User's Guide

Context Services 8.5.2

2/6/2024

Table of Contents

Context Services 8.5 User's Guide

The Context Services User’s Guide describes the Context Services functionality of the Conversation Manager Solution, and includes deployment information.

Every Genesys product includes a Release Note that provides any late-breaking product information that could not be included in the manual. This product information can often be important. To view it, open the `read_me.html` file in the application home directory, or follow the link under the Release Notes section of the product page to download the latest Release Note for this product.

Description

Context Services's REST APIs provide facilities to store your application data as Conversations. Each Conversation represents data collected when your application interacts with your Customer and then stored as Services, States, and Tasks via URI paths.

To use a REST API, your application will make an HTTP request and parse the response. By default, the response format is XML. If you wish, you can request JSON instead of XML. Because the REST API is based on open standards, you can use any web development language to access the API.

About GMS

Genesys Mobile Services (GMS) with Conversation Manager Solution brings business rules, context, conversation history, reporting, locations, and preferences to mobile interactions, enabling you to personalize every mobile experience.

GMS now embeds Context Services to enhance scalability and provide new interfaces in addition to new features such as Customer Journey. GMS/CS may be used when describing the Context Services capabilities in GMS, which include all the services, states, and tasks REST APIs. Note that GMS/CS does not provide profile management APIs.

About UCS

Universal Contact Server (UCS) interfaces with a database that stores data on contacts (customers). As the classic UCS, it works with Genesys eServices (Multimedia). With an optional set of additional capabilities known as Context Services, it works with other Genesys products and solutions, such as Genesys Voice Portal and Conversation Manager.

Note the UCS deployment is required only if your application need to handle profiles.

This part of the User's Guide covers:

- Installation and configuration of GMS and Context Services.
- Migration of your services from a UCS installation to a GMS installation.

What You Should Know

This guide is written for software developers and application architects who intend to create applications that interact with Genesys environments. Before working with Context Services, you should have an understanding of:

- computer-telephony integration (CTI) concepts, processes, terminology, and applications
- network design and operation
- your own network configurations
- Genesys Framework architecture

Scope of Use

Typical usage scenarios of Context Services include:

- Customer identification
- Service resumption
- Customer profile (retrieval and management)
- Callback offers
- Service resumption with an agent
- Proactive notification

New in This Document

The following topics have been added or changed in the GMS 8.5.205 release.

- This release includes no documentation updates for this guide.

Conversation Manager

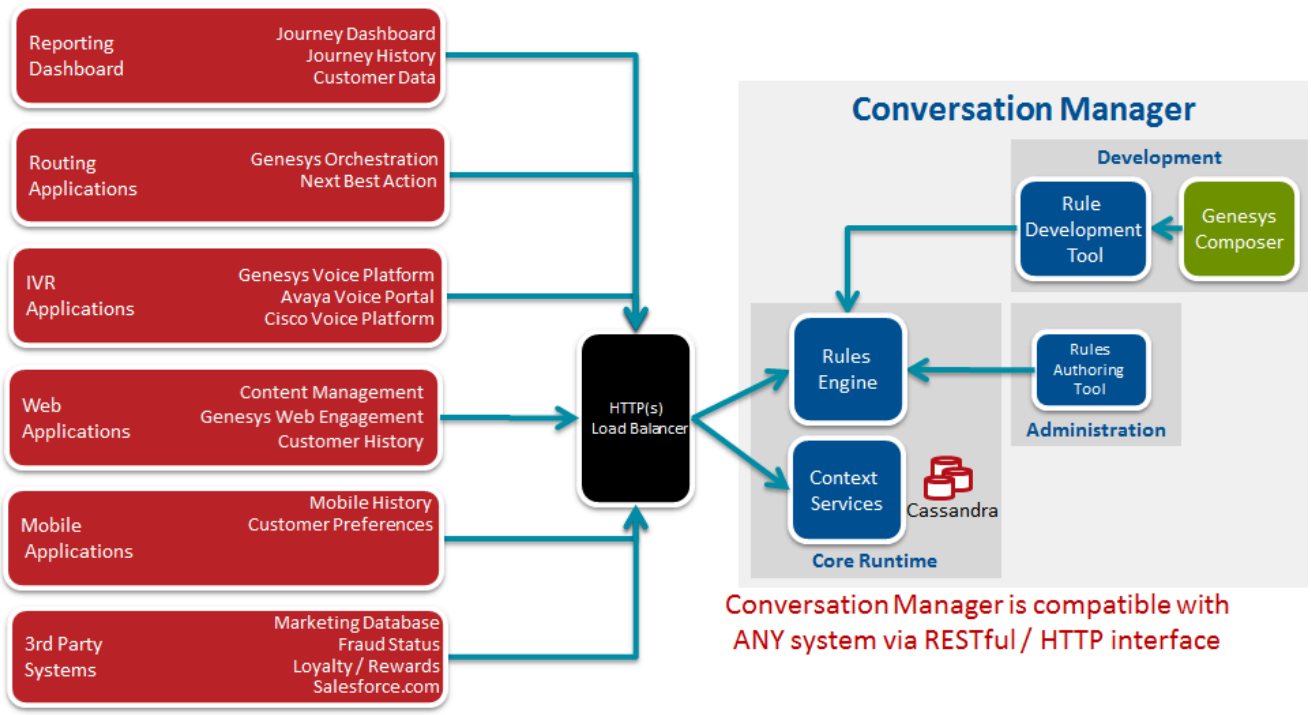
Describes the Conversation Manager and related components.

Genesys Conversation Manager takes Genesys' core capability of routing and extends it, generalizes it, and integrates it more tightly with other Genesys products. Rather than the call (T-Server) or the interaction (eServices/Multimedia), Conversation Manager takes the service as the basic entity. It orchestrates the service process across channels and over time, using dynamic data and business rules to make decisions about operations. For example;

A bank customer calls a toll-free number inquiring about mortgage preapproval. An IVR prompts him to enter his account number, then transfers him to an agent, who fills in an application form for him and asks him to fax some supporting documents. After he faxes the documents, he receives an SMS message thanking him and informing him that he will receive a response within 48 hours. The next day he receives an email congratulating him on the approval of his application.

This example involves voice, IVR, fax, SMS, and email channels. Conversation Manager is able to treat the entire sequence as a single service.

Architecture



Conversation Manager consists of the following components:

- **Context Services**—Provides contextual awareness of interactions. Know who the customer is, what they want, and where they are in this process. Context Services also comes with a tool to manage Service, State and Tasks.
- **Genesys Rules System**—Enables If-Then actions such as, “If we KNOW that the customer is a frequent user of our self-service tracking, THEN we offer self-service tracking as the first option in the menu.”
- **Journey Timeline**—A visual representation of the customer journey map. It depicts all the touch points of the customer for various services on different channels.
- **Journey Dashboard**—A visual representation of key performance indicators, rule executions, and journey metrics.

Services

Conversation Manager adds to Genesys the concept of service, which may be defined as follows:

- It represents a business process, which in turn may be seen as a communication or series of communications between a customer and an enterprise, and possibly also between various parts of the enterprise.
- It can span multiple interactions.
- It may include interactions in various media.
- It has a temporal beginning and end.
- It may be subdivided into states, which in turn may be subdivided into tasks (see also services, states, and tasks in the Developer's Guide).

> **Important**
>
> This term *state* does not have the same meaning as "SCXML state."

Orchestration Server

Orchestration Server has a function in Conversation Manager similar to the function of Universal Routing Server (URS) in Genesys voice and multimedia solutions. One of the main differences is that it operates based on business processes developed in State Chart XML (SCXML) rather than routing strategies written in IRL (Intelligent Routing Language, a Genesys proprietary language).

SCXML applications

SCXML applications can be written directly using any XML or plain text editor, or with Genesys

Composer, an Eclipse-based development environment. They are published on an application server such as JBoss or another Java-based application server, and are executed on Orchestration Server.

Genesys Composer

Composer also provides a set of function blocks that allow access to Context Services. These out-of-the box function blocks on the workflow diagram palette allow the developer to create applications that perform various actions, such as:

- Identify customers and update their profiles.
- Extend customer profiles with user-defined information.
- Query a customer's profile.
- Create, start, complete, and query customer services.
- Query customers' active services.
- Enter, complete, and query service states.

UCS in eServices and Conversation

UCS in eServices (Multimedia)

Genesys eServices (called Multimedia before release 8.0.1) is a cover term for Genesys components that work together to manage interactions whose media is something other than traditional telephonic voice (for example, email or chat). eServices includes some parts of the Genesys Customer Interaction Management (CIM) Platform, plus certain of the media channels that run on top of the Platform. UCS's function in eServices is to store and manage the following:

- Contact data
- Interaction data
  - The body of an interaction (plus associated metadata and user data) while it is being processed
  - The history of an interaction, including its place (if any) in a thread.
- Knowledge Management data: category systems, screening rules, standard responses, training objects, and models (training objects and models are available only with the Content Analyzer option).

In the context of eServices, clients communicate with UCS using RMI (Remote Method Invocation) and ESP (External Service Protocol, a Genesys protocol). For more details see the Preface and the "Overview" chapter in the *eServices 8.0 Deployment Guide.*

UCS in Conversation Manager

Central to Conversation Manager is the ability to maintain a unified view of the customer. Conversation Manager can use this knowledge in areas such as service personalization, enablement of service continuity, and in upsell/cross-sell campaigns. Context Services is the name of a group of additional capabilities that UCS provides. These capabilities can be invoked by any client, but most prominently by the components of the Conversation Manager solution. The Context Services functioning of UCS differs from its functioning in eServices in the following ways:

- In addition to interaction data and contact data (called customer data in the Context Services context), UCS/CS stores data on services. Services are the basic units in a model for business context used in customer service applications. See also services, states, and tasks in the Developer's Guide.
- Clients communicate with UCS using RESTful (HTTP) web services, not RMI or ESP.
- Context Services uses a different procedure for contact identification and creation.
- Context Services organizes data on contacts differently. See also profiles and identification in the Developer's Guide.

Prerequisites

To work with Context Services (CS), your system must meet the software requirements established in the Genesys Supported Operating Environment Reference Manual, as well as meeting the following minimum requirements.

> **Important**
> Make sure to estimate the size of solution that will be able to handle the expected user load. See Sizing Calculator.

Hardware Requirements

The following are minimum requirements:

- CPU: Quad core
- Memory: 4GB
- Disk: 160GB
- At least 2-3 nodes recommended for redundancy and availability

OS Requirements

- Genesys Supported Operating Environment Reference Guide

> **Important**
> For Linux installations, the Linux compatibility packages must be installed prior to installing the Genesys IPs.

Browser Support

- Genesys Supported Operating Environment Reference Guide

Java Requirements

- Context Services requires a JDK.

- Context Services supports JDK 8.

Genesys Environment

You must have a Genesys Mobile Services environment installed and running. See the the GMS prerequisites page for a list of Genesys components that are used with a GMS installation.

Sizing

You should also check the guidelines about sizing prior to your installation.

Sizing Calculator

Before deploying Context Services (CS) for your Conversation Manager solution, you must estimate the size of solution that will be able to handle the expected user load. Genesys recommends that you download the **CS Storage Sizing Calculator**, an Excel spreadsheet that you can use to help calculate the number of Context Services nodes required for your production deployment.

CS Storage Sizing Calculator
CS-StorageSizing.xlsx

Using the Sizing Calculator Spreadsheet

To use the sizing calculator, The following information is required:

- Approximate event size in bytes for event start/end/update of service/state/task (including extension data). You can extract this information from the JSON of the events.
- Number of update events per services/states/tasks
- Number of states/tasks per service
- Number of Cassandra nodes and replication factor
- Retention policy in days and number of conversation per day

The sizing calculator takes into account Cassandra storage specifics such as Replication Factor (storage size is multiplied by 3 as default replication), compression (ratio is estimated to 0.40 by default), and compaction overhead which requires more storage during cycles of data re-organization by Cassandra.

Input Example

Genesys Context Services - Sizing Input Parameters			
Events Parameter	**Event Size (bytes)**	**Number of Events**	**Comments**
service start event	246	1	
service end event	227		
service update event(s)	291	0	
state start event	210	5	
state end event	225		
state update event(s)	192	0	
task start event	208	10	
task end event	228		
task update event(s)	165	0	
Volume Parameter	**Value**	**Comments**	
Number of Conversations per Day	160000		
Conversations life (days)	90		
Number of storage nodes	4		
Replication Factor	3		
Constants	**Value**	**Comments**	
Compression ratio	0,50		
CommitLog Size (MB)	8192,00		

