process otherwise indicated by Avaya. Avaya monitors applicable mechanisms, forums, or processes but is under no obligation to implement any of Feedback, or be required to respond to any questions asked via the applicable mechanism, forum, or process. Licensee hereby assigns to Avaya all right, title, and interest in and to Feedback provided to Avaya.

2.12(a) Fees and Taxes. To the extent that fees are associated with the license of the SDK, Licensee agrees to pay to Avaya or pay directly to the applicable government or taxing authority, if requested by Avaya, all taxes and charges, including without limitation, penalties and interest, which may be imposed by any federal, state or local governmental or taxing authority arising hereunder excluding, however, all taxes computed upon Avaya's net income. If You move any Software, including the SDK, and as a result of such move, a jurisdiction imposes a duty, tax, levy or fee (including withholding taxes, fees, customs or other duties for the import and export of any such Software), then You are solely liable for, and agree to pay, any such duty, taxes, levy or other fees.

2.12(b) Audit. Avaya shall have the right, at its cost and expense, to inspect and/or audit (i) by remote polling or other reasonable electronic means at any time and (ii) in person during normal business hours and with reasonable notice Licensee's books, records, and accounts, to determine Licensee's compliance with this Agreement. In the event such inspection or audit uncovers non-compliance with this Agreement, then without prejudice to Avaya's termination rights hereunder, Licensee shall promptly pay Avaya any applicable license fees. Licensee agrees to keep a current record of the location of the SDK.

2.13 No Endorsement. Neither the name Avaya, Affiliates nor the names of contributors may be used to endorse or promote products derived from the Avaya SDK without specific prior written permission from Avaya.

2.14 High Risk Activities. The Avaya SDK is not fault-tolerant, and is not designed, manufactured or intended for use or resale as on-line control equipment or in hazardous environments requiring failsafe performance, such as in the operation of nuclear facilities, aircraft navigation or aircraft communications systems, mass transit, air traffic control, medical or direct life support machines, dedicated emergency call handling systems or weapons systems, in which the failure of the Avaya SDK could lead directly to death, personal injury, or severe physical or environmental damage ("high risk activities"). If Licensee uses the Avaya SDK for high risk activities, Licensee does so at Licensee's own risk and Licensee assumes all responsibility and liability for such use to the maximum extent such limitation or exclusion is permitted by applicable law. Licensee agrees that Avaya and its suppliers will not be liable for any claims or damages arising from or related to use of the Avaya SDK for high risk activities to the maximum extent such limitation or exclusion is permitted by law.

2.15 No Virus. Licensee warrants that (i) the applications, interfaces, value-added services and/or solutions, workflows or processes Licensee develops using this SDK will not contain any computer program file that includes time code limitations, disabling devices, or any other mechanism which will prevent the Avaya product (including other software, firmware, hardware), services and networks from being functional at all times (collectively “Time Bombs”); and (ii) the applications, interfaces, value-added services and/or solutions, workflows or processes Licensee develops using this SDK will be free of computer viruses, malicious or other harmful code, black boxes, malware, trapdoors, and other mechanisms which could: a) damage, destroy or adversely affect Avaya product, or services and/or end users; b) allow remote/hidden attacks or access through unauthorized computerized command and control; c) spy (network sniffers, keyloggers), and d) damage or erase such applications, interfaces, value-added services and/or solutions, workflows or processes developed using this SDK or data, or any computer files or systems of Avaya, Affiliates, and/or end users (collectively “Virus”). In addition to any other remedies permitted in the Agreement, if Licensee breaches its warranties under this Section, Licensee will, at its expense, take remedial action to eliminate any Time Bombs and/or Viruses and prevent re-occurrence (including implementing appropriate processes to prevent further occurrences) as well as provide prompt, reasonable assistance to Avaya to materially reduce the effects of the Time Bomb and/or Virus.

2.16 Disclaimer. Any software security feature is not a guaranty against malicious code, deleterious routines, and other techniques and tools employed by computer “hackers” and other third parties to create security exposures. Compromised passwords represent a major security risk. Avaya encourages You to create strong passwords using three different character types, change Your password regularly and refrain from using the same password regularly. You must treat such information as confidential. You agree to notify Avaya immediately upon becoming aware of any unauthorized use or breach of Your user name, password, account, API Key, or other credentials as provided by Avaya for use of the SDK, or subscription. You are responsible for ensuring that Your networks and systems are adequately secured against unauthorized intrusion or attack and regularly back up of Your data and files in accordance with good computing practices.

2.17 Third Party Licensed Software

A. “Commercial Third Party Licensed Software” is software developed by a business with the purpose of making money from the use of that licensed software. “Freeware Licensed Software” is software which is made available for use, free of charge and for an unlimited time, but is not Open Source Licensed Software. “Open Source Software" or "OSS" is as defined by the Open Source Initiative (“OSI”) https://opensource.org/osd and is software licensed under an OSI approved license as set forth at https://opensource.org/licenses/alphabetical (or such successor site as designated by OSI). These are collectively referred to herein as “Third Party Licensed Software”.

B. Licensee represents and warrants that Licensee, including any employee, contractor, subcontractor, or consultant engaged by Licensee, is to the Licensee’s knowledge, in compliance and will continue to comply with all license obligations for Third Party Licensed Software used in the Licensee application created using the SDK including providing to end users all information required by such licenses as may be necessary. LICENSEE REPRESENTS AND WARRANTS THAT, TO THE LICENSEE’S KNOWLEDGE, THE OPEN SOURCE LICENSED SOFTWARE EMBEDDED IN OR PROVIDED WITH LICENSEE APPLICATION OR SERVICES DOES NOT INCLUDE ANY OPEN SOURCE LICENSED SOFTWARE CONTAINING TERMS REQUIRING ANY INTELLECTUAL PROPERTY OWNED OR LICENSED BY AVAYA OR END USERS TO BE (A) DISCLOSED OR DISTRIBUTED IN SOURCE CODE OR OBJECT CODE FORM; (B) LICENSED FOR THE PURPOSE OF MAKING DERIVATIVE WORKS; OR (C) REDISTRIBUTABLE ON TERMS AND CONDITION NOT AGREED UPON BY AVAYA OR END USERS.

C. Subject to any confidentiality obligations, trade secret or other rights or claims of Licensee suppliers, Licensee will respond to requests from Avaya or end users relating to Third Party Licensed Software associated with Licensee's use of Third Party Licensed Software. Licensee will cooperate in good faith by furnishing the relevant information to Avaya or end users and the requester within two (2) weeks from the time Avaya or end user provided the request to Licensee.

3. OWNERSHIP.

3.1 As between Avaya and Licensee, Avaya or its licensors or suppliers shall own and retain all Intellectual Property rights, in and to the SDK and any corrections, bug fixes, enhancements, updates, improvements, or modifications thereto and Licensee hereby irrevocably transfers, conveys and assigns to Avaya, its licensors and its suppliers all of its right, title, and interest therein. Avaya or its licensors or suppliers shall have the exclusive right to apply for or register any patents, mask work rights, copyrights, and such other proprietary protections with respect thereto. Licensee acknowledges that the license granted under this Agreement does not provide Licensee with title or ownership to the SDK, but only a right of limited use under the terms and conditions of this Agreement.

3.2 Grant Back License to Avaya. Licensee hereby grants to Avaya an irrevocable, perpetual, non-exclusive, sublicensable, royalty-free, fully paid up, worldwide license under any and all of Licensee's Intellectual Property rights related to any Permitted Modifications, to (i) use, make, sell, execute, adapt, translate, reproduce, display, perform, prepare derivative works based upon, distribute (internally and externally) and sublicense the Permitted Modifications and their derivative works, and (ii) sublicense others to do any, some, or all of the foregoing.

4.0 SUPPORT.

4.1 No Avaya Support. Avaya will not provide any support for the SDK provided under this Agreement or for any Derivative Works, including, without limitation, modifications to the Source Code or applications built by Licensee using the SDK. Avaya shall have no obligation to provide support for the use of the SDK, or Licensee's application, services or solutions which may or may not include redistributable Client Libraries or Sample Application Code, to any third party to whom Licensee delivers such applications, services or solutions. Avaya further will not provide fixes, patches or repairs for any defects that might exist in the SDK or the Sample Application Code provided under this Agreement. In the event that Licensee desires support services for the SDK, and, provided that Avaya offers such support services (in its sole discretion), Licensee will be required to enter into an Avaya DevConnect Program Agreement or other support agreement with Avaya.

4.2 Licensee Obligations. Licensee acknowledges and agrees that it is solely responsible for developing and supporting any applications, interfaces, value-added services and/or solutions, workflows or processes developed under this Agreement, including but not limited to (i) developing, testing and deploying such applications, interfaces, value-added services and/or solutions, workflows or processes; (ii) configuring such applications, interfaces, value-added services and/or solutions, workflows or processes to interface and communicate properly with Avaya products; and (iii) updating and maintaining such applications, interfaces, value-added services and/or solutions, workflows or processes as necessary for continued use with the same or different versions of end user and/or third party licensor products, and Avaya products.

5.0 CONFIDENTIALITY.

5.1 Protection of Confidential Information. Licensee acknowledges and agrees that the SDK and any other Avaya technical information obtained by it under this Agreement (collectively, "Confidential Information") is confidential information of Avaya. Licensee shall take all reasonable measures to maintain the confidentiality of the Confidential Information. Licensee further agrees at all times to protect and preserve the SDK in strict confidence in perpetuity, and shall not use such Confidential Information other than as expressly authorized by Avaya under this Agreement, nor shall Licensee disclose any Confidential Information to third parties without Avaya's written consent. Licensee further agrees to immediately 1) cease all use of all Confidential Information (including copies thereof) in Licensee's possession, custody, or control; 2) stop reproducing or distributing the Confidential Information; and 3) destroy the Confidential Information in Licensee's possession or under its control, including Confidential Information on its computers, disks, and other digital storage devices upon termination of this Agreement at any time and for any reason. Upon request, Licensee will certify in writing its compliance with this Section. The obligations of confidentiality shall not apply to information which (a) has entered the public domain except where such entry is the result of Licensee's breach of this Agreement; (b) prior to disclosure hereunder was already rightfully in Licensee's possession; (c) subsequent to disclosure hereunder is obtained by Licensee on a non-confidential basis from a third party who has the right to disclose such information to the Licensee; (d) is required to be disclosed pursuant to a court order, so long as Avaya is given adequate notice and the ability to challenge such required disclosure.

5.2 Press Releases. Any press release or publication regarding this Agreement is subject to prior written approval of Avaya.

6.0 NO WARRANTY.

The SDK and Documentation are provided "AS-IS" without any warranty whatsoever. AVAYA SPECIFICALLY AND EXPRESSLY DISCLAIMS ANY WARRANTIES OR CONDITIONS, STATUTORY OR OTHERWISE, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT AND SATISFACTORY QUALITY. AVAYA DOES NOT WARRANT THAT THE SDK AND DOCUMENTATION ARE SUITABLE FOR LICENSEE'S USE, THAT THE SDK OR DOCUMENTATION ARE WITHOUT DEFECT OR ERROR, THAT OPERATION WILL BE UNINTERRUPTED, OR THAT DEFECTS WILL BE CORRECTED. FURTHER, AVAYA MAKES NO WARRANTY REGARDING THE RESULTS OF THE USE OF THE SDK AND DOCUMENTATION. NEITHER AVAYA NOR ITS SUPPLIERS MAKE ANY WARRANTY, EXPRESS OR IMPLIED, THAT THE SDK OR DOCUMENTATION IS SECURE, SECURITY THREATS AND VULNERABILITIES WILL BE DETECTED OR SOFTWARE WILL RENDER AN END USER'S OR LICENSEE'S NETWORK OR PARTICULAR NETWORK ELEMENTS SAFE FROM INTRUSIONS AND OTHER SECURITY BREACHES.

