



Avaya Solution & Interoperability Test Lab

Application Notes for eLoyalty Data Integration Service with Avaya Interaction Center 7.2 – Issue 1.0

Abstract

These Application Notes describe the configuration steps required to integrate the eLoyalty Data Integration Service with Avaya Interaction Center 7.2. eLoyalty Data Integration Service is used to import data from Avaya Interaction Center into the eLoyalty Behavioral Analytics database. The Behavioral Analytics Service is delivered as a managed service and is used with other contact center applications that collect data about customer interactions. This service allows companies to improve customer service by analyzing key attributes of customer service calls, such as call type, efficiency, first call resolution, and customer attitude. The objective of this interoperability compliance test was to verify the proper operation of the Data Integration Service with Interaction Center using a custom eLoyalty workflow. Other contact center applications were not included in the configuration.

Information in these Application Notes has been obtained through DevConnect compliance testing and additional technical discussions. Testing was conducted via the DevConnect Program at the Avaya Solution and Interoperability Test Lab.

1. Introduction

These Application Notes describe the configuration steps required to integrate the eLoyalty Data Integration Service with Avaya Interaction Center 7.2. eLoyalty Data Integration Service is used to import data from Avaya Interaction Center into the eLoyalty Behavioral Analytics database. The Behavioral Analytics Service is delivered as a manager service and is used with other contact center applications that collect data about customer interactions. This service allows companies to improve customer service by analyzing key attributes of customer service calls, such as call type, efficiency, first call resolution, and customer attitude. The Data Integration Service works in conjunction with other contact center applications, but only the interoperability between the eLoyalty Data Integration Service and Avaya Interaction Center were verified in this compliance test.

The Data Integration Service requires an eLoyalty custom workflow to be installed on Avaya Interaction Center (IC) and the Data Integration Service to be running on a separate server with Microsoft Message Queuing enabled. The Data Integration Service executes eLoyalty custom workflows via AIC WorkflowServices webservice. The custom eLoyalty workflow requests the contents of an Electronic Data Unit (EDU) corresponding to the EDU ID provided by the Data Integration Service.

The Data Integration Service works as follows:

- A call center application (not used in this compliance test) places a message with the call's EDU ID on Microsoft Message Queuing (MSMQ), which is enabled on the Data Integration Service server.
- Data Integration Service reads the MSMQ message and extracts the EDU ID.
- Data Integration Service requests the call data corresponding to the EDU ID using IC Web Services. The IC Web Service server must be configured on IC, which allows on-demand execution of custom workflows.
- IC fetches the eLoyalty custom workflow and the EDU data is retrieved. The requested EDU field values are then returned to the Data Integration Service.
- Data Integration Service formats the data and stores it in an XML file.
- The call data file (XML file) is imported into the Behavioral Analytics database for analysis. This database is hosted by eLoyalty. This step was not verified in the compliance test.

1.1. Interoperability Compliance Testing

Interoperability compliance testing included feature and serviceability testing. The feature testing focused on verifying that call data can be retrieved by the eLoyalty Data Integration Service from Avaya IC and stored in an XML file. The Data Integration Service used IC Web Services to request IC to run the custom workflow, which in turn will retrieve the call data as an EDU (Electronic Data Unit) and send the requested data fields to the Data Integration Service. The call data was requested by placing a manually generated EDUID message on the Data Integration Service MSMQ.

The serviceability testing focused on verifying that the Data Integration Service would start up automatically when the server is rebooted and that messages in the MSMQ are serviced if the Data Integration Service starts up after the message was queued.

1.2. Support

For technical support on the eLoyalty Data Integration Service, contact eLoyalty via phone, email, or internet.

- **Phone:** +1 877.2ELOYAL
- **Email:** BehavioralAnalytics@eloyalty.com
- **Web:** www.eloyalty.com

Reference Configuration **Figure 1** illustrates the configuration used for testing. In this configuration, a call center environment consisting of Avaya Interaction Center (IC), Avaya Application Enablement Services, and Avaya Aura™ Communication Manager is established. Communication Manager routes incoming calls to Avaya IC agents and Application Enablement Services is used to establish a CVLAN link between Communication Manager and IC. eLoyalty Data Integration Service uses the IC Web Services to request call data from IC, which runs the eLoyalty custom workflow to retrieve the call data (EDU) and return it the Data Integration Service.

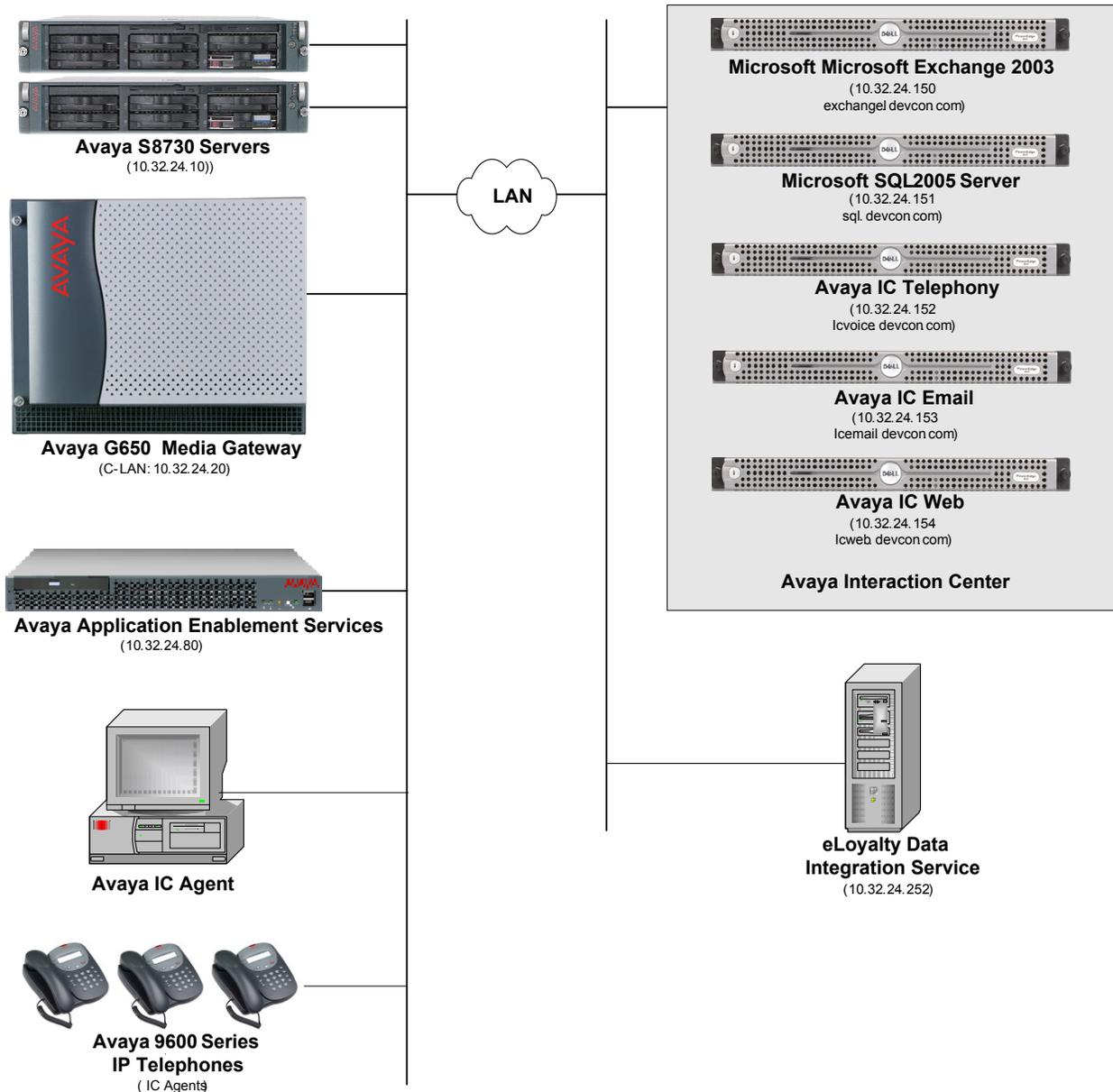


Figure 1: Configuration with Avaya IC and eLoyalty Data Integration Service

1.3. Equipment and Software Validated

The following equipment and software were used for the sample configuration:

