Avaya Solution & Interoperability Test Lab

Application Notes for eLoyalty Data Integration Service with Avaya Interaction Center 7.1 – Issue 1.0

Abstract

These Application Notes describe the configuration steps required to integrate the eLoyalty Data Integration Service with Avaya Interaction Center 7.1. 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 7.1. 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:** 2.ELOYAL
- **Email:** BehavioralAnalytics@eloyalty.com
- **Web:** www.eloyalty.com

Reference ConfigurationFigure 1 illustrates the configuration used for testing. In this configuration, a call center environment consisting of Avaya Interaction Center (IC), Avaya Aura™ 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.

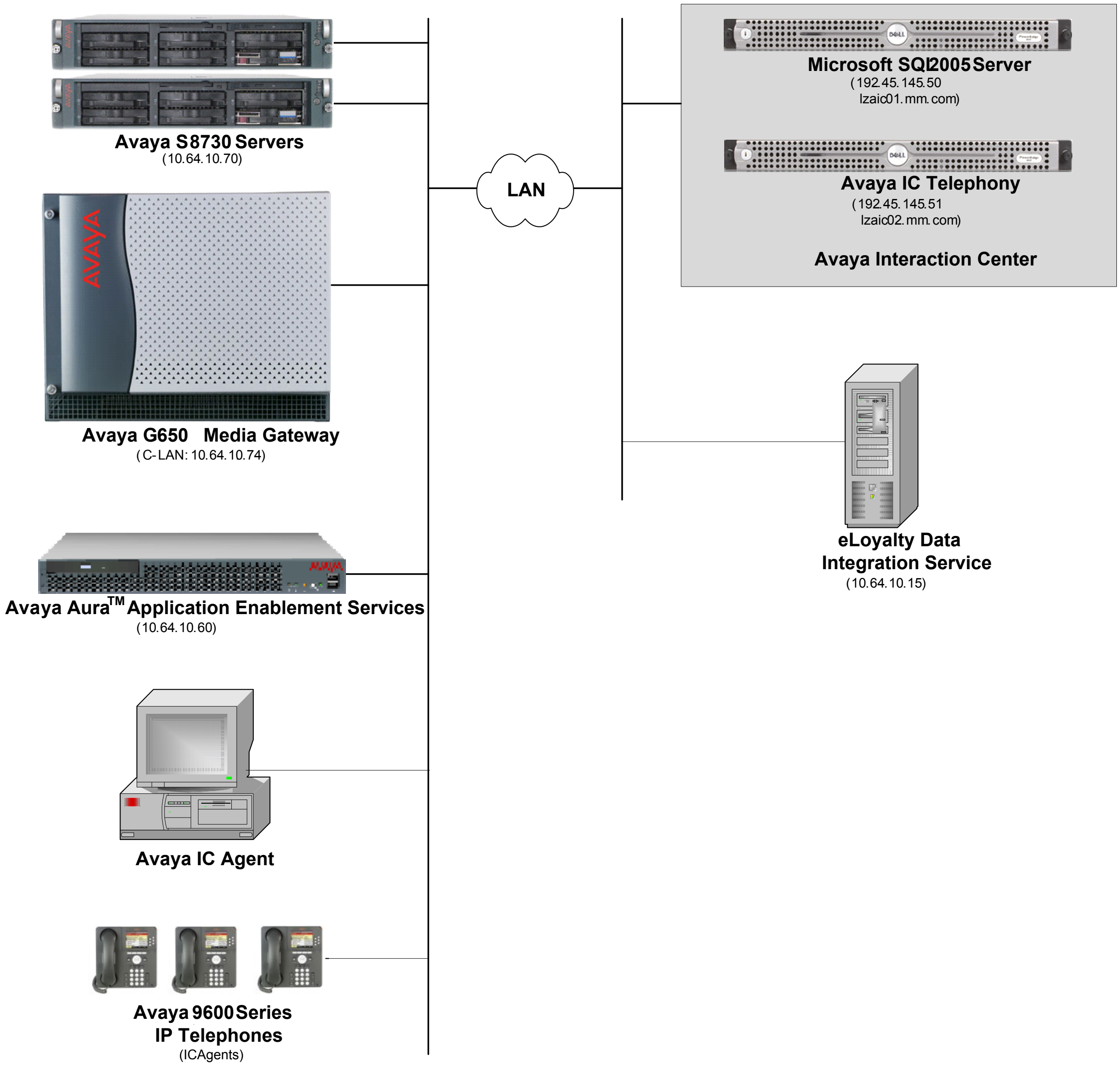


Figure 1: Configuration with Avaya IC and eLoyalty Data Integration Service

1.3. Equipment and Software Validated

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

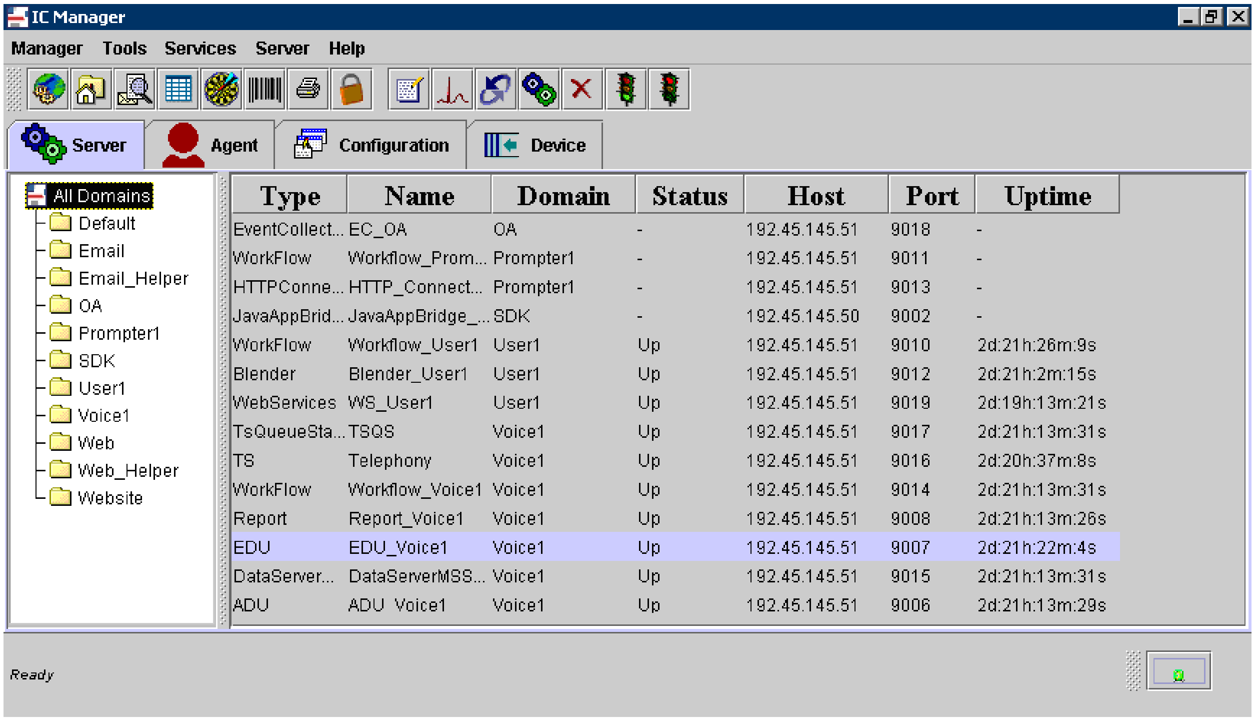
Equipment	Software
Avaya Interaction Center	7.1
Avaya Aura™ Application Enablement Services	5.2
Avaya S8730 Servers with an Avaya G650 Media Gateway	Avaya Aura™ Communication Manager 5.2.1 (R015x.02.1.016.4)
Avaya 9600 Series IP Telephones	3.0 (H.323)
eLoyalty Data Integration Service running on Microsoft 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.

