Creating Dynamic Tasks for Avaya Engagement Designer

Issue 2
July 2016

REVISED: March 24, 2016

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 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.

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 or to 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 "Open Source Software" or "OSS" is as defined by the Open Source Initiative ("OSI") and is software licensed under an OSI approved license as set forth at http://www.opensource.org/docs/osd (or such successor site as designated by OSI).

1.9 "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.

1.10 "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.

1.11 "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.

1.12 "Sample Application Code" means Software provided for the purposes of demonstrating functionality of an Avaya product through the Avaya Software Development Kit.

1.13 "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 the Sample Application Code with Permitted Modifications, provided that such distribution is subject to an end user license agreement 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.

C. Except as expressly authorized by this Agreement, and unless otherwise permitted by the applicable law, Licensee acknowledges and agrees that the foregoing license does not include any right to distribute, license, translate, publish, or display the SDK, Specification Documents or Documentation or any copy or part thereof. Licensee represents and warrants that it will not use, modify, or distribute the redistributable Client Libraries in any manner that causes any portion of the redistributable Client Libraries that is not already subject to an OSS license to become subject to the terms of any OSS license.

D. 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).

E. 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.

F. 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.

2.2 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, trade marks 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: https://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.

2.6 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.

2.7 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 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.

2.13 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 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, black boxes, malware, trapdoors, and other mechanisms to allow remote/hidden attacks or access through unauthorized computerized command and control, and will not contain any other computer software routines designed to spy, monitor traffic (network sniffers, keyloggers), 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, 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.

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, worldwide license under any and all of Licensee's Intellectual Property rights related to any Permitted Modifications, to use, employ, practice, make, have made, sell, and/or otherwise exploit any and all Permitted Modifications.

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 derivative 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 derivative 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 return to Avaya all Confidential Information (including copies thereof) in Licensee's possession, custody, or control upon termination of this Agreement at any time and for any reason. 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 AND WILLFUL MISCONDUCT, 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 AND WILLFUL MISCONDUCT, 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 indemnify and hold harmless Avaya, Affiliates and their respective officers, directors, agents, suppliers, customers and employees from and against all claims, damages, losses, liabilities, costs, expenses, and fees (including fees of attorneys and other professionals) 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, including, but not limited to, products liability claims and claims of infringement of third party Intellectual Property rights.

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 of this Agreement, Licensee will immediately cease using the SDK, and Licensee agrees to destroy all adaptations or copies of the SDK and Documentation, or return them to Avaya upon termination of this License.

10.5 The rights and obligations of the parties contained in Sections 2.3, 2.6, 2.7, 2.10, 2.11, 3, and 5 through 18 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.

Licensee shall comply with all applicable laws and regulations, including without limitation those applicable to data privacy, intellectual property, trade secret, fraud, music performance rights and the export or re-export of technology and will not export or re-export the SDK or any other technical information provided under this Agreement in any form in violation of the export control laws of the United States of America and of any other applicable country. For more information on such export laws and regulations, Licensee may refer to the resources provided in the websites maintained by the U.S. Commerce Department, the U.S. State Department and the U.S. Office of Foreign Assets Control.

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.

15.0 GOVERNING LAW AND DISPUTE RESOLUTION.

This Agreement and any dispute, claim or controversy arising out of or relating to this Agreement (“Dispute”), including without limitation those relating to 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.

Any Dispute shall be resolved in accordance with the following provisions. The disputing party shall give the other party written notice of the Dispute. The parties will attempt in good faith to resolve each Dispute within thirty (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. 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 these procedures and within these timeframes, 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 any or all other parties 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 this Agreement 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 each party will 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 shall 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.

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 above, 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 above with regard to arbitration of Disputes that arise anywhere other than in the United States or are based upon an alleged breach committed anywhere other than in the United States, each party to this Agreement consents to the exclusive jurisdiction of those courts, including their appellate courts, for the purpose of all actions and proceedings.

The parties agree that the arbitration provision in this section 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. Nothing in this section 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. In addition and notwithstanding the foregoing, Avaya shall be entitled to take any necessary legal action at any time, including without limitation seeking immediate injunctive relief from a court of competent jurisdiction, in order to protect Avaya's intellectual property and its confidential or proprietary information (including but not limited to trade secrets).

16.0 IMPORT/EXPORT CONTROL.

Licensee is advised that the SDK is of U.S. origin and subject to the U.S. Export Administration Regulations (“EAR”). The SDK also 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 SDK 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 SDK 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 SDK 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.