Output Example

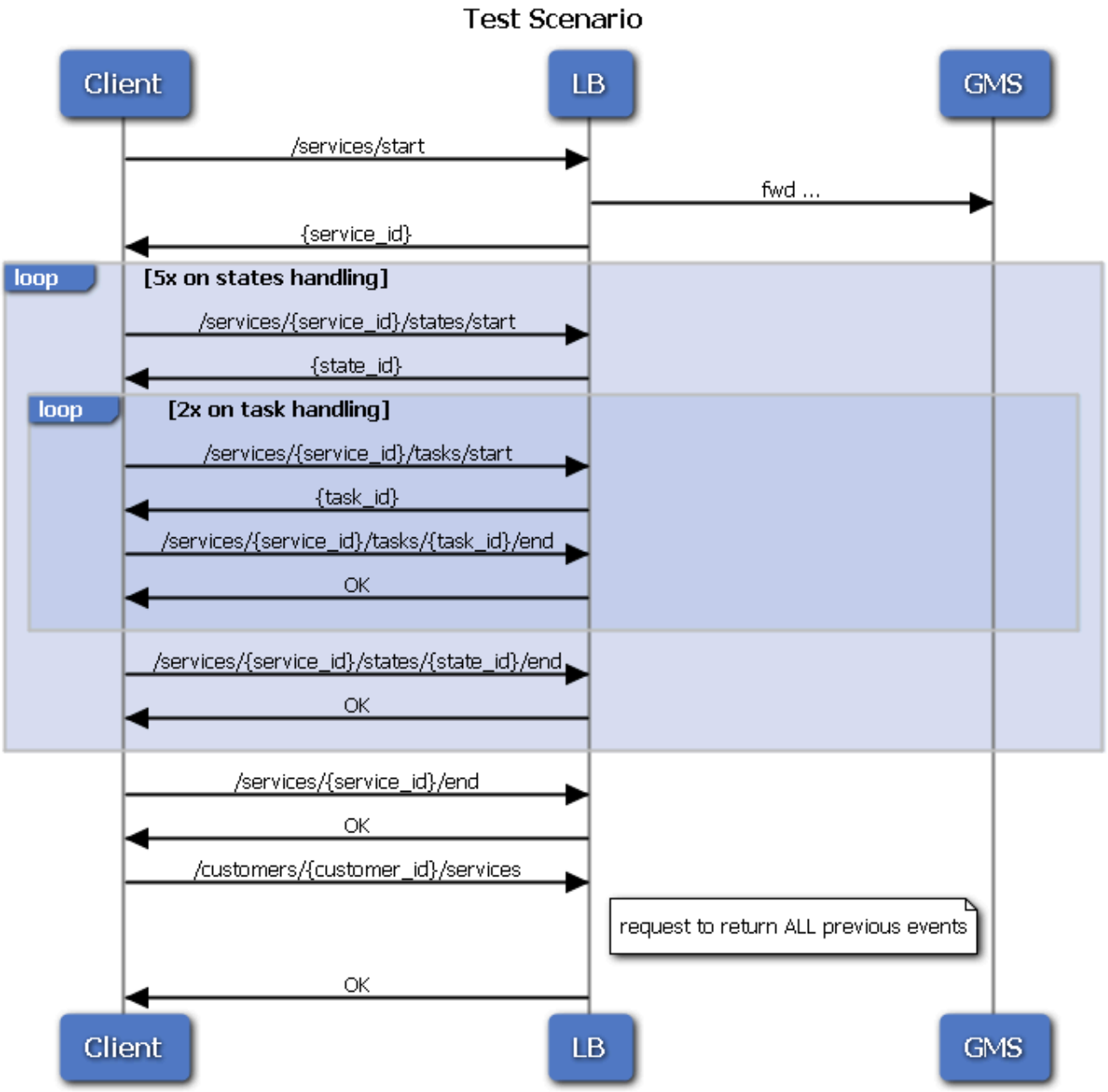
Calculations		
Description	**Value**	**Comments**
Number of events per conversation	32	
Size of conversation in storage (bytes)	7008	
Number of Activities	2560000	
Description	**Total**	**Comments**
Estimated disk size (MB)	80117,80	

Genesys Context Services - Sizing Output	
Data Size (MB)	80 118
Cluster Data Size (MB)	240 353
Data Size per node MB)	60 088
Disk usage per node (MB)	**128 369**
Grand Total (MB)	**513 475**

Performance and Sizing Data

This page provides an example of sizing and performance for Context Services in a GMS deployment.

Sizing Scenario

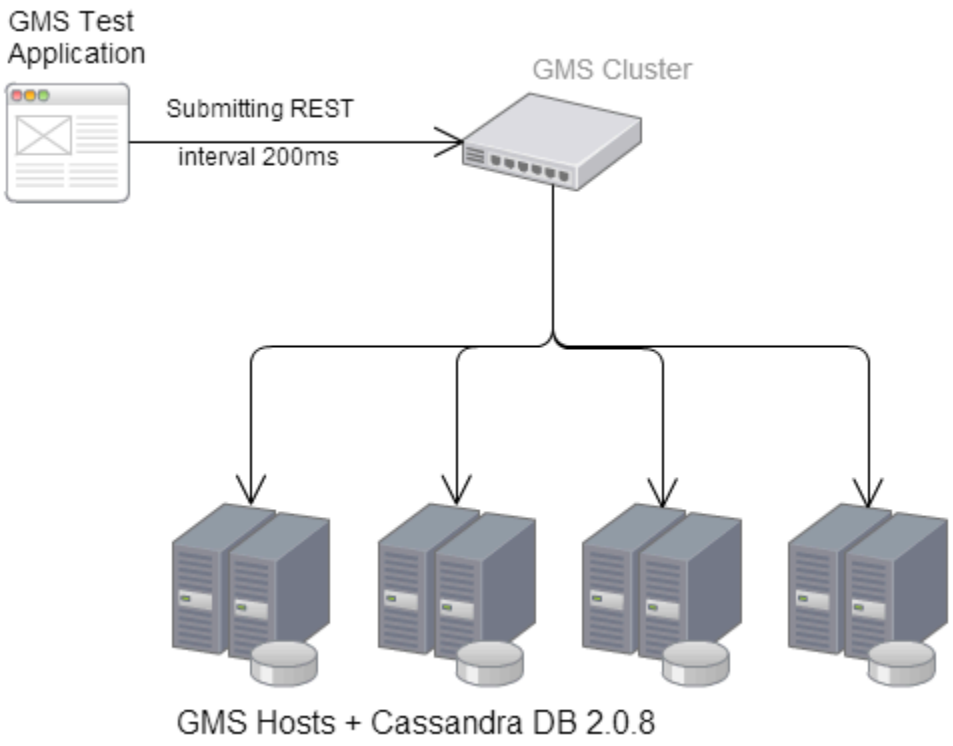


A Load Balancer (LB) receives the REST queries and dispatch them to GMS applications.

Our test scenario create a conversation implemented with one service including 5 states and tasks, which are equivalent to **33 REST queries** to handle on GMS side.

- 1 to 4 extension attributes are submitted in start events for services, states, and tasks.
- All the event attributes are set in Start and complete events using random values for integer a strings.

Sizing Settings



Number of hosts in use	4 (one per GMS cluster node)
Host Specifications	• Windows Server 2008 R2 Standard SP 1 • Intel Xeon X3440 @ 2.53 GHz, quad core • 8 GB RAM • Low Price disks
Components	• GMS 8.5.003.00+ • Cassandra 2.0.8 external
GMS Configuration	• Constant average CPU of 50% • Cassandra 2.0.8 (database of 2.5GB dat each node) • Replication Factor = 2

Sizing Results

Captured measure	Value	Comments
CPU	~20% to 60%	Each node was around an average of 30% with peaks to 60% from time to time.
Database size	68 GB / node (after 120h) = 272 GB	272 GB corresponds to 16.5 M conversations or 264 M service/ state/task events, for a ratio of 0.54KB per conversation (1 service, 5 states, 10 tasks).
I/O Disk	~3 MB /s (Cassandra measure)	Disk I/Os are fluctuating a lot (between 0 MB/s and 20 MB/s)
I/O Network	~3 MB /s	Between 20Mbps and 60Mbps.
Memory	2.4GB	For each GMS processes. Less than 2 GB for Cassandra processes.
Size of conversation (1 service, 5 states, 10 tasks)	~10kB in JSON	This matches a conversation (1 service, 5 states, 10 tasks), for 33 REST queries events, including extension data returned with queries by CustomerId.

Captured measure	Value	Comments
Throughput	~1000 queries/s (~30,5 scenario/s)	Constant through 24 hours of testing

Installing Context Services

This page details how you can install, then enable the Context Services APIs.

If your application needs the Customer Profile API due to backward compatibility issues, refer the migration page of this guide.

Installing the Genesys Mobile Services

Context Services uses the GMS platform to store data and implement additional features. In order to install the GMS/CS component, you must deploy GMS first. See the GMS Deployment Guide for details.

The Context Services installation CD contains a single installation package for both GMS and CS components. Follow GMS instructions prior to your installation. In particular, you must create a GMS application that you will use to enable and configure the Context Services.

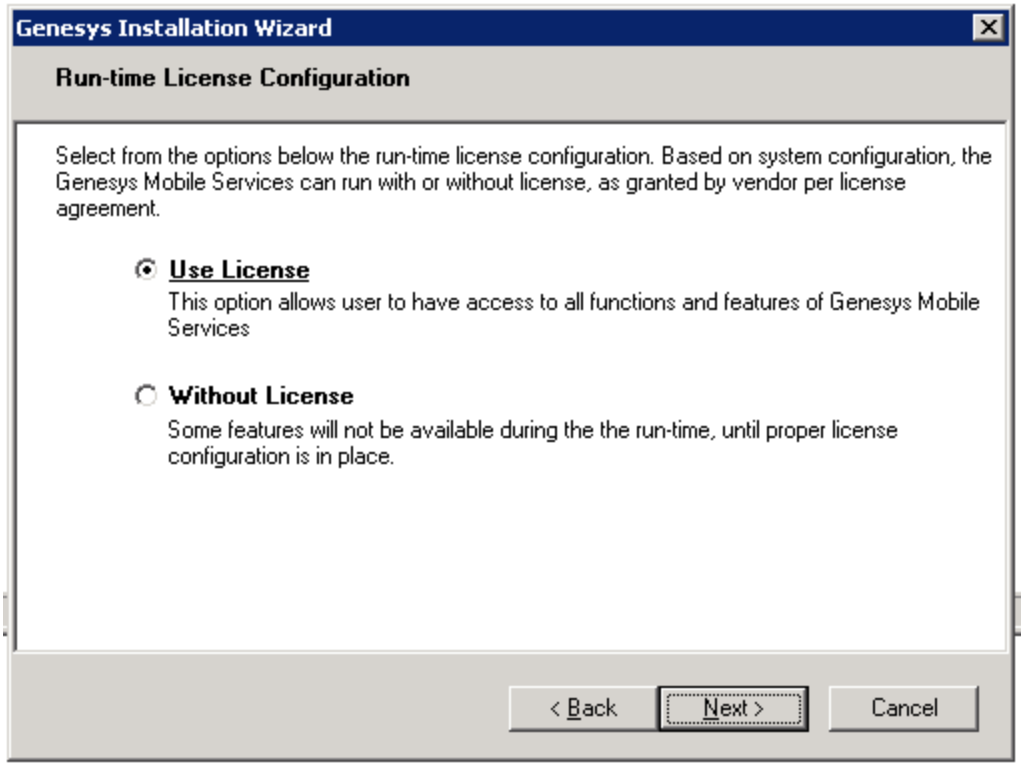
Run the installation package of your GMS installation CD:

1. Navigate to the installation directory `\windows\b1\ip`
2. Double-click `install.exe` which is located in this directory.
3. Follow instructions and enter the destination folder for the GMS installation.

> **Important**
>
> If you already installed a version of GMS older than 8.5.006.07, you must upgrade to GMS version 8.5.006.07 or higher.

Licensing

The GMS installation includes all Context Services materials, including licensing. The installation wizard displays the following information for licensing:



The licensing materials include Context Services. If you select the `Use License` option, your users will have access to Genesys Mobile Services Functions, including Context Services.

Enabling Context Services

Procedure:

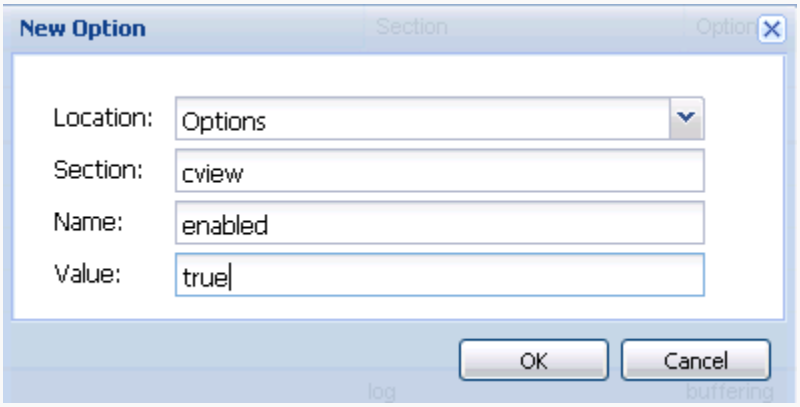
Purpose: To configure the `cview` variable which enables the Context Services.

Steps

1. Start Genesys Administrator the Configuration Manager and navigate to **PROVISIONING > Environment > Applications**.
2. Edit your GMS application.
3. Select the `Options` tab, and click **New** to create a new option.
   - Enter `cview` for the **Section**.

- Enter `enabled` for the **Name**.
- Enter `true` for the **Value**.

4. Click **OK**.



Accessing Context Services

Once Context Services is installed, enabled, and started, the services are available at the `<GMS_HOST_BASE_URL>/genesys/1/cs` base URL of your GMS host. Note that the `<GMS_HOST_BASE_URL>` must map the `server/external_url_base` option set in your application.

Checking that your Installation is Successful

The Genesys Mobile Services platform sees the Context Services as custom items and if your installation is successful, you can see Context Services in the home page of the Service Management User Interface.

- You can get detailed information about the Service Management User Interface here.
- You can also use the Context Services interface to manage your services.

Migrating from 8.1 to 8.5

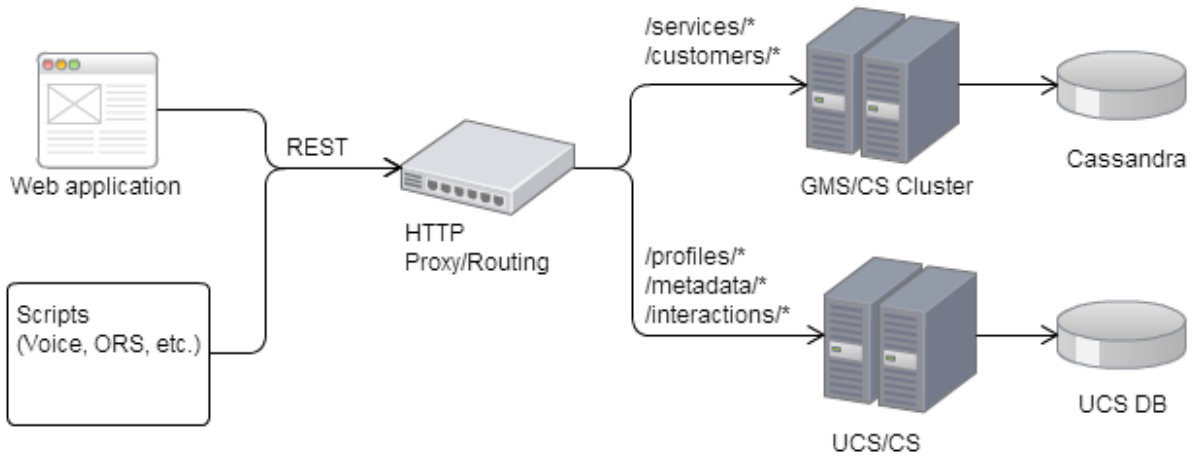
> **Warning**
>
> Starting from version 8.5.239, GMS no longer supports the migration process and the tool has been removed from the installation package.