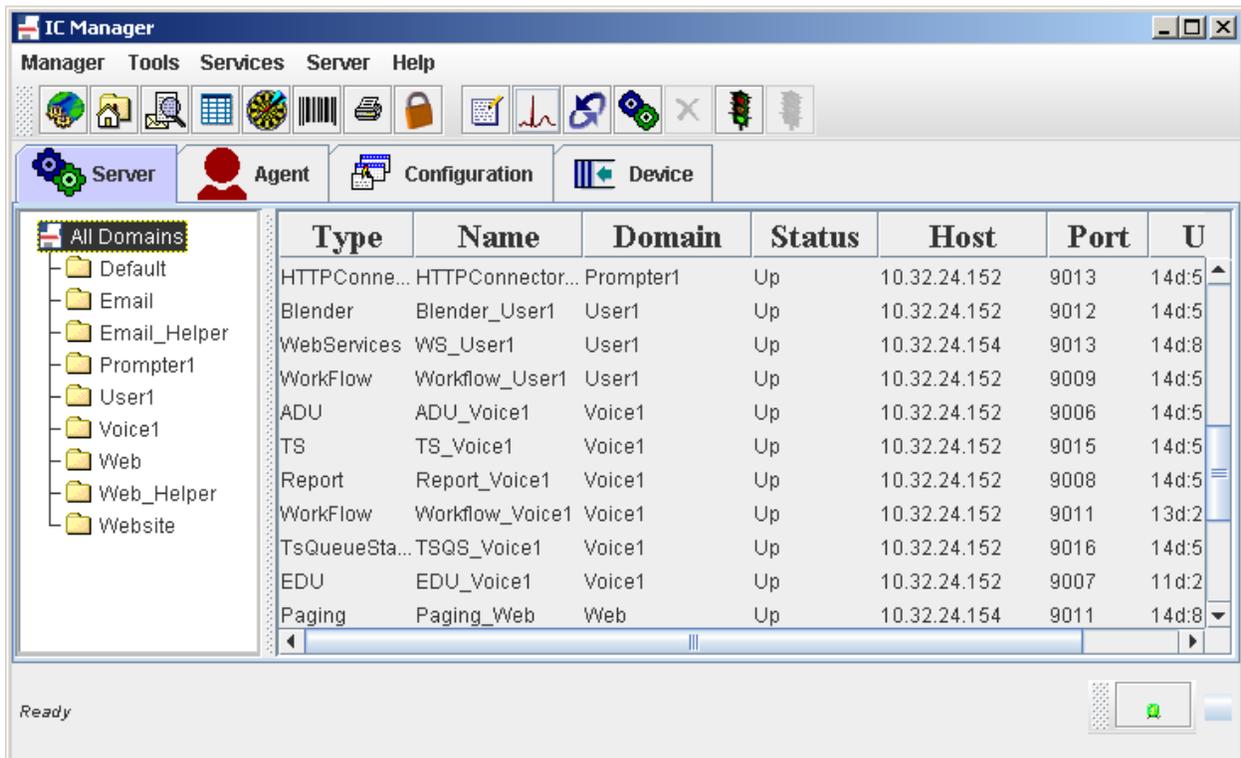
Equipment	Software
Avaya Interaction Center	7.2
Avaya Application Enablement Services	4.2.3
Avaya S8730 Servers with an Avaya G650 Media Gateway	Avaya Aura™ Communication Manager 5.2 (R015x.02.0.947.3) with Service Pack 3 (Patch 17579)
Avaya 9600 Series IP Telephones	3.0 (H.323)
eLoyalty Data Integration Service running on Microsoft Windows 2003 Server with Microsoft Message Queuing Enabled	3.0.0.4993

2. Configure Avaya Interaction Center

This section covers the configuration of Avaya IC. It is assumed that Avaya IC has been installed and configured as described in [1] and [2]. This solution only requires the voice media channel to be configured, including IC Web Services. IC Web Services allows on-demand execution of custom workflows and is used by the eLoyalty Data Integration Service. In addition, IC requires a CVLAN link to Application Enablement Services, which in turn has a CVLAN link to Communication Manager. Communication Manager receives incoming ACD calls and routes the call to an IC agent. The call data (EDU) is stored in IC memory and can be retrieved using the EDU ID. The duration that the data spends in IC memory is configurable in the EDU server.

Once IC is up and running, the eLoyalty custom workflow needs to be copied to the server running Avaya Workflow Designer and then the custom workflow needs to be built. This section covers how to build the eLoyalty custom workflow. Before building the custom workflow, verify that the IC servers are up and running via the IC Manager and that IC Web Services is operational as described below.

First, check the status of the IC servers and verify that they are all “Up”.

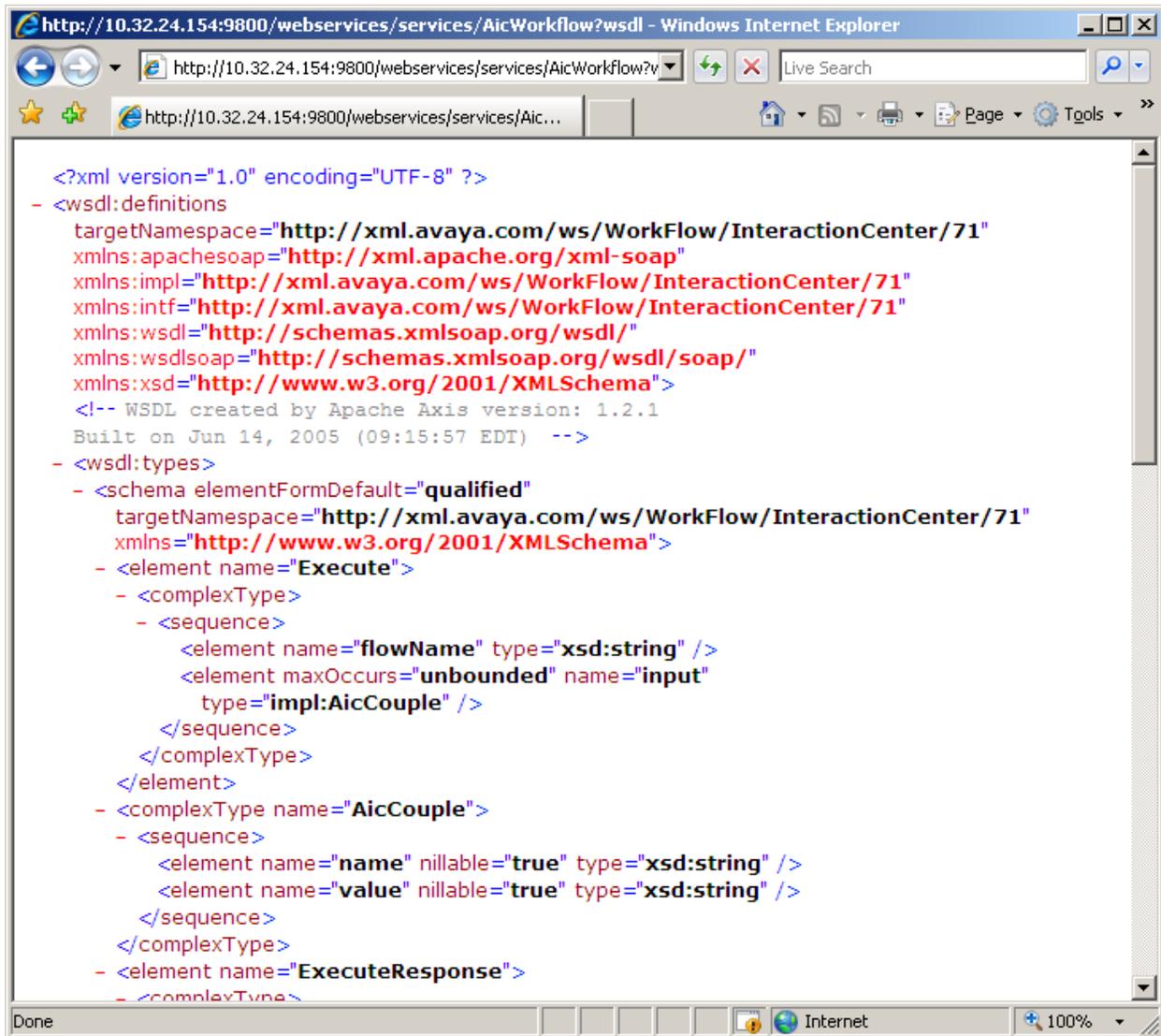


The screenshot shows the IC Manager application window. The window title is "IC Manager" and it has a menu bar with "Manager", "Tools", "Services", "Server", and "Help". Below the menu bar is a toolbar with various icons. The main area is divided into tabs: "Server", "Agent", "Configuration", and "Device". The "Server" tab is selected, and it shows a tree view on the left with "All Domains" expanded to show a list of folders: "Default", "Email", "Email_Helper", "Prompter1", "User1", "Voice1", "Web", "Web_Helper", and "Website". The main pane displays a table with the following columns: "Type", "Name", "Domain", "Status", "Host", "Port", and "U". The table contains the following data:

Type	Name	Domain	Status	Host	Port	U
HTTPConne...	HTTPConnector...	Prompter1	Up	10.32.24.152	9013	14d:5
Blender	Blender_User1	User1	Up	10.32.24.152	9012	14d:5
WebServices	WS_User1	User1	Up	10.32.24.154	9013	14d:8
WorkFlow	Workflow_User1	User1	Up	10.32.24.152	9009	14d:5
ADU	ADU_Voice1	Voice1	Up	10.32.24.152	9006	14d:5
TS	TS_Voice1	Voice1	Up	10.32.24.152	9015	14d:5
Report	Report_Voice1	Voice1	Up	10.32.24.152	9008	14d:5
WorkFlow	Workflow_Voice1	Voice1	Up	10.32.24.152	9011	13d:2
TsQueueSta...	TSQS_Voice1	Voice1	Up	10.32.24.152	9016	14d:5
EDU	EDU_Voice1	Voice1	Up	10.32.24.152	9007	11d:2
Paging	Paging_Web	Web	Up	10.32.24.154	9011	14d:8

The status bar at the bottom of the window shows "Ready" and a small green icon.