7.0 CONSEQUENTIAL DAMAGES WAIVER.

EXCEPT FOR PERSONAL INJURY CLAIMS, AVAYA SHALL NOT BE LIABLE FOR ANY INCIDENTAL, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH, ARISING OUT OF OR RELATING TO THIS AGREEMENT OR USE OF THE SDK, OR FOR THE LOSS OR CORRUPTION OF DATA, INFORMATION OF ANY KIND, BUSINESS, PROFITS, OR OTHER COMMERCIAL LOSS, HOWEVER CAUSED, AND WHETHER OR NOT AVAYA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

8.0 LIMITATION OF LIABILITY.

EXCEPT FOR PERSONAL INJURY CLAIMS, IN NO EVENT SHALL AVAYA'S TOTAL LIABILITY TO LICENSEE IN CONNECTION WITH, ARISING OUT OF OR RELATING TO THIS AGREEMENT EXCEED FIVE HUNDRED DOLLARS ($500). THE PARTIES AGREE THAT THE LIMITATIONS SPECIFIED IN THIS SECTION WILL APPLY EVEN IF ANY LIMITED REMEDY PROVIDED IN THIS AGREEMENT IS FOUND TO HAVE FAILED OF ITS ESSENTIAL PURPOSE.

9.0 INDEMNIFICATION.

Licensee shall defend, indemnify and hold harmless Avaya, Affiliates and their respective officers, directors, agents, suppliers, customers and employees “Indemnified Parties”) from and against all claims, demand, suit, actions or proceedings (“Claims”) and damages, losses, liabilities, costs, expenses, and fees (including fees of attorneys and other professionals) (“Damages”) based upon an allegation pertaining to wrongful use, misappropriation, or infringement of a third party’s Intellectual Property right arising from or relating to Licensee’s use of the SDK, alone or in combination with other software, such as operating systems and codecs, and the, direct or indirect, use, distribution or sale of any software, Derivative Works or other products (including but not limited to applications, interfaces, and application programming interfaces) developed utilizing the SDK.

Licensee shall defend, indemnify and hold harmless the Indemnified Parties from and against all Claims and Damages arising out of or related to: (i) personal injury (including death); (ii) damage to any person or tangible property caused, or alleged to be caused by Licensee or Licensee’s application created by using the SDK; (iii) the failure by Licensee or Licensee’s application created by using the SDK to comply with the terms of this Agreement or any applicable laws; (iv) the breach of any representation, or warranty made by Licensee herein; or (v) Licensee’s breach of any obligation under the Licensee EULA.

10.0 TERM AND TERMINATION.

10.1 This Agreement will continue through December 31st of the current calendar year. The Agreement will automatically renew for one (1) year terms, unless terminated as specified in Section 10.2 or 10.3 below.

10.2 Either party shall have the right to terminate the Agreement, upon thirty (30) days written notice to the other party.

10.3 Notwithstanding language to the contrary, Avaya may terminate this Agreement immediately, upon written notice to Licensee for breach of Section 2 (License Grant), Section 5 (Confidentiality) or Section 12 (Compliance with Laws). Avaya may also terminate this Agreement immediately by giving written notice if a Change In Control should occur or if Licensee becomes insolvent, or voluntary or involuntary proceedings by or against Licensee are instituted in bankruptcy or under any insolvency law, or a receiver or custodian is appointed for Licensee, or proceedings are instituted by or against Licensee for corporate reorganization or the dissolution of Licensee, which proceedings, if involuntary, have not been dismissed within thirty (30) days after the date of filing, or Licensee makes an assignment for the benefit of its creditors, or substantially all of the assets of Licensee are seized or attached and not released within sixty (60) days thereafter, or if Licensee has ceased or threatened to cease to do business in the regular course.

10.4 Upon termination or earlier termination of this Agreement, Licensee will immediately cease a) all uses of the Confidential Information; b) Licensee agrees to destroy all adaptations or copies of the Confidential Information stored in any tangible medium including any document or work containing or derived (in whole or in part) from the Confidential Information, and certify its destruction to Avaya upon termination of this License. Licensee will promptly cease use of, distribution and sales of Licensee products that embody any such Confidential Information, and destroy all Confidential Information belonging to Avaya as well as any materials that embody any such Confidential Information. All licenses granted will terminate.

10.5 The rights and obligations of the parties contained in Sections 2.3, 2.6, 2.7, 2.10, 2.11, 2.12, 3, and 5 through 17 shall survive any expiration or termination of this Agreement.

11.0 ASSIGNMENT.

Avaya may assign all or any part of its rights and obligations hereunder. Licensee may not assign this Agreement or any interest or rights granted hereunder to any third party without the prior written consent of Avaya. The term "assign" includes, but is not limited to, any transaction in which there is a Change In Control or reorganization of Licensee pursuant to a merger, sale of assets or stock. This Agreement shall terminate immediately upon occurrence of any prohibited assignment.

12.0 COMPLIANCE WITH LAWS AND IMPORT/EXPORT CONTROL.

Licensee shall comply with all applicable laws and regulations, including without limitation those applicable to data privacy, intellectual property, trade secret, and fraud. Licensee is advised that the Technical Information is of U.S. origin and subject to the U.S. Export Administration Regulations (“EAR”) and may be subject to applicable local country import/export laws and regulations. Diversion contrary to U.S. and/or applicable local country law and/or regulation is prohibited. Licensee agrees not to directly or indirectly export, re-export, import, download, or transmit the Technical Information to any country, end user or for any use that is contrary to applicable U.S. and/or local country regulation or statute (including but not limited to those countries embargoed by the U.S. government). Licensee represents that any governmental agency has not issued sanctions against Licensee or otherwise suspended, revoked or denied Licensee's import/export privileges. Licensee agrees not to use or transfer the Technical Information for any use relating to nuclear, chemical or biological weapons, or missile technology, unless authorized by the U.S. and/or any applicable local government by regulation or specific written license. Additionally, Licensee is advised that the Technical Information may contain encryption algorithm or source code that may not be exported to government or military end users without a license issued by the U.S. Bureau of Industry and Security and any other country’s governmental agencies, where applicable.

13.0 WAIVER.

The failure to assert any rights under this Agreement, including, but not limited to, the right to terminate in the event of breach or default, will not be deemed to constitute a waiver of the right to enforce each and every provision of this Agreement in accordance with their terms.

14.0 SEVERABILITY.

If any provision of this Agreement is determined to be unenforceable or invalid, this Agreement will not be rendered unenforceable or invalid as a whole, and the provision will be changed and interpreted so as to best accomplish the objectives of the original provision within the limits of applicable law.

15.0GOVERNING LAW AND DISPUTE RESOLUTION.

15.1 Governing Law. This Agreement and any dispute, claim or controversy arising out of or relating to this Agreement (“Dispute”), including without limitation the formation, interpretation, breach or termination of this Agreement, or any issue regarding whether a Dispute is subject to arbitration under this Agreement, will be governed by New York State laws, excluding conflict of law principles, and the United Nations Convention on Contracts for the International Sale of Goods.

15.2 **Dispute Resolution.** Any Dispute will be resolved in accordance with the provisions of this Section 15. The disputing party shall give the other party written notice of the Dispute in accordance with the notice provision of this Agreement. The parties will attempt in good faith to resolve each controversy or claim within 30 days, or such other longer period as the parties may mutually agree, following the delivery of such notice, by negotiations between designated representatives of the parties who have dispute resolution authority.

15.3 **Arbitration of Non-US Disputes**. If a Dispute that arose anywhere other than in the United States or is based upon an alleged breach committed anywhere other than in the United States cannot be settled under the procedures and within the timeframe set forth in Section 15.2, it will be conclusively determined upon request of either party by a final and binding arbitration proceeding to be held in accordance with the Rules of Arbitration of the International Chamber of Commerce by a single arbitrator appointed by the parties or (failing agreement) by an arbitrator appointed by the President of the International Chamber of Commerce (from time to time), except that if the aggregate claims, cross claims and counterclaims by any one party against the other party exceed One Million US Dollars at the time all claims, including cross claims and counterclaims are filed, the proceeding will be held in accordance with the Rules of Arbitration of the International Chamber of Commerce by a panel of three arbitrator(s) appointed in accordance with the Rules of Arbitration of the International Chamber of Commerce. The arbitration will be conducted in the English language, at a location agreed by the parties or (failing agreement) ordered by the arbitrator(s). The arbitrator(s) will have authority only to award compensatory damages within the scope of the limitations of Section 8 and will not award punitive or exemplary damages. The arbitrator(s) will not have the authority to limit, expand or otherwise modify the terms of this Agreement. The ruling by the arbitrator(s)) will be final and binding on the parties and may be entered in any court having jurisdiction over the parties or any of their assets. The parties will evenly split the cost of the arbitrator(s)' fees, but Avaya and Customer will each bear its own attorneys' fees and other costs associated with the arbitration. The parties, their representatives, other participants and the arbitrator(s) will hold the existence, content and results of the arbitration in strict confidence to the fullest extent permitted by law. Any disclosure of the existence, content and results of the arbitration will be as limited and narrowed as required to comply with the applicable law. By way of illustration, if the applicable law mandates the disclosure of the monetary amount of an arbitration award only, the underlying opinion or rationale for that award may not be disclosed.

15.4 **Choice of Forum for US Disputes.** If a Dispute by one party against the other that arose in the United States or is based upon an alleged breach committed in the United States cannot be settled under the procedures and within the timeframe set forth in Section 15.2, then either party may bring an action or proceeding solely in either the Supreme Court of the State of New York, New York County, or the United States District Court for the Southern District of New York. Except as otherwise stated in Section 15.3 each party consents to the exclusive jurisdiction of those courts, including their appellate courts, for the purpose of all actions and proceedings arising out of or relating to this Agreement.

15.5 Injunctive Relief. Nothing in this Agreement will be construed to preclude either party from seeking provisional remedies, including, but not limited to, temporary restraining orders and preliminary injunctions from any court of competent jurisdiction in order to protect its rights, including its rights pending arbitration, at any time. The parties agree that the arbitration provision in Section 15.3 may be enforced by injunction or other equitable order, and no bond or security of any kind will be required with respect to any such injunction or order.

15.6 Time Limit. Actions on Disputes between the parties must be brought in accordance with this Section within 2 years after the cause of action arises.

16.0 AGREEMENT IN ENGLISH.

The parties confirm that it is their wish that the Agreement, as well as all other documents relating hereto, including all notices, have been and shall be drawn up in the English language only. Les parties aux présentes confirment leur volonté que cette convention, de même que tous les documents, y compris tout avis, qui s'y rattachent, soient rédigés en langue anglaise.

17.0 ENTIRE AGREEMENT.

This Agreement, its exhibits, schedules and other agreements or documents referenced herein, constitute the full and complete understanding and agreement between the parties and supersede all contemporaneous and prior understandings, agreements and representations relating to the subject matter hereof. No modifications, alterations or amendments shall be effective unless in writing signed by both parties to this Agreement.

18. REDISTRIBUTABLE CLIENT FILES.

The list of SDK client files that can be redistributed, if any, are in the SDK in a file called Redistributable.txt.

1 Table of Contents

2 Version History

Date	Revision #	Summary of Changes
December 2017	Version 1.0	Initial Oceana Release 3.4
April 2018	Version 1.1	Minor updates with Oceana 3.4.0.1
April 2018	Version 1.2	Updates with Oceana 3.5, primarily transfer to service and an eventing mechanism for Generic Channel contacts.
July 2018	Version 1.3	Further 3.5 Updates – Social handle, Routepoint being mandatory, Ability to specify to: address, Support for pre-existing context IDs, new authentication mechanism.
August 2018	Version 1.4	Minor updates based on feedback.
Sep 2018	Version 1.5	Minor updates based on feedback (routing to different groups).
Nov 2018	Version 1.6	Minor updates (re passing in WorkRequestID, CoreDataServices secure connection)
March 2019	Version 1.7	Additional parameters for createContact request - Account and CRM ID customer identifiers. [Alignment with Oceana 3.6]
Dec 2019	Version 1.8	Described facility to create Extra Data.
Dec 2020	Version 1.9	Updated supported capacity
Apr 2021	Version 1.10	Comment re CS storage of stored contacts

3 Overview

3.1 Description

The purpose of this guide is to delineate the steps involved in configuring and utilizing the Generic Channel Feature initially productised in Oceana 3.4. The Generic Channel feature allows third-party application developers the ability to queue their own contact-types in the Oceana Contact Centre to available agents. Customers can queue these third-party contact types into the Contact Centre to be routed to skilled idle agents using criteria specified by the application as part of the contact creation.