Describes how you can migrate your application from former 8.1 versions to 8.5 versions.

Why Migrating?

In release 8.5, services data are no longer stored in Universal Contact Server (UCS) database; they moved to Genesys Moblie Services (GMS) Cassandra database.

1. If you are upgrading from 8.1 to 8.5, you must first migrate your Context Services data using the Context Services migration tool.
2. If your application needs profiles, you can keep using data stored in UCS. You do not need to modify the 8.1.3 Context Services queries for profiles and interactions. In this scenario, you must also set up your proxy to correctly handle URLs, as shown in the architecture diagram below.



As detailed in the *Context Services Developer's Guide*, profiles, interactions, and additional metadata resources are no longer available in 8.5. If your application requirements include these resources, you can still use the UCS APIs to manage customer data. You must update the UCS configuration to point

to the same base URL than GMS. You should therefore edit your proxy configuration; see below for detailed instructions.

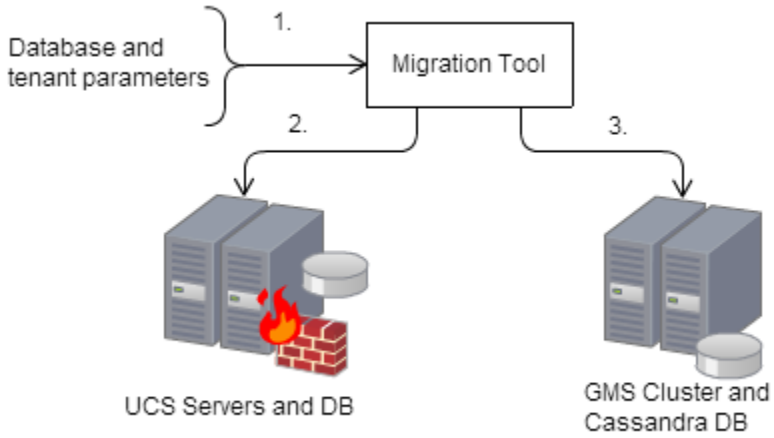
> **Important**
> If your application uses both GMS and UCS queries, you should make sure that it does not use deprecated methods.

Database Migration Process

> **Important**
> Before your start the migration, you must install Context Services.

The migration tool is a command line tool installed with the Context Services. This tool exports the services stored in UCS and then imports them in the Cassandra Database of the GMS Cluster:

- All service data, including state, tasks, and extensions are migrated.
- All the start/complete events are re-created.



Migration process:

1. Launching the tool with all required parameters.
2. Extracting Context Service data from UCS DB.
3. Importing Context Service data in GMS DB.

Database Migration Results

The migration tool creates the following files after the migration:

- `<Migration Tool Directory>/failure.log`;
- `<Migration Tool Directory>/success.log`.

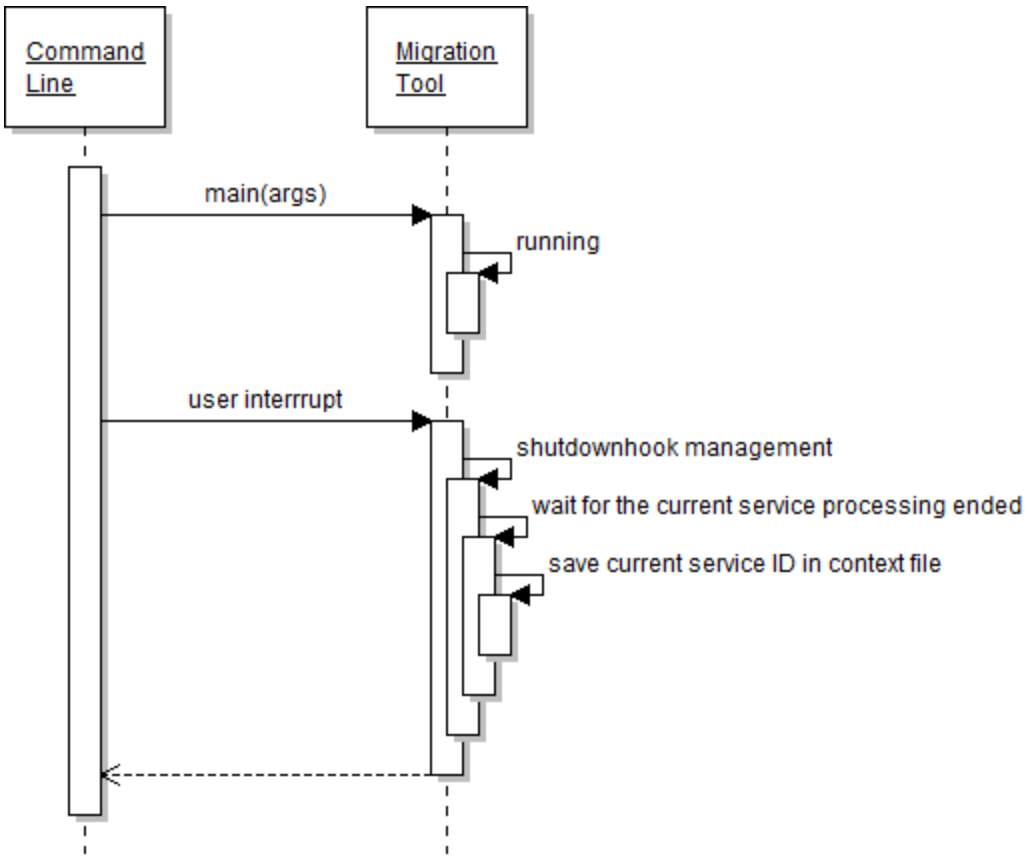
If no error occurs during the migration, the Context Services is then available in the Genesys Mobiles Services cluster.

> **Important**
>
> The Context Services data is not deleted from the UCS database.

Interruption during the Database Migration

You can abort the migration during the import stage by entering `CTRL-C` command at the console.. The migration process may need a few seconds to stop. Then, you will be able to restart the migration tool by specifying the last imported service ID with the `-continue-from` option.

Here is an interruption sequence diagram.



Migrating the Context Services from UCS to GMS Database

You must complete the following steps to perform the service data migration:

1. Checking the Business Attributes Mapping Options
2. Enabling the custom IDs
3. Running the Migration Tool

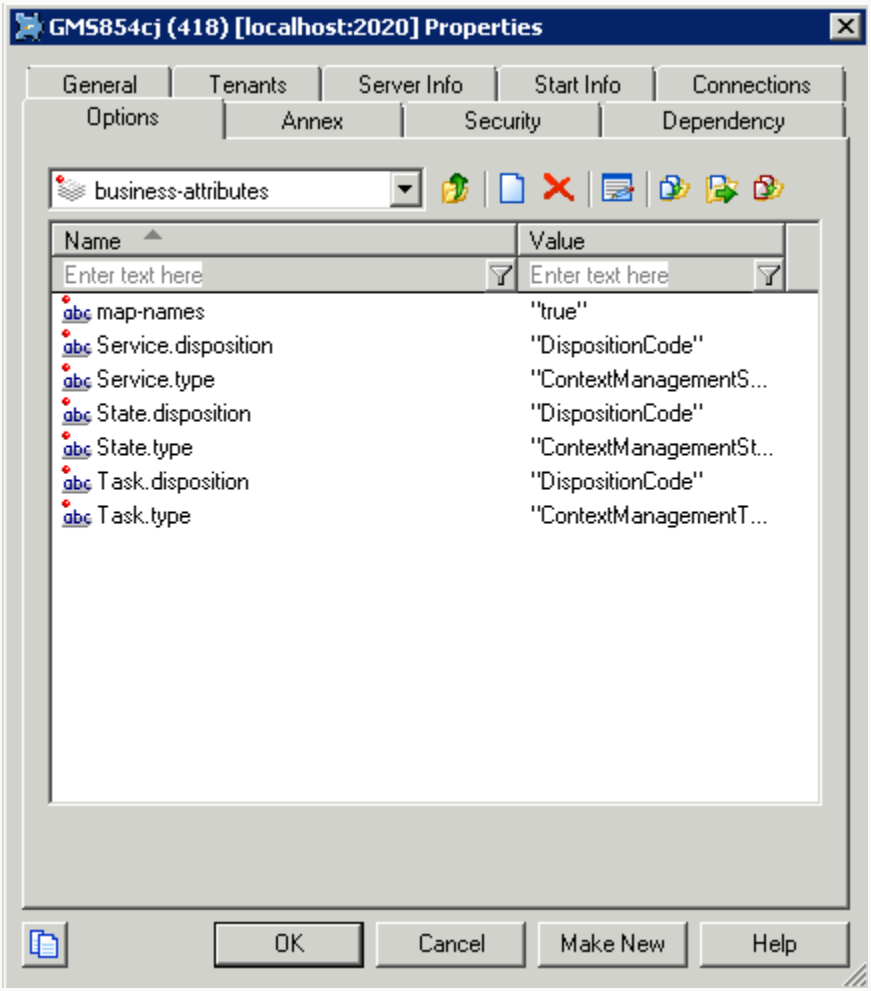
Checking the Business Attributes Mapping Options

Procedure: Checking the Business Attributes Mapping Options

Purpose: To make sure that your UCS and GMS applications have the same business attribute mapping. The mapping may be disabled during the migration process.

Steps

1. Open the Configuration Manager and edit both your GMS and UCS applications.
2. Make sure that the options defined in the `business-attributes` sections are identical in both applications.



Enabling the Custom IDs

Procedure: Enabling the Custom IDs

Purpose: To configure the `allow-custom-ids` option which allows the migration tool to replicate the UCS service IDs in the GMS Cassandra database. This option allows to keep identical IDs in the new storage location. Note that further services will be created with distinct UUID-type IDs.

Steps

1. Open the Configuration Manager and edit your GMS application.
2. Select the `Options` tab, and select the `cview` section.
3. Click **Add** to create Add the option `allow-custom-ids` and set its value to `true`.
4. Click **OK** to apply changes.

Running the Migration Tool

Procedure: Running the Migration Tool

Purpose: To migrate services from UCS database to Cassandra.

Before you run the tool, make sure that:

- You installed the Context Services.
- The Migration Tool is available in the `<GMS installation directory>/tools/cs_migration_tool` folder.
- You already set the `cview/allow-custom-ids` option to `true` and your `business-attributes` options are set correctly for both UCS and GMS.

The command line tool includes two migration modes:

- The `DB` mode, which migrates all the services from the UCS database to the GMS database;
- The `FILE` mode, which migrates a restricted list of service IDs from the UCS database to the GMS database.

Steps

1. Open a console.
2. Enter the migration command line:

```
$ startClient.bat [DB_OPTIONS] -tenantid <tenantID> -ucsurl <UCS_URL> -gmsurl <GMS_URL>
[ADDITIONAL_OPTIONS]
or
$ startClient.bat  -file <PATH_TO_FILE> -tenantid <tenantID> -ucsurl <UCS_URL> -gmsurl
<GMS_URL>  [ADDITIONAL_OPTIONS]
```

See the tables below for information about the parameters.

The parameters are described in the following table:

Command Line Parameters

Parameters	Scope	Mandatory	Description
-dbtype	DB only	Y	Sets the type of Database used for UCS ('oracle' or 'mssql') `-dbtype mssql`
-dbhost	DB only	Yes	Sets the host for the UCS Database `-dbhost demo_srv`
-dbport	DB only	Yes	Sets the port of the UCS Database. `-dbport 1433`
-dbname	DB only	Y	Sets the name of the UCS Database to migrate. In this case, all the services are migrated to the GMS database. `-dbname UCS`
-dbuser	DB only	Yes	Sets the user name of the UCS Database. `-dbuser sa`
-dbpassword	DB import only	Yes	Sets the password of the UCS Database.

Parameters	Scope	Mandatory	Description
			`-dbpassword mypass`

-file	FILE only	Yes	Sets the migration file which contains the list of ServiceIds to migrate.
			This text file (`.txt`) must contain one `service_id` per line; for example, you can create a file named `listOfIds.txt` containing the following list of IDs:
			`10001`
			`10002`
			`10003`

-ucsurl	ALL modes	Yes	Sets the UCS URL. `-ucsurl http://<host>:<port>/genesys/1/c`
-gmsurl	ALL modes	Yes	Sets the GMS URL. `-gmsurl http://<host>:<port>/genesys/1/cs`
-tenantid	ALL modes	Yes	Sets the GMS tenant DBID. `-tenantid 102`
-continue_from	ALL modes	No	In case of restart, specifies from which `service_id` to continue the migration. `-continue_from 10003`

In addition, the migration tool supports a set of additional options which help you to fine-tune your migration. Each option matches the following syntax:

```
-D<option>="<value>"
```

where <value> can either be a number or a string.

Special Options

Option	Scope	Mandatory	Description
EXTRACTOR_SELECT_QUERY	DB import only	No	Sets a specific selection query to migrate data from UCS Database.

Option	Scope	Mandatory	Description

			The default value is:
			
			`SELECT ServiceId, StartTime FROM ServiceStarted UNION SELECT ServiceId, StartTime FROM ServiceStartedAnonymous ORDER BY StartTime`
			
			The query must return the service ids in the first column; for example:
			
			`// selecting a range of services -DEXTRACTOR_SELECT_QUERY="SELECT ServiceId FROM ServiceStarted WHERE ServiceId >= 822184 AND ServiceId < 922184 ORDER BY StartTime ASC"`
			
			or
			
			`// selecting all the associated services which are not completed -DEXTRACTOR_SELECT_QUERY="SELECT ServiceId FROM ServiceStarted WHERE (ServiceId NOT IN (SELECT ServiceId FROM ServiceCompleted)) ORDER BY StartTime ASC"`

THREAD_POOL_SIZE	Any	No	Sets the number of services to process in parallel. Default is 30. The default value should be fine in most cases. If you modify this value, you change the number of requests that will be in process in case of user termination on demand (Ctrl-C). `-DTHREAD_POOL_SIZE=50`

Additional Configuration for UCS Backward Compatibility

Now, you must also configure the new Context Services application to ensure compatibility with your UCS installation. See Configuring CS for UCS Compatibility.