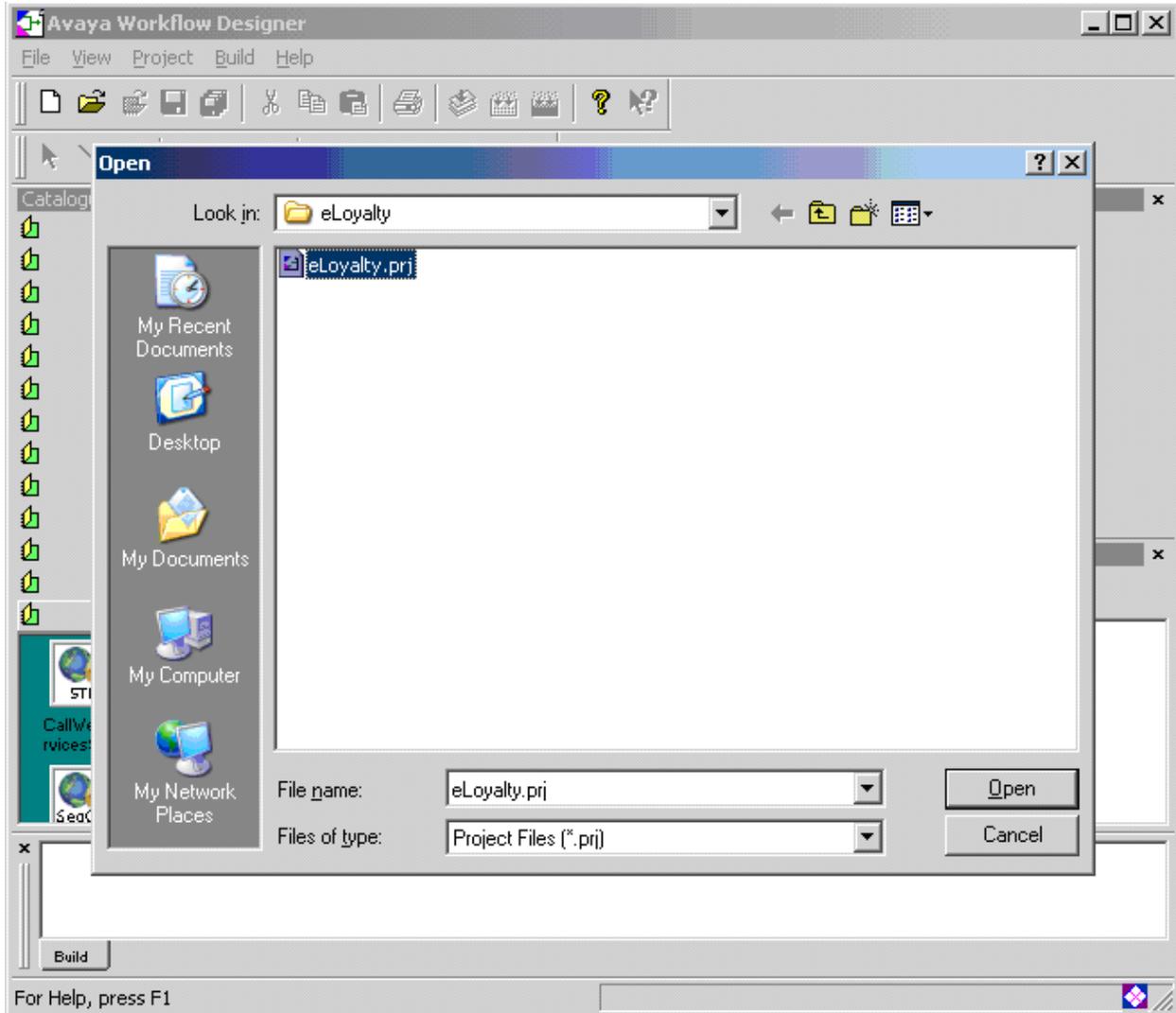
Next, verify that IC Web Services is operational by entering the following URL in a web browser: <http://<AICWSaddress>:9800/webservices/services/AicWorkflow?wsdl>, where <AICWSaddress> is the IP address of the server hosting IC Web Services. 9800 is the standard installation port for IC Web Services. If IC Web Services is operational, the web browser should display the WSDL for IC Workflow Services as shown below.