Users can configure one Generic Channel in Oceana, but it is possible to route different types of contacts by routing to different routepoints or by assigning different sets of attributes to different groups of contacts. So if for example a customer wants to route tickets of types 'Sales' and 'Support', they could create attributes 'Sales' and 'Support', assign these attributes to different sets of agents and create the contacts assigning the appropriate attributes.

At a high level, there are 2 distinct parts that end-customers must develop:
1) Acquiring an authentication token via the Breeze Authorization Service (running on the Oceana Breeze UAC cluster)
2. Consuming the provided API to create/drop Generic Channel contacts via the GenericChannelAPI svar (running on the Oceana Breeze OCP cluster).

Please note it is not permitted to deploy any customer developed Generic Channel API consuming application on a Breeze node that is part of an Oceana deployment. The customer developed software must be deployed on a separate server.

3.2 Contents

The package consists of:

- This guide.
- Installed reference client.
- Source Java code for the installed reference client.

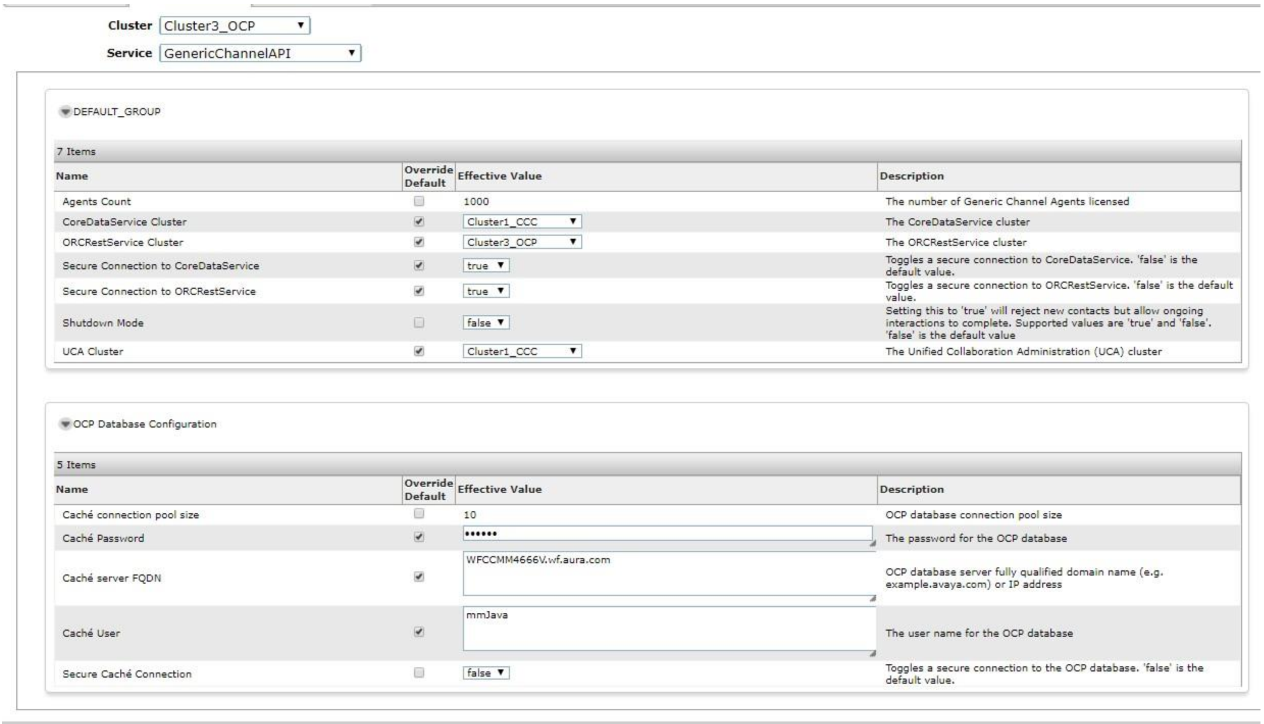
3.3 Related Oceana/Breeze Documentation

- Deploying Avaya Oceana™ Solution.
- External Client Authentication with the Authorization Service (Breeze).
- Eventing Framework documentation component of Breeze APIs.

4 Oceana Configuration

4.1 Svar Configuration

The new GenericChannelAPI svar is deployed on the OCP cluster. Attributes need to be configured here – this is documented in the Oceana Deployment Guide but sample attribute values are displayed here.



Cluster Cluster3_OCP ▼

Service GenericChannelAPI ▼

DEFAULT_GROUP

7 Items

Name	Override Default	Effective Value	Description
Agents Count	□	1000	The number of Generic Channel Agents licensed
CoreDataService Cluster	☑	Cluster1_CCC ▼	The CoreDataService cluster
ORCRestService Cluster	☑	Cluster3_OCP ▼	The ORCRestService cluster
Secure Connection to CoreDataService	☑	true ▼	Toggles a secure connection to CoreDataService. 'false' is the default value.
Secure Connection to ORCRestService	☑	true ▼	Toggles a secure connection to ORCRestService. 'false' is the default value.
Shutdown Mode	□	false ▼	Setting this to 'true' will reject new contacts but allow ongoing interactions to complete. Supported values are 'true' and 'false'. 'false' is the default value
UCA Cluster	☑	Cluster1_CCC ▼	The Unified Collaboration Administration (UCA) cluster

OCP Database Configuration

5 Items

Name	Override Default	Effective Value	Description
Caché connection pool size	□	10	OCP database connection pool size
Caché Password	☑	••••••	The password for the OCP database
Caché server FQDN	☑	WFCCMM4666V.wf.aura.com	OCP database server fully qualified domain name (e.g. example.avaya.com) or IP address
Caché User	☑	mmJava	The user name for the OCP database
Secure Caché Connection	□	false ▼	Toggles a secure connection to the OCP database. 'false' is the default value.

Agent Count : the number of GenericChannel agents licensed.

CoreDataServices Cluster : the IP/FQDN of the cluster hosting the OceanaCoreDataServices svar.

ORCRestService Cluster : the IP/FQDN of the cluster hosting the ORCRestService svar.

Secure Connection to CoreDataervices: This indicates if the connection to OceanaCoreDataService is over http or https.

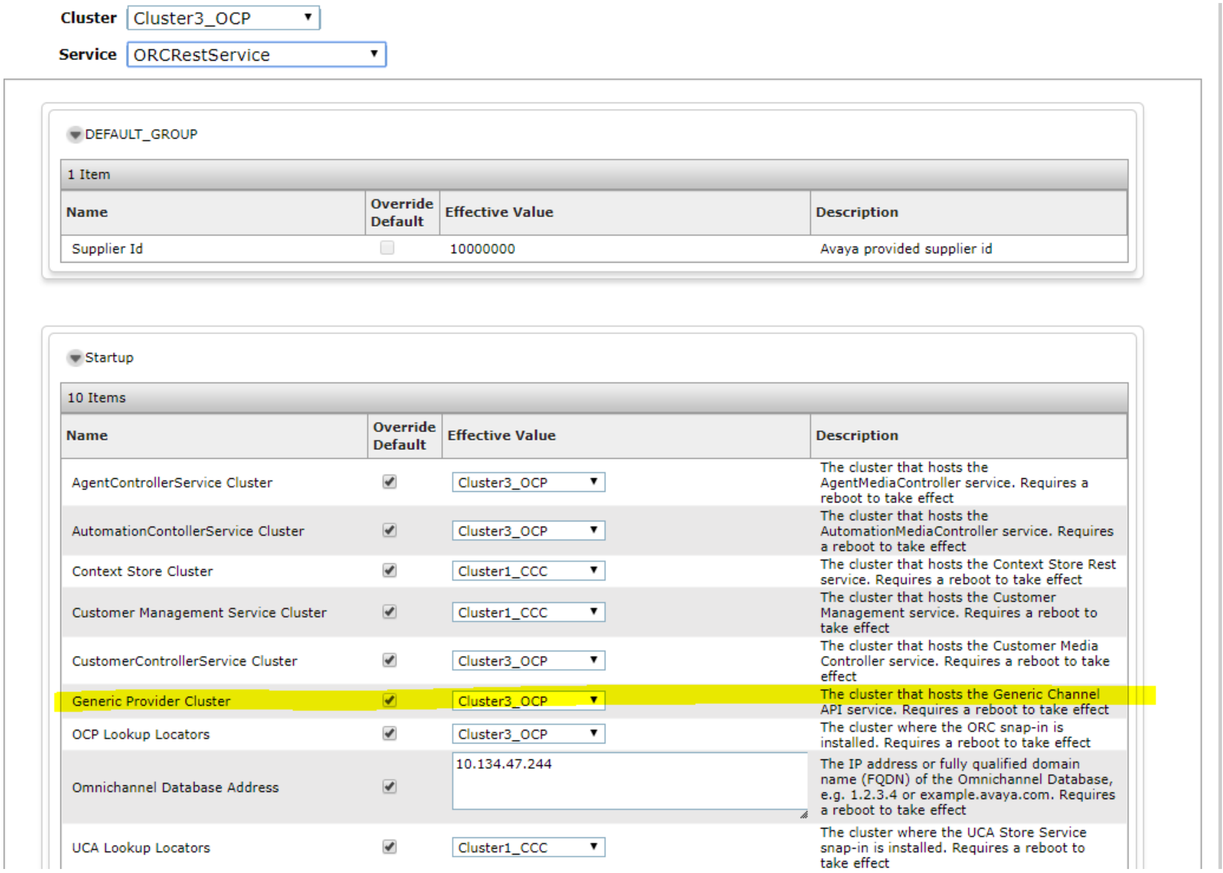
Secure Connection to ORCRestService: This indicates if the connection to ORCRestService is over http or https.

Shutdown Mode: When this is enabled, Oceana will reject any new incoming create requests to create Generic Channel contacts.

UCA Cluster: the cluster where the UCAService is running.

OCP Database Configuration: These database settings refer to the OCP database and connecting to it.

Additionally a new attribute needs to be configured for the ORCRestService SVAR:



Cluster: Cluster3_OCP

Service: ORCRestService

DEFAULT_GROUP

1 Item

Name	Override Default	Effective Value	Description
Supplier Id	□	10000000	Avaya provided supplier id

Startup

10 Items

Name	Override Default	Effective Value	Description
AgentControllerService Cluster	☑	Cluster3_OCP	The cluster that hosts the AgentMediaController service. Requires a reboot to take effect
AutomationContollerService Cluster	☑	Cluster3_OCP	The cluster that hosts the AutomationMediaController service. Requires a reboot to take effect
Context Store Cluster	☑	Cluster1_CCC	The cluster that hosts the Context Store Rest service. Requires a reboot to take effect
Customer Management Service Cluster	☑	Cluster1_CCC	The cluster that hosts the Customer Management service. Requires a reboot to take effect
CustomerControllerService Cluster	☑	Cluster3_OCP	The cluster that hosts the Customer Media Controller service. Requires a reboot to take effect
Generic Provider Cluster	☑	Cluster3_OCP	The cluster that hosts the Generic Channel API service. Requires a reboot to take effect
OCP Lookup Locators	☑	Cluster3_OCP	The cluster where the ORC snap-in is installed. Requires a reboot to take effect
Omnichannel Database Address	☑	10.134.47.244	The IP address or fully qualified domain name (FQDN) of the Omnichannel Database, e.g. 1.2.3.4 or example.avaya.com. Requires a reboot to take effect
UCA Lookup Locators	☑	Cluster1_CCC	The cluster where the UCA Store Service snap-in is installed. Requires a reboot to take effect

This attribute is the cluster on which the Generic Channel API svar is installed.

4.2 Engagement Designer Flow Creation

Please see the section “Deploy Engagement Designer workflows” in the “Deploying Avaya Oceana ™ Solution” guide. A similar approach needs to be followed for the GenericChannel contact type. When the steps are complete, please verify the flow is deployed via the following steps.