Configuration

This chapter covers configurations for your Context Services installation.

Page	Summary
Configuring Business Attributes	How to map Context Service keys to Business Attribute key-value pairs.
Assigning Roles	How to configure and assign roles to your Context Services application.
Configuring Pulse	How to configure Pulse for your Context Services application.
Configuring Tenancy	How to configure Pulse for your Context Services application.
Purging Services	How to configure purge tasks your Context Services application.
Set Conversation Expiration	How to configure expiration dates by type of conversation instead of instancing purge scenarios.
Options reference	All the GMS options available to configure Context Services in GMS.

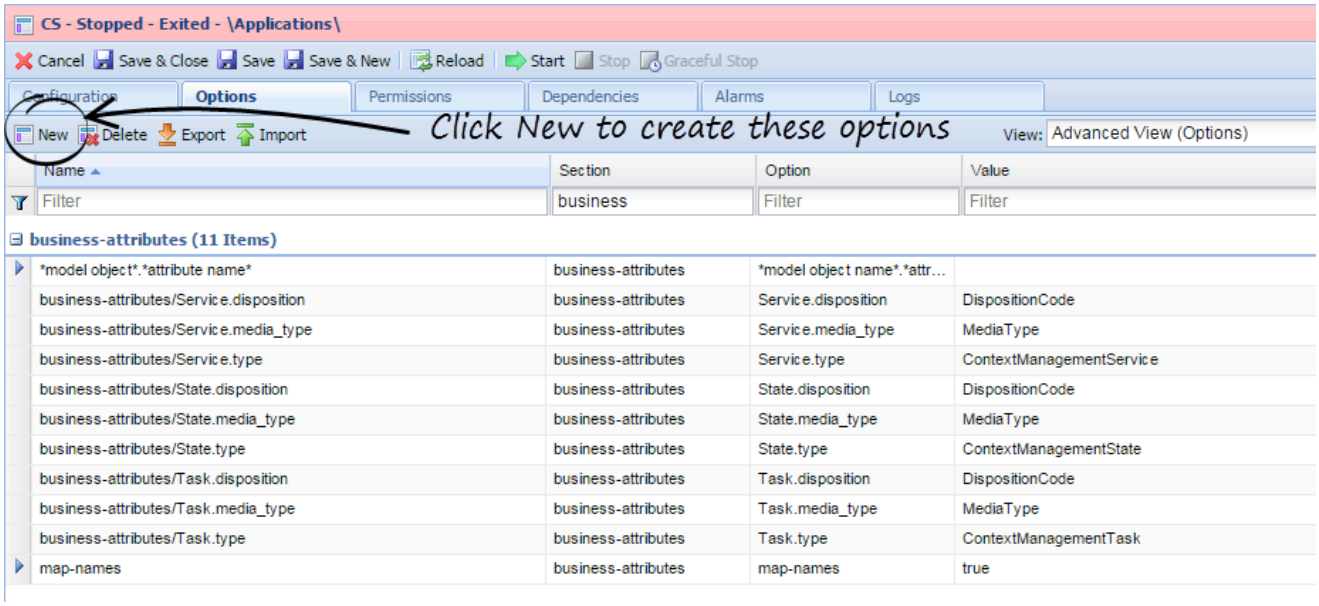
Configure Business Attributes

To enable Customer Journey, you must map Context Service keys to Business Attribute key-value pairs as described in this page.

The Business Attribute values are defined in the Tenant. Check the options reference for additional details.

> **Important**
>
> If you change business attribute values in your configuration, users will need to refresh the Context Services Interface to see the changes taken into account.

Map Context Services with Business Attributes



1. Edit your Context Services application in Genesys Administrator or Configuration Manager.
2. In the **Options** tab, select the **business-attributes** section or the **business-attributes.<TenantID>** section if you are in a multi-tenant configuration.
   [+] Tell me why.
   You must create a `business-attributes` section per tenant if you are in a multi-tenant environment. Click here for configuration details.
3. Create the following options as follows.
   - Enter `${resource name}.${field name}` for the Name such as, for instance:

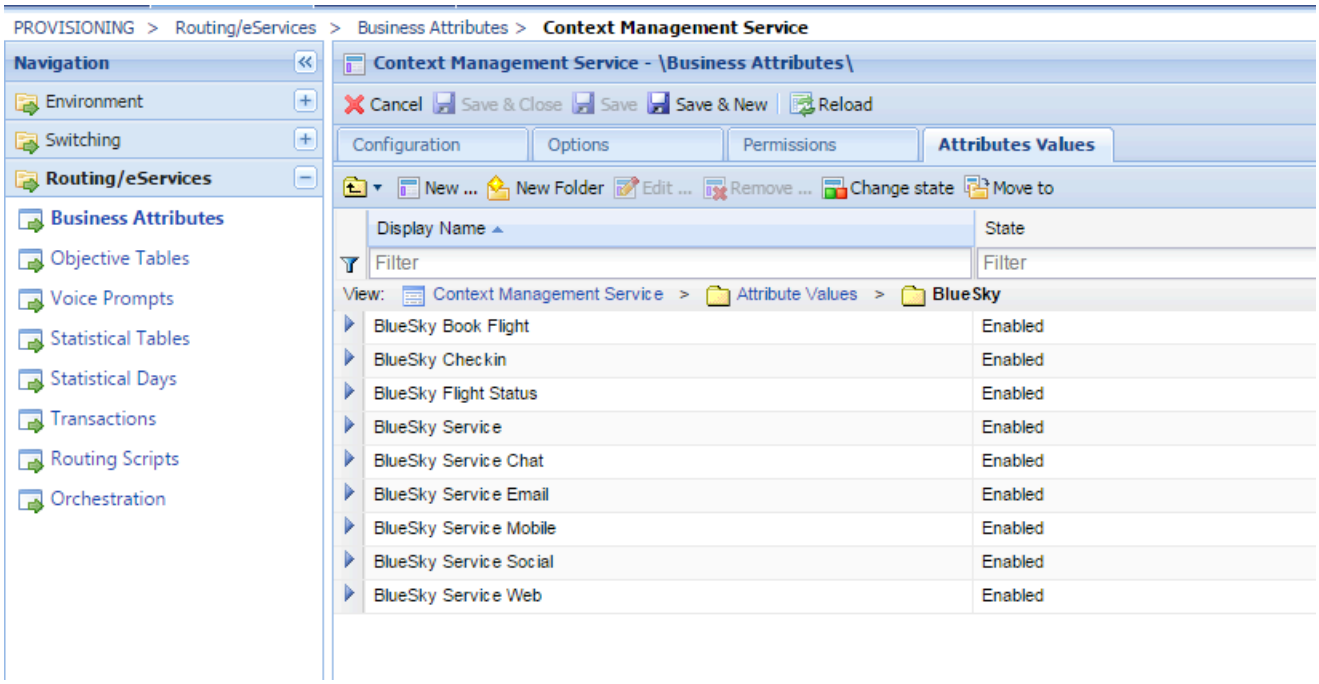
`Service.service_type`, `Task.disposition`, `State.media_type`.
Possible `${resource name}` values are:

- `Service`
- `State`
- `Task`

- Possible `${field name}` values to map are:
  - `type` (for service type)
  - `disposition`
  - `application_type`
  - `resource_type`
  - `media_type`
- For the **Value**, enter the name of the Business Attribute configured in the proper tenant. A Business Attribute can be mapped to several resource fields. For instance, the `Service.media_type` and `Task.media_type` string can both point to `MediaType` Business Attributes.

4. Click **Add** to create a `map-names` option.
   - Enter `map-names` for the **Name**.
   - For the **Value**, enter:
     - `true` to return the Names of Business Attribute Values instead of DB IDs in the responses for GET operations.
     - `false` (default) to return the DB IDs of Business Attribute Values in the responses for GET operations.

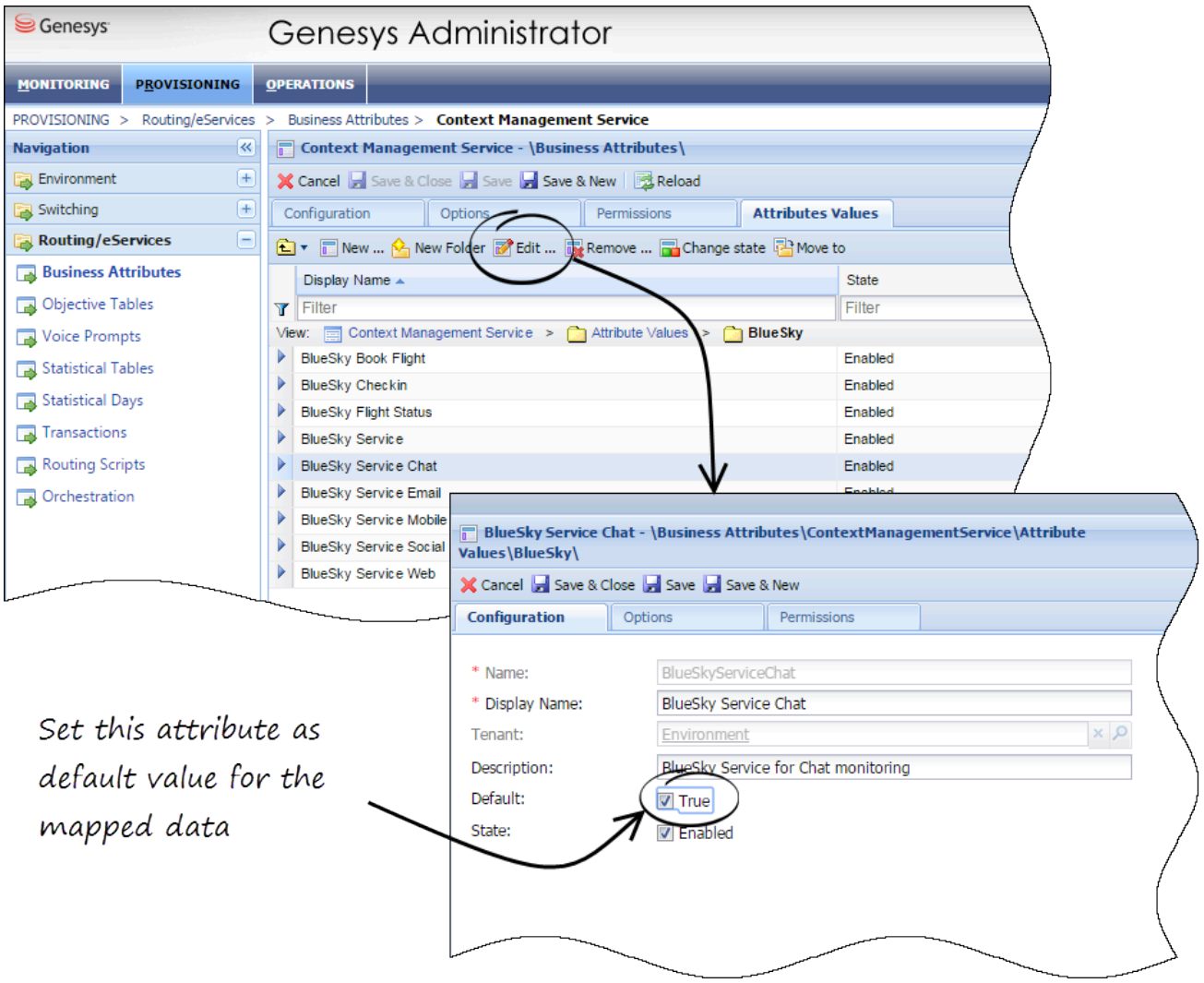
Example of Mapped Values



In the example above, the list of service types is defined as the value of the `ContextManagementService` business attributes.

You can use the mapped business attributes, key and associated values, to retrieve resources through GET operations by setting to `true` the `map-names` options, as stated above.

Set Default Values for your Business Attributes



If you set a default business attribute value for one of your mapped business attributes, this value will be used by default for the mapped data in your Context Services queries.

Let's consider that you mapped the `ContextManagementService` attribute with the `service_type` data, and you set the `BlueSkyServiceChat` value as default. Then, if you perform a Service start:

```
POST /services/start
```

You do not have to specify the `service_type` property; `service_type` is automatically set by default to `BlueSkyServiceChat`.

> **Tip**
> Starting 8.5.105, the Context Services Interface uses display names in panels and lists. Make sure to set meaningful display names when defining Business Attributes and their attributes.

Assigning Roles

Steps to assign Context Services roles to users.

Role-Based Access Control

If you need role-based access control for your Context Services queries, you must define a user which owns the Tasks privileges related to Context Services:

Name	Description
Administrator	Specifies write access for all CS APIs.
Administrator or Supervisor	Specifies read access for all Context Services APIs.

You can get more information about roles and queries here.

Enabling the Role Options for Context Services

Purpose: Enable the Role-Based access for your Context Services API.

Prequisites: You already enabled the Context Services.

