



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Genesys Mobile Services Developer's Guide

Genesys Mobile Engagement 8.1.1

Table of Contents

Welcome to the Developer's Guide!	3
Sample Application Overview	4
InsureCo Sample	5
ORS Samples Overview	7
Custom Reporting	8
Sample Resources	12

Welcome to the Developer's Guide!

This Developer's Guide contains information that will help you understand what sample applications are included with your Genesys Mobile Services installation, how to test those samples in your own environment, and what you can do to jump start your own development based on the example these applications provide. The [overview](#) page provides a quick outline of samples included with the release, or you can go directly to the sample-specific documentation using the following links:

- [InsureCo Sample](#)
- [ORS Samples Overview](#)
- [Custom Reporting](#)

Downloads for each sample are provided on the respective documentation pages, and can also be found on the [Sample Resources](#) page.

Other Tasks and Resources

Before reading this material you may want to:

- [Install](#) Genesys Mobile Services.
- Ensure you have access to the latest version of the [API Reference](#).
- Download the latest version of the Release Note (using links on the [Genesys Mobile Services Product Page](#)) to see the most recent news and updates about this product.

Sample Application Overview

This chapter provides an brief overview of the different sample applications included with Genesys Mobile Services. As additional sample applications are made available, this chapter will be updated to provide supporting information. For additional details, please refer to appropriate pages below.

- [InsureCo Sample](#)

InsureCo Sample

InsureCo Sample Overview

This sample application allows a user to create an insurance claim, after which an option is given to either talk with an agent as soon as the claim is submitted or to wait and be notified when an agent becomes available. These scenarios demonstrate how to implement one basic (request-interaction) and one advanced (request-inbound-delay) service, as well as using APNS Push messages and Google C2DM from Genesys Mobile Services. The intended audience for this sample are Enterprise iPhone/Android developers who plan to implement mobile services using the Genesys Mobile Services platform.

Scenarios

1. In the basic scenario a call is placed immediately after the user creates and submits a claim, allowing that user to talk with an agent.
2. In the advanced scenario, a user creates a claim but is unable to contact an agent right away. When an agent becomes available, the application notifies the user, reserves the available agent, and offer appropriate call options.

In addition to these scenarios, you can use this sample to explore other features supported by Genesys Mobile Services such as sending the geolocation coordinates or storing and uploading images as part of your request to the Genesys Mobile Services server.

Related Information

InsureCo Sample System Requirements

Platform	Build Requirements	Runtime Requirements
iPhone	iOS 5.0 SDK	iOS 5.0
Android	<ul style="list-style-type: none"> • Android SDK (API level 13 and more) • Apache Maven 3.x 	Android OS 2.2.1

Downloadable Files

-  [Genesys Mobile Services Sample InsureCo Native iOS](#)
-  [Genesys Mobile Services Sample InsureCo Native Android](#)

-  [Genesys Mobile Services Sample ORS Samples](#)

ORS Samples Overview

Downloadable Files

-  [Genesys Mobile Services Sample ORS Samples](#)

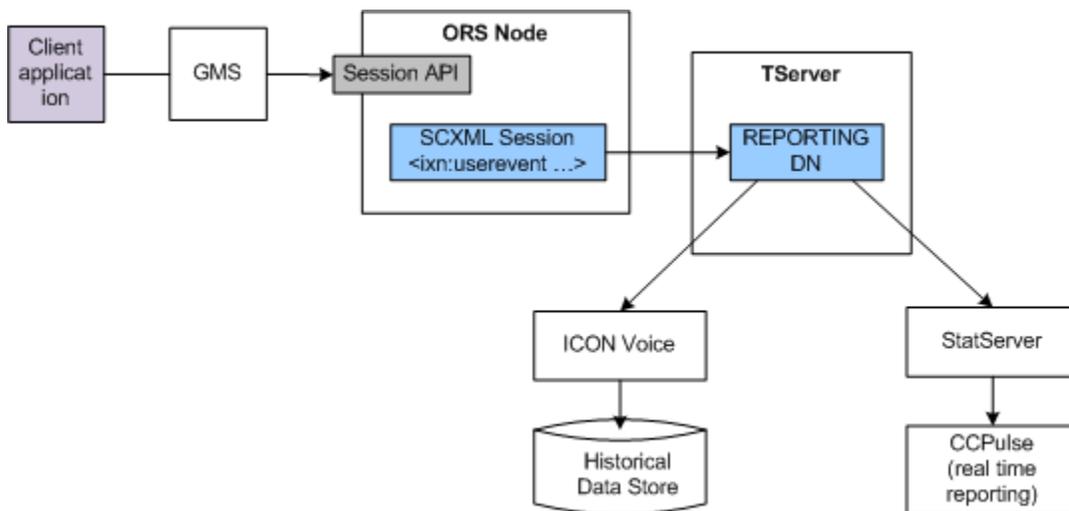
Custom Reporting

Basic Configuration for Real-Time and Historical Reporting Based on T-Server's UserEvent Mechanism