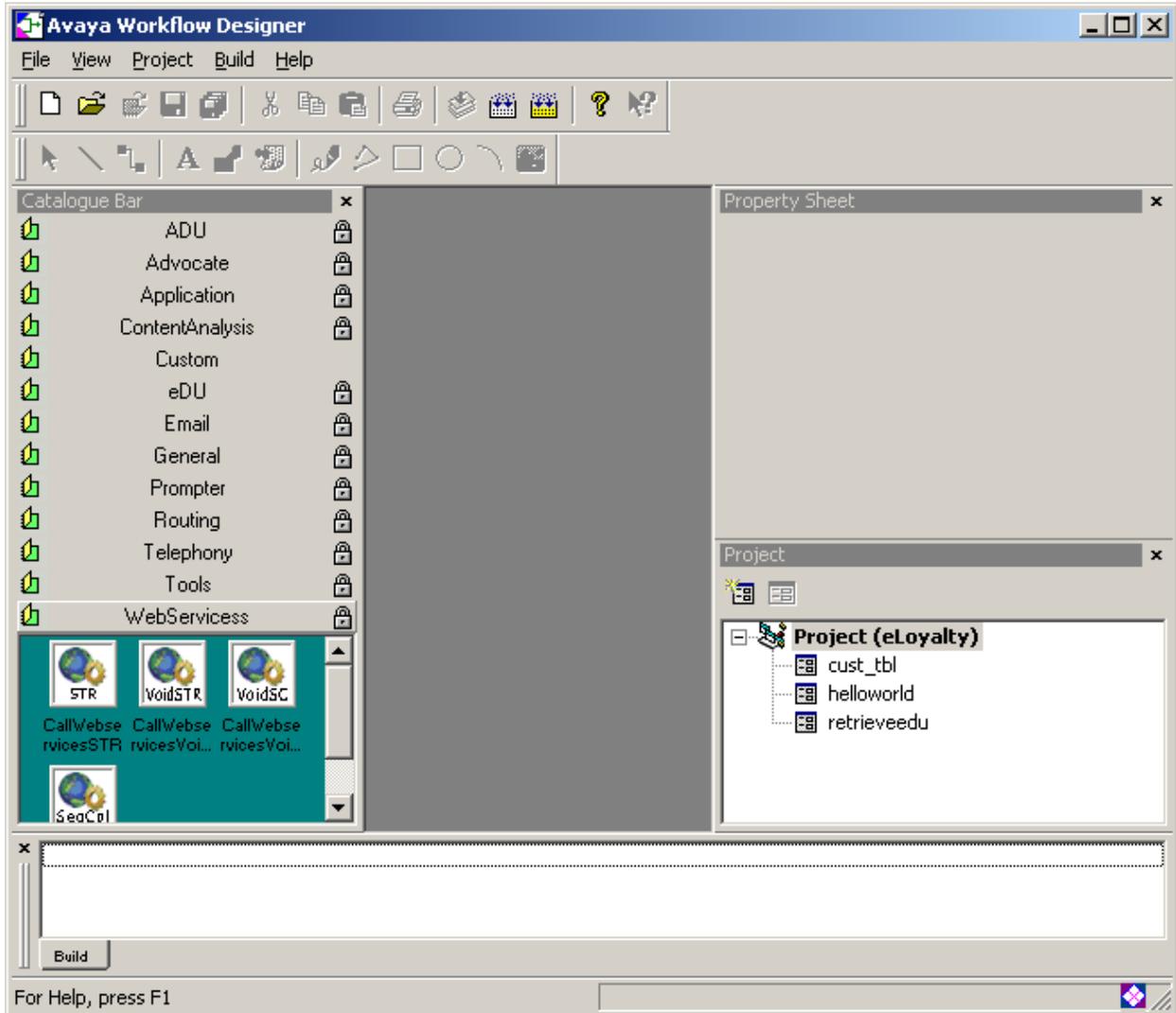
The screenshot shows a Windows Internet Explorer browser window displaying the WSDL for IC Workflow Services. The address bar shows the URL: <http://10.32.24.154:9800/webservices/services/AicWorkflow?wsdl>. The main content area displays the following XML code:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions
  targetNamespace="http://xml.avaya.com/ws/WorkFlow/InteractionCenter/71"
  xmlns:apachesoap="http://xml.apache.org/xml-soap"
  xmlns:impl="http://xml.avaya.com/ws/WorkFlow/InteractionCenter/71"
  xmlns:intf="http://xml.avaya.com/ws/WorkFlow/InteractionCenter/71"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <!-- WSDL created by Apache Axis version: 1.2.1
  Built on Jun 14, 2005 (09:15:57 EDT) -->
- <wsdl:types>
- <schema elementFormDefault="qualified"
  targetNamespace="http://xml.avaya.com/ws/WorkFlow/InteractionCenter/71"
  xmlns="http://www.w3.org/2001/XMLSchema">
- <element name="Execute">
- <complexType>
- <sequence>
  <element name="flowName" type="xsd:string" />
  <element maxOccurs="unbounded" name="input"
    type="impl:AicCouple" />
</sequence>
</complexType>
</element>
- <complexType name="AicCouple">
- <sequence>
  <element name="name" nillable="true" type="xsd:string" />
  <element name="value" nillable="true" type="xsd:string" />
</sequence>
</complexType>
- <element name="ExecuteResponse">
- <complexType>
```

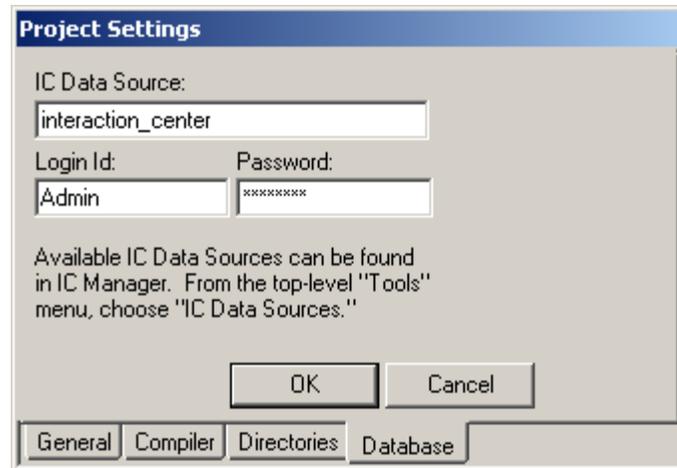
After verifying that Avaya IC is up and running, including IC Web Services, copy the eLoyalty custom workflow to a directory accessible by Avaya Workflow Designer. Launch Workflow Designer. In Workflow Designer, navigate to File→Open Project and open the eLoyalty custom workflow as shown below.



After opening the eLoyalty custom workflow, the Workflow Designer window appears as follows.

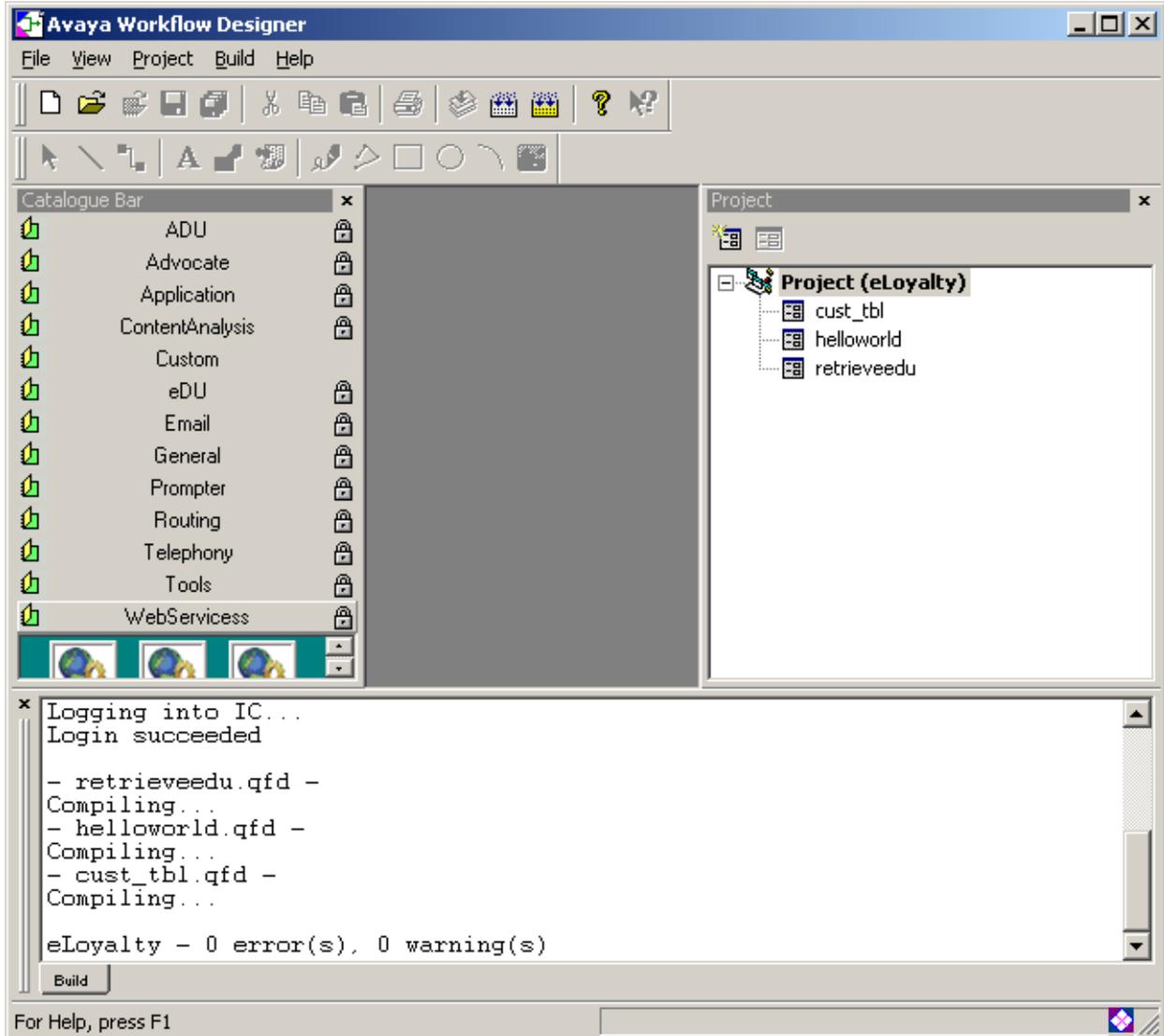


Next, select Project→Project Settings from the menu and go to the **Database** tab in the **Project Settings** window. Set the **IC Data Source** field to the appropriate value. In this configuration, it was set to *interaction_center*, which was used as the IC data source in the **Workflow_Voice1** server. Provide the login credentials for IC Manager and then click **OK**.

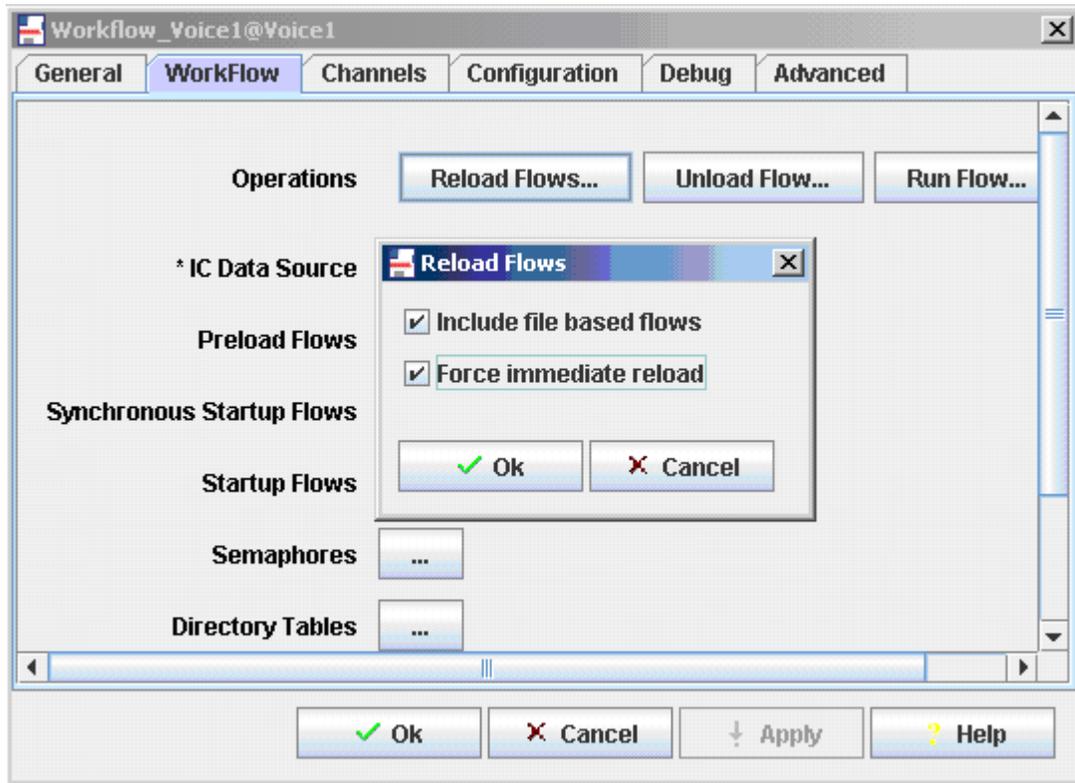


The image shows a screenshot of the "Project Settings" dialog box, specifically the "Database" tab. The dialog has a title bar "Project Settings" and a blue header. Below the header, there are three input fields: "IC Data Source:" with the value "interaction_center", "Login Id:" with the value "Admin", and "Password:" with the value "*****". Below these fields is a text box containing the instruction: "Available IC Data Sources can be found in IC Manager. From the top-level 'Tools' menu, choose 'IC Data Sources.'" At the bottom of the dialog are two buttons: "OK" and "Cancel". At the very bottom, there are four tabs: "General", "Compiler", "Directories", and "Database", with "Database" being the active tab.