Start

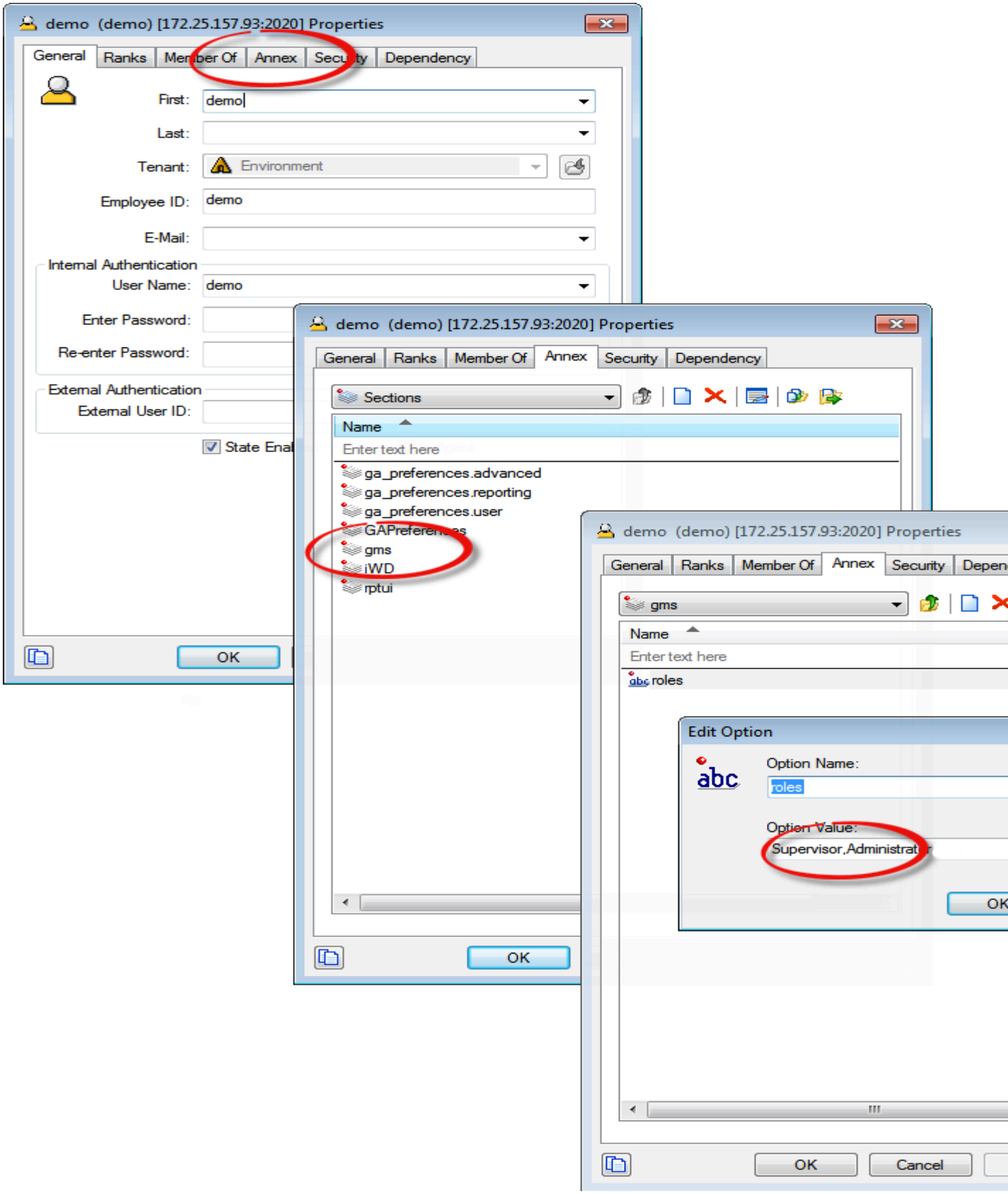
1. Start Genesys Administrator (or the Configuration Manager) and navigate to **PROVISIONING > Environment > Applications**.
2. Edit your Context Services application.
3. Select the `Options` tab, and click `New` to create a new option.
   - Enter `cview` for the `Section`.
   - Enter `use-role` for the `Name`.
   - Enter `true` for the `Value`.
4. Click OK.

Assigning Roles to the Context Services User

Prequisites: You already enabled Context Services.

Start

1. Start Genesys Administrator (or the Configuration Manager) and edit your user's properties.#
2. Select the `Annex` tab, and create or edit the `gms` section.
3. Create a `roles` option and enter the list of roles separated with commas.
4. Click OK.


demo (demo) [172.25.157.93:2020] Properties
General
Ranks
Member Of
Annex
Security
Dependency
First:
demo
Last:
Tenant:
Environment
Employee ID:
demo
E-Mail:
Internal Authentication
User Name:
demo
Enter Password:
Re-enter Password:
External Authentication
External User ID:
OK
Sections
Name
Enter text here
ga_preferences.advanced
ga_preferences.reporting
ga_preferences.user
gms
iWD
rptui
gms
roles
Edit Option
Option Name:
roles
Option Value:
Supervisor,Administrat
Cancel

Adding roles to a Context Services user.

Configuring Pulse

Steps to enable the Pulse features available for Context Services in the GAX interface. If you configure Pulse options, you will get widgets including Context Services statistics in your Pulse Dashboard.

Pulse is a widget-driven, graphical user application, which is accessible from a web browser as a Genesys Administrator Extension (GAX) plug-in application. Using a direct communication link to a real-time metrics engine, Stat Server, Pulse enables at-a-glance views of real-time contact center statistics within the GAX user interface.

Context Services installation already includes default templates for Pulse. If you configure Pulse options, as detailed below, Context Services pushes events to the Stat Server for Pulse which are then available for users in the Pulse dashboards and widgets.

> **Important**
> See the Pulse portal page for detailed documentation on this tool.

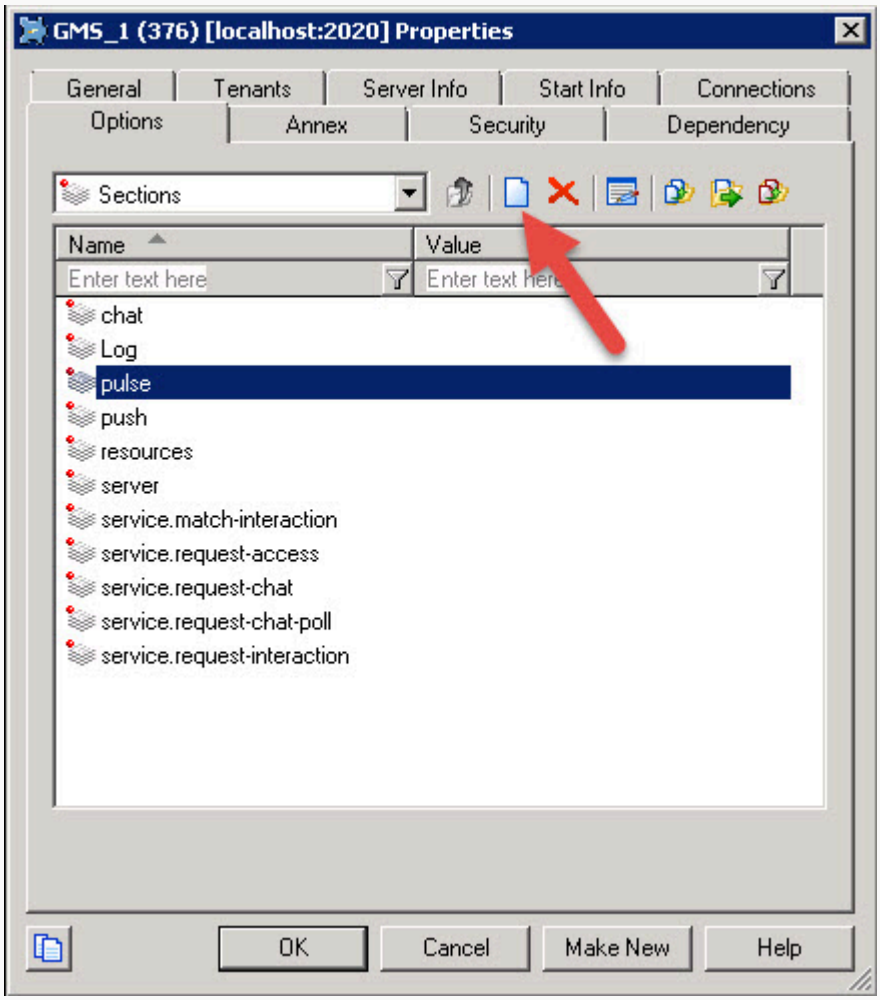
Procedure: Configuring Pulse Options

Purpose: To create and configure the `pulse` section

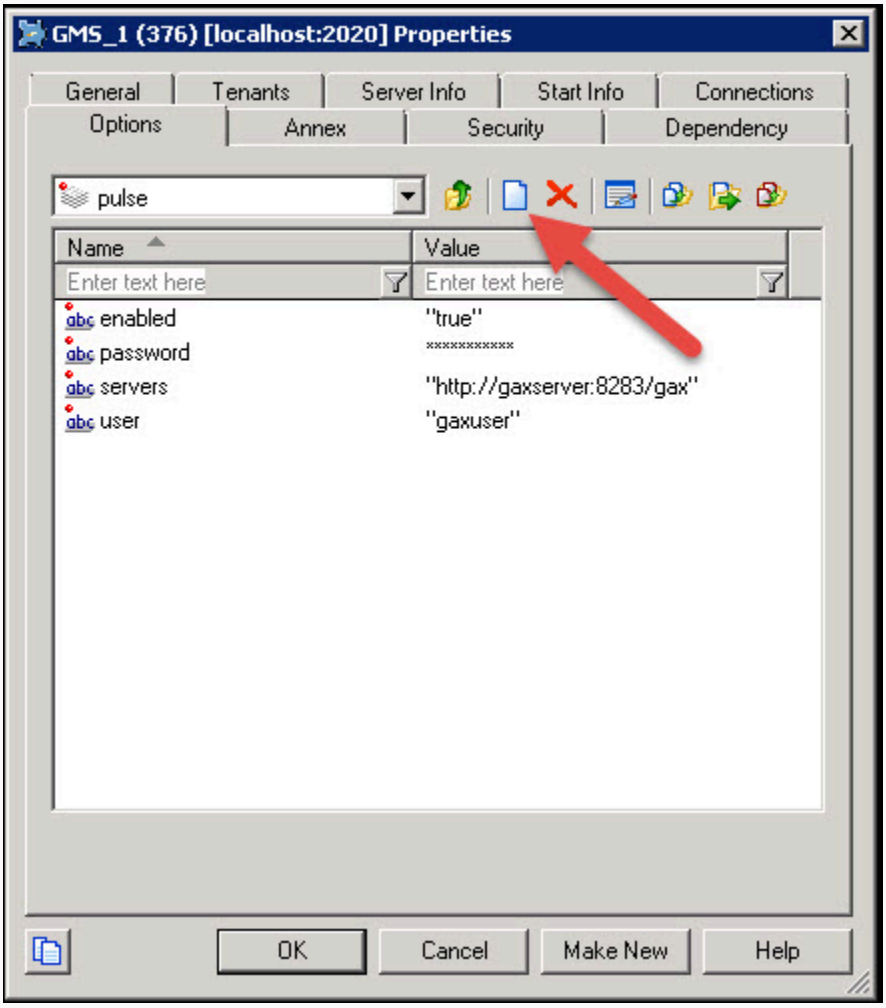
- You already activated Context Services.
- You already deployed Pulse.

Steps

1. Start the Configuration Manager and navigate to **Applications**.
2. Edit your Context Services application.
3. Select the `Options` tab, and click the `New` button to create a new section.
   - Enter `pulse` for the Section.



4. Select the section and click the New button to create the following options:
   - Enter `enabled` for the `Name`, then `true` for the `Value`. Click **OK**.
   - Enter `user` for the `Name`, then a username who has Pulse authorizations for the `Value`. Click **OK**.
   - Enter `password` for the `Name`, then the username's password for the `Value`. Click **OK**.
   - Enter `servers` for the `Name`, then a list of one or more URLs separated by semicolons which point to Pulse applications for the `Value`; for instance: "`http://gax1dev:8283/gax;http://gax2dev:8283/gax`". Click **OK**.
   - Enter `version` for the `Name`, then "2" for the `Value`. Click **OK**.


GMS_1 (376) [localhost:2020] Properties
General
Tenants
Server Info
Start Info
Connections
Options
Annex
Security
Dependency
pulse
Name
Value
Enter text here
Enter text here
enabled
"true"
password
****
servers
"http://gaxserver:8283/gax"
user
"gaxuser"
OK
Cancel
Make New
Help

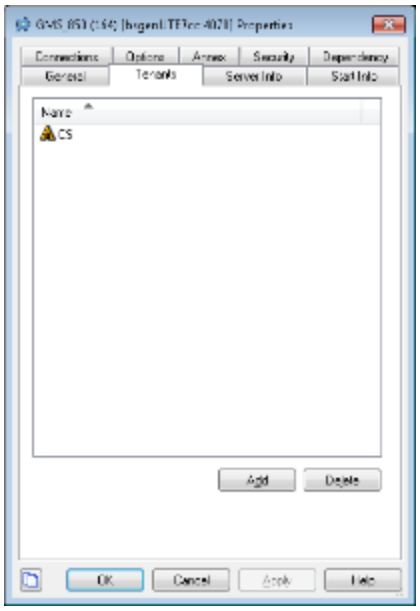
Configuring Tenancy

Steps to configure tenancy (single-tenancy and multi-tenancy).

Configuring Single-Tenancy

To configure a single tenant for your Context Services application:

1. Edit your Context Services application with Configuration Manager or with Genesys Administrator.
2. In the Tenants tab, click Add to select your tenant.



Important

If you configure a single tenant:

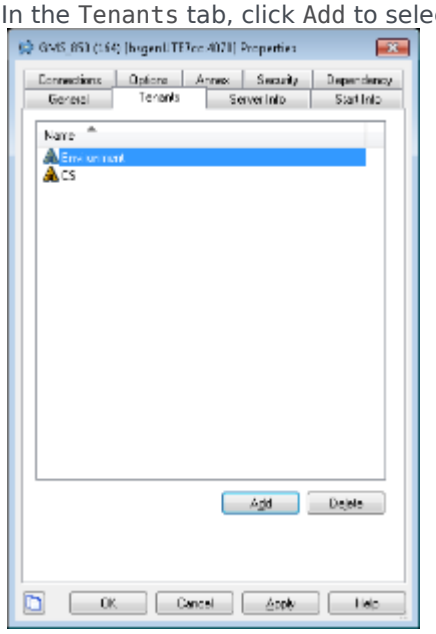
- The configuration for the `business-attributes` section is compatible with the former 8.1 UCS configuration.
- Do not modify your application to use the additional HTTP headers available for multi-tenancy. By default, Context Services uses the tenant ID defined in your Tenants tab.