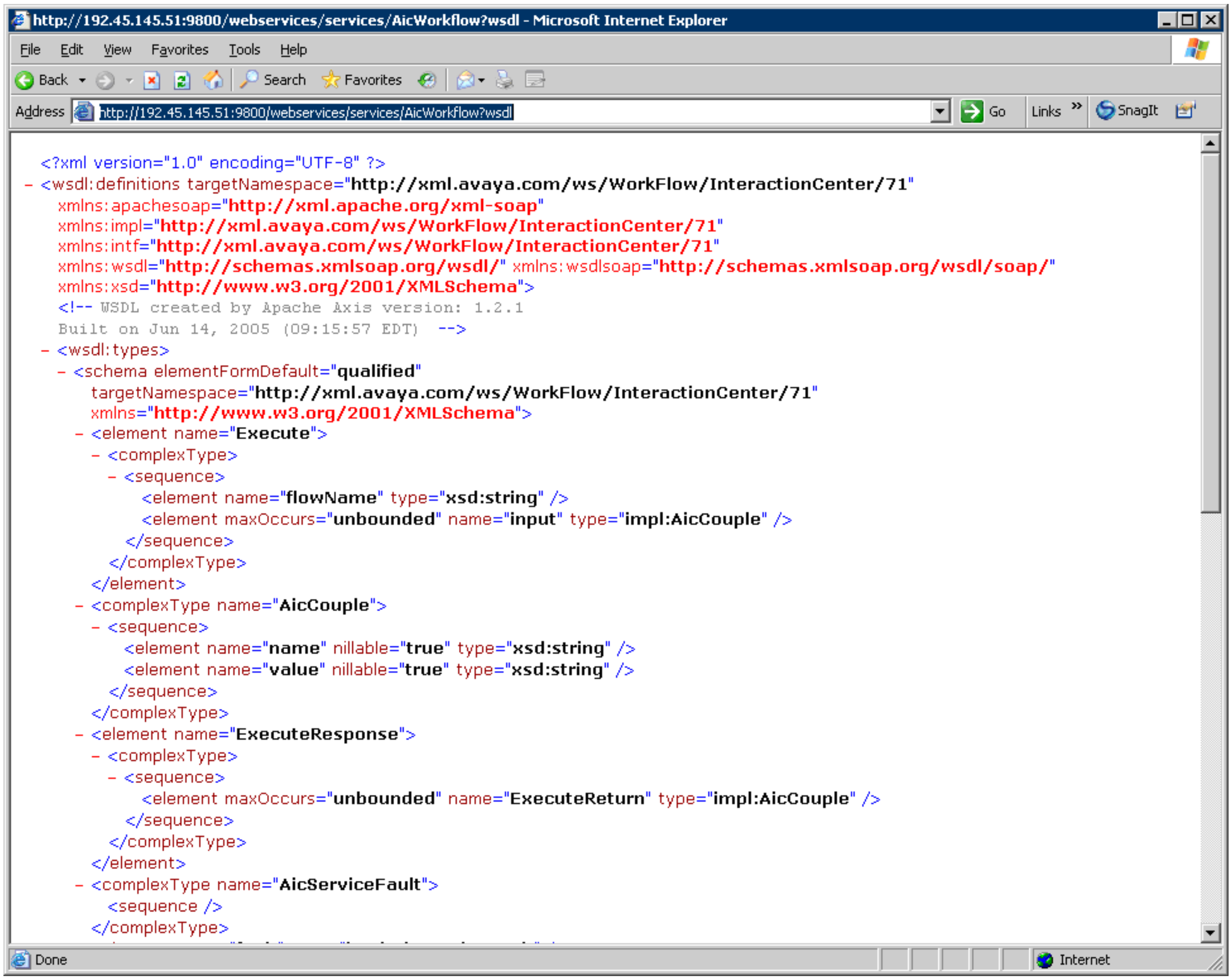
Once IC is up and running, the eLoyalty custom workflow needs to be copied to the server running Avaya Workflow Designer and then the custom workflow needs to be built. This section covers how to build the eLoyalty custom workflow. Before building the custom workflow, verify that the IC servers are up 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".



Type	Name	Domain	Status	Host	Port	Uptime
EventCollect...	EC_OA	OA	-	192.45.145.51	9018	-
WorkFlow	Workflow_Prom...	Prompter1	-	192.45.145.51	9011	-
HTTPConne...	HTTP_Connect...	Prompter1	-	192.45.145.51	9013	-
JavaAppBrid...	JavaAppBridge_...	SDK	-	192.45.145.50	9002	-
WorkFlow	Workflow_User1	User1	Up	192.45.145.51	9010	2d:21h:26m:9s
Blender	Blender_User1	User1	Up	192.45.145.51	9012	2d:21h:2m:15s
WebServices	WS_User1	User1	Up	192.45.145.51	9019	2d:19h:13m:21s
TsQueueSta...	TSQS	Voice1	Up	192.45.145.51	9017	2d:21h:13m:31s
TS	Telephony	Voice1	Up	192.45.145.51	9016	2d:20h:37m:8s
WorkFlow	Workflow_Voice1	Voice1	Up	192.45.145.51	9014	2d:21h:13m:31s
Report	Report_Voice1	Voice1	Up	192.45.145.51	9008	2d:21h:13m:26s
EDU	EDU_Voice1	Voice1	Up	192.45.145.51	9007	2d:21h:22m:4s
DataServer...	DataServerMSS...	Voice1	Up	192.45.145.51	9015	2d:21h:13m:31s
ADU	ADU_Voice1	Voice1	Up	192.45.145.51	9006	2d:21h:13m:29s

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.

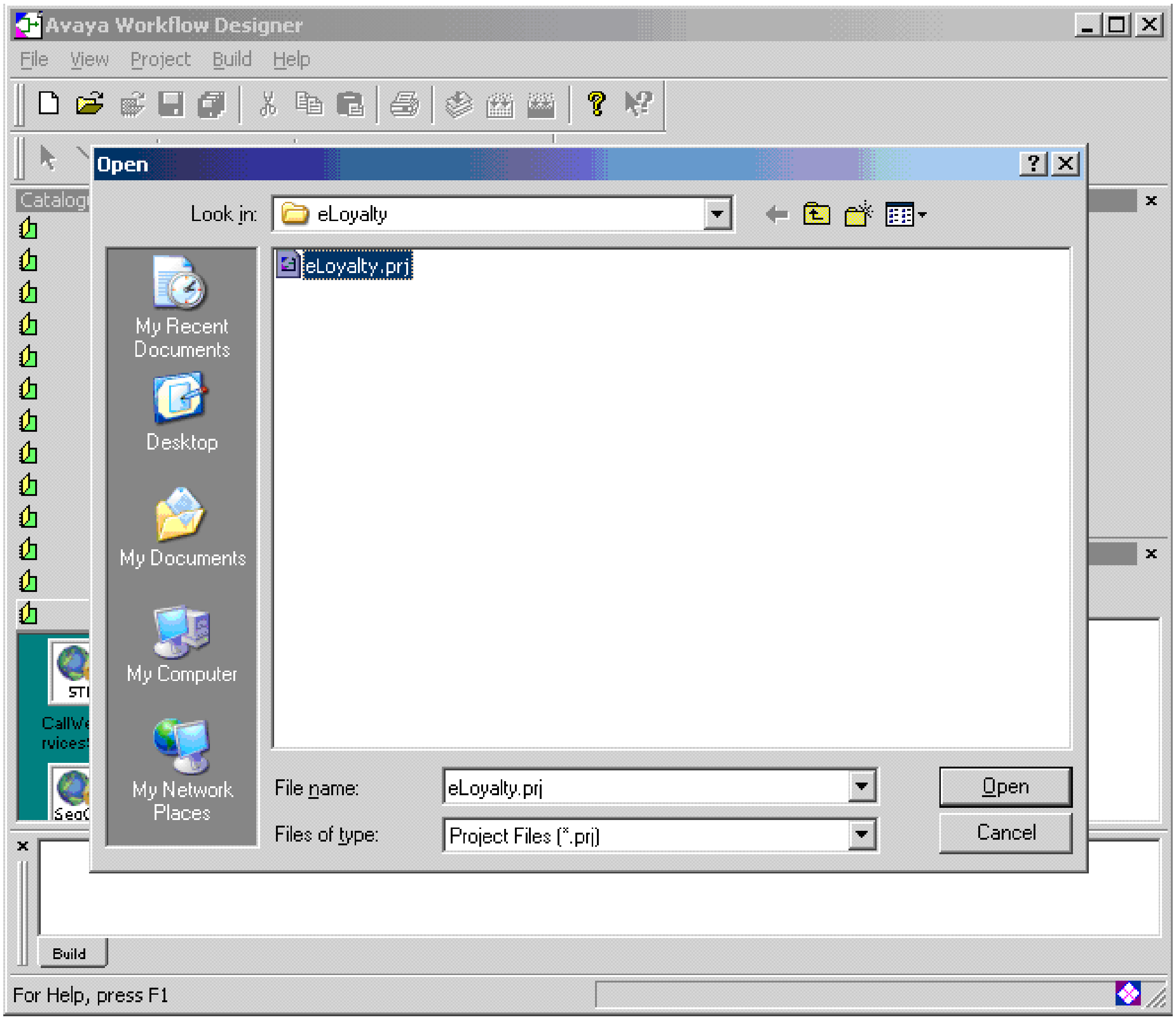


http://192.45.145.51:9800/webservices/services/AicWorkflow?wsdl - Microsoft Internet Explorer