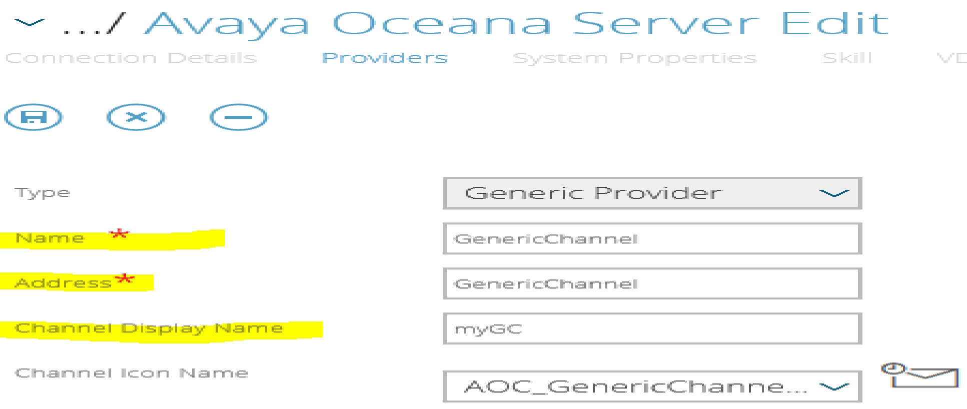
1. In your web browser, enter the following URL to open the Engagement Designer Admin Console: https://<AOC1 FQDN>/services/EngagementDesigner/admin.html.
2. On the Workflows tab, verify that the OceanaGenericAssistedService workflow is available in the list of deployed workflows.
3. In Oceana 3.5 support for transfer to service has been introduced, so there is an additional flow that needs to be deployed (OceanaGenericTransfer).

5 Oceana Generic Channel Configuration via ACM

5.1 Configure provider in ACM

Initial steps are to configure the provider in ACM.

On ACM, go to Configuration - Oceana - Server Details [select the UCA server] - Providers and Create a new Generic Channel Provider. Note: the value for Name must be GenericChannel.

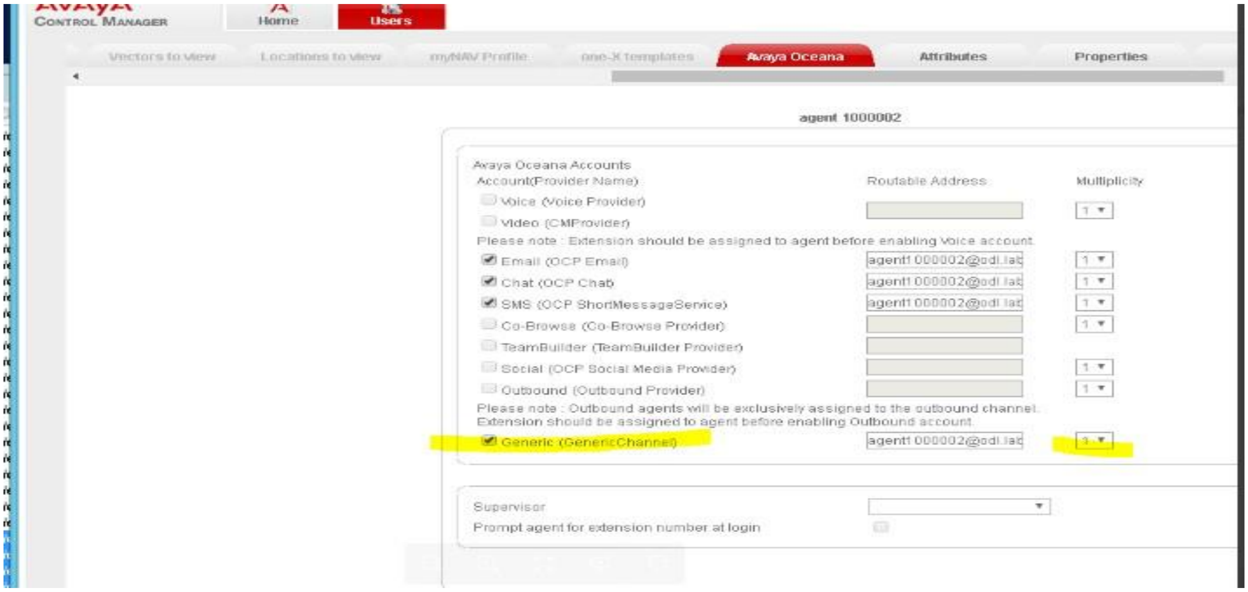


Type	Generic Provider
Name *	GenericChannel
Address*	GenericChannel
Channel Display Name	myGC
Channel Icon Name	AOC_GenericChanne…

To make the new GenericChannel Provider available to agents, you must restart all the clusters.

5.2 Assign Contact Type to Agents

On ACM – go to Users - Select an individual user - Select Avaya Oceana - and assign the GenericProvider Type with the desired multiplicity level.



agent 1000002

Avaya Oceana Accounts

Account(Provider Name)	Routable Address	Multiplicity
□ Voice (Voice Provider)		1 ▾
□ Video (CMProvider)		

Please note : Extension should be assigned to agent before enabling Voice account.

Account(Provider Name)	Routable Address	Multiplicity
☑ Email (OCP Email)	agent1000002@odl.lab	1 ▾
☑ Chat (OCP Chat)	agent1000002@odl.lab	1 ▾
☑ SMS (OCP ShortMessageService)	agent1000002@odl.lab	1 ▾
□ Co-Browse (Co-Browse Provider)		1 ▾
□ TeamBuilder (TeamBuilder Provider)		
□ Social (OCP Social Media Provider)		1 ▾
□ Outbound (Outbound Provider)		1 ▾

Please note : Outbound agents will be exclusively assigned to the outbound channel. Extension should be assigned to agent before enabling Outbound account.

Account(Provider Name)	Routable Address	Multiplicity
☑ Generic (GenericChannel)	agent1000002@odl.lab	1 ▾

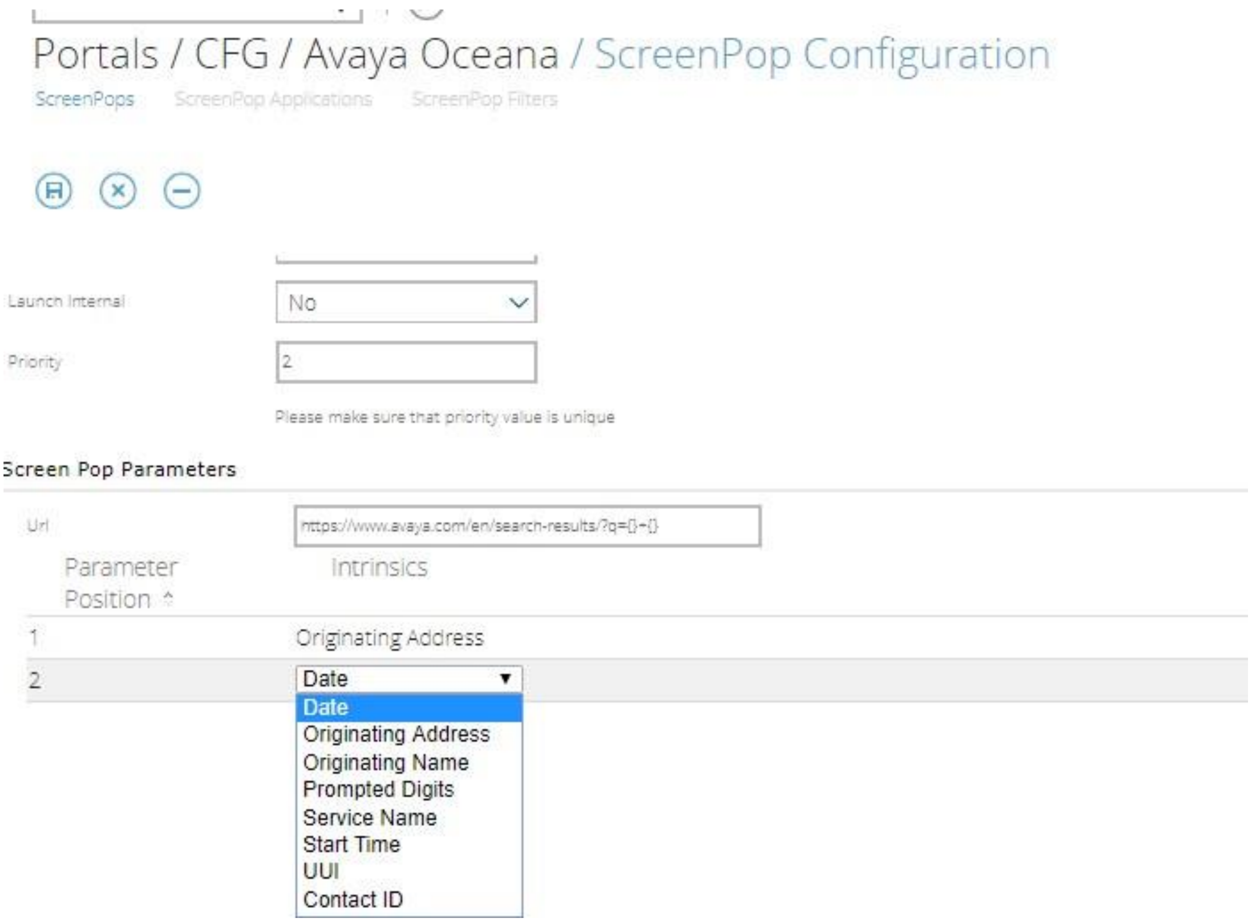
Supervisor ▾

Prompt agent for extension number at login □

5.3 Configure screen-pops, disposition code, work codes

If desired, channel specific disposition codes and work codes can be configured. Screenpops can be configured to handle the media of the GenericChannel contact (Oceana is not involved in the transport of media for GenericChannel contacts).

On ACM-Portals, Configuration, Oceana, Screenpop Configuration



Portals / CFG / Avaya Oceana / ScreenPop Configuration

Launch Internal: No

Priority: 2

Please make sure that priority value is unique

Screen Pop Parameters

Url: https://www.avaya.com/en/search-results/?q={}-{}

Parameter Position ^	Intrinsics
1	Originating Address
2	Date

Note Contact ID is now available as a screen-pop parameter, this is the unique identifier passed in for contact create requests.

5.4 Context Store Data Storage Configuration

If customers are planning to re-open closed Generic Channel contacts, care should be taken with the value set for the Context Store Manager “EDM: Keep closed contacts” attribute particularly if extra contextual information is passed into Oceana with the initial contact create request.

6 Client Authentication via the Breeze Authorization Service

6.1 Authentication Mechanisms

Generic Channel clients use the Breeze Authorization service to authenticate.

There are two mechanisms that can be adopted here:

1) A confidential client based approach.

2) An optional public client based approach(new with Oceana 3.5).

6.2 Confidential Client Authentication

This connection mechanism is described in detail at https://www.devconnectprogram.com/fileMedia/download/3049b344-b223-4644-a06f-fef6d220091d but the step involved are outlined below.

Pre-requisites for these configuration steps are to have Openssl and the Java Keytool application installed on the system you plan to utilize for Generic Channel creation and removal. Openssl is freely available on the internet and binaries for different platforms are available. The Keytool application comes as part of the standard Java JDK install.