17.0 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.

18.0 ENTIRE AGREEMENT.

This Agreement, its exhibits 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.

19. 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.

Contents

Chapter 1: Creating dynamic tasks for Avaya Engagement Designer

Introduction to dynamic tasks

Dynamic tasks allow you to program unique tasks to meet your business workflow requirements. You can add dynamic tasks to an existing drawer on the Engagement Designer palette, or create a new drawer for your tasks. Dynamic tasks can be added to the palette any time after Engagement Designer has been installed. The underlying technology for dynamic tasks is OSGi, which is a standard equivalent to Java EE. Avaya Breeze™ supports the Enterprise Bundle Service (eba) svar file for Engagement Designer and Composite Bundle Archive (cba) svar files for tasks.

You must have Java programming experience to create dynamic tasks.

Prerequisites

- Install and configure Avaya Breeze™. For instructions see *Deploying Avaya Breeze™*.
- Install and configure Engagement Designer. For instructions see *Avaya Engagement Designer Snap-in Reference*.
- Install and configure Eclipse or any development IDE.
- Install and configure Apache Maven.
- Install the Avaya Breeze™ SDK. For instructions see *Getting Started with the Avaya Breeze™ SDK*.

Installing the Engagement Designer SDK

About this task

To install the Engagement Designer SDK, you must first download it. If you have an Avaya SSO login and access to PLDS, you can download the SDK from PLDS. Or, you can download it from Avaya DevConnect. For information about downloading software from PLDS, see *Deploying Avaya Breeze™*. To download the SDK from the DevConnect site, following the steps in this procedure.

Procedure

1. Download the Engagement Designer SDK from DevConnect.

   a. Go to DevConnect: Avaya Engagement Designer.

   b. In the left navigation pane, select **Releases**.

   c. Select **Downloads**.
   d. Click **Avaya Engagement Designer Dynamic Tasks SDK**.
   e. Accept the license agreement.
   f. Click the **Download** link to download the SDK.

2. Unzip the SDK to a directory or folder, for example, `Avaya-ED-SDK`.
3. Navigate to the SDK directory.
4. For Linux or Mac OS, run bash `install.sh`. For Windows, run `install.bat`.
5. Accept the EULA agreement by entering y.

   The SDK components are set up into your local repository.

Creating the scaffolding

Procedure

1. Navigate to the directory where the dynamic task must be created.
2. Run the following command: **`mvn archetype:generate -DarchetypeCatalog=local`**
3. Choose the archetype, by selecting `local->com.avaya.workflow.sdk:task-archetype (Engagement Designer Task Archetype).`
4. Define a value for property 'groupId' (for example: com.avaya).

   The groupid property identifies the project uniquely across all projects. The value must follow the Java package name rules; it must be at least a domain name that the user controls.
5. Define value for property 'artifactId' (for example: watasks).

   The artifactId is the name of the jar without version.
6. Define value for property 'version’ (for example: 1.0).
7. Define value for property 'package' (for example: com.avaya.watasks).
8. Confirm properties configuration.

   For these examples:

   groupId: com.avaya
   artifactId: watasks
   version: 1.0
   package: com.avaya.watasks
   serviceName: SampleTask
   serviceVersion: 1.0.0.0.0
   Y: :

   This procedure generates the following sample directory structure:

A parent directory named `watasks` with the following contents:

`tasks` folder
`task-snapin` folder
pom.xml file (generated by Maven)

Defining the task properties and icon

About this task

This is a multi-module project. The tasks folder is the project and contains the code. The tasks-snapin folder generates the cba (composite bundle archive) and svar (format supported by Avaya Breeze™).

Procedure

1. Open the pom.xml file in the parent directory and define common properties, like your unique `task-name`.
2. In the parent directory, open the `/tasks/pom.xml` file and define any other dependencies of your task.
3. Navigate to the `<parent directory>/tasks/resources` directory and open the `tasks.json` file.
4. Edit properties in the `tasks.json` file.

   In this file edit properties like the task name, task drawer name, banner color, and others. See Task properties description on page 16 for a description of the different task properties that you can edit. See Sample dynamic task.json file on page 10 for sample code you can use as a reference.
5. Optionally, update the png icon file for the task in the `<parent directory>/tasks/resources` directory.
6. Optionally, update the HTML files for task help and the index.html file for the entire bundle in the `<parent directory>/tasks/resources` directory.