Address: http://192.45.145.51:9800/webservices/services/AicWorkflow?wsdl

```
  <?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>
          - <sequence>
              <element maxOccurs="unbounded" name="ExecuteReturn" type="impl:AicCouple" />
            </sequence>
          </complexType>
        </element>
      - <complexType name="AicServiceFault">
          <sequence />
        </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.

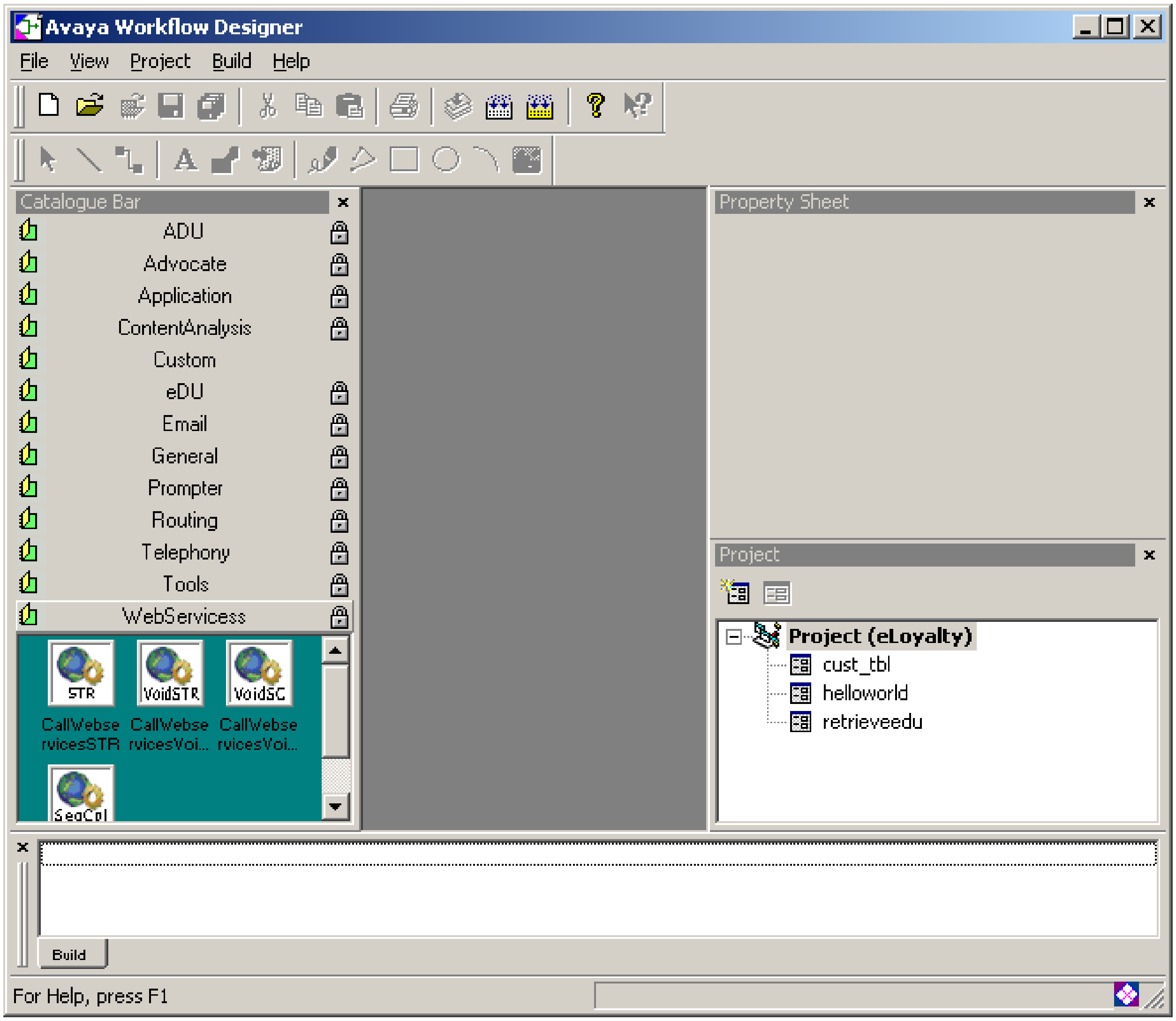


eLoyalty.prj

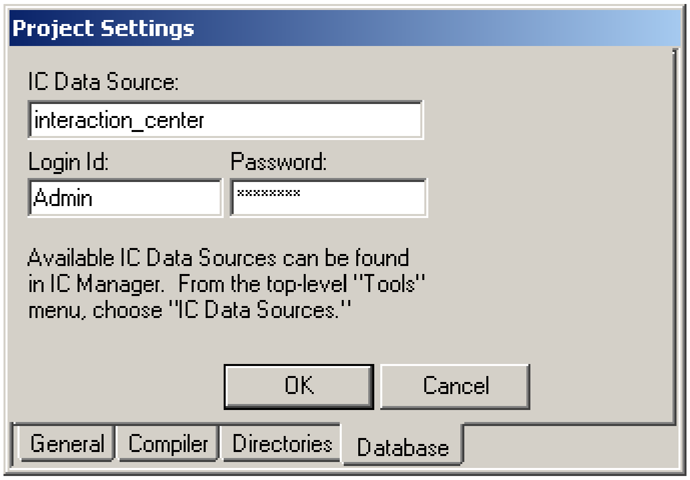
File name: eLoyalty.prj

Files of type: Project Files (*.prj)

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 the click **OK**.



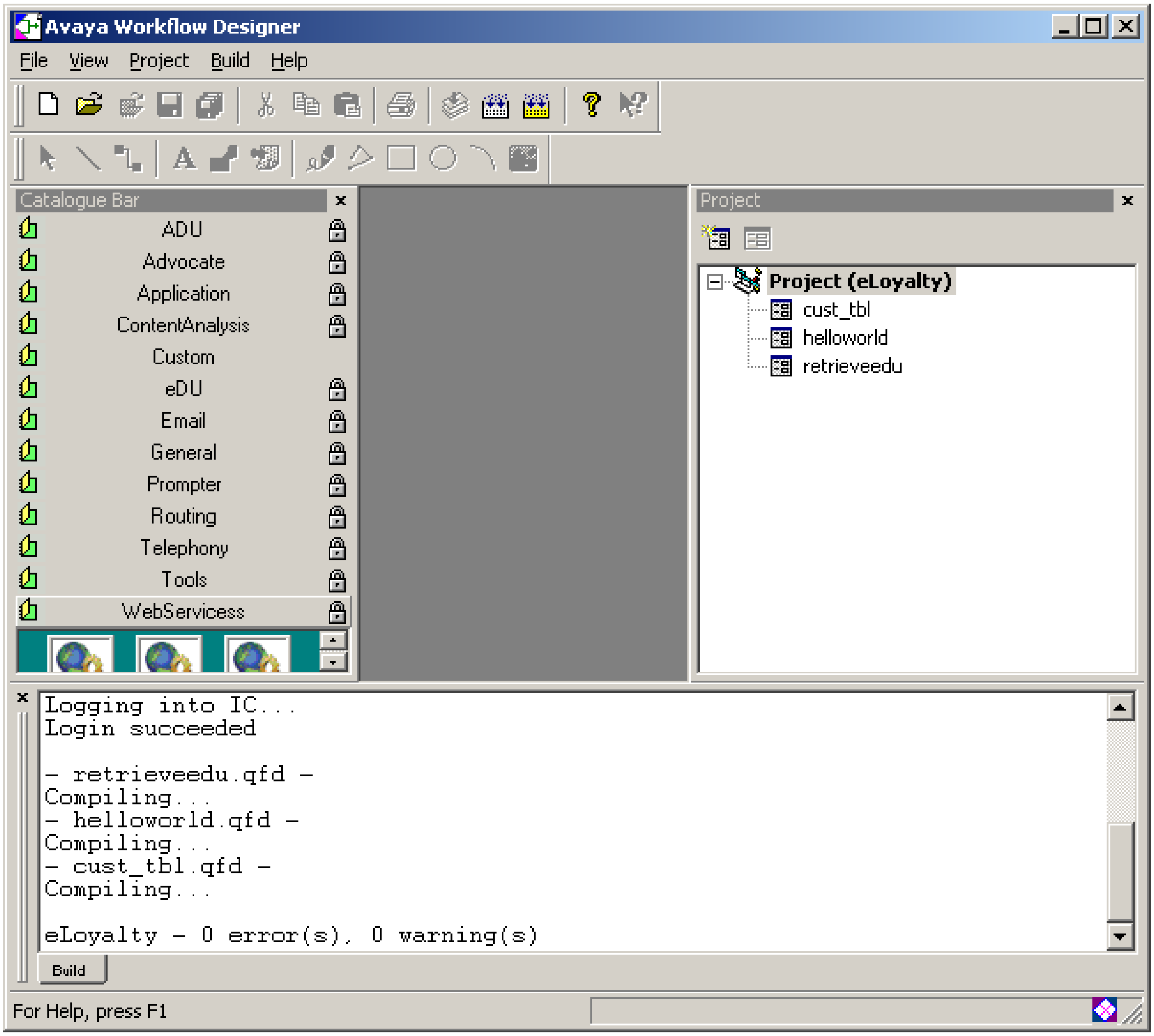
IC Data Source: interaction_center

Login Id: Admin

Password: ********

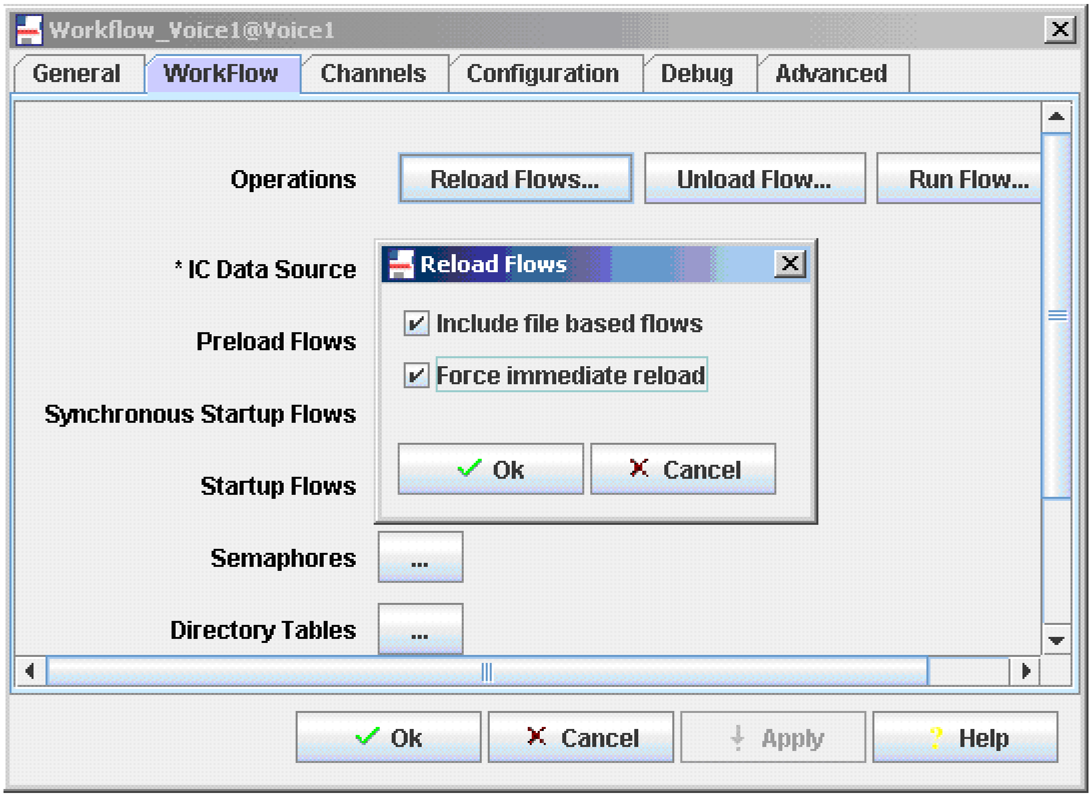
Available IC Data Sources can be found in IC Manager. From the top-level "Tools" menu, choose "IC Data Sources."

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.