6.2.1 Authentication Steps

Note: it is highly recommended to use the same password for all passwords required below. [Note in the example below the generated files are all created in the local directory]

1. As the Breeze Authorization Service expects valid signed JSON Web Tokens from clients to authenticate them, a client would first need to have a private key. Openssl is used here to generate an example key pair:

```
# Generates a key pair for the application
 openssl genrsa -aes256 -out client.key 2048
```

2. Next, generate a CSR to get it signed by a CA:

```
# Generates a CSR using the above key
 openssl req -x509 -sha256 -new -key client.key -out client.csr
```

3. You will be asked to enter a few fields (for creating the CSR and DN), an example is shown below:
Country Name (2 letter code) [AU]: State or Province Name (full name) [Some-State]: Locality Name (e.g., city) []: Organization Name (e.g., company) [Internet Widgits Pty Ltd]: Organizational Unit Name (e.g., section) []: Common Name (e.g. server FQDN or YOUR name) []: Email Address []:

4. Once the CSR is generated, get it signed by a CA. One approach could be to self-sign the CSR:

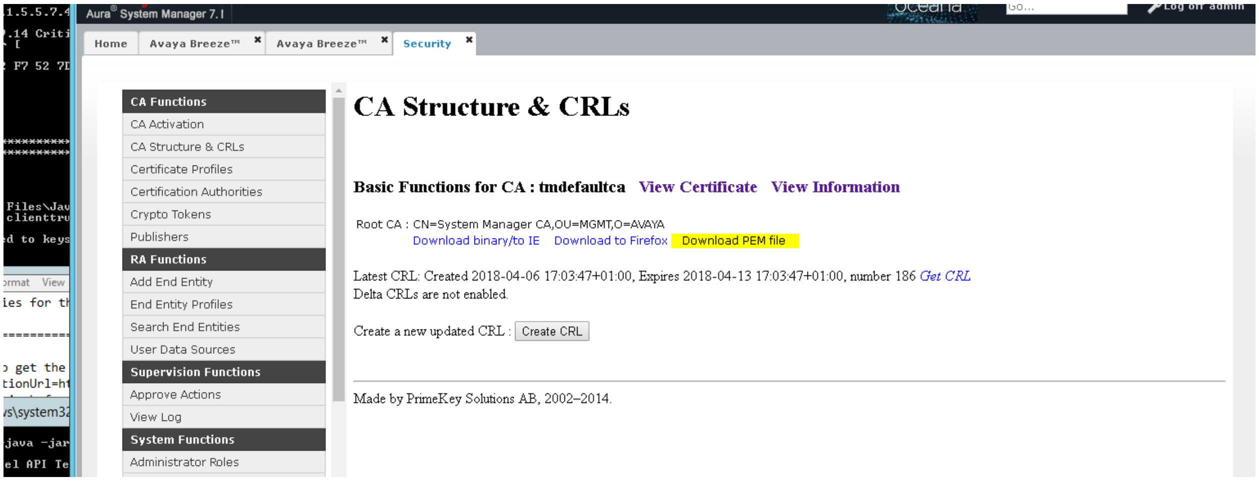
```
# Self-sign the CSR
openssl x509 -sha256 -days 3652 -in client.csr -signkey client.key -out client.crt
```

Take a note of client.crt - This certificate will be used further while adding an external client in SMGR

5. Next, create a PKCS#12 store out of the generated key and certificate:
# Takes the cert and the private key to create a PKCS#12 keystore openssl pkcs12 -export -name clientcert -in client.crt -inkey client.key -out keystore.p12

6. This step converts the PKCS#12 store to a Java KeyStore:
# Converts a PKCS#12 store to JKS keytool -importkeystore -destkeystore clientkeystore.jks -srckeystore keystore.p12 -srcstoretype pkcs12 -alias clientcert

7. Download the SMGR CA Cert which is Active and is the one which is trusted by the Breeze node onto the local machine where these steps are being carried out and copy to the current folder.
This is available to download from SMGR at:



CA Structure & CRLs

Basic Functions for CA : tmdefaultca

Root CA : CN=System Manager CA,OU=MGMT,O=AVAYA

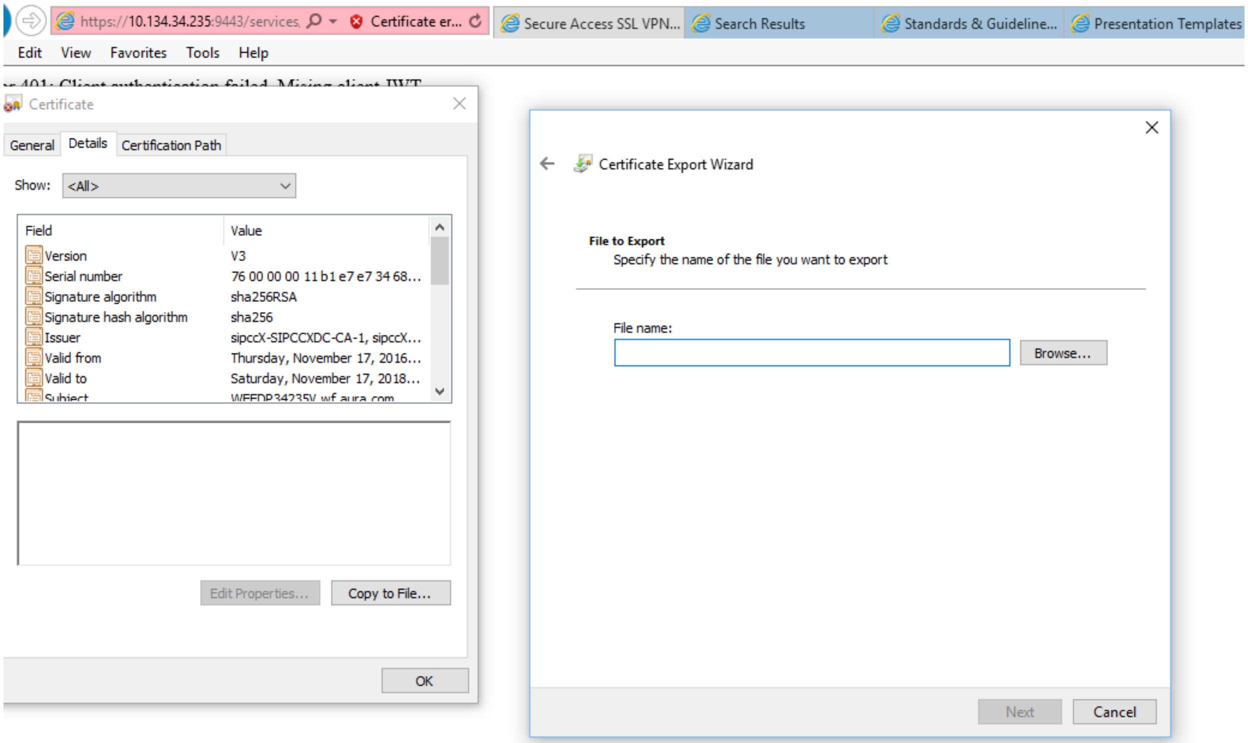
Latest CRL: Created 2018-04-06 17:03:47+01:00, Expires 2018-04-13 17:03:47+01:00, number 186
Delta CRLs are not enabled.

Create a new updated CRL :

Made by PrimeKey Solutions AB, 2002–2014.

Alternatively,download the cert using following command:
openssl s_client -servername LabFQDNorIP -connect LabFQDNorIP:9443/services/AuthorizationService/token </dev/null | sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' >/home/USER/SystemManagerCA.cacert.pem

Finally this certificate can also be found by browsing with IE to the URL of the UCA cluster in the Oceana environment (e.g. https://<FQDN:9443>/services/AuthorizationService/token) where the Authorization Service is running and viewing and downloading the certificate. This approach was tested using the DER encoded binary X.509 (.CER) format in the Export File Format screen.



Field	Value
Version	V3
Serial number	76 00 00 00 11 b1 e7 e7 34 68...
Signature algorithm	sha256RSA
Signature hash algorithm	sha256
Issuer	sipccX-SIPCCXDC-CA-1, sipccX...
Valid from	Thursday, November 17, 2016...
Valid to	Saturday, November 17, 2018...
Subject	WFFDP34235V.wf.aura.com

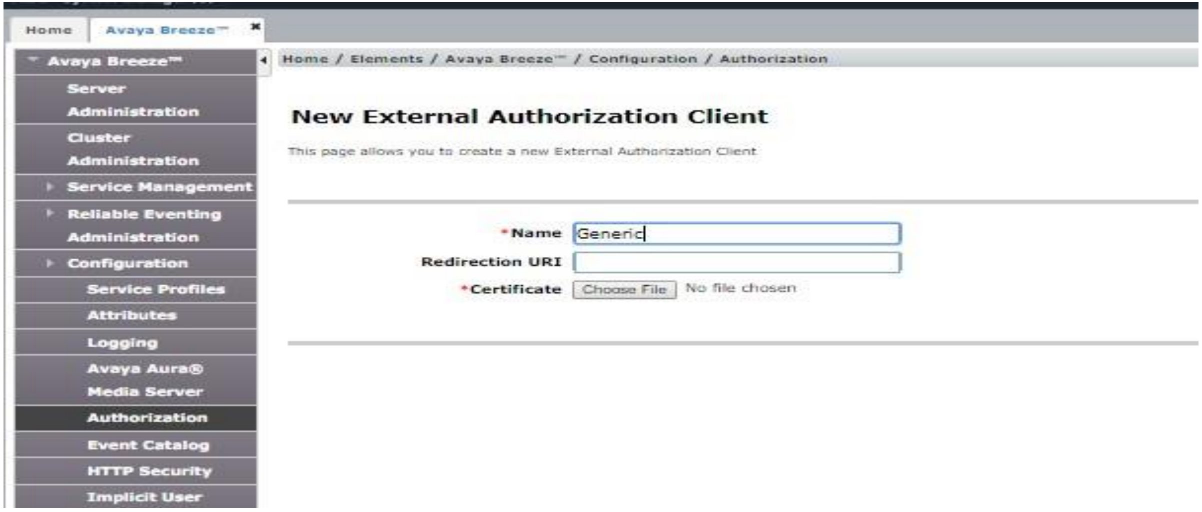
File to Export
Specify the name of the file you want to export

File name:

8. Next, import this certificate into a Java trust store:
# Import SMGR CA cert onto a trust store keytool -import -noprompt -alias smgrca -keystore clienttruststore.jks - file SystemManagerCA.cacert.pem -storepass password

9. The following files are of interest in further steps:
client.crt - Certificate to be used while provisioning the external client in SMGR clientkeystore.jks - The client keystore clienttruststore.jks - The client truststore

10. Go to Breeze > Configuration > Authorization > Client tab > Select New:



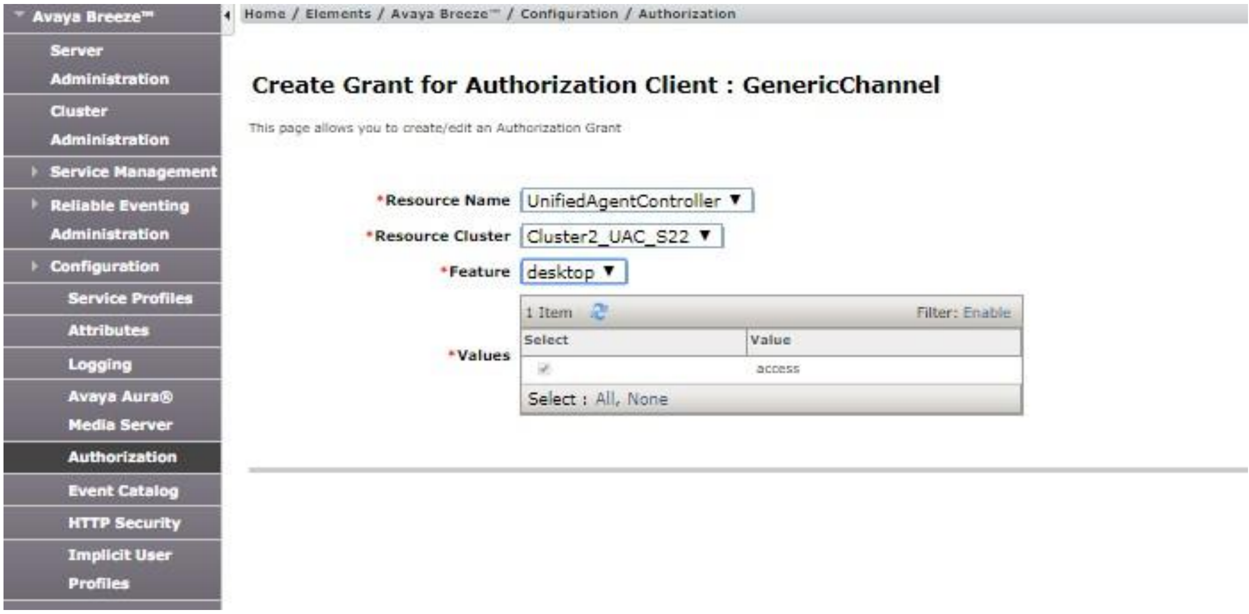
New External Authorization Client

This page allows you to create a new External Authorization Client

*Name: Generic
Redirection URI:
*Certificate: No file chosen

and load the certificate created above.