Configuring Multi-Tenancy

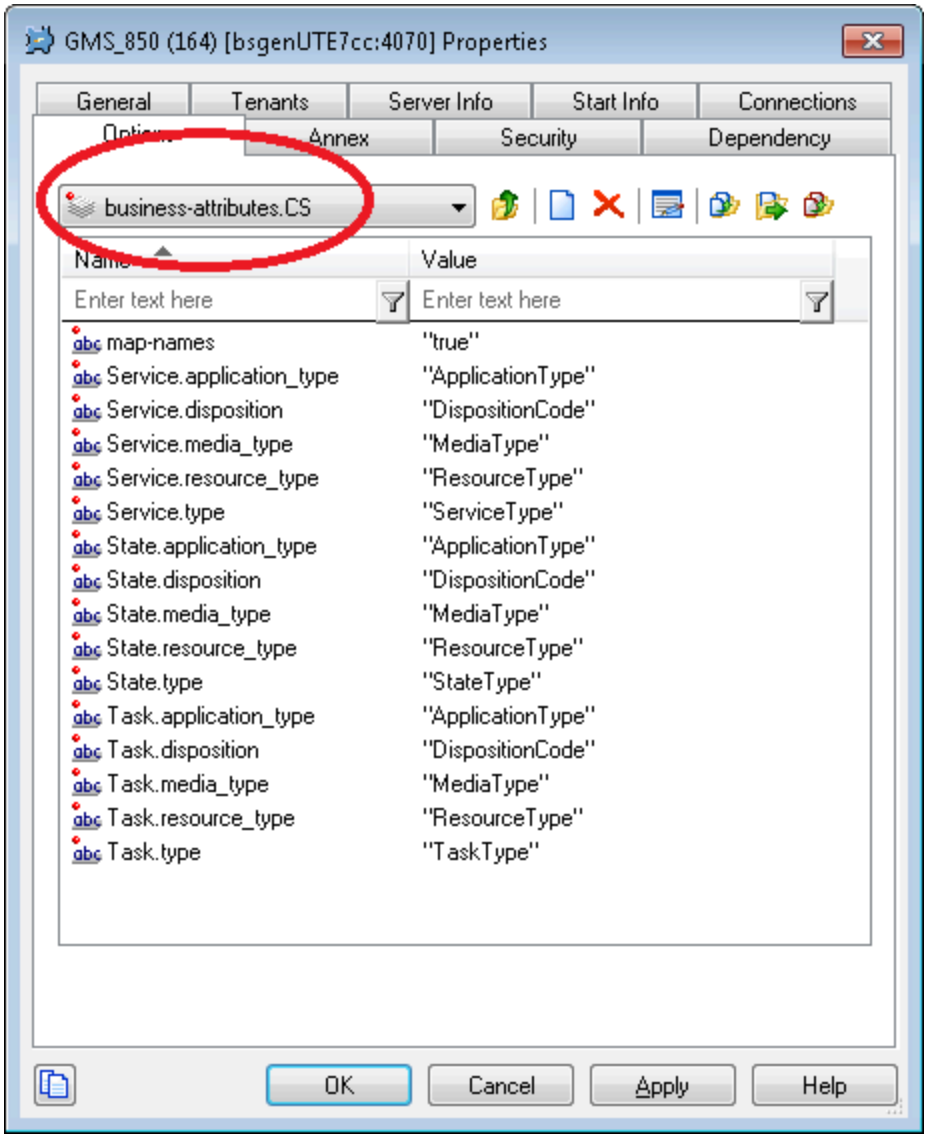
To configure multiple tenants for your Context Services application:

> **Important**
> This feature is available for Context Services applications only.

1. Edit your Context Services application with Configuration Manager or with Genesys Administrator.
2. In the Tenants tab, click Add to select your tenants.



3. For each tenant, define the associated `business-attributes.<tenantID>` section. For example, the `business-attributes.CS` section below defines the configuration for the CS tenant.



Adding ContactCenterId and GroupId HTTP headers

In order to support multi-tenancy and business units, your Context Services queries need to include additional HTTP headers:

- `ContactCenterId` contains the Tenant DBID or the Tenant Name that you set in your configuration..
- `GroupId` contains an optional business unit name.

> **Important**
> You must make sure that you provide these IDs in HTTP headers, not as part of the

> JSON Body content.

If your request does not include these headers, the Context Services handles the request in single-tenant mode and uses the provisioned tenant ID with no business unit name of your Context Service application.

Read also Configuring Tenancy.

ContactCenterId Errors

If you provide an incorrect `ContactCenterId` string, your application receives the following error message:

```
{
    "message": "Invalid tenant specified : specified tenant in HTTP header
'ContactCenterId'='wrong value' not configured. Verify configuration.",
    "exception":
"com.genesyslab.gsg.services.contextservices.exceptions.ContextServicesExceptionResource"
}
```

If you do **not** provide the `ContactCenterId` string in a multi-tenant configuration, the error message is:

```
{
    "message": "Invalid tenant specified : no tenant specified in multi-tenant environment.
Verify configuration and 'ContactCenterId' HTTP header.",
    "exception":
"com.genesyslab.gsg.services.contextservices.exceptions.ContextServicesExceptionResource"
}
```

Purge Services

Steps to schedule purges of the service database. You can configure purges either from the Genesys Administrator or from the Configuration Manager interfaces. Your application can also perform purges by itself with a purge query.

Usage of Purging Capabilities

> **Important**
>
> The purging capabilities of Context Services are not available for Profile management.

The purge features delete all the service information, including nested task, state records, and extensions. Options enable you to select the type of services to delete (started, completed, or both), and to set up a time limit (or expiration date) which is compared to the `Started time` or `Completed time` field of the services.

You can schedule tasks to periodically purge the database in the Configuration Server and Genesys Administrator interfaces. All the scheduled tasks are stored in the Configuration Server. Administrators should use these interfaces to:

- Schedule purging jobs for all or for anonymous services.
- Schedule the purge of started services which are not completed at the purge date time.
- Schedule the purge of started services which are completed at the purge date time.

You should be aware that:

- You cannot select the services to delete according to their nested states and tasks.
- Context Services does not check data integrity during the purge job.
- Context Services does not check that the sub-states or sub-tasks are all completed.
- There is no interface to display or retrieve information about the purge progress.
- You cannot interrupt a purge task.
- You can set up multiple purging tasks.

> **Important**
>
> Starting 8.5.105, you can update your services with a TTL or an expiration date by using the Update

Expiration Time query. You can also modify Start Events and End Events to add one of these parameters.

Purging Criterias

If you schedule a purge, you must choose one of the criteria listed below:

- `purge.service.all` to purge all the services which received a started event prior to the limit date.
- `purge.service.started.anonymous` to purge the anonymous services which received a started event prior to the limit date and are not completed at the date of the purge.
- `purge.service.started` to purge the services which received a started event prior to the limit date and are not completed at the date of the purge.
- `purge.service.completed` to purge the services which received a started event prior to the limit date and are completed at the date of the purge.
- `purge.service.completed.anonymous` to purge the anonymous services which received a started event prior to the limit date and are completed at the date of the purge.

Important

You can use these criteria in the Genesys Administrator and Configuration Manager interfaces, and in the purge query.

Schedule Purge Jobs

You can add configuration options to schedule the purge of service records .

1. Open your Context Services application in the Genesys Administrator interface or the Configuration Manager interface.
2. In the `Annexes` tab, create one or more sections called `scheduled-job-XX`, where *XX* is any convenient identifier.
3. Create options and assign values to them, as described in the table below.

Important

If you have multiple `scheduled-job-XX sections`, be careful to have no overlap of the scheduled times (specified by the `cron-expression` option). Only one purge operation can be executed at a given time: if one operation is not finished when a

> second one should start, the second operation does not start at all.

Service Purging Options

You can create as many `scheduled-job-XX` sections as you need with the following options.

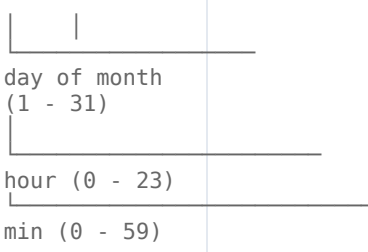
Purging Options

Option name	Mandatory	Default value	Valid values	Changes Take Effect	Description
enabled	No	`false`	`true`, `false`	Immediately	`true` to enable the scheduled job.
organization	No	N/A	`<ContactCenterId>[.<GroupId>]`	Immediately	Organization ID in the form `<ContactCenterId>[.<GroupI` where `ContactCenterId` refers to the tenant (DBID or Name) and `GroupId` is a sub-tenant. **Tip** You should set the same values as for the HTTP POST event operation headers used in the Context Services queries. .
action	Yes	`purge.service.all`	• `purge.service.all` • `purge.service.started` • `purge.service.completed.anonymous` • `purge.service.started` • `purge.service.comple`	Immediately	Specifies the type of purge to perform. The time limit is set through the `period` and `period-type` options. • `purge.service.all` to purge all the services which received a started event prior

Option name	Mandatory	Default value	Valid values	Changes Take Effect	Description
			ted.anonymous		to the configured period. • purge.service.started.anonymous to purge the anonymous services which received a started event prior to the configured period and are not completed at the date of the purge. • purge.service.started to purge the services which received a started event prior to the configured period and are not completed at the date of the purge. • purge.service.completed to purge the services which received a started event prior to the configured period are

Option name	Mandatory	Default value	Valid values	Changes Take Effect	Description
					completed at the date of the purge. • `purge.service.completed.anonymous` to purge the anonymous services which received a started event prior to the configured period and are completed at the date of the purge.
period	Yes	5	Any positive integer from 1 to 9999	Immediately	Sets the time frame for the purge depending on the `period-type` option. If you set `period` to 6 and `period-type` to days, the purge deletes all the services older than 6 days. See also the `period-type` option.
period-type	Yes	`months`	• `hours` • `days` • `months` • `years`	Immediately	Specifies the units to use for the `period` option calculation. • `hours` to purge services older than `n` hours, where `n` is the `period` option's

Option name	Mandatory	Default value	Valid values	Changes Take Effect	Description
					value. • days to purge services older than n days, where n is the period option's value. • **months (default)** to purge services older than n months, where n is the period option's value. • years to purge services older than n years, where n is the period option's value.
cron-expression	Yes	0 20 * * 5 (i.e. Fire at 8pm every Friday)	Cron expression as described at Cron Expression	Immediately	Explanation of the provided sample: 0 20 * * 5 ┬ ┬ ┬ ┬ ┬ │ │ │ │ │ └───── day of week (0 - 7) (Sunday=0 or 7) └────────── month (1 - 12)

Option name	Mandatory	Default value	Valid values	Changes Take Effect	Description
					

Setting Options in Cluster Mode

If you deploy your Context Services application in cluster mode (which is the default deployment), you must also set the purge options in your Cluster Server application. Because Context Services is a custom component from an architecture stand-point, Cluster deployment is detailed in the GMS Deployment Guide.

Set Conversation Expiration

Context Services includes granular expiration capabilities for conversations. You can configure expiration dates by type of conversation identified in the `service_type` attribute of a service instance, instead of instancing purge scenarios.

The conversation data remains valid from the date where you start or update the associated service, till its expiration date. At expiration, the conversation data (all its service, states, and tasks) is removed from the database.

> **Important**
>
> By default, expiration date is set to 5 years (5y). Starting 8.5.105, you can also update the expiration time with a POST API query.

More about the Expiration Option

You can configure the expiration date in the Context Service application or in the Business Attributes mapped with services (see below). The `expiration` attribute must match the following formatting:

```
expiration = <integer><letter>
```

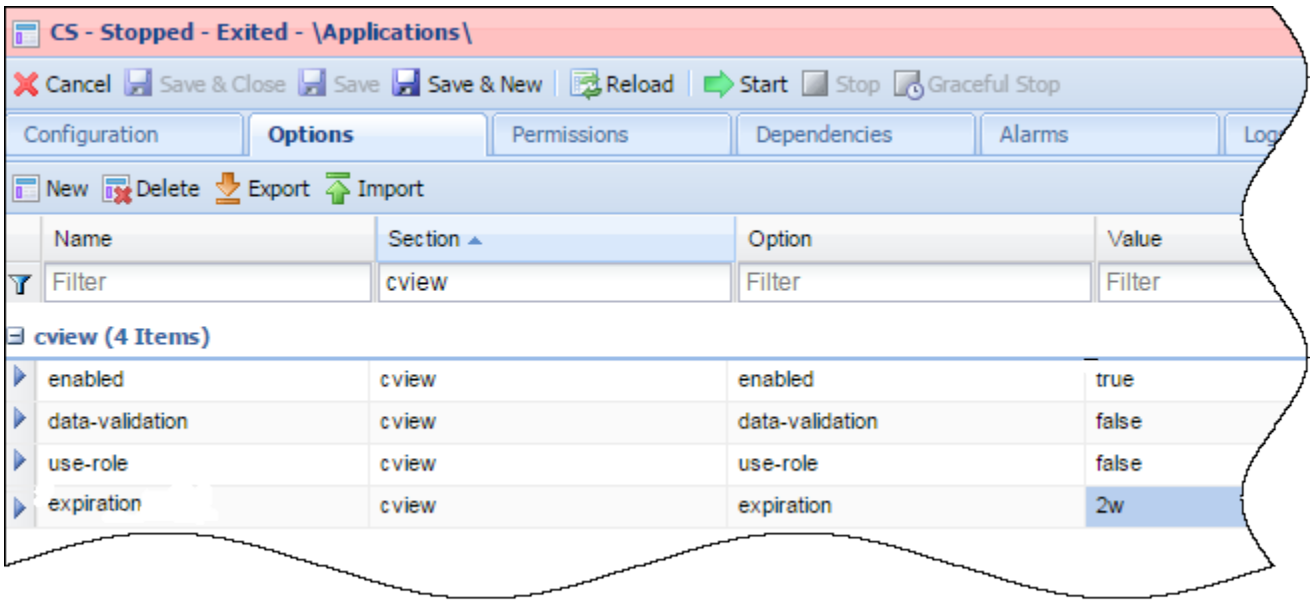
where:

- `<integer>` specifies the time period
- `<letter>` is one of the following period type:
  - `s`: seconds
  - `m`: minutes
  - `h`: hours
  - `d`: days
  - `w`: weeks (7 days)
  - `M`: months (30 days)
  - `y`: years (365 days)

For example, setting `expiration=2w` means that data inserted or updated using the Context Services API will last 2 weeks.