```
Logging into IC...
Login succeeded

- retrieveedu.qfd -
Compiling...
- helloworld.qfd -
Compiling...
- cust_tbl.qfd -
Compiling...

eLoyalty - 0 error(s), 0 warning(s)
```

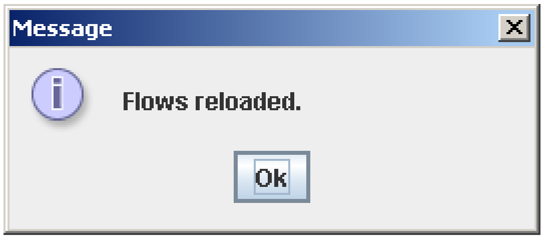
After successfully building the workflow, open the **Workflow_Voice1** server from IC Manager and select the **WorkFlow** tab. Click the **Reload Flows…** button and select both checkboxes in the **Reload Flows** window. Click **Ok**.



☑ Include file based flows

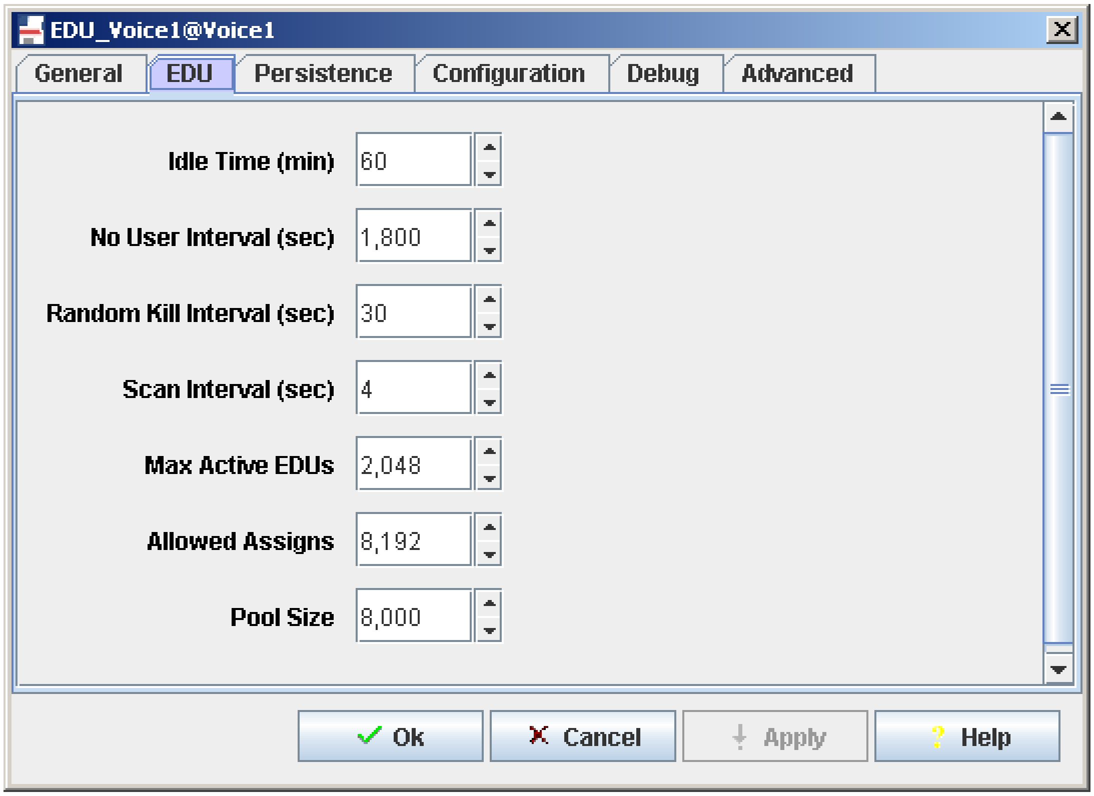
☑ Force immediate reload

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



Flows reloaded.

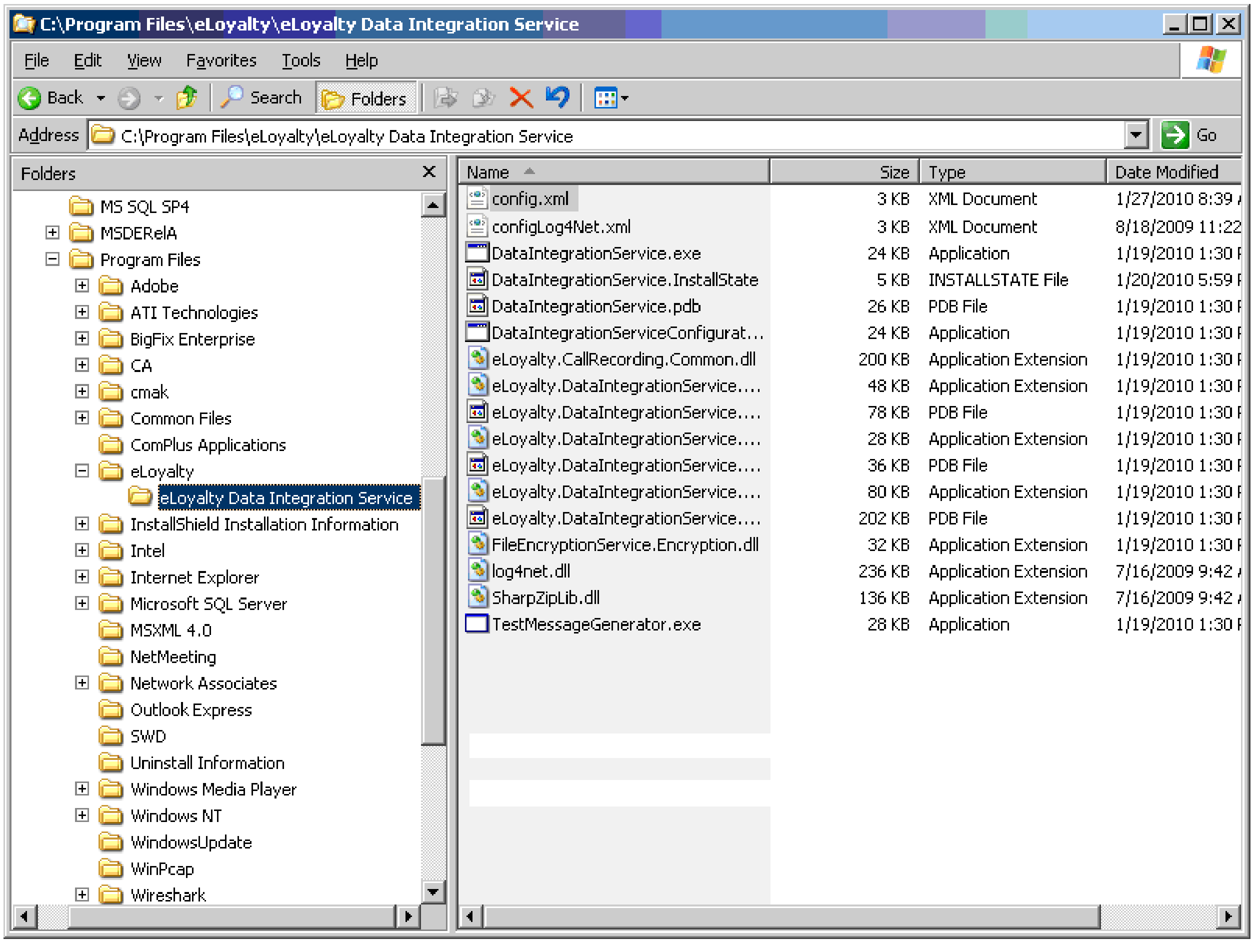
The following window shows the **EDU** tab of the **EDU_Voice1** server. It is shown for illustrative purposes to show the **Idle Time (min)** field, which is set to *60* by default. This field dictates how long the call data (EDU) will be kept in IC memory. This field may be modified as needed.



Idle Time (min)	60
No User Interval (sec)	1,800
Random Kill Interval (sec)	30
Scan Interval (sec)	4
Max Active EDUs	2,048
Allowed Assigns	8,192
Pool Size	8,000

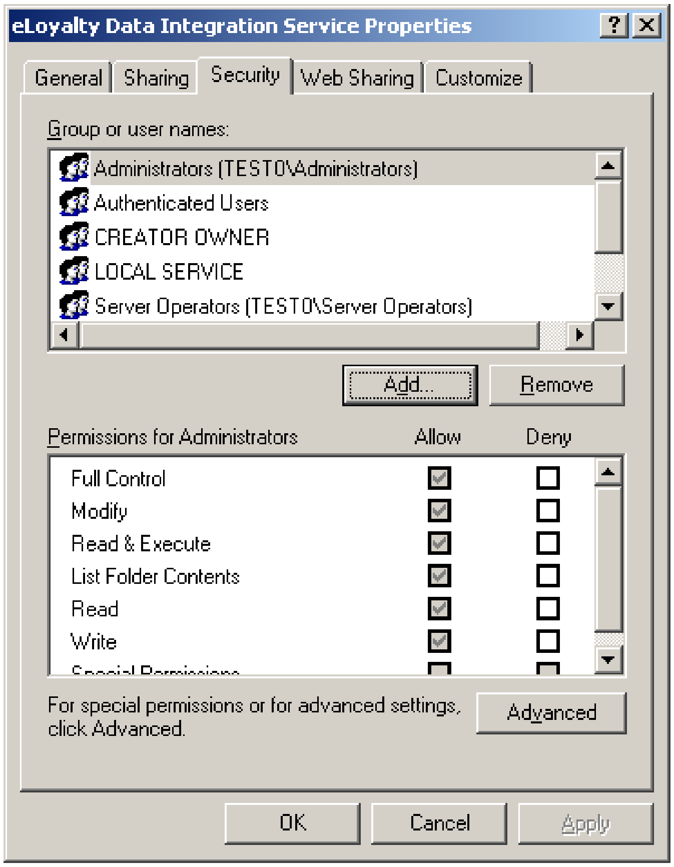
3. Configure eLoyalty Data Integration Service

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.



Name	Size	Type	Date Modified
config.xml	3 KB	XML Document	1/27/2010 8:39
configLog4Net.xml	3 KB	XML Document	8/18/2009 11:22
DataIntegrationService.exe	24 KB	Application	1/19/2010 1:30
DataIntegrationService.InstallState	5 KB	INSTALLSTATE File	1/20/2010 5:59
DataIntegrationService.pdb	26 KB	PDB File	1/19/2010 1:30
DataIntegrationServiceConfigurat...	24 KB	Application	1/19/2010 1:30
eLoyalty.CallRecording.Common.dll	200 KB	Application Extension	1/19/2010 1:30
eLoyalty.DataIntegrationService....	48 KB	Application Extension	1/19/2010 1:30
eLoyalty.DataIntegrationService....	78 KB	PDB File	1/19/2010 1:30