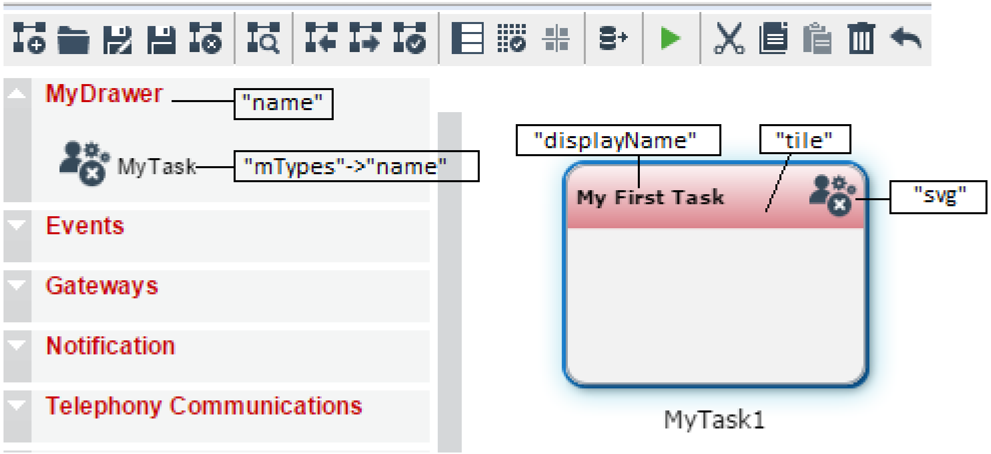
Sample dynamic task.json file

The following is a sample of a task.json file that defines task properties. It creates a new drawer "MyDrawer" on the Engagement Designer palette. The new drawer includes the new task type "MyTask," which displays with the name "My First Task." Two Java classes, HelloModel.java and HelloExecution.java are generated in the example.

```
[
{
        "name": "MyDrawer",
        "mTypes": [
            {

                "name": "MyTask",
                "snapinVersion": "3.0",
                "displayName": "My First Task",
                "tag": "hello",
                "svg": "hello",
                "helpUrl": "Hello.html",
                "tile": "red",
                "serviceName": "CollabDesigner",
                "custom": "true",
                "nodeType": "activity",
                "modelClass": "com.avaya.watasks.HelloModel",
                "executionClass": "com.avaya.watasks.HelloExecution",
                "properties": [
                ],
                 "input_schema": "{title: InputSchema,
                                  type: object,
                                  properties: {title
                                  :{type: string}
                                  }
                                 }",
                 "output_schema": "{title: OutputSchema,
                                   type: object,
                                   properties: {concatenatedStrings:{type: string}
                                   }
                              }"
                         }
                    }"
             }
        ]
    }
]
```

This file defines the following task. See Task properties description on page 16 for a definition of the use of each property used in this sample.



Sample dynamic task – ReadNumbers

This sample ReadNumbers dynamic task is useful for several solutions. It reads the digits entered individually. For example, 8421 is read as eight, four, two, one instead of as the default eight thousand four hundred twenty-one. The task has many use case, for example, reading an account number to bank customers, or confirming a phone number associated with an account.

Every task has a tasks.json file, model, and execution class, like the samples provided here. You can use these samples as a reference to create properties for your task and to write your model and execution classes.