> **Important**
>
> If you set or change the expiration option, it applies only to further created conversations. The option is not retroactive and does not apply to conversations created or updated prior to the configuration change.

Configuring Default Expiration Time

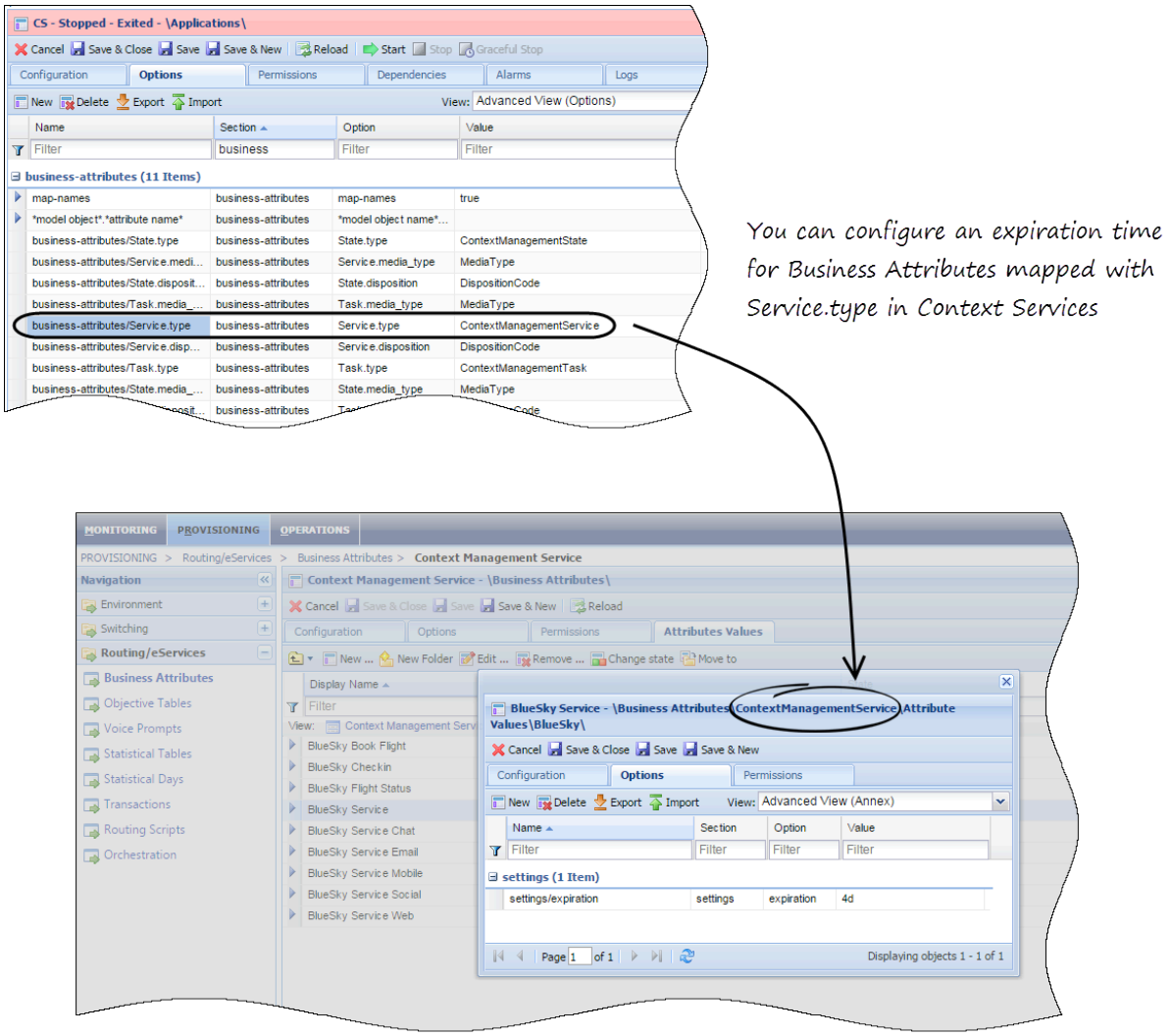


In Genesys Administrator, open your application and in the **Options** tab, edit the `cview/exp` attribute.

Set its property to the appropriate value.

Click **Save** to apply the change to new conversations.

Configuring Expiration for a Specific Conversation Type



To overwrite the default expiration for a given service type, you must first set up Business At mapping for `Service.type` resources. See there for details. Then, you can edit the Business mapped with your service type and add the `setting/expiration` option to its configuration.

In our example, Service types are mapped with `ContextManagementService` Business Attribu values:
`business-attributes/Service.type=ContextManagementService`

To set a specific expiration time of your Business Attribute value, edit its configuration and ac
`settings/expiration=<integer><letter>` to the **Options** tab.

> **Important**
>
> The expiration time set in Business Attribute values is taken into account after restarting Context Services or making configuration changes in Context Services.

Auto-Completing Services

Added in 8.5.111

Instead of manually completing a service, you can enable the auto-completion by setting the auto-complete-enabled option to true and by configuring an auto-complete-after time in your application configuration. In this scenario, the system will wait the given auto-complete-after time to complete the service after its last update and/or the last updated to its nested states and tasks.

For example, to auto-complete a service that is inactive more than 20 seconds, use the following configuration in your Context Services application:

```
auto-complete-enabled= true
auto-complete-after=20
```

Note that you can override this configuration in the REST query that starts the service.

> **Important**
>
> If you modify the `auto-complete-after` configuration parameter at runtime, only the new services will use its new value for auto-completion. Services that were created before the value changed will keep on using the former default value.

Configuration Options Reference

The configuration options described below only apply to the Context Services applications which do not include Customer Profile capabilities. If your application uses Customer Profile capabilities provided by UCS, refer to the 8.1.3 configuration options.

- For backward compatibility issues, refer to the migration page.
- For GMS configuration, refer to the GMS options reference page, where you will also find the configuration options described below that only apply to the Context Services applications.

[cview] Section

allow-custom-ids

Section: cview
Default Value: false
Valid Values: true, false
Changes Take Effect: Immediately

If `true`, enables custom IDs; this option is for migration purpose only and allows Context Services to replicate the UCS service identifier into Context Services storage. If false, Context Services generate new identifiers in response of "start" events.

auto-complete-after

Section: cview
Default Value: No default value
Valid Values: long
Changes Take Effect: Immediately
Introduced: 8.5.111.04

Time in seconds to wait to auto-complete the service after its last update if `auto-complete-enabled = true`. This option value is used if it is not explicitly provided in the Start request by setting the `auto_complete_after parameter` of the Service Start Event. If you do not provide a value for `auto_complete_after` in the configuration or in the Start request, then only an explicit API call to complete the service will be able to terminate the service.

auto-complete-enabled

Section: cview
Default Value: false
Valid Values: true, false
Changes Take Effect: Immediately
Introduced: 8.5.111.04
Related Feature: auto-complete

If `true`, enables the auto-complete feature. If a service or its states or its tasks are not updated within the period specified by the option `auto-complete-after`, the service (including states and tasks) is terminated.

auto-complete-organizations

Section: cview
Default Value: ""
Valid Values: String
Changes Take Effect: Immediately
Introduced: 8.5.111.04

Comma-separated list of organization IDs (the Contact Center ID concatenated with the Group ID separated by a period) for which the auto-completion is enabled.

data-validation

Section: cview
Default Value: false
Valid Values: true, false
Changes Take Effect: Immediately

If `true`, allows additional checks of consistency during production. For example, if your application creates a State or a Task, the server checks that the service exists before it creates the inner object; if not, your application receives a `Service Not Found Exception`.

> **Warning**
> This data validation feature is a costly process that requires additional storage read access.

enabled

Section: cview
Default Value: false
Valid Values: true, false
Changes Take Effect: Immediately

Enables (true) or disables (false) the Context Services feature. If set to FALSE, the other options are ignored.

expiration

Section: cview
Default Value: 5y
Valid Values: <integer><letter>
Changes Take Effect: Immediately

Defines expiration time for conversation (service, states, tasks) after creation or last API update. The expiration time is formatted as follows: `<integer><letter>` where:

- `<integer>` specifies the time period
- `<letter>` is one of the following period type:
  - `s`: seconds
  - `m`: minutes
  - `h`: hours
  - `d`: days
  - `w`: weeks (7 days)
  - `M`: months (30 days)
  - `y`: years (365 days)

use-role

Section: cview
Default Value: false
Valid Values: true, false
Changes Take Effect: Immediately

Enables (true) or disables (false) Role-Based Access Control.

[elasticsearch] Section

allowedServiceTypes

Section: elasticsearch
Default Value: '"*"
Valid Values: String
Changes Take Effect: Immediately

Comma-separated list of the service types allowed to publish conversations to the elastic server. The service types are String or DBIDs matching the Business Attributes mapping; for example: `Identification,Special Offers`.

enabled

Section: elasticsearch
Default Value: false
Valid Values: true, false
Changes Take Effect: Immediately

Set to `true` to push conversations to elastic search when services are completed.

server

Section: elasticsearch
Default Value: No default value
Valid Values: String
Changes Take Effect: Immediately

Elastic Search server URL; for instance: `http://gax1dev:1664`.

urlPattern

Section: elasticsearch
Default Value:
Valid Values:
Changes Take Effect:

Pattern to append to the server URL; for instance: `${service.type}-${date.year}.${date.month}.${date.day}/service/${service.id}`. The possible pattern variables are:

- `date.year`
- `date.month`
- `date.day`
- `service.type`
- `service.id`

[pulse] Section

enabled

Section: pulse
Default Value: false
Valid Values: true, false
Changes Take Effect: Immediately

If `true`, enables the Pulse feature in Context Services.

user

Section: pulse
Default Value: No default value
Valid Values: String
Changes Take Effect: Immediately

Name of a user who has pulse authorizations.

password

Section: pulse
Default Value: No default value
Valid Values: String
Changes Take Effect: Immediately

Password of the user who has pulse authorizations.

servers

Section: pulse
Default Value:
Valid Values: String
Changes Take Effect: Immediately

A list of one or more URLs separated by semicolons which point to Pulse applications for the Value; for instance: `http://gax1dev:8283/gax;http://gax2dev:8283/gax`.

version

Section: pulse
Default Value: 2
Valid Values: 1 or 2
Changes Take Effect: Immediately
Introduced: 8.5.103
Modified: 8.5.110.07

Enables version 2 of Pulse Dashboards starting in 8.5.103.

Starting in 8.5.110, 2 is the default value for this option.

[business-attributes] Section

This section defines the mapping between Context Services and the Business Attributes configured in the Genesys Configuration Server. The Business Attribute values are defined in the Tenant.

> **Important**
>
> If your application is multi-tenant, you should define a `business-attributes.<tenantId>` section per tenant.

resourcename.fieldname

Section: business-attributes
Default Value:
Valid Values:
Changes Take Effect: Immediately

Associates a Business Attribute key with the name of the Business Attribute configured in the proper tenant.

- Possible `resource name` values are:
  - `Service`
  - `State`
  - `Task`
- Possible `field name` values to map are:
  - `type` (for service type)
  - `disposition`
  - `application_type`
  - `resource_type`
  - `media_type`
- Such as, for instance: `Service.service_type`, `Task.disposition`, `State.media_type`.

> **Important**
>
> - If there is no configuration for a given field, Context Services automatically allows

any valid integer value for this field. In this case, your application is responsible for the value's validity.

- A Business Attribute can be mapped to several resource fields. For instance, the `Service.media_type` and `Task.media_type` string can both point to the "MediaType" Business Attributes.

map-names

Section: business-attributes
Default Value: false
Valid Values: String
Changes Take Effect: Immediately

Set to `true` to return the Names of Business Attribute Values instead of DB IDs in the responses for GET operations; `false` (default) to return the DB IDs of Business Attribute Values in the responses for GET operations.

Enabling Customer Profiles in UCS

> **Important**
>
> This step is optional. If you don't need to keep backward compatibility or Customer profiles in your application, you don't have to enable customer profiles in UCS. UCS is intended to be used for profile management only. If your application used to handled services in UCS using former releases of Context Services, you should migrate these services to GMS, as detailed in the migration page.

You cannot manage profiles in Context Services if you don't deploy UCS. Starting 8.5.103.16, you must deploy UCS to access the **Journey Timeline** interface.

- For a description of deploying UCS, see the *eServices documentation*.
- For a description of deploying Context Services in UCS, refer to the guidelines in this chapter.
  - Prepare Your Deployment in UCS
  - Learn about how UCS works with Context Services
- Finally, you must perform specific configuration steps in your Context Services application to ensure compatibility with your UCS installation.

See Configuring CS for UCS Compatibility.

Prepare Your Deployment in UCS

Describes all of the required procedures for deploying UCS and its Context Services capabilities for the Customer Profile API. For a description of deploying UCS as part of an eServices solution, see the *eServices Deployment Guide*.

Prerequisites

Functioning environment including:

- Management Framework: DB Server, Configuration Server.
- Genesys Administrator or Configuration Manager.
- RDBMS, either Oracle or Microsoft SQL.
- Java Environment and Libraries for eServices and UCS. This is a single component provided on your UCS product CD.

Setting up the UCS Database

Purpose: To set up the database or databases that UCS will use.

Prerequisites

RDBMS, either Oracle or Microsoft SQL. See also the eServices 8.0 Deployment Guide. See the Deployment category page for overall prerequisites for deploying UCS.

Procedure

1. Create a database in your RDBMS.
2. Locate scripts in `\Universal Contact Server\<application-name>\sql_scripts\<RDBMS-type>`.
3. Run `ucs-<RDBMS-type>.sql` for a new installation or choose the proper upgrade script for your RDBMS type.
   For an existing UCS database, run all scripts that cover your existing version, the current version, and all versions in between. For example, to upgrade from 7.6.1 to 8.0.2, you must run
   1. `upgrade_<RDBMS-type>_7.6.1_to_8.0.0.sql`
   2. `upgrade_<RDBMS-type>_8.0.0_to_8.0.1.sql`
   3. `upgrade_<RDBMS-type>_8.0.1_to_8.0.2.sql`

   Genesys supplies upgrade scripts for all releases starting with 7.0.1.