eLoyalty.DataIntegrationService....	28 KB	Application Extension	1/19/2010 1:30
eLoyalty.DataIntegrationService....	36 KB	PDB File	1/19/2010 1:30
eLoyalty.DataIntegrationService....	80 KB	Application Extension	1/19/2010 1:30
eLoyalty.DataIntegrationService....	202 KB	PDB File	1/19/2010 1:30
FileEncryptionService.Encryption.dll	32 KB	Application Extension	1/19/2010 1:30
log4net.dll	236 KB	Application Extension	7/16/2009 9:42
SharpZipLib.dll	136 KB	Application Extension	7/16/2009 9:42
TestMessageGenerator.exe	28 KB	Application	1/19/2010 1:30

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.



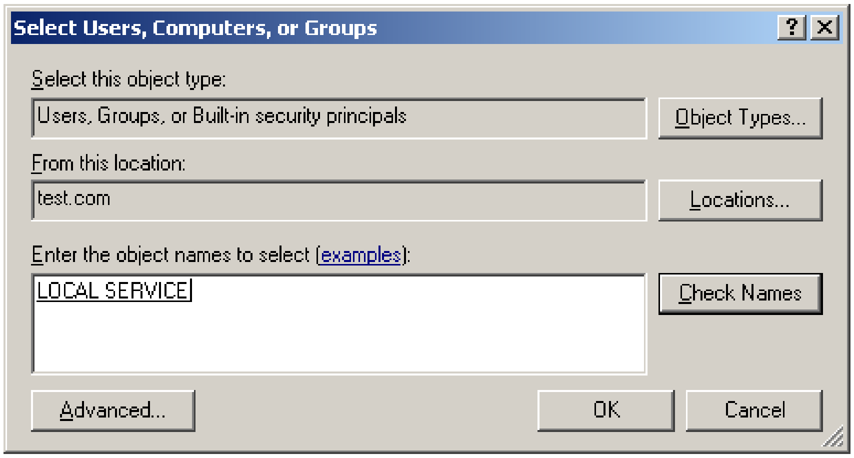
Group or user names:

- Administrators (TEST0\Administrators)
- Authenticated Users
- CREATOR OWNER
- LOCAL SERVICE
- Server Operators (TEST0\Server Operators)

Permissions for Administrators	Allow	Deny
Full Control	☑	□
Modify	☑	□
Read & Execute	☑	□
List Folder Contents	☑	□
Read	☑	□
Write	☑	□
Special Permissions	□	□

For special permissions or for advanced settings, click Advanced.

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**.



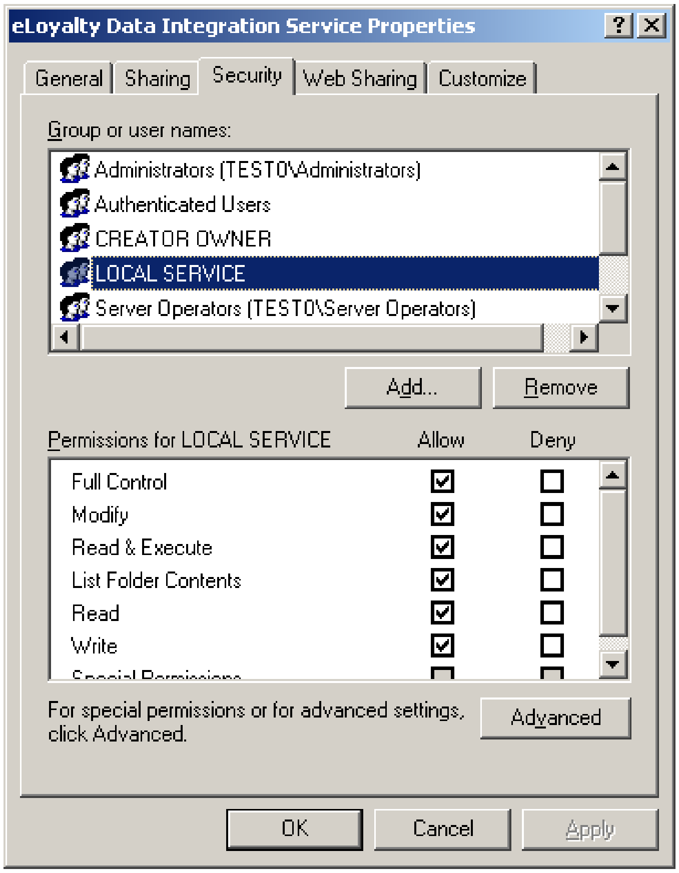
Select Users, Computers, or Groups

Select this object type: Users, Groups, or Built-in security principals

From this location: test.com

Enter the object names to select (examples): LOCAL SERVICE

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



eLoyalty Data Integration Service Properties

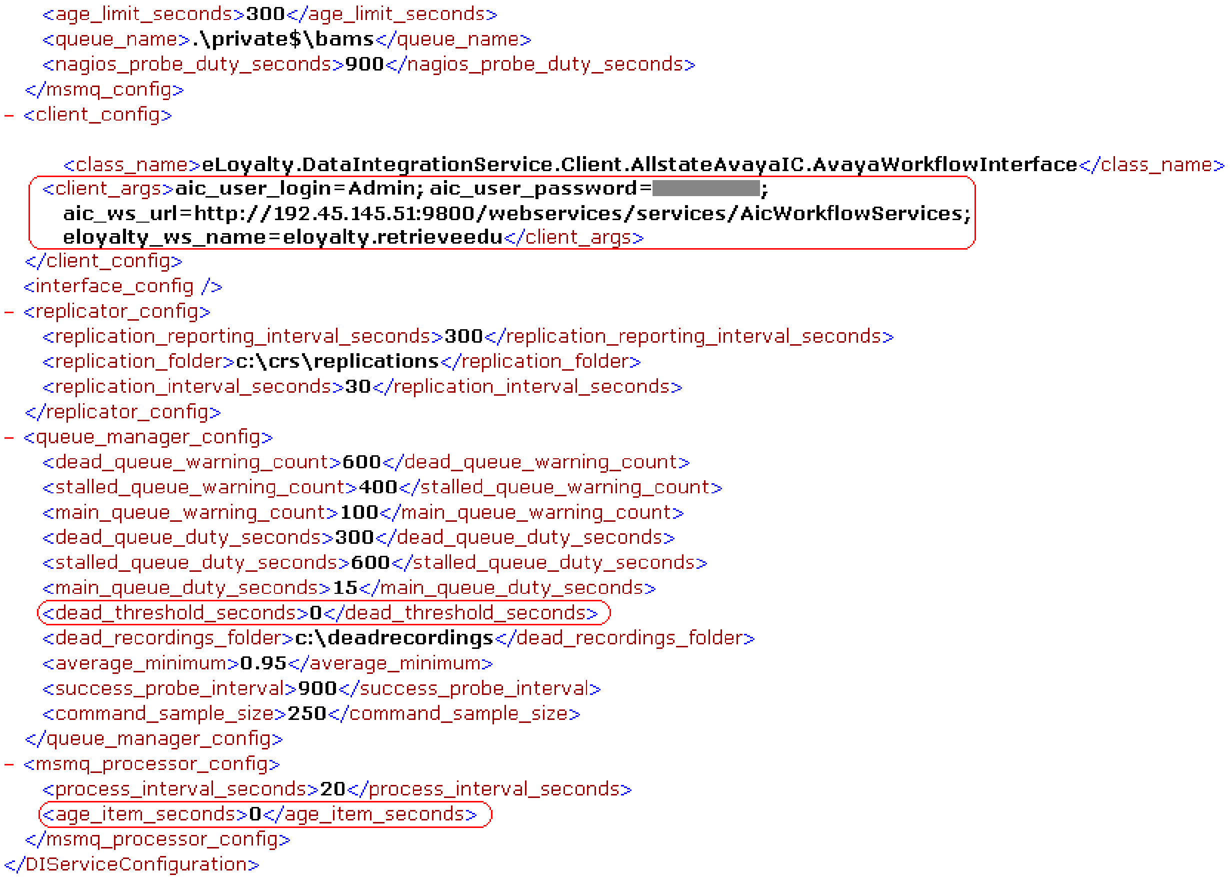
Group or user names:

- Administrators (TESTO\Administrators)
- Authenticated Users
- CREATOR OWNER
- LOCAL SERVICE
- Server Operators (TESTO\Server Operators)

Permissions for LOCAL SERVICE	Allow	Deny
Full Control	☑	□
Modify	☑	□
Read & Execute	☑	□
List Folder Contents	☑	□
Read	☑	□
Write	☑	□
Special Permissions	□	□

For special permissions or for advanced settings, click Advanced.

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.

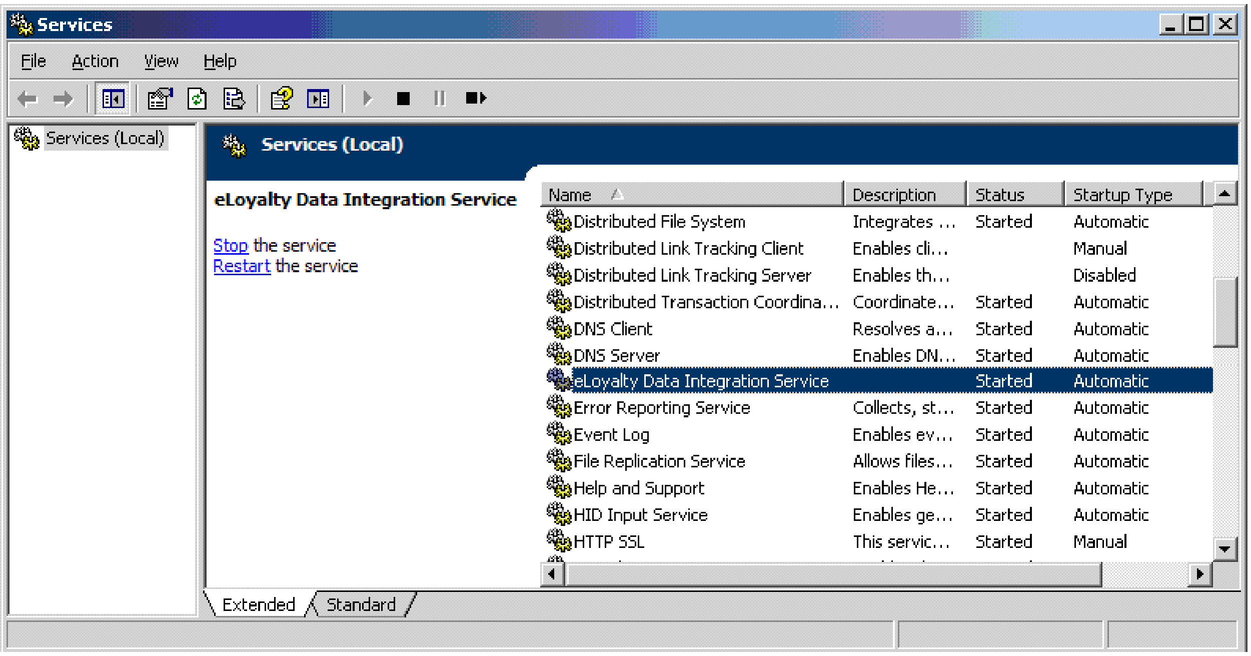