11. Assign a grant to this authorization client.



Create Grant for Authorization Client : GenericChannel

This page allows you to create/edit an Authorization Grant

*Resource Name: UnifiedAgentController
*Resource Cluster: Cluster2_UAC_S22
*Feature: desktop
*Values

1 Item — Filter: Enable

Select	Value
☑	access

Select : All, None

6.2.2 Sample code for Access to authorization Token

The following is sample Java code to access the authorization token:

```
KeyStore clientKeyStore = KeyStore.getInstance("JKS");
clientKeyStore.load(new FileInputStream("keystore_location"), "pass-word".toCharArray());

KeyStore clientTrustStore = KeyStore.getInstance("JKS");
clientTrustStore.load(new FileInputStream("truststore_location"), "pass-word".toCharArray());

AuthorizationClientHelper clientHelper = new AuthorizationClientHelper.Builder()
.tokenEndpoint("https://<FQDN:9443>/services/AuthorizationService/token") .clientIdentifier("client_id")
.keyStore(clientKeyStore, "keystore_password", "key_alias") .trustStore(clientTrustStore) .build();

AccessToken token = clientHelper.getAccessToken();
```

There are a few things which this code is doing before making a call to get an access token:
The first two lines prepare the client keystore. The keystore_location would be the absolute path of "clientkeystore.jks" which was created in Step 6 in the previous section.

The next two lines load the client truststore. The truststore_location would be the absolute path of "clienttruststore.jks" created in Step 8 of the previous section.

The builder requires these arguments (in exact order):

- The Authorization Service token endpoint location
- Client identifier got from Step 11 in the previous section.

- Keystore initialized on the first two lines with the key alias mentioned in Step 6 command of the previous section.
- Truststore initialized in the third and fourth lines.

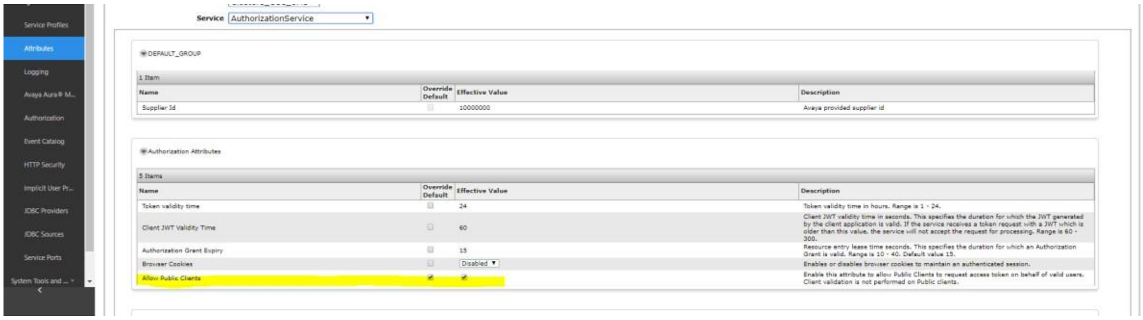
Note: Please note that currently the time on the client PC must match the time on the Breeze Node that you are authenticating against. Please note that currently it is advisable to use the IP address instead of the FQDN of the Breeze cluster you are authenticating against.

Note: Please also note that the tokens provided by the Authorization Service are of defined time duration and consuming clients will need to deal with this timeout in their implementation.

Note: Please note that if you are using domains that are not listed at http://data.iana.org/TLD/tlds-alpha-by-domain.txt (e.g domains like .local), you will need to enter the IP address instead of the FQDN as the authorization endpoint. The authorization code only supports domains in the list referenced above.

6.3 Public Client Authentication

Support for public clients is not enabled by default. This is administrable via a service attribute on Authorization Service.



Service: AuthorizationService

DEFAULT_GROUP

1 Item

Name	Override Default	Effective Value	Description
Supplier Id	□	10000000	Avaya provided supplier id

Authorization Attributes

5 Items

Name	Override Default	Effective Value	Description
Token validity time	□	24	Token validity time in hours. Range is 1 - 24.
Client JWT Validity Time	□	60	Client JWT validity time in seconds. This specifies the duration for which the JWT generated by the client application is valid. If the service receives a token request with a JWT which is older than this value. the service will not accept the request for processing. Range is 60 - 300.
Authorization Grant Expiry	□	15	Resource entry lease time seconds. This specifies the duration for which an Authorization Grant is valid. Range is 10 - 40. Default value 15.
Browser Cookies	□	Disabled ▾	Enables or disables browser cookies to maintain an authenticated session.
Allow Public Clients	☑	☑	Enable this attribute to allow Public Clients to request access token on behalf of valid users. Client validation is not performed on Public clients.

When this is administered to true, a client which doesn't provide its authorization details in Authorization header is treated as a public client.

Public clients do not request for client scopes in the token request. In case a public client requests for the same, the request is rejected.

The user's credentials are then provided in the requested body. An example is shown below:

```
POST <HOST>/services/AuthorizationService/token HTTP/1.1
Content-Type: application/x-www-form-urlencoded

grant_type=password&
username=johndoe&
password=EyT54Jut
```

Note - There is no Authorization header present in the request.

This request returns a token which can then be used in Generic Channel requests (as per the token described in section 6.2.2).

7 API Functions

The API supports functions for the creation and dropping of Generic Channel contacts and 2 helper functions to return the available attributes and Routepoints available in the system. For each of the methods supported, the authorization token retrieved from the Breeze Authorization service above must be passed as a parameter. Requests will be rejected if the passed in authorization token is invalid.

7.1 CreateContact

https://FQDN_or_IP/services/GenericChannelAPI/api/createcontact

Support Post method

This supports passing in of Customer PhoneNumber, EmailAddress, social handle, a list of attributes, a Routepoint and the actual contact ID. This contact ID is what will be made available for screenpop and WidgetAPI on Workspaces.

Sample JSON for this is:

```
{{"identifier":"emailAddress","emailAddress":"Sprint46@demoAvaya.com","destinationAddress":"testD
estinationAddress","contactID":"test2018Jul24155318702","routePoint":"SystemDefaultDigitalRoutepoi
nt","attributes":"Location.GalwayE,Language.English","ssoToken":"eyJhbGciOiJSUzI1...","socialHandle":"
sprint46demo","socialPlatform":"twitter", "workRequestID":"" }}
```

Please note that the attribute here is just a sample value – for the contact to route successfully there must be an available agent with the GeneriChannel contact type and the entered attribute(s) assigned to them.

Please note the system enforces uniqueness of ContactID in the contacts that are currently in queue or active at agents. The system will not allow creation of a contact with a ContactID that is already in queue or active at an agent.

The contactID, ssoToken and a customer identifier, (be that email address/telephone number/social handle&platform/account type&ID/crmId),Routepoint fields must be specified. The "identifier" field allowed values are "emailAddress","telephoneNumber","social", "account", "crmId".

If these fields are not present the contact create request will be rejected.

Other fields are optional, but not specifying any attributes would lead to routing based on Generic Channel contact type only.

Please note when passing in ContactID values only alphanumeric characters are permitted.

Additional features in Oceana 3.5

Routepoint as a mandatory field is a breaking change introduced in Oceana 3.5. A default Routepoint – SystemDefaultDigitalRoutepoint is now present on all Oceana installations.

Oceana 3.5 introduced support for a social handle to be a customer identifier – this is specified via two parameters – socialHandle and socialPlatform.

The Oceana 3.5 release also introduced support for passing in an existing WorkRequestID/ContextID as a parameter in the createContact request. This allows users to prepopulate information into ContextStore which is then merged with the other parameters that are passed in with the createContact request. This will allow the consumer of the GenericChannel API to insert supporting contextual information into Oceana prior to CreateContact and then to correlate the CreateContact request with this ContextID. The parameter used to pass this parameter into the request is “workRequestID” (as per above).

Additional features in Oceana 3.6

Customer Identifiers:

Note, with Oceana 3.6, Oceana now also supports passing in Account/CRM details as the customer identifier. Please note that to pass in account details, two parameters must be supplied – acccountID and accountType.

Sample JSON:

```
{{"identifier":"account","accountID":"123435","accountType":"ACCOUNT_ID","contactID":"test2018Jul24155318702","routePoint":"SystemDefaultDigitalRoutepoint","attributes":"Location.GalwayE,Language.English","ssoToken":"eyJhbGciOiJSUzI1...","socialHandle":"sprint46demo","socialPlatform":"twitter", "workRequestID":"" }}
```

The accountType specified needs to exist in the Oceana OCP Database – these account type are populated via the Oceana Customer Management Tool (OCMT). Attempts to create a contact with a non-existent account type will be rejected.

To pass in CRM details, a crmID must be supplied in the createContact request:

Sample JSON:

```
{{"identifier":"crmId","crmId":"123435",routePoint":"SystemDefaultDigitalRoutepoint","attributes":"Location.GalwayE,Language.English","ssoToken":"eyJhbGciOiJSUzI1...","socialHandle":"sprint46demo","socialPlatform":"twitter", "workRequestID":"" }}
```

Priority:

Generic Channel supports passing in priority as an optional parameter in createContact requests.

Priority is a numeric value which can be between 1 and 10. It can be used by Oceana to route contacts with order of importance.

Sample JSON:

```
{"crmId":"123456_789","identifier":"crmid","destinationAddress":"testDestinationAddress","contactID":"RND2019Mar23131107713","extraData":"","routePoint":"SystemDefaultDigitalRoutepoint","attributes":"Location.Inhouse,Language.English","ssoToken":"eyJhbGciOiJSUzI1...","priority":"10","workRequestID":""}
```

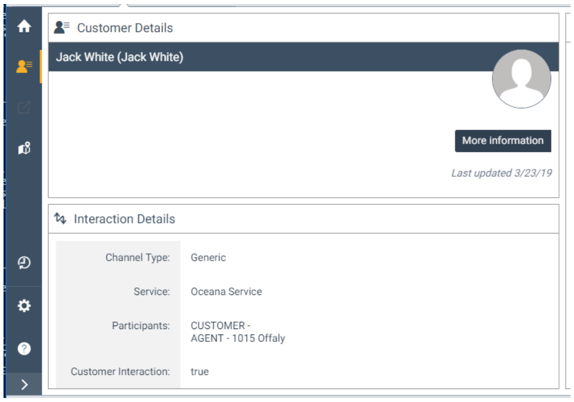
First/Last Name parameters:

It is also possible to pass in the first and last names of the customer creating the contact as optional parameters in the createContact request:

Sample JSON:

```
{"identifier":"crmid","lastName":"White","destinationAddress":"testDestinationAddress","contactID":"RND2019Mar23132900806","extraData":"","isUseRPAsCalledParty":"true","priority":"10","crmId":"123456_789","firstName":"Jack","routePoint":"SystemDefaultDigitalRoutepoint","attributes":"Location.Inhouse,Language.English","ssoToken":"eyJhbGciOiJSUzI1...","workRequestID":""}
```

These will appear in Workspaces in the customer details display:



Customer Details

Jack White (Jack White)

Last updated 3/23/19

Interaction Details

Channel Type: Generic

Service: Oceana Service

Participants: CUSTOMER -
AGENT - 1015 Offaly

Customer Interaction: true

ExtraData being passed into ContextStore:

It is possible to pass in extra data with the contact creation request that will be stored with the context in the Context Store. This data can be exposed to Agents using Workspaces if bespoke Workspaces widgets are developed accessing the Context Store data.

Sample JSON:

```
{"identifier":"emailAddress","lastName":"Pale","destinationAddress":"testDestinationAddress","contactID":"Contact1234","extraData":"extraData:mytestExtraData","isUseRPAsCalledParty":"true","priority":"1","firstName":"Jack","emailAddress":"enda@edtest1.com","routePoint":"SystemDefaultDigitalRoutepoint","overwriteContextDetails":"false","ssoToken":"eyJhbGciOiJSUzI1...","workRequestID":"MyContextID"}
```

Note the WorkRequestID is being passed in with the contact creation request and the format of the data is a key value pair.

7.2 DropContact

https://FQDN_or_IP services/GenericChannelAPI/api/dropcontact - supports POST method.