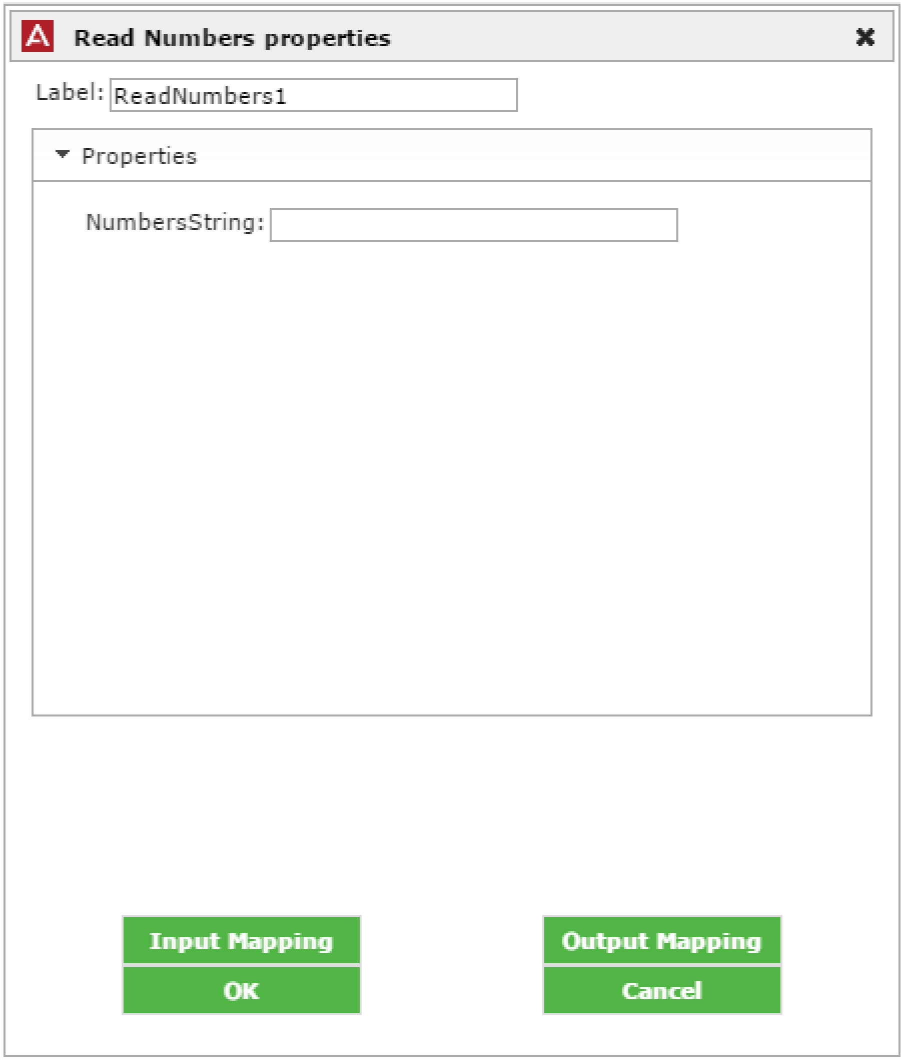
Tasks.json

```
[
  {
    "name": "MyDrawer",
    "mTypes": [
       {
        "name": "Read Numbers",
        "snapinVersion": "3.0",
        "displayName": "Read Numbers",
        "tag": "ReadNumbers",
        "svg": "ReadNumbers",
        "tile": "green",
        "serviceName": "CollabDesigner",
        "custom": "true",
        "nodeType": "activity",
        "modelClass": "com.avaya.readnumbers.ReadNumbersModel",
        "executionClass": "com.avaya.readnumbers.ReadNumbersExecution",
        "properties": [{
          "name": "NumbersString",
          "displayPropName": "NumbersString",
          "type": "String",
          "controlType": "text",
          "modelName": "setNumbersString",
          "mapping": "numbersString"
        }
        ],
        "mapping": "[\"input\", \"output\"]",
        "input_schema": "{title: ReadNumbersInputSchema,
                          type: object,
                          properties: {  numbersString:{type:string}  } }",
        "output_schema": "{title: ReadNumbersOutputSchema,
                           type: object,
                           properties: { result: { type: string  }  }  }"
         }
       ]
   }
 ]
```

The properties tag defines the following task properties.

```
"properties": [{
          "name": "NumbersString",
          "displayPropName": "NumbersString",//display name
          "type": "String",
          "controlType": "text",
          "modelName": "setNumbersString",//setter method in model class
          "mapping": "numbersString" //name of mapping in model and execution class
        }
```

This is how the NumbersString property looks:



Model class

```
package com.avaya.readnumbers;
//Dependencies can be added in pom, as described at the end of this section.
import com.roobroo.bpm.model.BpmNode;
import com.roobroo.bpm.util.WFUtil;
import org.apache.commons.lang.StringUtils;
import java.util.List;
@SuppressWarnings("serial")

//Detail on BpmNode is provided in the section BpmNode API class.
public class ReadNumbersModel extends BpmNode {

        public ReadNumbersModel(String name, String id) { super(name, id);}

        private String numbersString;

        //This is the setter method as defined under “modelName” of the property
        “NumbersString” in tasks.json
        public void setNumbersString(String numbers){this.numbersString=numbers;}

        public String getNumbersString(){ return numbersString;}
        @Override
        public boolean validateProperties(List<String> w, List<String> e){
                boolean isValid = true;

               //Mandatory parameter numbersString, checked that the property numberString
               has an assigned value either through mapping or as an input to the task.
               //WFUtil.validateMapping has been used below with the last parameter
               "numbersString" as defined in tasks.json's properties under 'mapping'
               //WFUtil.validateEmptyProperty has been used below with second parameter
               as the private member "numbersString" of this class.
                 if((! WFUtil.validateMapping(w, e, getDataInputAssociations(),
                 "numbersString"))&&(! WFUtil.validateEmptyProperty(numbersString,
                 "numbersString", e))){

                        isValid = false;
                        return super.validateProperties(w, e) && isValid;
                 }
                 if(StringUtils.isNotEmpty(getNumbersString()) &&
                 (! WFUtil.validateMapping(w, e,
                 getDataInputAssociations(), "numbersString")))
                 {
                        try
                        {
                              Integer.parseInt(getNumbersString());
                        }
                        catch(NumberFormatException nfe)
                        {
                              e.add("Invalid number format");
                              isValid=false;
                         }
                  }
        return super.validateProperties(w, e) && isValid;
        }
}
```

Execution class

```
package com.avaya.readnumbers;

import org.apache.commons.lang.StringUtils;
import org.json.JSONObject;
import com.avaya.app.entity.Instance;
import com.avaya.app.entity.NodeInstance;
import com.roobroo.bpm.model.BpmNode;

//Detail on NodeInstance is provided in the section NodeInstance API class.

@SuppressWarnings("serial")
public class ReadNumbersExecution extends NodeInstance {

       public ReadNumbersExecution(Instance instance, BpmNode node) {
              super(instance, node);
       }

       public Object execute() throws Exception {
              //Get node from Model Class
              ReadNumbersModel rnm=(ReadNumbersModel)getNode();

              //Get the value form tasks.json
              String num = (String) get("numbersString");

              //This is the json object that is returned after execution of task and can
              be viewed by clicking on the task in the instance generated under admin
              console.
              JSONObject obj=new JSONObject();

              //Mapping is given priority in this case, if it is empty, then the value
              entered in the task(coming from model class) is used.
              if(num == null || num.isEmpty())
              { num = rnm.getNumbersString();}

              if(StringUtils.isNotEmpty(num))
              { try
                { Integer.parseInt(num);}
                catch(NumberFormatException nfe)
                { throw new IllegalArgumentException("The number is not valid");}

                String[] a;
                a=num.split("");
                String b=StringUtils.join(a,",",1,a.length);
                obj.put("result",b); }
                else
                { obj.put("result","");
                  throw new IllegalArgumentException("Empty string");}
                return obj;}
}
```

Add dependencies

You can add dependencies under `<parent directory>/tasks/pom.xml.`

- ReadNumbersModel class has a dependency on org.apache.commons.lang.StringUtils. Add the following under the <dependencies> tag:

```
           <dependency>
                 <groupId>commons-lang</groupId>
                 <artifactId>commons-lang</artifactId>
                 <version>2.6</version>
           </dependency>
```

- ReadNumbersModel imports packages com.roobroo, ReadNumbersExecution imports com.avaya.app.entity,javax.persistence. These dependencies are there by default in the scaffold project generated by the SDK. You can add this using:

```
<Import-
Package>javax.persistence,org.json,com.avaya.app.entity,com.avaya.workflow.*,com.roob
roo.*;
!*</Import-Package>
```

Task properties description

The following properties are defined in the tasks.json file in the `/task/resources` directory.

Task properties

Property	Description
name	The task name displayed on the palette.
snapinVersion	The task snap-in version number