To build the custom workflow, select Build→Build Flowset from the menu of Workflow Designer. The workflow should compile successfully as shown in the bottom section of the window.



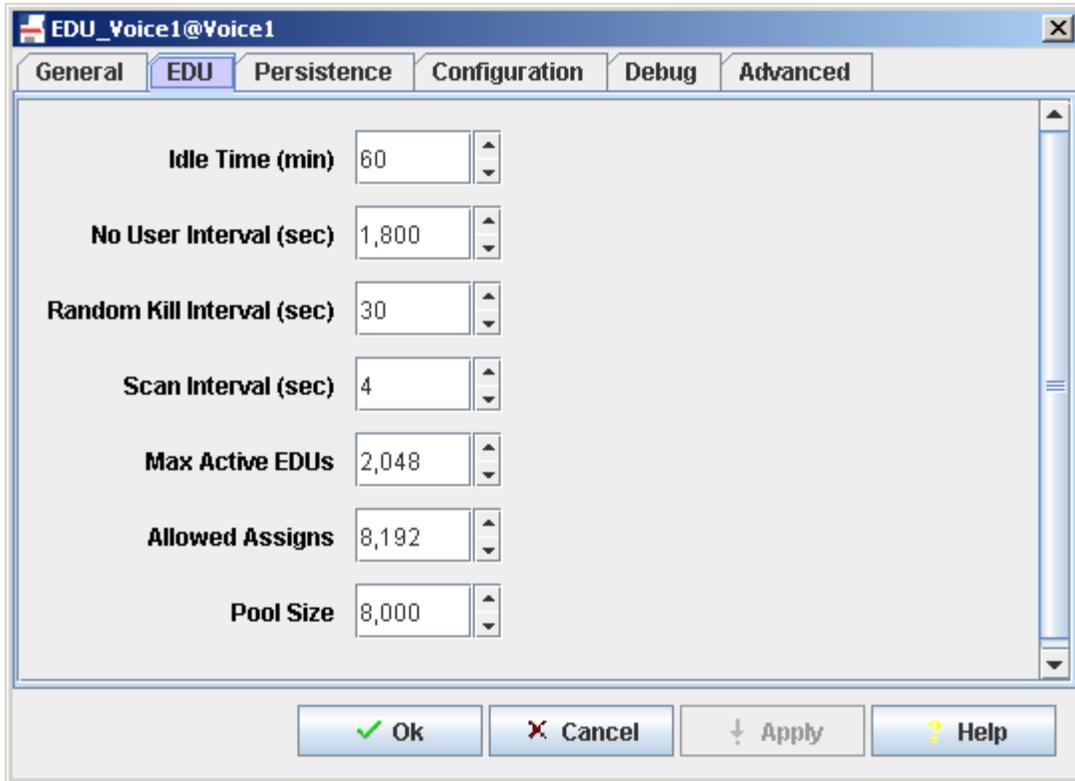
After successfully building the workflow, open the **Workflow_Voice1** server from IC Manager and select the **WorkFlow** tab. Click the **Reload Flows...** button and select both checkboxes in the **Reload Flows** window. Click **Ok**.



A message indicating that the flows were reloaded should be displayed as shown below. Click **Ok** and close the **Workflow_Voice1** server.

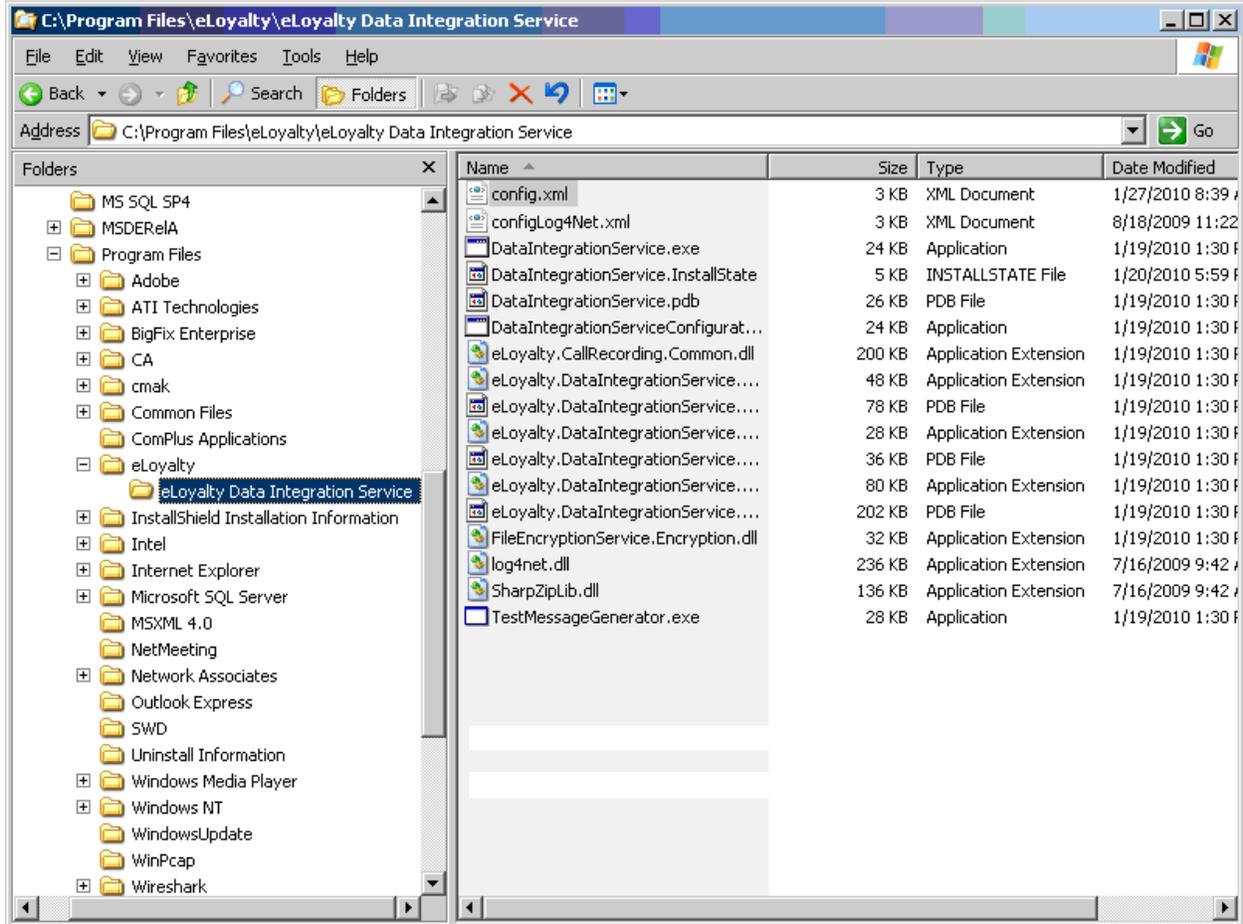


The following window shows the **EDU** tab of the **EDU_Voice1** server. It is shown for illustrative purposes to show the **Idle Time (min)** field, which is set to *60* by default. This field dictates how long the call data (EDU) will be kept in IC memory. This field may be modified as needed.

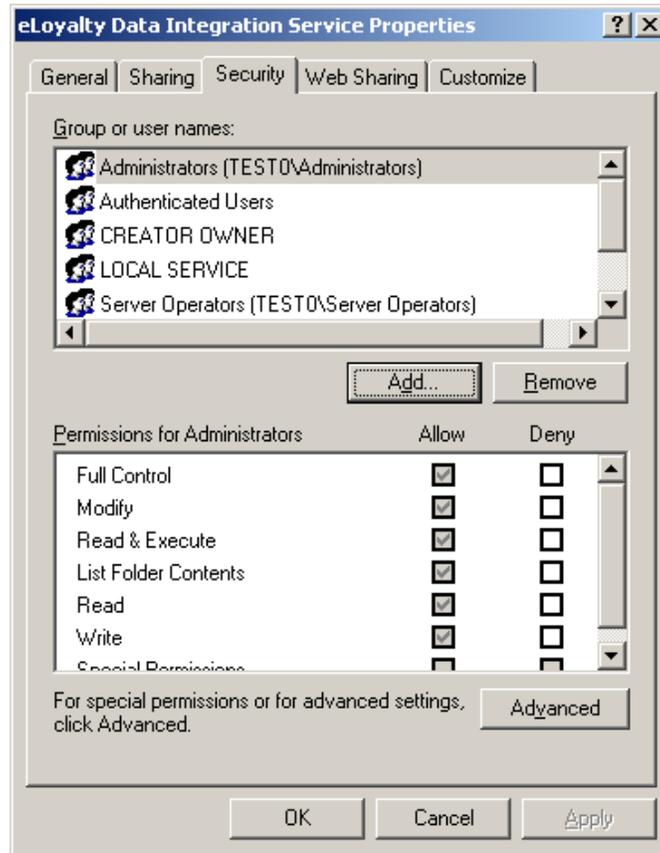


3. Configure eLoyalty Data Integration Service

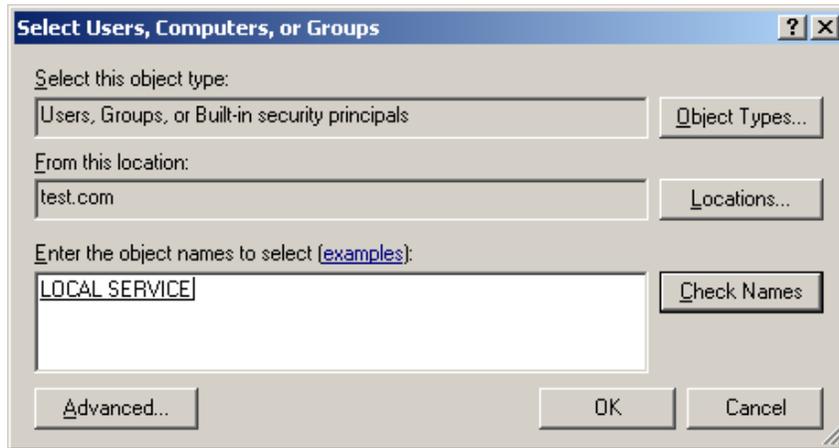
This section covers the procedure for configuring the eLoyalty Data Integration Service. It is assumed that the Data Integration Service has already been installed on a separate server. The following screen shows that the software was installed in the C:\Program Files\eLoyalty\eLoyalty Data Integration Service directory.



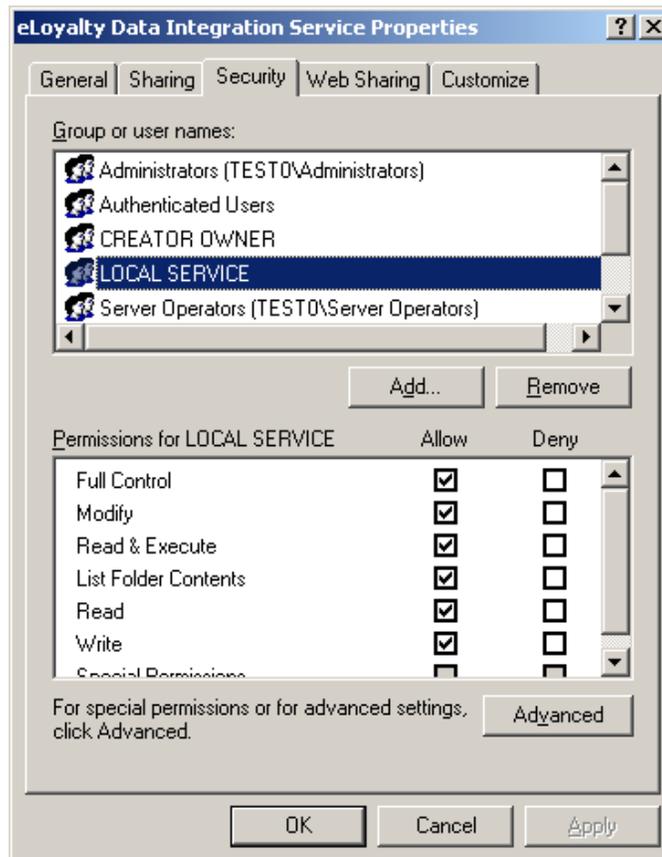
After the software has been successfully installed, verify that LOCAL SERVICE has full control of the C:\Program Files\eLoyalty\eLoyalty Data Integration Service directory. The software runs under LOCAL SERVICE. Right-mouse-click on this directory and select **Properties** from the pop-up menu. If LOCAL SERVICE does not appear in the **Group or user names** field, click the **Add** button.



In the **Select Users, Computers, or Groups** window, enter LOCAL SERVICE in the **Enter the object name to select** textbox as shown below. Click **Check Names** to verify that it is a valid entry. Click **OK**.



In the **eLoyalty Data Integration Service Properties** window, highlight LOCAL SERVICE and allow permissions as shown below. Click **OK**.



Edit the `config.xml` file located in Data Integration Service installation directory. Configure the login credentials for IC Manager and the appropriate URL for the Avaya IC Workflow Services as shown in the first red box below. The `dead_threshold_seconds` and `age_item_seconds` parameters may be modified as shown below. Please check with eLoyalty on the proper settings for these parameters in the customer environment.