Sample JSON:

{{"contactID":"850000000","ssoToken":"eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiI0bWJSTmJrVFEzUzR5WGhQcy
01R25nIiwiY2xpZW50U2NvcGVMaXN0IjpbeyJmZWF0dXJlTmFtZSI6IlVuaWZpZWRBZ2VudENvbnRyb2xsZX
IuZGVza3RvcCIsImZlYXR1cmVWYWx1ZXMiOlsiYWNjZXNzIl0sImNsdXN0ZXJJZCI6NzEwMX1dLCJpc3MiOiIw
OWZkMmQ4MC1jY2UyLTRlZDQtYjY3Yi01NTYyZjljNDc4N2IiLCJleHAiOjE1MTExMjUyODksImlhdCI6MTUx
MTAzODg4OSwianRpIjoiMWMzZTU0Y2QtMDRmZS00YTNhLWE5MTItMDZiYjQ4N2YxNGRmIn0.R3bSSMI
NB6AZ3oPrazcuX - 1845EUttrOz01 -
Rr2m0TaNg1hgvv3tbiqsjCOVBsbe4CNUhL8U5qElqHoDIy2feXxFFURH3NH3aJ1IAxtEwFAao6V6s_yrqMmB
sLoPkJgJ2OlHHO2jQ2ZvVRDB -
wH4MStkqdhwNsc04YM3aFnALvrdMejDk1MhOPlcWshLrMy3MVNmOv_f0YH5FZRnWFYgvV1lWPVOHW
Kg3HFRUobpRij9LEd_ps1XuWnxPnKmmaQsxq4cleHB_iMNoHbHm315PwCi_QNEPGYLriIFhonTnlPv5dfvm
gwrAwTltEfoNmA1aqu9sgqvnj5mecf5I7KQug"}}

Please note that the DropContact command drops the contact at all phases of routing – i.e. if the contact is in queue waiting for an agent, alerting at an agent or active at an agent, the contact will be dropped.

7.3 GetAttributes

https://FQDN_or_IP services/GenericChannelAPI/api/getlistofattributes – supports Post method

1 parameter – the authorization token

7.4 GetRoutepoints

https://FQDN_or_IP services/GenericChannelAPI/api/getlistofroutepoints - Supports POST method

1 parameter – the authorization token

7.5 Sample code:

The code below is in C# and is designed to show example usage of the API.

7.5.1 CreateContact

```
static void createContact(RestClient Client2,string token, string contactID)
        {
            try
            {
                var request2 = new
              RestRequest("services/GenericChannelAPI/api/createcontact", Method.POST);
                request2.AddHeader("Content-Type", "application/json");
                JsonObject jS1 = new JsonObject();
                jS1.Add("contactID", contactID);
                jS1.Add("ssoToken", token);
                jS1.Add("emailAddress", "mMouse@avaya.com");
                jS1.Add("telephoneNumber", "5566777");
                jS1.Add("identifier", "emailAddress");
                jS1.Add("routePoint","myGenericRP2");
                jS1.Add("attributes", "Location.Inhouse");
                jS1.Add("workRequestID", "PreviouslyCreatedWRID");
                request2.AddJsonBody(jS1);
                IRestResponse response2 = Client2.Execute(request2);
                var content2 = response2.Content; // raw content as string
                log.Debug("Create" + contactID + " Output: " + content2);
            }
            catch(Exception ex1)
            {
                log.Debug(ex1.Message);
            }
        }
```

7.5.2 Drop Contact:

```
  static void dropContact(RestClient Client2,string token, string contactID)
        {
            try
            {
                var request2 = new
RestRequest("services/GenericChannelAPI/api/dropcontact", Method.POST);
                request2.AddHeader("Content-Type", "application/json");

                JsonObject jS1 = new JsonObject();
                jS1.Add("contactID", contactID);
                jS1.Add("ssoToken", token);

                request2.AddJsonBody(jS1);
                IRestResponse response2 = Client2.Execute(request2);
                var content2 = response2.Content; // raw content as string
                log.Debug("DR" +contactID + " Output: " +content2);
            }
            catch (Exception ex)
            {
                log.Debug(ex.Message);
            }
        }
```

7.5.3 Attributes:

```
static void getListOfAttributes(RestClient Client2, string token)
        {
            try
            {
                var request2 = new
RestRequest("services/GenericChannelAPI/api/getlistofattributes", Method.POST);
```

```
            request2.AddHeader("Content-Type", "application/json");

            JsonObject jS1 = new JsonObject();
            jS1.Add("ssoToken", token);
            request2.AddJsonBody(jS1);
            IRestResponse response2 = Client2.Execute(request2);
            var content2 = response2.Content; // raw content as string
            log.Debug("GetAttributes" + " Output: " + content2);

        }
        catch (Exception ex)
        {
            log.Debug(ex.Message);
        }
    }
```

7.5.4 Routepoints:

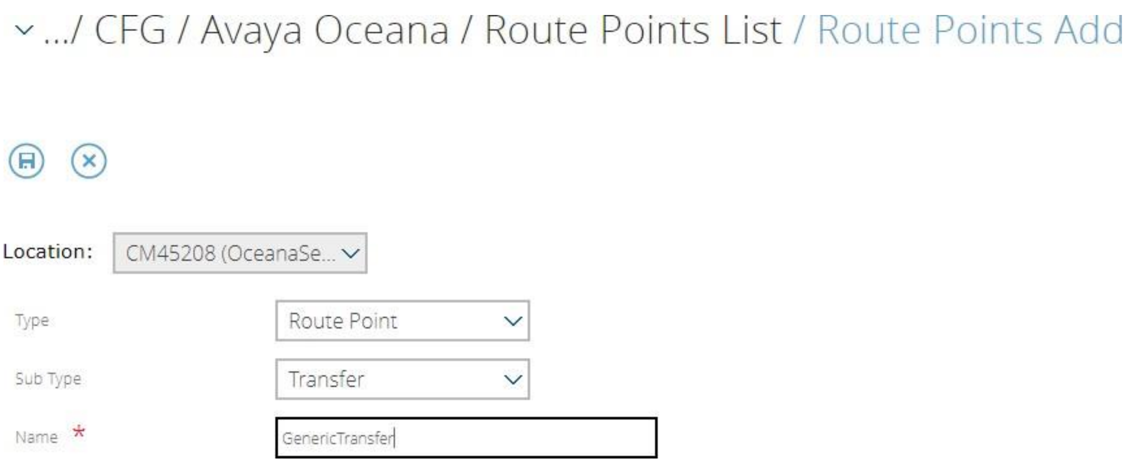
```
static void getListOfRoutepoints(RestClient Client2, string token)
    {
        try
        {
            var request2 = new
RestRequest("services/GenericChannelAPI/api/getlistofroutepoints", Method.POST);

            request2.AddHeader("Content-Type", "application/json");

            JsonObject jS1 = new JsonObject();
            jS1.Add("ssoToken", token);
            request2.AddJsonBody(jS1);
            IRestResponse response2 = Client2.Execute(request2);
            var content2 = response2.Content; // raw content as string
            log.Debug("GetAttributes" + " Output: " + content2);

        }
        catch (Exception ex)
        {
            log.Debug(ex.Message);
        }
    }
```

8 Transfer to Service for Generic Channel

With the launch of Oceana 3.5, support for transfer to service for Generic Channel has been introduced.

There are a number of steps in ACM required for this (same approach as being used for other channel types).

Initially, a transfer Routepoint must be created- ACM-> Configuration -> Oceana -> Route Points-> Add

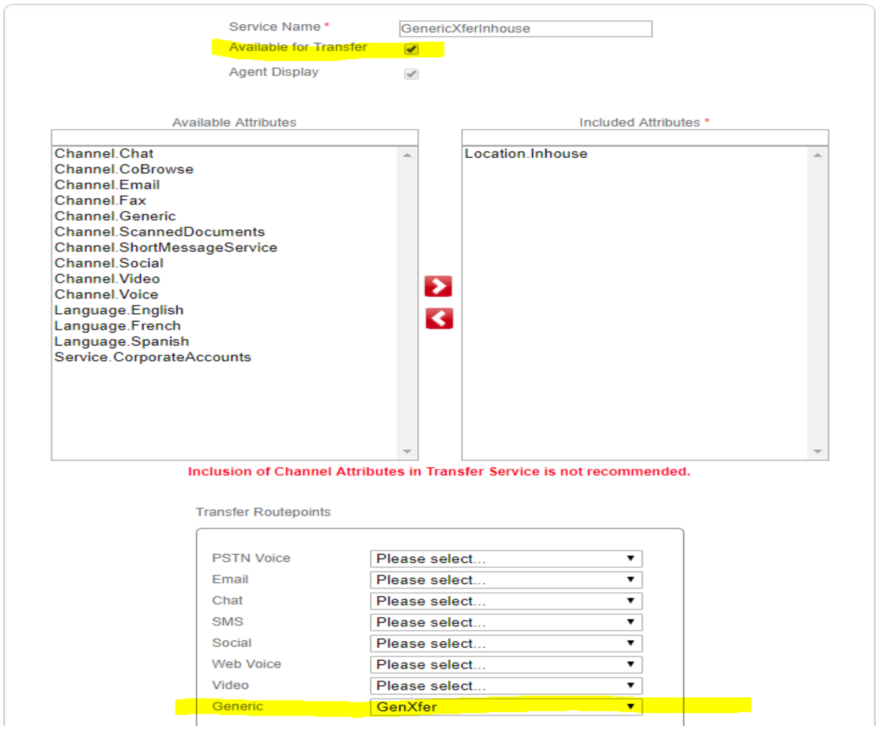


⌄ .../ CFG / Avaya Oceana / Route Points List / Route Points Add

Location: CM45208 (OceanaSe...

Type	Route Point
Sub Type	Transfer
Name *	GenericTransfer

Then a transfer to service service must be created. ACM -> Oceana ->WorkAssignment -> Services



Service Name * GenericXferInhouse

Available for Transfer ☑

Agent Display ☑

Available Attributes:

Channel.Chat
Channel.CoBrowse
Channel.Email
Channel.Fax
Channel.Generic
Channel.ScannedDocuments
Channel.ShortMessageService
Channel.Social
Channel.Video
Channel.Voice
Language.English
Language.French
Language.Spanish
Service.CorporateAccounts

Included Attributes *:

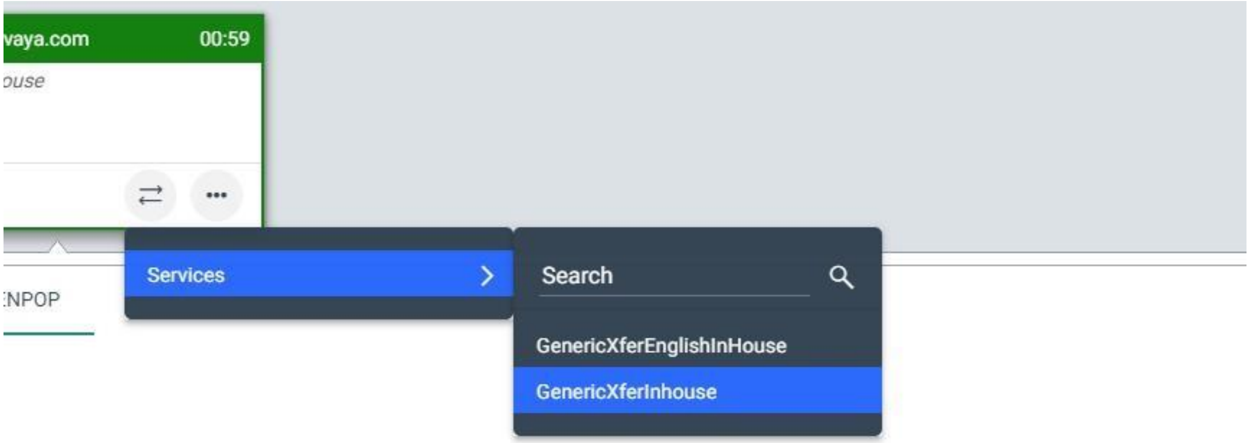
Location.Inhouse

Inclusion of Channel Attributes in Transfer Service is not recommended.

Transfer Routepoints

PSTN Voice	Please select...
Email	Please select...
Chat	Please select...
SMS	Please select...
Social	Please select...
Web Voice	Please select...
Video	Please select...
Generic	GenXfer

When an agent answers a Generic Channel contact after the transfer service has been created, the agent will have the option to transfer the contact to service.