Table continues…

Property	Description
displayName	The task name displayed when the task is dragged onto the canvas.
tag	How the task appears in the bpmn file. Spaces are not allowed in the tag.
svg	The name of the png file that has the icon image for the task. Do not include the file extension in the name. Add the icon png file to the resources folder.
helpUrl	The reference to the help file in the `resources/help` folder. The help is accessed by clicking the help icon on the task properties window. (You may need to create the `resources/help` folder.)
tile	Color of the top band of the task displayed on the canvas. Choose from the following colors: red, green, purple, gold, or blue.
serviceName	CollabDesigner. This value remains constant.
custom	Indicates if this is a custom task. For a dynamic task, this is always “true.”
nodeType	The type of task: activity.
modelClass	The name of model class for the task. It is defined by user under `<parent directory/tasks/src/<package name defined during creation>`.
executionClass	The name of execution class. It is defined by user under `<parent directory/tasks/src/<package name defined during creation>`.
properties	All the properties that display in the task properties window.
input schema	JSON schema that defines the expected input for the task.
output schema	JSON schema that defines the expected output of the task.
displayProperties	JSON definition of indicating what is shown in the display tile of the activity. That is, when the task is dragged and dropped on the palette, what is shown before clicking the task to open the properties window. There are two subsections: mapping and properties. If a mapping is available for the input schema variable 'mediauri', it will be used as the display value for the property "Media-URI" defined in the task type. The 'properties' section indicates that if there are no mappings done, then the content of the first available property from this list will show in the tile display.

Values that describe properties of fields displayed in the task properties window

Property	Description
name	Name of the property displayed in the task properties window. Spaces are not supported.
controlType	The type of control used for the property. See list of User Interface controls below.
type	The type of data the property stores.
modelName	A data variable name in the model class where the entered value is stored in the model. For example, a setter function that sets the value of this property.
fillerModel	The model class name that supplies the entries in the drop down box.
mapping	The variable in the input schema associated with the control.
textLines	The number of lines in a multiline text box. The default is 7.
clickerModel	Model class that handles the button click.
clicker	A function name in the model class that gets executed when the button is clicked.
clickerParam	Parameters in JSON format passed to the clicker function.
clickerLabel	The name that appears on the button.
clickerTextField	Another field name defined in the same property dialog that can display any return strings from the clicker function.

User interface controls supported for task properties

Control type	Property type	Description
"combo"	"String"	Indicates that the property's control is a drop-down combo box.
"text"	"String"	Indicates that the property's control is a simple text box.
"multilineText"	"String"	Indicates that the property is a multi line text box.
"password"	"String"	Indicates that the property is a text box that accepts passwords (characters are masked).
"button"		Indicates that the property is a clickable button.
"checkbox"		Indicates that the property is a checkbox.

Table continues…

"datetime"	"String"	Indicates the property uses a calendar widget to get the date and time. The property value would be in the format of MM/DD/YYYY HH:MM (e.g. 12/24/2014 15:30). You can use the following Java code to convert the datetime string to Date object. Date date = new SimpleDateFormat("MM/dd/yyyy HH:mm")).parse("12/24/2014 15:30");

BpmNode API class

The HelloModel.java class extends the BpmNode class.

```
import com.roobroo.bpm.model.BpmNode;
public class HelloModel extends BpmNode {
       public HelloModel(String name, String id) {
             super(name, id);
             // TODO Auto-generated constructor stub
       }
}
```

This class defines the fields corresponding to the properties of the task. All the fields must be ‘public’.

NodeInstance API class

HelloExecution extends NodeInstance and implements the execute method.

```
import org.json.JSONObject;
import com.roobroo.bpm.im.Instance;
import com.roobroo.bpm.im.NodeInstance;
import com.roobroo.bpm.model.BpmNode;

public class HelloExecution extends NodeInstance {
        public HelloExecution(Instance instance, BpmNode node) {
                super(instance, node);
        }
        public Object execute() throws Exception {
                return new JSONObject("{title:Hello}");
    }
}
```

Execute can throw an exception that can be processed by Engagement Designer using an Error Boundary Event. The execute method returns an output that can be used in output mapping.

The NodeInstance API class has the following methods.

public Object get(String name);– Gets a particular part of input data, the part is specified in the json path syntax.

public BpmNode getNode(); – Returns the bpm node for which this is the executor.

public Instance getInstance(); – Returns the workflow instance.

public boolean isReconstruction(); – Checks whether the executor is being run a second time because there was a failover.

public void log(String message); – Log using the Engagement Designer logger.

public String subscribe*(String family, String type) throws JSONException; – A convenience methods to allow to subscribe to Eventing Framework.

public BpmNode getParentNode() throws JSONException; – If there is a parent workflow instance which called this instance, this is the reference to the parent bpm node.

OSGi task bundles

The task author can add as many tasks as wanted in a single bundle by adding the model and execution classes to the tasks.json directory. Also, add a png file for each task icon to the resources directory.