```

<?xml version="1.0" ?>
- <DIServiceConfiguration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <resubmission_directory>c:\resubmissions</resubmission_directory>
  <remote_configuration_port>8087</remote_configuration_port>
- <msmq_config>
  <age_limit_seconds>300</age_limit_seconds>
  <queue_name>.\private$\bams</queue_name>
  <nagios_probe_duty_seconds>900</nagios_probe_duty_seconds>
</msmq_config>
- <client_config>
  <class_name>eLoyalty.DataIntegrationService.Client.AllstateAvayaIC.AvayaWorkflowInterface</class_name>
  <client_args>aic_user_login=Admin; aic_user_password=██████████;
  aic_ws_url=http://10.32.24.154:9800/webservices/services/AicWorkflowServices;
  eloyalty_ws_name=eloyalty.retrieveedu</client_args>
</client_config>
  <interface_config />
- <replicator_config>
  <replication_reporting_interval_seconds>300</replication_reporting_interval_seconds>
  <replication_folder>c:\crs\replications</replication_folder>
  <replication_interval_seconds>30</replication_interval_seconds>
</replicator_config>
- <queue_manager_config>
  <dead_queue_warning_count>600</dead_queue_warning_count>
  <stalled_queue_warning_count>400</stalled_queue_warning_count>
  <main_queue_warning_count>100</main_queue_warning_count>
  <dead_queue_duty_seconds>300</dead_queue_duty_seconds>
  <stalled_queue_duty_seconds>600</stalled_queue_duty_seconds>
  <main_queue_duty_seconds>15</main_queue_duty_seconds>
  <dead_threshold_seconds>0</dead_threshold_seconds>
  <dead_recordings_folder>c:\deadrecordings</dead_recordings_folder>
  <average_minimum>0.95</average_minimum>
  <success_probe_interval>900</success_probe_interval>
  <command_sample_size>250</command_sample_size>
</queue_manager_config>
- <msmq_processor_config>
  <process_interval_seconds>20</process_interval_seconds>
  <age_item_seconds>0</age_item_seconds>
</msmq_processor_config>
</DIServiceConfiguration>

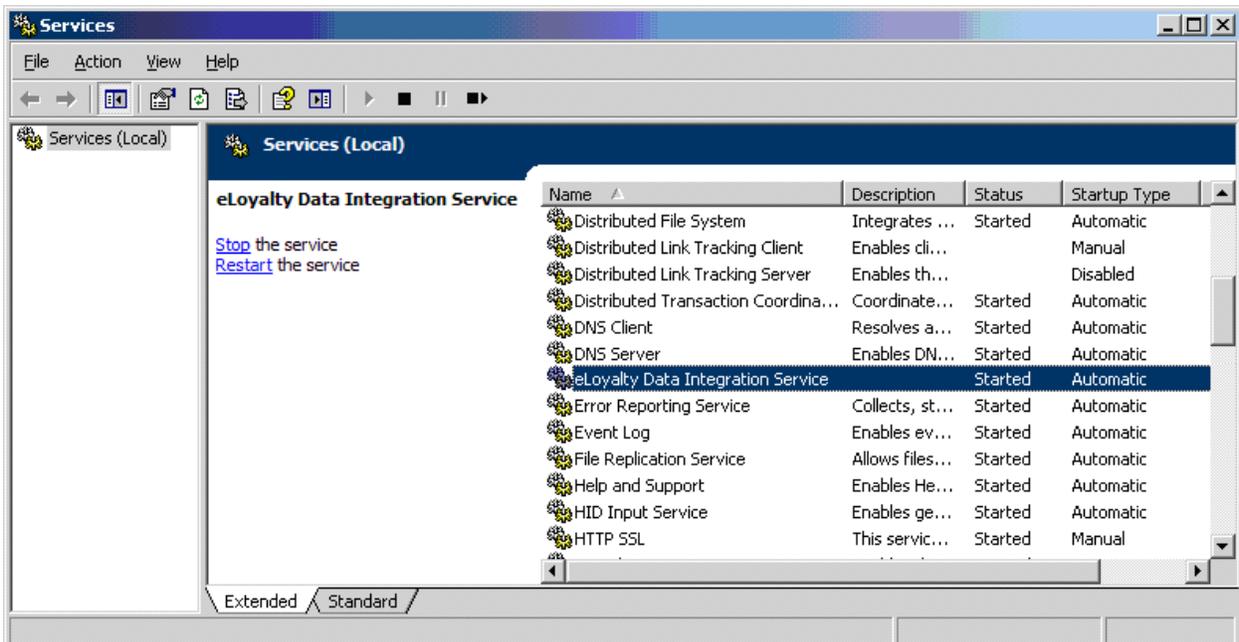
```

During system startup, there is a startup dependency between Data Integration Service and MSMQ. To ensure that MSMQ is started before Data Integration Service, add the following registry entry by running a registry script or using the Registry Editor.

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\eLoyalty Data  
Integration Service]  
"DependOnService"=hex(7):4d,00,53,00,4d,00,51,00,00,00,00,00
```

Finally, open **Microsoft Windows Services** under **Administrative Tools** and set the **Startup Type** for eLoyalty Data Integration Service to “Automatic”, and then start the service. The service should start successfully.



4. General Test Approach and Test Results

This section describes the interoperability compliance testing used to verify eLoyalty Data Integration Service with Avaya IC. This section covers the general test approach and the test results.

Interoperability compliance testing included feature and serviceability testing. The feature testing focused on verifying that call data can be retrieved by the eLoyalty Data Integration Service from Avaya IC and stored in an XML file. The Data Integration Service used IC WorkflowServices webservice to run the custom workflow, which in turn retrieved the call data as an EDU (Electronic Data Unit) and sent the requested data fields to the Data Integration Service. The call data was requested manually using a test application provided by eLoyalty. The test application simulated the behavior of the eLoyalty Behavioral Analytics application, which was not covered in the compliance test.

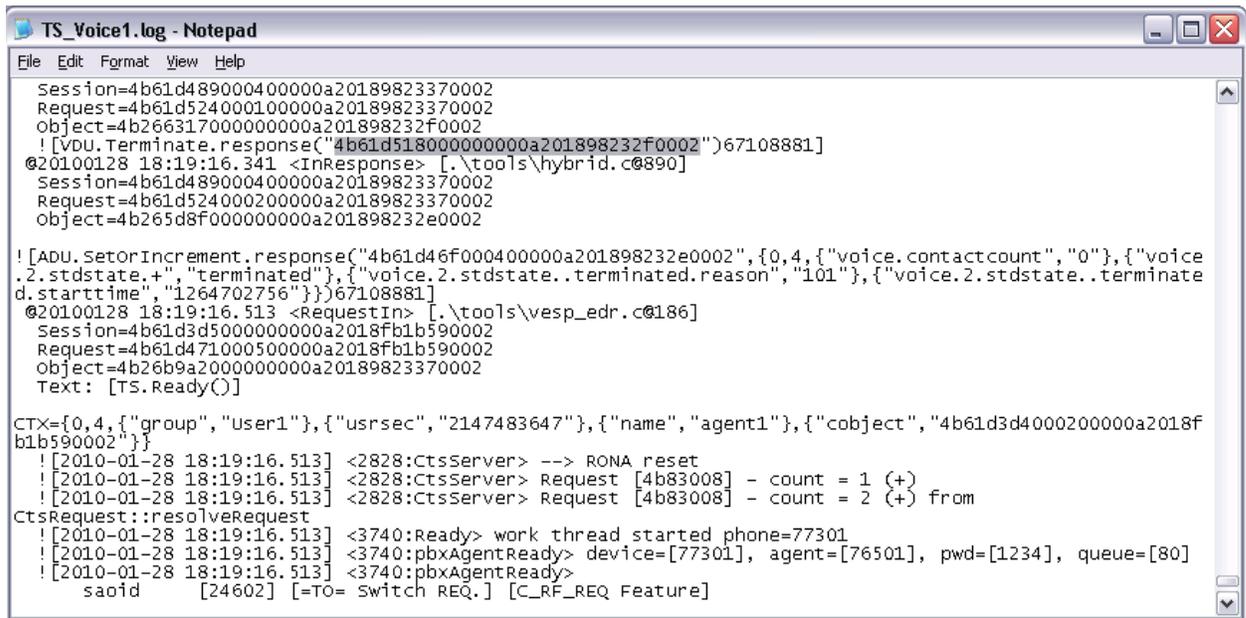
The serviceability testing focused on verifying that the Data Integration Service would start up automatically when the server is rebooted and that messages in the MSMQ are serviced if the Data Integration Service starts up after the message was queued.