Prerequisites

1. ORS is connected to T-Server
2. StatServer is connected to the same T-Server (if you need real-time reporting)
3. Icon is connected to T-Server and configured to store user events to G_CUSTOM_DATA_P table (if you need historical reporting)

Architecture



Configuration instructions

- Create a new DN of type **Extension**. The name of the DN is not important, but it is used inside SCXML scripts so should be meaningful and recognizable.
Example: Sip_Switch -> DN -> REPORTING
- Make sure Icon and StatServer are connected to the T-Server that is servicing the switch specified in step 1.
Example: Sip_Server.
- In Icon configuration in Configuration Manager, add the 'custom-states' section under Options and create the **GlobalData** option there. List attached data fields you want to capture preceded with data

type.

Example:

```
custom-states/GlobalData=char,gms_SessionId, char,gms_SessionEventSeq, char,gms_ServiceName,
char,gms_UserId, char,gms_externalId,
char,gms_ServiceStartAt, char,gms_WaitingForAgent,
char,gms_AgentAvailable, char,gms_UserConnected,
char,gms_AgentConnected, char,gms_IxnCompleted,
char,gms_ServiceStoppedAt
```

- Start Icon and StatServer (if not already started) and use logs to verify they registered on REPORTING DN.
- Add the following block of code to the beginning of your SCXML flow. This code will setup the `_data.userevent_u_data_to_send` variable to store all significant state changes you want to capture from the point of view of reporting. Names should match Icon's GlobalData configuration option and StatServer/CCPulse reporting and statistics templates.
Example:

```
<datamodel>

</datamodel>
<script>
    _data.userevent_u_data_to_send = {
        'gms_SessionId':_sessionid,
        'gms_SessionEventSeq':0,
        'gms_ServiceName':'your service name here',
        'gms_UserId':,
        'gms_externalId':,
        // service state change timestamps
        'gms_ServiceStartAt': ,
        'gms_WaitingForAgent':,
        'gms_AgentAvailable':,
        'gms_UserConnected':,
        'gms_AgentConnected':,
        'gms_IxnCompleted':,
        'gms_ServiceStoppedAt':
    };
</script>
```

- Add following block of code into your SCXML flow where significant state change is happening:

```
<script>
    _data.userevent_u_data_to_send.gms_WaitingForAgent = new Date().getTime().toString();
    _data.userevent_u_data_to_send.gms_SessionEventSeq =
_data.userevent_u_data_to_send.gms_SessionEventSeq + 1;
</script>
<ixn:userevent requestid="_data.userevent_reqid" resource="{ 'switch': 'SIP_Switch',
'dn': 'REPORTING' }"
    udata="_data.userevent_u_data_to_send"/>
```

Example:

```
<state id="waitForAgent">
    <onentry>
        <script>
            _data.userevent_u_data_to_send.gms_WaitingForAgent = new
Date().getTime().toString();
            _data.userevent_u_data_to_send.gms_SessionEventSeq =
_data.userevent_u_data_to_send.gms_SessionEventSeq + 1;
```

```

    </script>
    <ixn:userevent requestid="_data.userevent_reqid"
resource="{ 'switch': 'SIP_Switch', 'dn': 'REPORTING' }"
              udata="_data.userevent_udata_to_send"/>
    <queue:submit route="false" timeout="_data.queueSubmitTimeout">
      <queue:targets type="agentgroup">
        <queue:target name="_data.defaultAgentGroup"/>
      </queue:targets>
    </queue:submit>
  </onentry>

  <transition event="queue.submit.done" target="agentAvailable"/>
  <transition event="error.queue.submit" target="error">
</transition>
  <transition event="service.ttl.expired" target="error">
</transition>
</state>