The tasks bundle contains the OSGi service, which implements the interface. This code is generated and generally does not need to be modified.

```
public interface TaskService {
       public String getName();
       Category[] getTaskCategories();  // for listing tasks available
       BpmNode getModel(String model, String name, String id);
       NodeInstance getExecutor(Instance instance, BpmNode node,
         String whatYouWantFromWhatever);
       byte[] getResource(String resourceName); // to get png, help files
       public ArrayList<String> getFillerData(String modelName, String filler,
         String strParam);
       public String executeClicker(String clickerModel, String clickerFunc,
         String strParam);
}
```

The task bundle is registered with OSGi using the OSGi blueprint file of their bundle. This code does not need to be modified.

```
<?xml version="1.0" encoding="UTF-8"?>
<blueprint xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0">
   <service id="myService">
       <interfaces>
        <value>com.roobroo.TaskService</value>
       </interfaces>
      <bean class="com.avaya.MyTaskService"/>
   </service>
</blueprint>
```

The code to make this an OSGi bundle is also generated in `<parent directory>task/tasks/pom.xml`.

```
                       <plugin>
                         <groupId>org.apache.felix</groupId>
                         <artifactId>maven-bundle-plugin</artifactId>
                         <version>2.4.0</version>
                         <extensions>true</extensions>
                         <configuration>
                            <manifestLocation>META-INF</manifestLocation>
                            <rebuildBundle>true</rebuildBundle>
                           <instructions>
                             <Export-Package>!*</Export-Package>
                             <Import-Package>com.avaya.workflow.*,com.roobroo.*;!*
                             </Import-Package>
                              <Bundle-Blueprint>OSGI-INF/blueprint.xml
                              </Bundle-Blueprint>
                             <Embed-Dependency>*;scope=!runtime;artifactId=!engine
                             </Embed-Dependency>
                             <Embed-Transitive>true</Embed-Transitive>
                              <Embed-Directory>target/dependency</Embed-Directory>
                              <Embed-StripGroup>true</Embed-StripGroup>
                           </instructions>
                         </configuration>
                       </plugin>
                        <plugin>
                          <artifactId>maven-dependency-plugin</artifactId>
                          <executions>
                            <execution>
                              <id>copy-dependencies</id>
                              <phase>package</phase>
                              <goals>
                                <goal>copy-dependencies</goal>
                              </goals>
                            </execution>
                          </executions>
                        </plugin>
```

The OSGi classloader loads classes from two places:

- When another bundle is exporting the package, it can be imported in the `<Import-Package>` instruction. This applies to packages that are exported by Avaya Breeze™ as part of the edp-osgi-api bundle as well.
- The jar can be included providing the class as a dependency in the pom. Only those jars will be included that are specified by the filter in `<Embed-Dependency>`.