All test cases passed.

5. Verification Steps

This section provides the verification steps that may be performed to verify that the eLoyalty Data Integration Service can retrieve call data from Avaya IC 7.2.

1. Place a call to Communication Manager that gets routed to an IC agent. Terminate and wrap-up the call.
2. Open the `TS_Voice1.log` located in the **logs** directory under the IC 7.2 installation directory. Retrieve and save the EDU ID (or VDU ID) in the log.

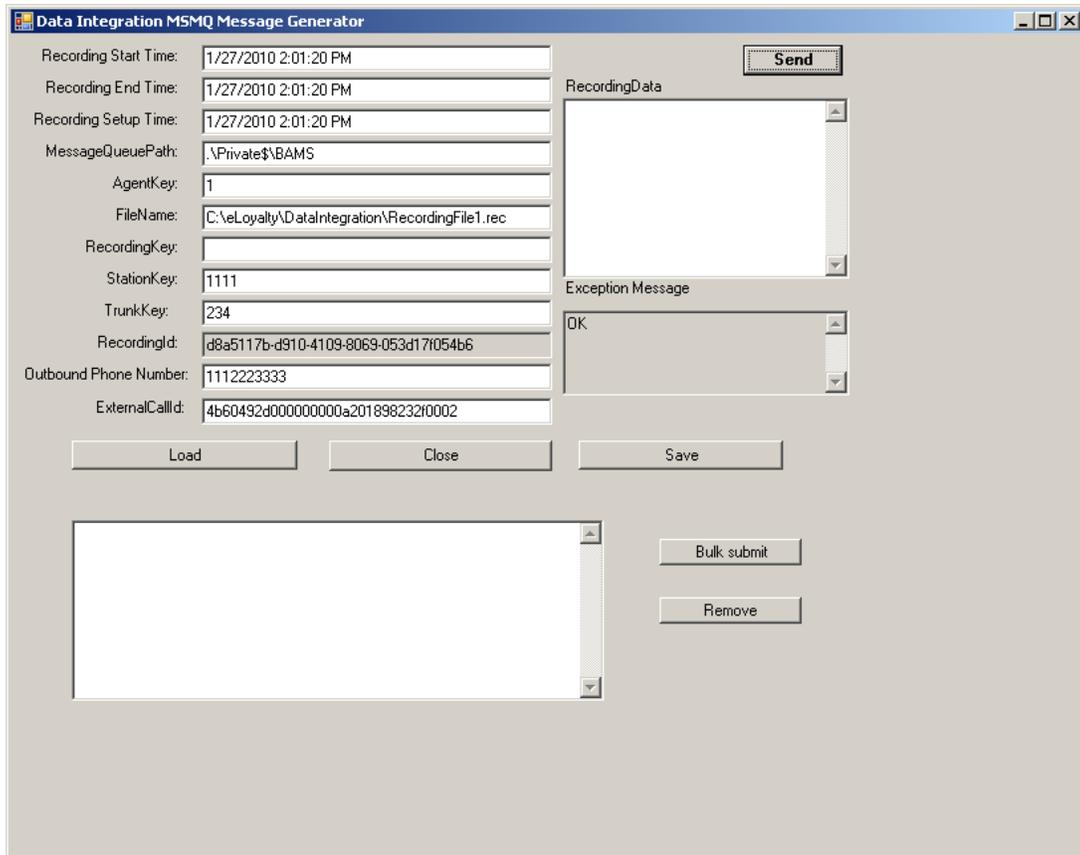


```
TS_Voice1.log - Notepad
File Edit Format View Help
Session=4b61d489000400000a20189823370002
Request=4b61d524000100000a20189823370002
Object=4b266317000000000a201898232f0002
! [VDU.Terminate.response("4b61d518000000000a201898232f0002")67108881]
@20100128 18:19:16.341 <InResponse> [.\tools\hybrid.c@890]
Session=4b61d489000400000a20189823370002
Request=4b61d524000200000a20189823370002
Object=4b265d8f000000000a201898232e0002

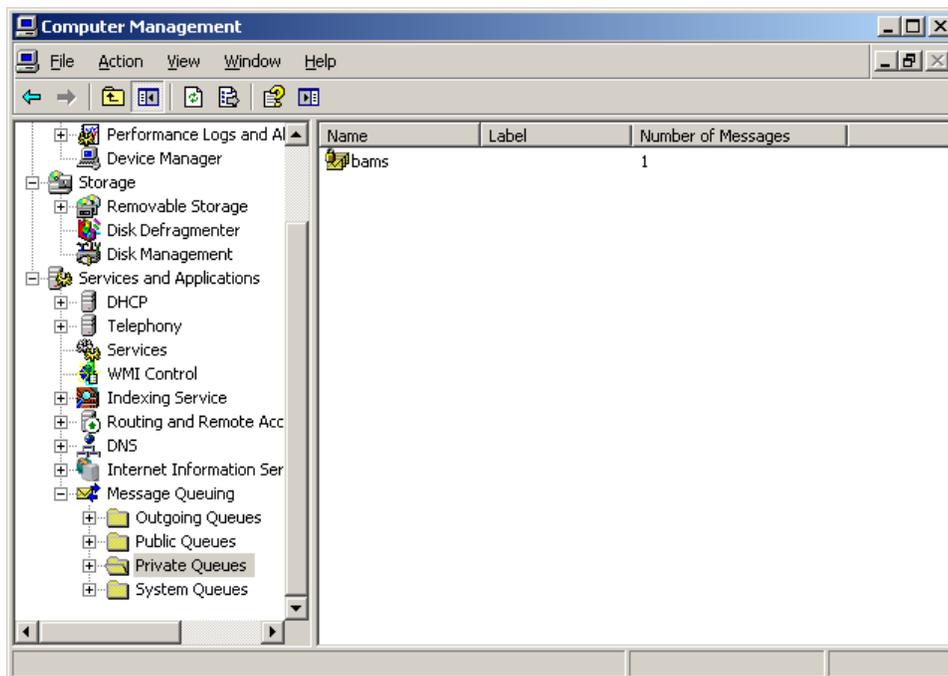
! [ADU.SetOrIncrement.response("4b61d46f000400000a201898232e0002", {0,4, {"voice.contactcount", "0"}, {"voice.2.stdstate.", "terminated"}, {"voice.2.stdstate..terminated.reason", "101"}, {"voice.2.stdstate..terminate.d.starttime", "1264702756"}})67108881]
@20100128 18:19:16.513 <RequestIn> [.\tools\vesp_edr.c@186]
Session=4b61d3d5000000000a2018fb1b590002
Request=4b61d471000500000a2018fb1b590002
Object=4b26b9a2000000000a20189823370002
Text: [TS.Ready()]

CTX={0,4, {"group", "User1"}, {"usrsec", "2147483647"}, {"name", "agent1"}, {"cobject", "4b61d3d4000200000a2018fb1b590002"}}
! [2010-01-28 18:19:16.513] <2828:CtsServer> --> RONA reset
! [2010-01-28 18:19:16.513] <2828:CtsServer> Request [4b83008] - count = 1 (+)
! [2010-01-28 18:19:16.513] <2828:CtsServer> Request [4b83008] - count = 2 (+) from
CtsRequest::resolveRequest
! [2010-01-28 18:19:16.513] <3740:Ready> work thread started phone=77301
! [2010-01-28 18:19:16.513] <3740:pbxAgentReady> device=[77301], agent=[76501], pwd=[1234], queue=[80]
! [2010-01-28 18:19:16.513] <3740:pbxAgentReady>
  saoid [24602] [=TO= Switch REQ.] [C_RF_REQ Feature]
```