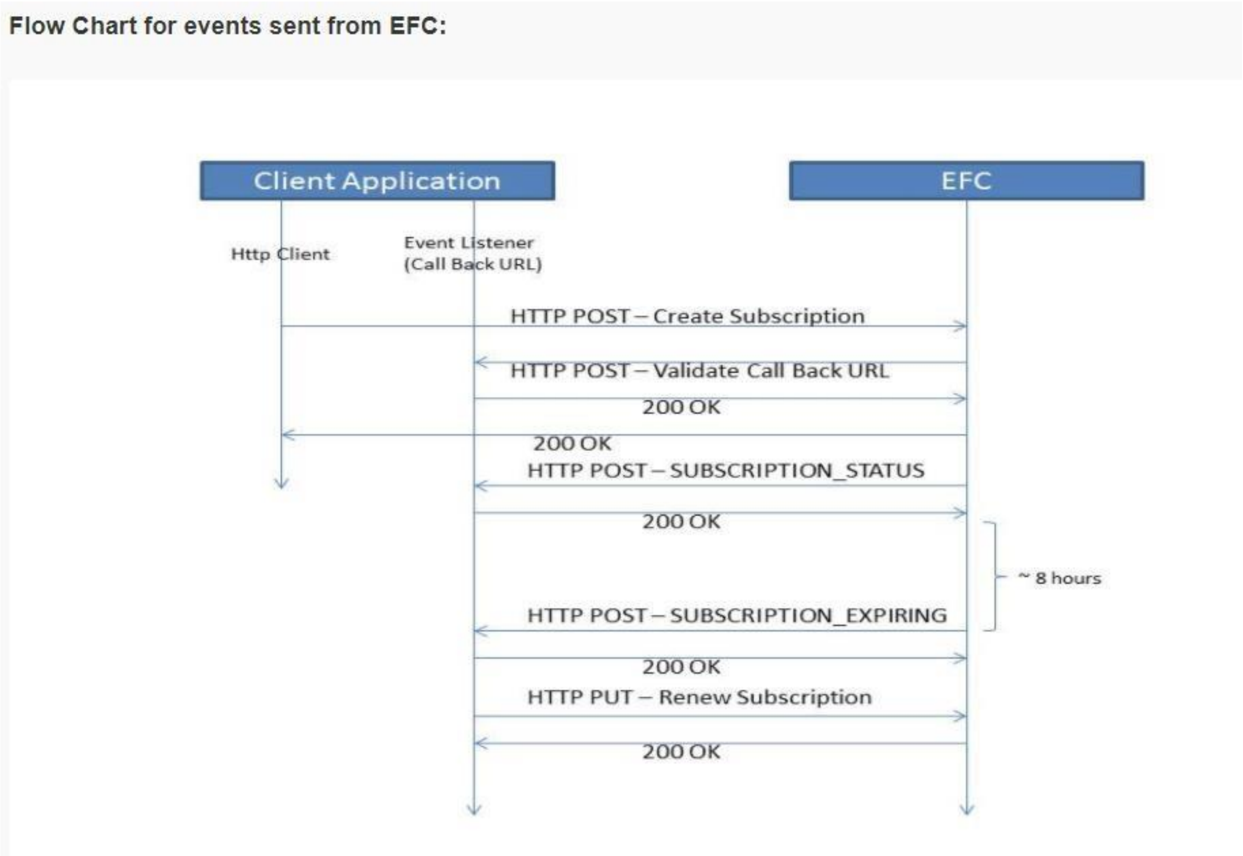


9 Subscribing for Generic Channel Events

With Oceana 3.5, Avaya is documenting a method for client to subscribe for events related to the life cycle of Generic Channel contacts (e.g. Contact Routed, Contact Answered, Contact Ended etc.).

The mechanism being used is the Eventing Framework feature already utilised inside Oceana for event generation between Oceana Internal components.

This is the flow that is used for the client to subscribe to the EF:



Flow Chart for events sent from EFC:

Client Application
EFC
Http Client
Event Listener
(Call Back URL)
HTTP POST – Create Subscription
HTTP POST – Validate Call Back URL
200 OK
200 OK
HTTP POST – SUBSCRIPTION_STATUS
200 OK
~ 8 hours
HTTP POST – SUBSCRIPTION_EXPIRING
200 OK
HTTP PUT – Renew Subscription
200 OK

This operates by passing in a URL to which the EF sends the events that the client has subscribed for. Thus the client needs to have a httpServer running at the URL specified that will return 200 ok to requests.

Note – the client machine must allow the events from Oceana pass through so it, so there may be configuration required from a firewall perspective to allow this network traffic through.

For more information on this, please review the Eventing Framework information available from Devconnect at:

https://www.devconnectprogram.com/site/global/api-refs/breeze-3_5/index.html

A sample subscription would be:

{{"callbackContentType":"application/json","callbackUrl":"http://XX.YY.ZZ.WW:1234/","family":"OmniCenter"}}

Sample C# code:

```
  public static void StartEventSubscription()
        {
            try
            {
                log.Debug("*********************Start Event Subscription
******************************");

                string strEFIPAddress =
ConfigurationManager.AppSettings["EVENT_CONNECTOR_HOST"];

                string strFormattedEFAddress = "http://" + strEFIPAddress + "/";

                var client1 = new RestClient(strFormattedEFAddress);


                var request2 = new
RestRequest("services/EventingConnector/subscriptions", Method.POST);
                request2.AddHeader("Content-Type", "application/json");


                string strEventFamily = ConfigurationManager.AppSettings["EVENT_FAMILY"];
                string strCallbackIP =
ConfigurationManager.AppSettings["EVENT_CALLBACK_SERVER"];
                string strCallbackPort =
ConfigurationManager.AppSettings["EVENT_CALLBACK_PORT"];

                string strCallbackURL = "http://" + strCallbackIP + ":" + strCallbackPort
+ "/";

                Thread work = new Thread(new ParameterizedThreadStart(startHttpServer));
                work.Start(strCallbackURL);

                //wait 5 seconds after starting the webserver before requesting the
subscription
                Thread.Sleep(5000);
                JsonObject jS1 = new JsonObject();
                jS1.Add("callbackContentType", "application/json");
                jS1.Add("callbackUrl", strCallbackURL);
                jS1.Add("family", strEventFamily);


                request2.AddJsonBody(jS1);
                IRestResponse response2 = client1.Execute(request2);
                var content2 = response2.Content; // raw content as string

                if (content2.Contains("subscription"))
                {
                    Console.WriteLine("Created a subscription");
                }
            }
            catch (Exception ex12)
            {
                Console.WriteLine(ex12.Message);
            }
        }
```

App.config extract matching the above:

```
 <!-- IP Address of EF server-->
   <add key="EVENT_CONNECTOR_HOST" value="10.134.47.209"/>
   <!-- IP Address of EF server-->
   <add key="EVENT_FAMILY" value="OmniCenter"/>
     <!-- IP Address of Callback server-->
   <add key="EVENT_CALLBACK_SERVER" value="135.60.160.198"/>
   <!-- Callback server port-->
   <add key="EVENT_CALLBACK_PORT" value="2221"/>
   <!-- EventMetaDataFiltering-->
   <add key="EVENT_METADATA_FILTERING" value="channelType = Generic"/>
```

10 Shutdown behavior

Please note that the GenericChannel svar supports the concept of a shutdown mode of operation controlled via an svar attribute. If shutdown mode is enabled, new contact create requests are rejected. Note that contact drop requests are processed whilst this mode is enabled. This mode could be enabled prior to a planned shutdown to help 'drain' all Generic Channel contacts from Oceana.

If there are any Generic Channel contacts present in a queueing state or active at agents and the OCP cluster is rebooted, these contacts will be re-queued when the OCP cluster comes back up.

11 Sample Client

The sample client provides a mechanism to test the functionality provided by the API.

Please note that the steps in chapter 6 above regarding authorization need to be performed on the machine where the test client is to be run. The test is written in Java so there is a requirement that java be installed on the machine running the test client. Note the operation of this client has been verified with Java 1.8.

The client is invoked via the command:

```
java.exe -jar GenericChannelTestClient-11.0-SNAPSHOT.jar -p test.properties
```

where the test.properties contains all the details of the clusters that the client is connecting to and also details the certs involved. Launching the sample clients puts the user into a command line environment with assorted options.

Please review the Readme.txt in the client folder for more information on the available commands.

11.1 Sample client commands

11.1.1 GetToken

If the certs have been configured and the test.properties file populated properly – a token can be acquired using the following command:

```
getToken
```

Similarly, if public client authentication is configured, a token can be acquired using:

```
getPublicToken
```

11.1.2 Create

```
usage: create -n integer [-r integer] [>|>> file]

    create [-r integer] << file

    create contactId
```

options:

```
 -n <integer> Number of contacts to create

 -r <integer> Request rate in contacts per second, default is 1
```

So, for example to create 3 contacts, issue the command create –n,3

11.1.3 Drop

```
usage: drop [-r integer] << file
```

drop contactId

options:

-r <integer> Request rate in contacts per second, default is 1

11.1.4 Attributes

This returns all the available attributes.

11.1.5 Routepoints

This returns all the available Routepoints.

11.1.6 Event subscriptions

Usage : events –s new

This creates a new subscription to the event parameters defined in the properties file.

Usage : events

This pumps the received events to the screen.

Usage : events –s del

This deletes the event subscription.

11.1.7 Pass in Existing WorkRequestID

Usage: create -c ExistingContext -i NewContact

where the -c passes in an Existing Context/WorkRequestID

12 Troubleshooting

A basic test would be to see if the same problem exists with the reference client provided.

A new logfile is created for the GenericChannelAPI svar. This is found at /var/log/Avaya/dcm/services/GenericChannel on each node the GenericChannelAPI svar is deployed on.

The GenericChannelAPI calls directly into the ORCRestService API so the ORCRestService logs would be useful to investigate in the event of an error.

For contact routing the standard Oceana logs – UCM, Engagement Designer, WAE, Contact Center Service would be worthy of examination – Opening the instances of the GenericChannel flow in Engagement Designer would also be useful.

For authentication issues, an examination of the AuthorizationService logs on the UAC nodes would be useful. Examination of the contents of the clienttruststore and clientkeystore using the keytool –list command would be important. Examine carefully if the cert from UAC node is present in the client trust store. Please note as per described above that the time on any client machine needs to be synched exactly with the time on the Breeze UCA nodes where the Authorization Service is running.

For evening framework issues, an examination of the eventing connector logs would be useful. An examination of the logs of the local httpserver would also be useful.

13 Limitations of currently implemented solution

1) Oceana only supports 1 Generic Channel provider per customer site but customers can differentiate between groups of contacts by assigning different attributes to different sets of contacts (and also configuring different screenpops for these different sets of attributes).
2) No transport of media for Generic Channel contacts is supported – customers need to handle media themselves.
3) No support for hold/unhold/transfer to user/conference/coach/observe/barge-in of Generic Channel contacts. Capabilities will be defined and be unchangeable ie agents can answer, transfer to service and drop Generic Channel contacts.
4) An individual agent can only be connected to Oceana via Workspaces OR a custom client – the same agent cannot be connected via simultaneous Workspaces and custom clients. Note Oceana will support Workspaces and custom clients for different agents at the same customer site. So if a customer wants to write a custom client for handling Generic Channel contacts in a particular manner, the custom client will need to handle all contact types if the agent has more than one channel type assigned to them.
5) Oceana has implemented limits on rate of contact creation and numbers of Generic Channel contacts in the system - these are dependent on the GenericChannel svar attribute regarding number of Generic Channel agents licensed.

Oceana 3.6	**On-Prem only**	**On-Prem only**	**AWS and On-Prem**	
Capacity Metric	3000 Agent Oceana footprint	2000 Agent Oceana footprint	1000-Agent Oceana Footprint	100-Agent Oceana Footprint
Max supported generic channel contacts per hour (Assumes no other channel is active)	12000	12000	6000	600
Max Number of concurrent generic channel sessions per agent	3	3	3	3
Max Number of Active generic channel Agents	3000	2000	1000	100
Max queued generic channel contacts	10000	10000	10000	1000
Max allowed rate of incoming create/drop contact requests per node	10	10	10	4

6) No out-of-the-box support for integration with Generic Channel desktop contact recording solutions. Customers have the facility to create their own widgets and screen-pops/desktop clients.
7) No support of additional information other than the Contact ID being passed in to the contactCreate command – this limitation is tempered by the possibility of passing in a pre-existing ContextID in to the createContact request.

LAST PAGE