```
      <age_limit_seconds>300</age_limit_seconds>
      <queue_name>.\private$\bams</queue_name>
      <nagios_probe_duty_seconds>900</nagios_probe_duty_seconds>
    </msmq_config>
  - <client_config>

      <class_name>eLoyalty.DataIntegrationService.Client.AllstateAvayaIC.AvayaWorkflowInterface</class_name>
      <client_args>aic_user_login=Admin; aic_user_password=[redacted];
        aic_ws_url=http://192.45.145.51:9800/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.



eLoyalty Data Integration Service

Stop the service
Restart the service

Name	Description	Status	Startup Type
Distributed File System	Integrates ...	Started	Automatic
Distributed Link Tracking Client	Enables cli...		Manual
Distributed Link Tracking Server	Enables th...		Disabled
Distributed Transaction Coordina...	Coordinate...	Started	Automatic
DNS Client	Resolves a...	Started	Automatic
DNS Server	Enables DN...	Started	Automatic
eLoyalty Data Integration Service		Started	Automatic
Error Reporting Service	Collects, st...	Started	Automatic
Event Log	Enables ev...	Started	Automatic
File Replication Service	Allows files...	Started	Automatic
Help and Support	Enables He...	Started	Automatic
HID Input Service	Enables ge...	Started	Automatic
HTTP SSL	This servic...	Started	Manual

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.