```

Verifying reporting data

- Run your scenario by triggering Genesys Mobile Services and Orchestration Server (ORS) APIs directly.
- Make sure user events are being delivered to StatServer and Icon applications by checking T-Server logs. You should see something like this:

```

00:34:20.757 Int 04543 Interaction message "RequestDistributeUserEvent" received from 516
("OrchestrationServer")
-- Absent ThisDN, REPORTING was used
@00:34:20.7570 [0] 8.1.000.62 send_to_client: message EventACK
  AttributeEventSequenceNumber      0000000000000ef8
  AttributeCustomerID                'Environment'
  AttributeTimeinUsecs               757000
  AttributeTimeinSecs                1348817660 (00:34:20)
  AttributeReferenceID               431
  AttributeThisDN                    'REPORTING'
  AttributeUserEvent                 RequestDistributeUserEvent
00:34:20.757 Trc 04542 EventACK sent to [516] (00000003 OrchestrationServer
192.168.27.50:40727)
@00:34:20.7570 [0] 8.1.000.62 distribute_user_event: message EventUserEvent
  AttributeEventSequenceNumber      0000000000000ef9
  AttributeCustomerID                'Environment'
  AttributeTimeinUsecs               757000
  AttributeTimeinSecs                1348817660 (00:34:20)
  AttributeUserEvent                 EventUserEvent
  AttributeUserData                  [347] 00 0c 00 00..
    'gms_AgentAvailable'            '1348817660755'
    'gms_AgentConnected'
    'gms_IxnCompleted'
    'gms_ServiceName'              'inbound-delay'
    'gms_ServiceStartAt'            '1348817660553'
    'gms_ServiceStoppedAt'
    'gms_SessionEventSeq'           3
    'gms_SessionId'                 '65UA6ISSJH76R340BNDQ2DG0DG000036'
    'gms_UserConnected'
    'gms_UserId'
    'gms_WaitingForAgent'            '1348817660744'
    'gms_externalId'
  AttributeANI                       '777'
  AttributeDNIS                       '333'
  AttributeReferenceID                431

```

```

AttributeThisDN      'REPORTING'
00:34:20.758 Trc 04542 EventUserEvent sent to [508] (0000000c Icon_Voice 192.168.27.50:42678)
00:34:20.758 Trc 04542 EventUserEvent sent to [588] (00000004 Stat_Server
192.168.27.50:40728)
00:34:20.758 Trc 04542 EventUserEvent sent to [592] (00000005 Universal_Routing_Server
192.168.27.50:40744)
    
```

- Check your Icon log and G_CUSTOM_DATA_P table and make sure data is recorded properly. Examples:

Icon log:

```

00:39:19.569 Int 04543 Interaction message "EventUserEvent" received from 65200
("SIP_Server@REPORTING")
00:39:19.751 Int 04543 Interaction message "EventUserEvent" received from 65200
("SIP_Server@REPORTING")
00:39:19.766 Int 04543 Interaction message "EventUserEvent" received from 65200
("SIP_Server@REPORTING")
00:39:19.987 Trc 25016 Persistent Queue GUD: transaction 10929 is committed. 5 records
written into the queue
00:39:19.987 Trc 25003 Database queue [GUD]: persistent queue transaction 10929 is being
processed.
00:39:20.001 Trc 25004 Database queue [GUD]: persistent queue transaction 10929 is
processed, committed and removed. 5 records are written.
    
```

Icon's G_CUSTOM_DATA_P table:

```

select * from dbo.G_CUSTOM_DATA_P

8          0          830          REPORTING          0          101          1          2012-09-28
07:43:09.443          1348818189          4496060          65UA6ISSJH76R340BNDQ2DG0DG000038
1          inbound-delay          1348818189441
9          0          830          REPORTING          0          101          1          2012-09-28
07:43:09.590          1348818189          4496060          65UA6ISSJH76R340BNDQ2DG0DG000038
2          inbound-delay          1348818189441          1348818189590
10         0          830          REPORTING          0          101          1          2012-09-28
07:43:09.600          1348818189          4496060          65UA6ISSJH76R340BNDQ2DG0DG000038
3          inbound-delay          1348818189441          1348818189590
1348818189596
    
```

- Start CCPulse and create a reporting template for monitoring REPORTING DN.

Congratulations: you are done!

Sample Resources

List of Genesys Mobile Services Code Samples

Sample	Description	Documentation	Resource
Orchestration Server Samples	A package of sample files (DFM, SCXML, and VXML) showcasing how Genesys Mobile Services and Orchestration Server interact.	ORS Samples Overview	 Genesys Mobile Services Sample ORS Samples
InsureCo Sample Application	A sample application allows users to create an insurance claim and receive the option to either talk with an agent as soon as the claim is submitted, or to wait and be notified when an agent becomes available. Demonstrates how to implement one basic (request-interaction) and one Advanced (request-inbound-delay) service, as well as using Push messages from Genesys Mobile Services.	InsureCo Sample	<ul style="list-style-type: none">  Genesys Mobile Services Sample InsureCo Native iOS  Genesys Mobile Services Sample InsureCo Native Android
Interaction Workspace Test Plugin	Unzip the file in the <gms install dir>/webapps folder.	Genesys Mobile Services Configuration	 Genesys Mobile Services Interaction Workspace Test Plugin