Predefined task properties

The generated properties.xml file in the svar has predefined properties that are required for the cba to link to the Engagement Designer eba. There is no need to change these properties – you can add more properties if needed.

```
<?xml version="1.0" encoding="UTF-8"?>
<service xmlns="http://archiveschemas.aus.avaya.com/properties"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://archiveschemas.aus.avaya.com/properties properties.xsd"
        name="${serviceName}" version="${serviceVersion}"
application="${serviceName}-${serviceVersion}">
        <smgr>
                <description>My Service</description>
                <!--  orig_order>1</orig_order-->
                <!--  orig_group>1</orig_group-->
                <fs_component>true</fs_component>
                <attribute name="AssociatedBusinessLevelApplicationName">
```

```
<displayName>AssociatedBusinessLevelApplicationName</displayName>
                        <helpInfo>
                        </helpInfo>
                        <global>false</global>
                        <validation name="AnyString">
                                <type>STRING</type>
                        </validation>
                        <admin_visible>false</admin_visible>
                        <admin_changeable>false</admin_changeable>
                        <factory>
                                <value>EngagementDesigner</value>
                                <user_changeable>false</user_changeable>
                        </factory>
                        </attribute>
                        <attribute name="AssociatedBusinessLevelApplicationVersion">

<displayName>AssociatedBusinessLevelApplicationVersion</displayName>
                        <helpInfo>
                        </helpInfo>
                        <global>false</global>
                        <validation name="AnyString">
                                <type>STRING</type>
                        </validation>
                        <admin_visible>false</admin_visible>
                        <admin_changeable>false</admin_changeable>
                        <factory>
                                <value>3.1.0.0.9000</value>
                                <user_changeable>false</user_changeable>
                        </factory>
              </attribute>
              <attribute name="AssociatedCompositionUnitName">

<displayName>AssociatedCompositionUnitName</displayName>
                        <helpInfo>
                        </helpInfo>
                        <global>false</global>
                        <validation name="AnyString">
                                <type>STRING</type>
                        </validation>
                        <admin_visible>false</admin_visible>
                        <admin_changeable>false</admin_changeable>
                        <factory>
                                <value>engine-snapin-eba</value>
                                <user_changeable>false</user_changeable>
                        </factory>
                </attribute>
                <attribute name="CompositeBundleName">
                        <displayName>CompositeBundleName</displayName>
                <helpInfo>
                Composite bundle name that is the symbolic name of the cba which will
                be the name used by the local repository
                       </helpInfo>
                       <global>false</global>
                       <validation name="AnyString">
                               <type>STRING</type>
                               <type>STRING</type>
                       <admin_visible>false</admin_visible>
                       <admin_changeable>false</admin_changeable>
                       <factory>
                               <value>tasks-snapin-cba</value>
                               <user_changeable>false</user_changeable>
                       </factory>
                </attribute>
                <attribute name="CompositeBundleVersion">
```

```
                        <displayName>CompositeBundleVersion</displayName>
                        <helpInfo>
                Composite bundle version that is the bundle version of the
                cba which will be the name used by the local repository
                        </helpInfo>
                        <global>false</global>
                        <validation name="AnyString">
                                <type>STRING</type>
                        </validation>
                        <admin_visible>false</admin_visible>
                        <admin_changeable>false</admin_changeable>
                        <factory>
                                <value>3.1.0-SNAPSHOT</value>
                                <user_changeable>false</user_changeable>
                        </factory>
                </attribute>
        </smgr>
</service>
```

Building and deploying the task

Before you begin

Engagement Designer must be installed and configured before you can deploy new tasks.

About this task

When a new task is deployed on System Manager, Engagement Designer adds the task to the list of tasks available in the Engagement Designer palette. When a task is uninstalled, it is removed from the palette, and workflow definitions using that task show an error. When a new svar is uploaded that overrides an existing task, the new task is used for all existing and new workflow instances.

Procedure

1. Run the mvn package at the top level.

   This step builds the svar package for the task bundle. The task svar file is generated at the location: `watasks/tasks-snapin/tasks-snapin-svar/target`. The default name of the task svar file is: `SampleTask-1.0.0.0.0.svar`.

2. Load and install the task on System Manager.

   For instructions for installing a snap-in on System Manager see *Quick Start to Deploying Avaya Breeze™ Snap-ins*.

Chapter 2: Related resources

Documentation

See the following related documents at http://support.avaya.com.

Title	Use this document to:
Deploying Avaya Breeze™	Deploy and configure Avaya Breeze™.
Getting Started with the Avaya Breeze™ SDK	Install and use the Avaya Breeze™ SDK.
Avaya Engagement Designer Snap-in Reference	Deploy and configure Engagement Designer.
Avaya Engagement Designer Developer's Guide	Understand and use Engagement Designer.
Getting Started with the Avaya Engagement Designer	Learn the basics of Engagement Designer and how to create simply Workflow Definitions.

Related links

Finding documents on the Avaya Support website on page 24

Finding documents on the Avaya Support website

About this task

Use this procedure to find product documentation on the Avaya Support website.

Procedure

1. Use a browser to navigate to the Avaya Support website at http://support.avaya.com/.
2. At the top of the screen, enter your username and password and click **Login**.
3. Put your cursor over **Support by Product**.
4. Click **Documents**.
5. In the **Enter your Product Here** search box, type the product name and then select the product from the drop-down list.
6. If there is more than one release, select the appropriate release number from the **Choose Release** drop-down list.
7. Use the **Content Type** filter on the left to select the type of document you are looking for, or click **Select All** to see a list of all available documents.

For example, if you are looking for user guides, select **User Guides** in the **Content Type** filter. Only documents in the selected category will appear in the list of documents.

8. Click **Enter**.

Related links

Documentation on page 24

Avaya DevConnect

Avaya DevConnect provides additional resources for Engagement Designer developers. You can access documentation, videos, webinar recordings, tier 1 and tier 2 support as well as a developer forum. View the Engagement Designer DevConnect website at: DevConnect: Avaya Engagement Designer.

Index