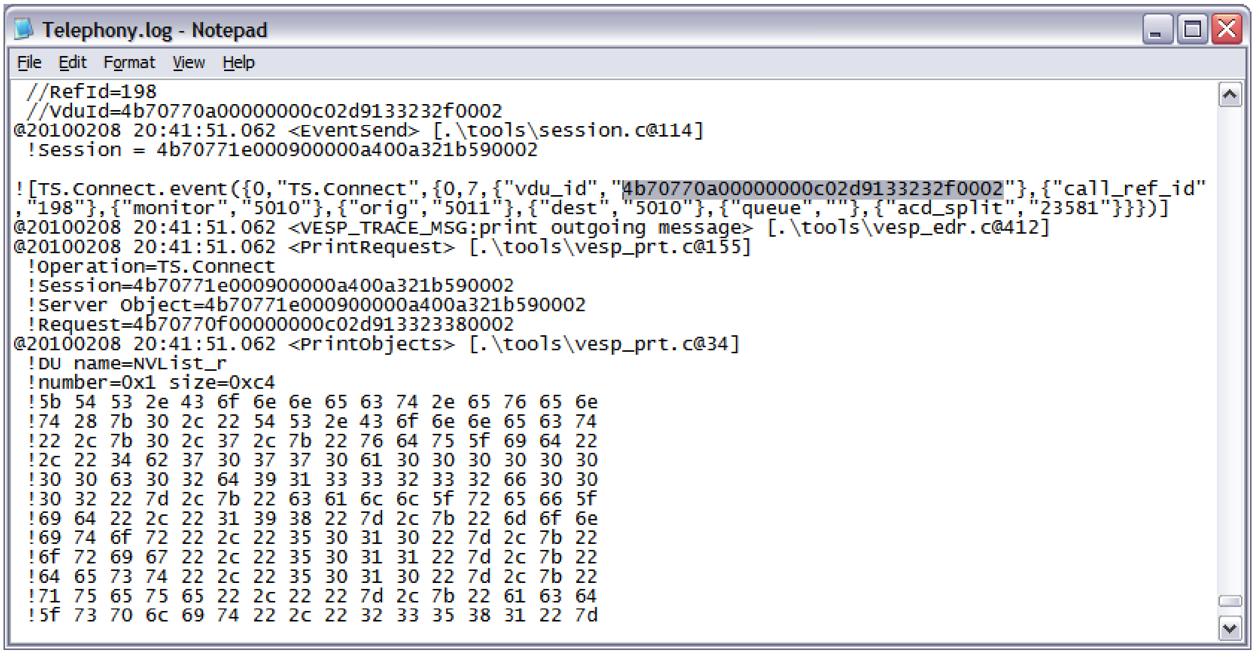
All test cases passed.

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 7.1.

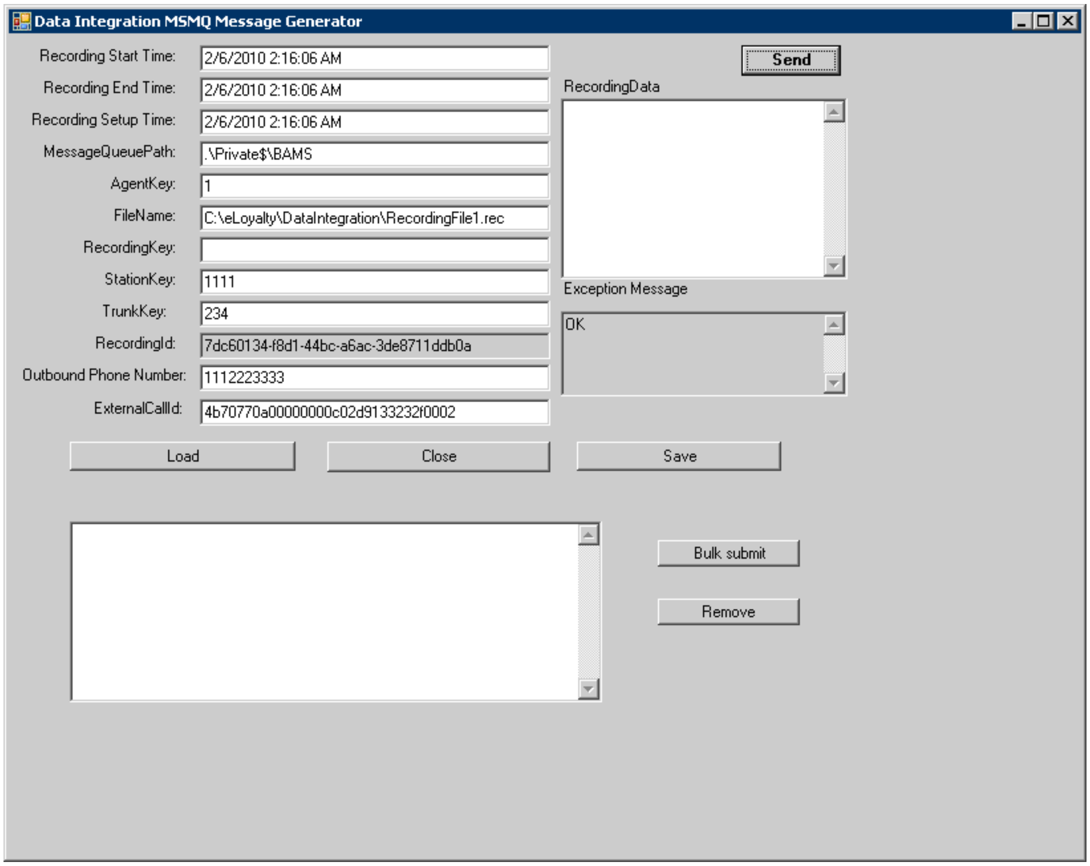
1. Place a call to Communication Manager that gets routed to an IC agent. Terminate and wrap-up the call.

2. Open the `Telephony.log` located in the **logs** directory under the IC 7.1 installation directory. Retrieve and save the EDU ID (or VDU ID) in the log. The name of the log is based on the name of the TS server.



```
 //RefId=198
 //VduId=4b70770a00000000c02d9133232f0002
@20100208 20:41:51.062 <EventSend> [.\tools\session.c@114]
 !Session = 4b70771e000900000a400a321b590002

![TS.Connect.event({0,"TS.Connect",{0,7,{"vdu_id","4b70770a00000000c02d9133232f0002"},{"call_ref_id"
,"198"},{"monitor","5010"},{"orig","5011"},{"dest","5010"},{"queue",""},{"acd_split","23581"}}})]
@20100208 20:41:51.062 <VESP_TRACE_MSG:print outgoing message> [.\tools\vesp_edr.c@412]
@20100208 20:41:51.062 <PrintRequest> [.\tools\vesp_prt.c@155]
 !Operation=TS.Connect
 !Session=4b70771e000900000a400a321b590002
 !Server Object=4b70771e000900000a400a321b590002
 !Request=4b70770f00000000c02d913323380002
@20100208 20:41:51.062 <PrintObjects> [.\tools\vesp_prt.c@34]
 !DU name=NVList_r
 !number=0x1 size=0xc4
 !5b 54 53 2e 43 6f 6e 6e 65 63 74 2e 65 76 65 6e
 !74 28 7b 30 2c 22 54 53 2e 43 6f 6e 6e 65 63 74
 !22 2c 7b 30 2c 37 2c 7b 22 76 64 75 5f 69 64 22
 !2c 22 34 62 37 30 37 37 30 61 30 30 30 30 30 30
 !30 30 63 30 32 64 39 31 33 33 32 33 32 66 30 30
 !30 32 22 7d 2c 7b 22 63 61 6c 6c 5f 72 65 66 5f
 !69 64 22 2c 22 31 39 38 22 7d 2c 7b 22 6d 6f 6e
 !69 74 6f 72 22 2c 22 35 30 31 30 22 7d 2c 7b 22
 !6f 72 69 67 22 2c 22 35 30 31 31 22 7d 2c 7b 22
 !64 65 73 74 22 2c 22 35 30 31 30 22 7d 2c 7b 22
 !71 75 65 75 65 22 2c 22 22 7d 2c 7b 22 61 63 64
 !5f 73 70 6c 69 74 22 2c 22 32 33 35 38 31 22 7d
```