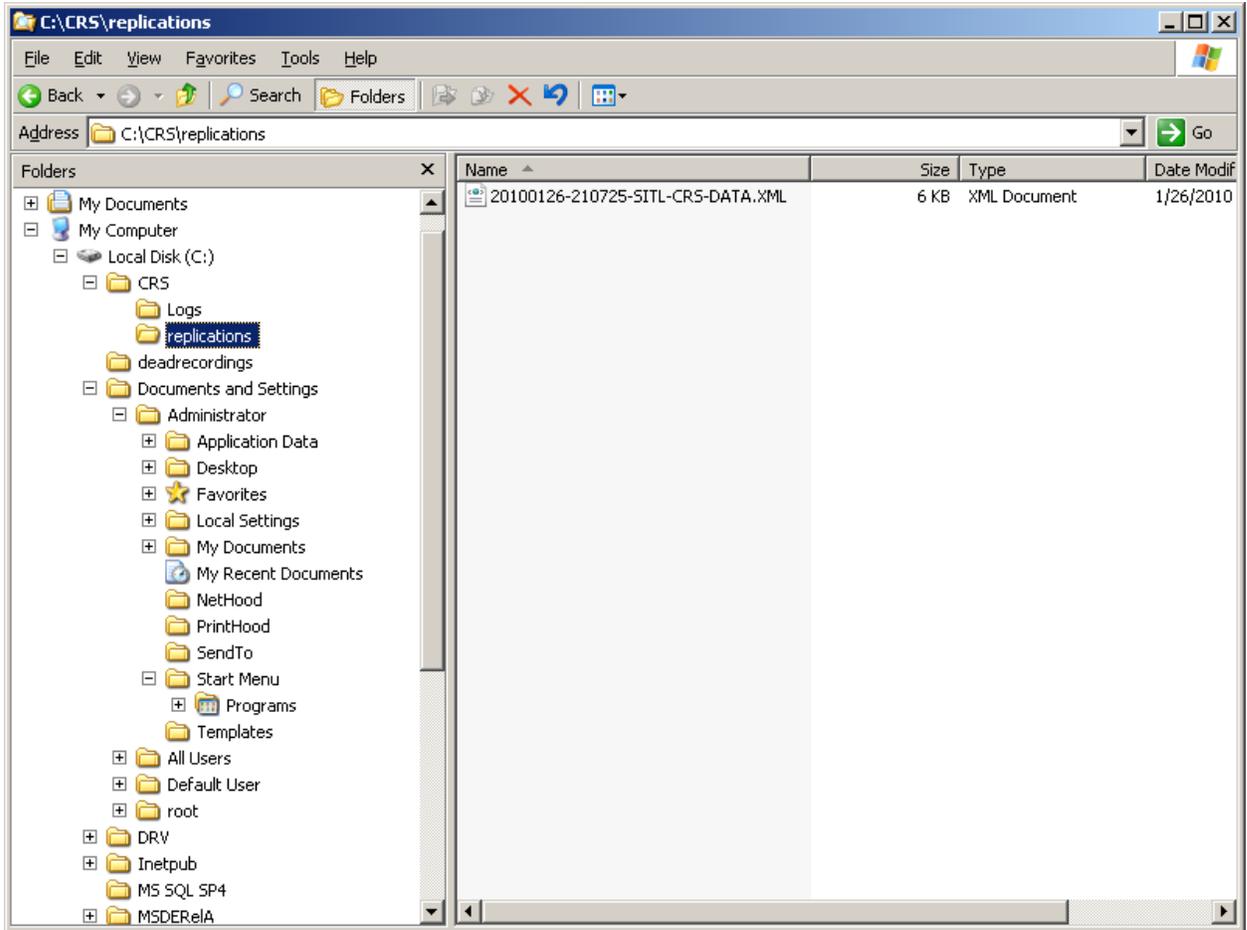
3. Execute the test application provided by eLoyalty by entering the EDU ID into the GUI of the test application.



- From Windows Computer Management, it can be seen that there is a message on the queue until it is serviced by the Data Integration Service.



5. After the eLoyalty custom workflow is run and the EDU is retrieved by Avaya IC and returned to the Data Integration Service, the call data (EDU) is stored in an XML file in the C:\CRS\replications directory.



6. The content of the XML file is shown below.

```
- <CRSEventReplication>
- <Commands>
  <Command name="InsertClientData">insert into ClientDataEvent ( RecordingId,
    ClientCallId, ClientEventTime, ClientData ) values ( @RecordingId, @ClientCallId,
    @ClientEventTime, @ClientData )</Command>
</Commands>
- <Data>
- <Execute command="InsertClientData">
  <Parameter name="@RecordingId">767f7810-5f45-4406-ba69-
    dda41ee21759</Parameter>
  <Parameter
    name="@ClientCallId">4b61d518000000000a201898232f0002</Parameter>
  <Parameter name="@ClientEventTime">2010-01-28 18:09:04.000</Parameter>
  <Parameter name="@ClientData"><AicEduRecord
    eduid="4b61d518000000000a201898232f0002"><Parameter name="vdu_id"
    value="4b61d518000000000a201898232f0002" /><Parameter name="ani"
    value="77304" /><Parameter name="dnis" value="28900" /><Parameter
    name="primary_ani" value="77304" /><Parameter name="primary_dnis"
    value="28900" /><Parameter name="loginid" value="agent1" /><Parameter
    name="agent_key" value="100007" /><Parameter name="voice_direction"
    value="inbound" /><Parameter name="calltype"
    value="direct" /><Parameter name="ctype" value="direct" /><Parameter
    name="ucid" value="00001003551264702682" /><Parameter name="type"
    value="voice" /><Parameter name="owner"
    value="4b5dc244000100000a20189823370002" /><Parameter
    name="createtime" value="2010-01-28 18:19:04" /><Parameter
    name="createtimet" value="1264702744" /><Parameter name="phone"
    value="76501" /><Parameter name="dest" value="77301" /><Parameter
    name="orig" value="77304" /><Parameter name="ext"
    value="77301" /><Parameter name="agent" value="" /><Parameter
    name="agent.1"
    value="4b26ba35000000000a20189823380002" /><Parameter
    name="contactduration" value="8" /><Parameter name="contactendtime"
    value="1264702752" /><Parameter name="last_termination"
    value="4b61d489000400000a20189823370002" /><Parameter
    name="duration" value="12" /><Parameter name="endtime" value="2010-
    01-28 18:19:16" /><Parameter name="queue" value="28900" /><Parameter
    name="acd_split" value="28905" /><Parameter name="agent.2"
    value="" /><Parameter name="voice" value="" /><Parameter
    name="voice.1" value="0" /><Parameter name="voice.1.loginid"
    value="agent1" /><Parameter name="voice.1.leg_id"
    value="4b61d518000b00000a20189823370002" /><Parameter
    name="voice.1.agent_key" value="100007" /><Parameter
```

```
name="voice.1.destination" value="77301" /><Parameter
name="voice.1.origin" value="77304" /><Parameter
name="voice.1.direction" value="inbound" /><Parameter
name="voice.1.acdname" value="7f7" /><Parameter name="voice.1.ucid"
value="00001003551264702682" /><Parameter name="voice.1.queue_key"
value="11" /><Parameter name="voice.1.stdstate" value="" /><Parameter
name="voice.1.stdstate.1" value="created" /><Parameter
name="voice.1.stdstate.1.created" value="" /><Parameter
name="voice.1.stdstate.1.created.reason" value="" /><Parameter
name="voice.1.stdstate.1.created.starttime"
value="1264702744" /><Parameter name="voice.1.stdstate.2"
value="alerting" /><Parameter name="voice.1.stdstate.2.alerting"
value="" /><Parameter name="voice.1.stdstate.2.alerting.reason"
value="" /><Parameter name="voice.1.stdstate.2.alerting.starttime"
value="1264702744" /><Parameter name="voice.1.queue"
value="0" /><Parameter name="voice.1.queue_number"
value="28900" /><Parameter name="voice.1.queue_time"
value="0" /><Parameter name="AgentDesktop" value="" /><Parameter
name="AgentDesktop.CHBrowser" value="" /><Parameter
name="AgentDesktop.CHBrowser.value" value="77304" /><Parameter
name="AgentDesktop.media_type" value="voice" /><Parameter
name="AgentDesktop.contact_label" value="77304" /><Parameter
name="AgentDesktop.ScreenPop" value="" /><Parameter
name="AgentDesktop.ScreenPop.type" value="voice" /><Parameter
name="AgentDesktop.media_type_additional"
value="inbound" /><Parameter name="AgentDesktop.CHBrowser.field"
value="mediainteraction.ani" /><Parameter
name="AgentDesktop.ScreenPop.value" value="(000) 007-
7304" /><Parameter name="voice.1.connect" value="3" /><Parameter
name="voice.1.ringtime" value="3" /><Parameter
name="voice.1.stdstate.3" value="active" /><Parameter
name="voice.1.stdstate.3.active" value="" /><Parameter
name="voice.1.stdstate.3.active.reason" value="" /><Parameter
name="voice.1.stdstate.3.active.starttime"
value="1264702747" /><Parameter name="voice.1.talktime"
value="5" /><Parameter name="voice.1.exit_reason"
value="normal" /><Parameter name="voice.1.stdstate.4"
value="wrapup" /><Parameter name="voice.1.stdstate.4.wrapup"
value="" /><Parameter name="voice.1.stdstate.4.wrapup.reason"
value="101" /><Parameter name="voice.1.stdstate.4.wrapup.starttime"
value="1264702752" /><Parameter name="voice.1.stdstate.5"
value="terminated" /><Parameter name="voice.1.stdstate.5.terminated"
value="" /><Parameter name="voice.1.stdstate.5.terminated.reason"
value="101" /><Parameter name="voice.1.stdstate.5.terminated.starttime"
value="1264702756" /></AicEduRecord></Parameter>
</Execute>
</Data>
</CRSEventReplication>
```

6. Conclusion

These Application Notes describe the configuration steps required to integrate the eLoyalty Data Integration Service with Avaya Interaction Center 7.2. The Data Integration Service was able to use IC Web Services to request that Avaya IC run a custom workflow and return call data (EDU) based on an EDU ID. An eLoyalty test application was used to exercise the Data Integration Service. All test cases passed.

7. Additional References

This section references the product documentation that is relevant to these Application Notes.

- [1] *Avaya Interaction Center Release 7.2 Installation Planning and Prerequisites*, Release 7.2, May 2009, available at <http://support.avaya.com>.
- [2] *Avaya Interaction Center Release 7.2 Installation and Configuration*, Release 7.2, May 2009, available at <http://support.avaya.com>.

©2010 Avaya Inc. All Rights Reserved.

Avaya and the Avaya Logo are trademarks of Avaya Inc. All trademarks identified by ® and ™ are registered trademarks or trademarks, respectively, of Avaya Inc. All other trademarks are the property of their respective owners. The information provided in these Application Notes is subject to change without notice. The configurations, technical data, and recommendations provided in these Application Notes are believed to be accurate and dependable, but are presented without express or implied warranty. Users are responsible for their application of any products specified in these Application Notes.

Please e-mail any questions or comments pertaining to these Application Notes along with the full title name and filename, located in the lower right corner, directly to the Avaya DevConnect Program at devconnect@avaya.com.