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



Data Integration MSMQ Message Generator

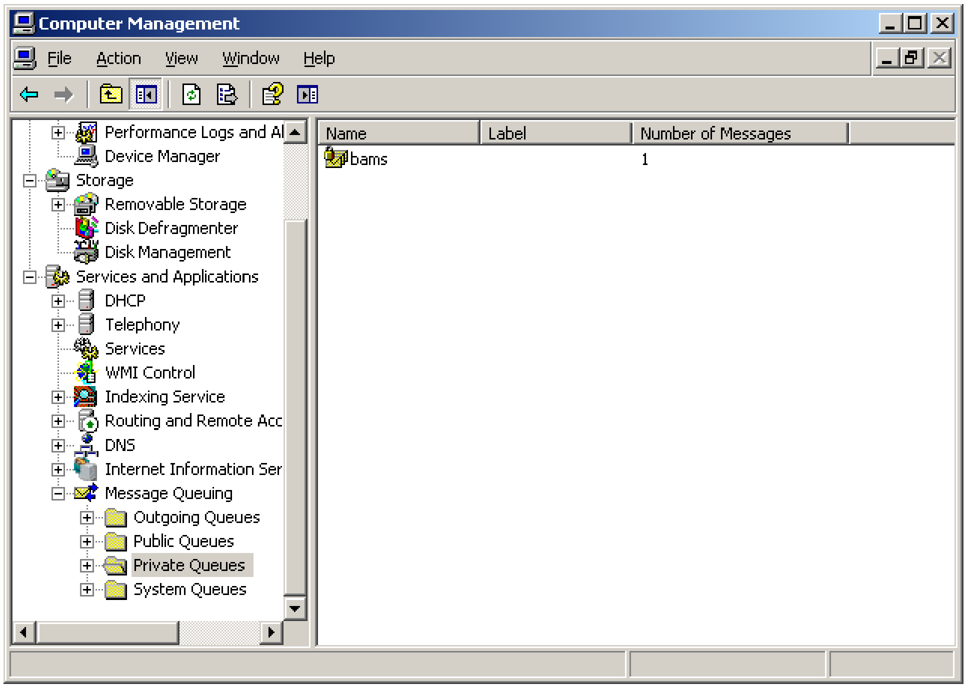
Recording Start Time: 2/6/2010 2:16:06 AM
Recording End Time: 2/6/2010 2:16:06 AM
Recording Setup Time: 2/6/2010 2:16:06 AM
MessageQueuePath: .\Private$\BAMS
AgentKey: 1
FileName: C:\eLoyalty\DataIntegration\RecordingFile1.rec
RecordingKey:
StationKey: 1111
TrunkKey: 234
RecordingId: 7dc60134-f8d1-44bc-a6ac-3de8711ddb0a
Outbound Phone Number: 1112223333
ExternalCallId: 4b70770a00000000c02d9133232f0002

RecordingData

Exception Message

OK

4. 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.

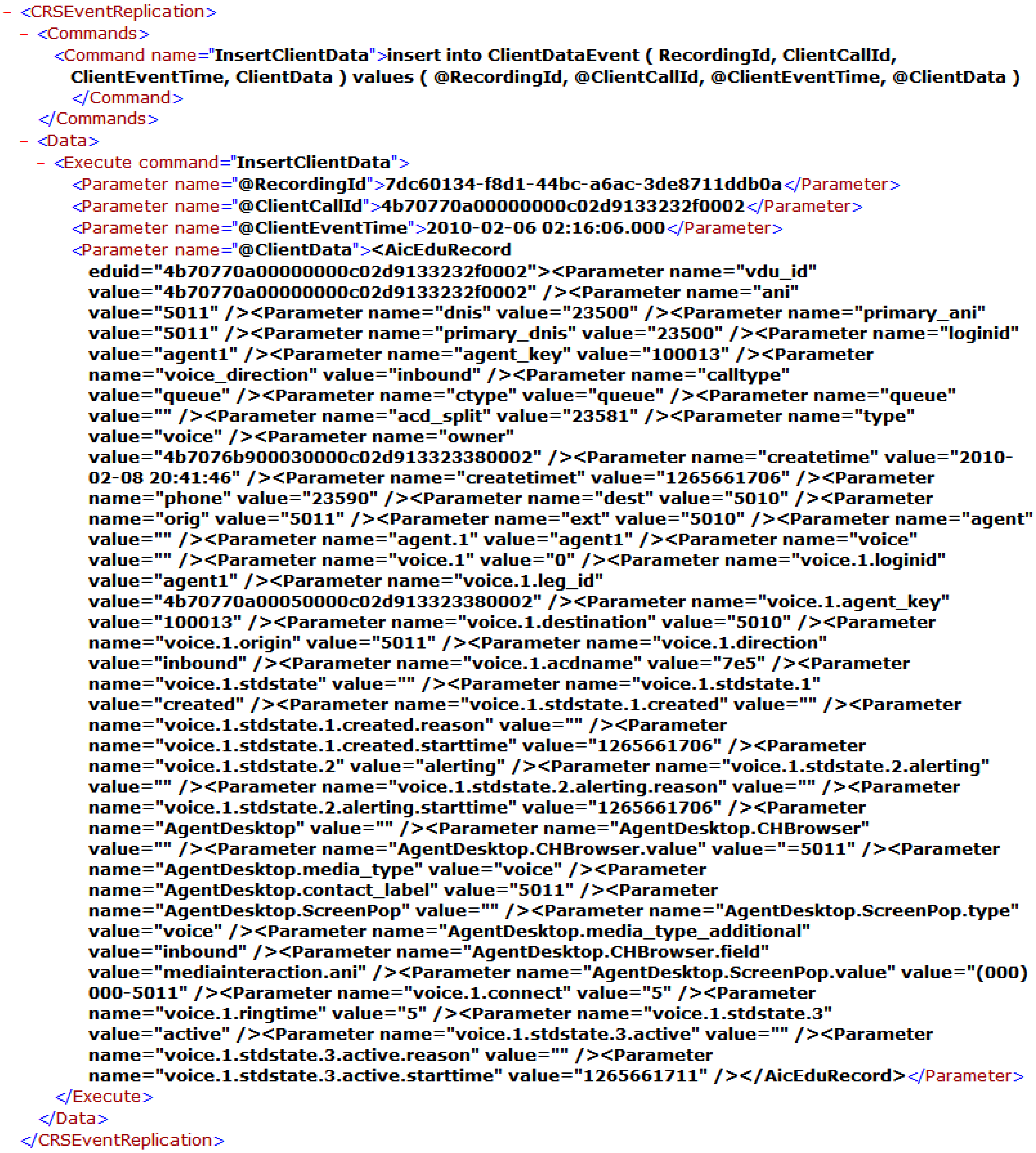


Computer Management

Name	Label	Number of Messages
bams		1

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">7dc60134-f8d1-44bc-a6ac-3de8711ddb0a</Parameter>
        <Parameter name="@ClientCallId">4b70770a00000000c02d9133232f0002</Parameter>
        <Parameter name="@ClientEventTime">2010-02-06 02:16:06.000</Parameter>
        <Parameter name="@ClientData"><AicEduRecord
          eduid="4b70770a00000000c02d9133232f0002"><Parameter name="vdu_id"
          value="4b70770a00000000c02d9133232f0002" /><Parameter name="ani"
          value="5011" /><Parameter name="dnis" value="23500" /><Parameter name="primary_ani"
          value="5011" /><Parameter name="primary_dnis" value="23500" /><Parameter name="loginid"
          value="agent1" /><Parameter name="agent_key" value="100013" /><Parameter
          name="voice_direction" value="inbound" /><Parameter name="calltype"
          value="queue" /><Parameter name="ctype" value="queue" /><Parameter name="queue"
          value="" /><Parameter name="acd_split" value="23581" /><Parameter name="type"
          value="voice" /><Parameter name="owner"
          value="4b7076b900030000c02d913323380002" /><Parameter name="createtime" value="2010-
          02-08 20:41:46" /><Parameter name="createtimet" value="1265661706" /><Parameter
          name="phone" value="23590" /><Parameter name="dest" value="5010" /><Parameter
          name="orig" value="5011" /><Parameter name="ext" value="5010" /><Parameter name="agent"
          value="" /><Parameter name="agent.1" value="agent1" /><Parameter name="voice"
          value="" /><Parameter name="voice.1" value="0" /><Parameter name="voice.1.loginid"
          value="agent1" /><Parameter name="voice.1.leg_id"
          value="4b70770a00050000c02d913323380002" /><Parameter name="voice.1.agent_key"
          value="100013" /><Parameter name="voice.1.destination" value="5010" /><Parameter
          name="voice.1.origin" value="5011" /><Parameter name="voice.1.direction"
          value="inbound" /><Parameter name="voice.1.acdname" value="7e5" /><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="1265661706" /><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="1265661706" /><Parameter
          name="AgentDesktop" value="" /><Parameter name="AgentDesktop.CHBrowser"
          value="" /><Parameter name="AgentDesktop.CHBrowser.value" value="=5011" /><Parameter
          name="AgentDesktop.media_type" value="voice" /><Parameter
          name="AgentDesktop.contact_label" value="5011" /><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)
          000-5011" /><Parameter name="voice.1.connect" value="5" /><Parameter
          name="voice.1.ringtime" value="5" /><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="1265661711" /></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 7.1. 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 7.1 Installation Planning and Prerequisites*, Release 7.1, Document ID 07-300568, May 2006, available at http://support.avaya.com.

[2] *Avaya Interaction Center Release 7.1 Installation and Configuration*, Release 7.1, Document ID 07-300569, May 2006, available at http://support.avaya.com.

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.