Special Information for Oracle RAC

DAP Configuration

To connect UCS to an Oracle Real Application Cluster (RAC), configure a DAP for UCS as follows:

- Use the first node's host and port settings on the `Server info` tab.
- For the host, port, and ONS settings of each additional node, create options in the `settings` section, as follows. Note that the ONS settings are optional.
  - Name: ONSConfiguration
    Value: nodes=node1:node1port,node2:node2port, ... where port is the ONS port, usually 6251
  - Name: hostx, where x is a positive integer
    Value: host of RAC
  - Name: portx, where x is a positive integer matching one of the hostx options

Value: DB port of RAC, usually 1521

- Create a service option to specify the network service name of the Oracle database.

Here is an example configuration for three nodes, named rac1, rac2, and rac3. The DB port is 1521 and the ONS port is 6251 for all nodes.

```
[ServerInfo]
host: rac1
ports: default, 1521
[Options > settings]
ONSConfiguration: nodes=rac1:6251,rac2:6251,rac3:6251
host1: rac2
port1: 1521
host2: rac3
port2: 1521
```

UCP Library

Starting with the 8.1 release, support of Oracle RAC also requires that you deploy the Universal Connection Pool library, which Genesys is not able to deliver with the installation package. To deploy the UCP library:

1. Download the `ucp.jar` file, version 11.2.0.1.0 (higher versions are not supported) from the Oracle web site: [1]
2. Copy the jar file to the UCS home folder in `./lib/db/oracle`

When connected to an Oracle RAC configuration, the UCS database layer uses this jar file for connection handling. If UCS is started against an Oracle RAC without `ucp.jar`, it will fail to start.

Next Steps

Configure a Database Access Point (DAP).

Configure DAP

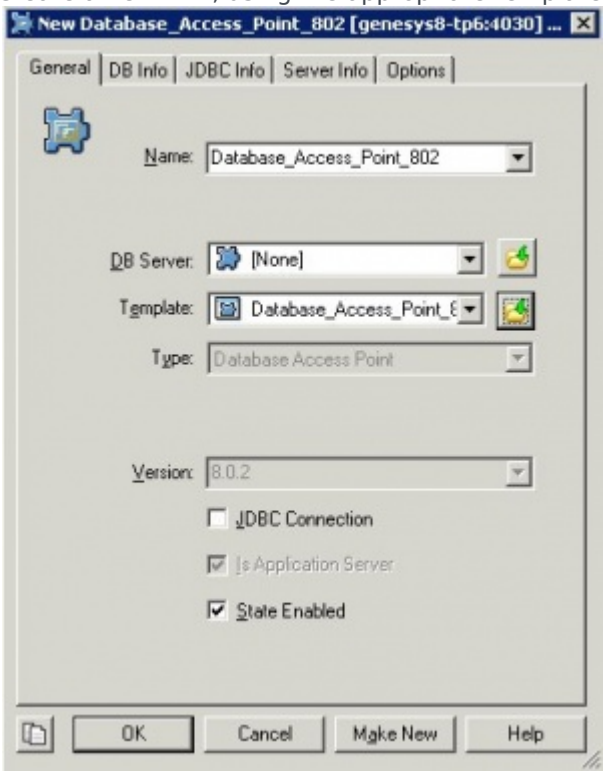
Purpose: To set up the DAP (Database Access Point) that UCS will use.

Prerequisites

RDBMS, either Oracle or Microsoft SQL. See also the eServices 8.0 Deployment Guide.

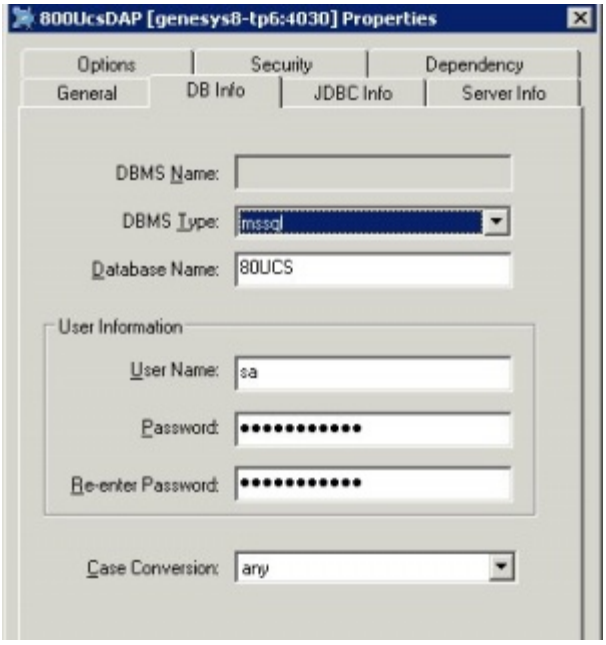
Procedure

1. Create a new DAP, using the appropriate template. On the General tab:



   a. Enter a name for the DAP.
   b. Do not enter anything in the DB Server field.
   c. Select `Enable JDBC access.`

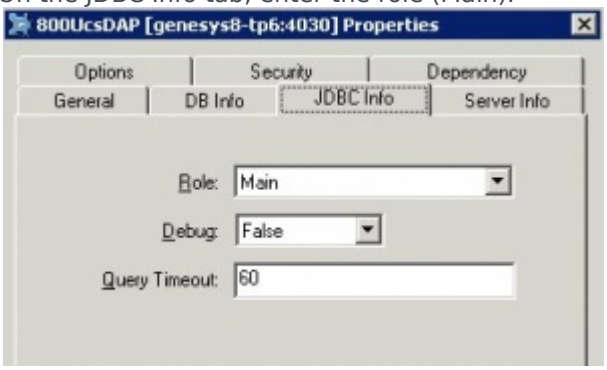
2. On the Database Information tab:



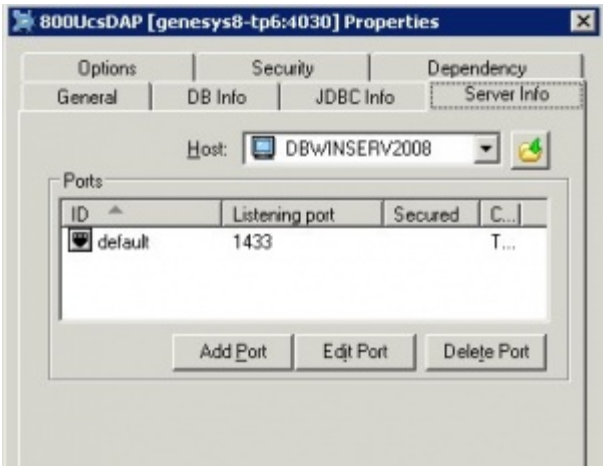
a. Enter the DBMS type, database name, user name, and password.

b. Set Case Conversion to any, and leave the DBMS Name field clear.

3. On the JDBC Info tab, enter the role (Main).



4. On the Server Info tab, enter the host name and port number.



> **Important**
> To connect to an Oracle RAC (Real Application Cluster), see this additional information.

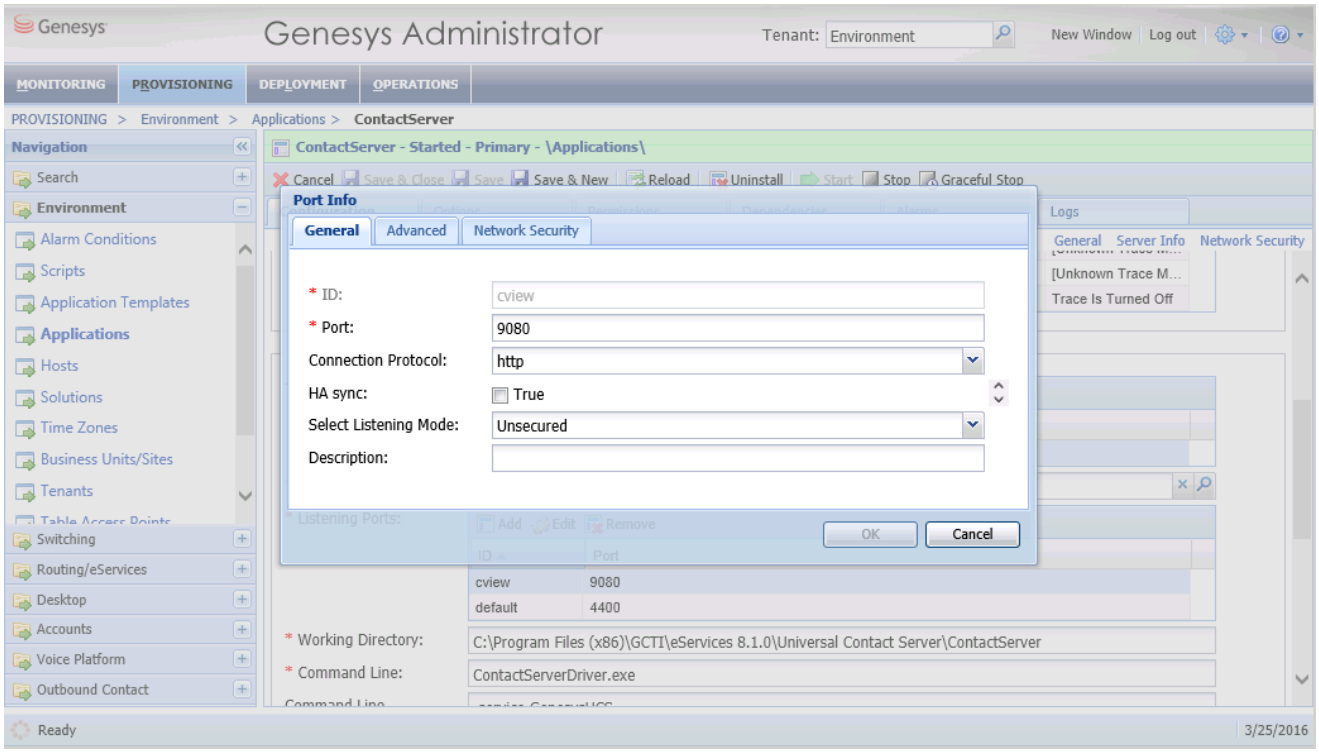
Next Steps

Configure a UCS Application object.

Configure UCS Application

Purpose: To configure a UCS Application object.

Configure UCS Application for Context Services



Prerequisites: It is preferable to set up the database and configure a DAP before creating the Application object.

1. On the **General** tab, enter a name.
2. On the **Server Info** tab, enter a host name and add a standard `cview` port with its protocol set
3. On the **Start Info** tab, enter an arbitrary character. The real values will be entered during insta
4. On the **Connections** tab, add connections to the UCS DAP, Message Server, and Stat Server.

5. On the **Options** tab, in the `cview` section:
    1. Set `enabled` to `true`.
    2. Set `tenant-id` to the identifier of the tenant with which UCS will be associated.
    3. Set other configuration options as needed. All options are described on the Configuration O page.

> **Important**
>
> The underscore character (_) is not supported for host names that UCS connects to. Having this character in a host name can result in unstable behavior, such as inability to connect to the target host. Note that RFC 1123, section 2.1 "Host Names and Numbers" limits host names to letters, digits, and hyphen. If a host that UCS connects to contains underscore in its name, Genesys recommends that you create an alias and change the host name in the Configuration Layer.

Next Steps

1. Optionally, configure UCS to use TLS.
2. Optionally, configure role privileges for UCS.
3. Install UCS. Installing UCS is a simple matter of launching the installation entering Configuration Server login information.

Export Certificates

Purpose: Describes using Microsoft Managment Console to export digital certificates.

Procedure

If you have generated a Windows certificate, as described in the "Certificate Generation and Installation" chapter of the *Genesys Security Guide,* you must use Microsoft Management Console to make the certificate usable by UCS, as follows:

1. From the Windows Start menu, select `Run`, then execute the `mmc` command to start Microsoft Management Console.
2. In the Trusted Publishers folder, select the certificate that you assigned to your host in the Genesys configuration environment. Right-click and select `Export` to launch a wizard.
3. Click `Next` in the first pane of the wizard.
4. Select `Yes, export the private key.`
5. Select `Personal Information Exchange - PKCS #12 and Enable strong protection.`
6. Enter a password.
7. Enter a file name (such as `certificate.pfx`) and select the location to save it.

Next Steps

Configure UCS to use the certificate, as described in Using TLS with UCS.

Using TLS with UCS

Purpose: To set up UCS to use TLS.

Overview

This page describes setting up UCS to use TLS for secure connections. The procedure can also be used with E-mail Server, a component of Genesys eServices. For clients of UCS, see Using TLS with UCS Clients. This page refers to keytool, which is a key and certificate management utility included in JDK or JRE installations. For instance, when you install JDK, keytool is placed in the `\bin` directory.

> **Important**
> Starting with release 8.1.3, the TLS options are configured as described in the *Framework 8.1 Configuration Options Reference Manual*.

Procedure

1. Generate a certificate, in any of the following ways:
   - Use Windows Certificate Services, as described in the "Certificate Generation and Installation" chapter of the *Genesys 8.1 Security Deployment Guide.*
   - Use keytool with the–genkey parameter; for example:

```
keytool -genkey -v -alias hostname.example.com
 -dname "CN=hostname.example.com,OU=IT,O=ourcompany,C=FR" -keypass theKeyPassword
 -keystore certificate.jks -storepass theKeystorePassword -keyalg "RSA" -sigalg
 "SHA1withRSA"
 -keysize 2048 -validity 3650
```

   - Use any other tool, such as openSSL.
2. In the Genesys configuration environment, assign the certificate to the Host on which UCS is running, as described in the "Genesys TLS Configuration" chapter of the *Genesys 8.1 Security Deployment Guide.*
3. If you generated a Windows certificate, you must use Microsoft Management Console to make the certificate usable by UCS.
4. Locate the certificate and copy it to a selected location on UCS's host.
5. Set configuration options in your UCS Application object. Starting with release 8.1.3, the TLS options are configured as described in the *Genesys 8.1 Security Deployment Guide.*

Next Steps

Optionally, configure the clients of UCS to use TLS, as described on the Using TLS with UCS Clients page.

8.1.0 Maintenance Release

The 8.1.0 maintenance release of October 2011 adds the possibility of performing the following TLS-related configuration on the `Server Info` tab (Configuration Manager) or section (Genesys Administrator):

- Configure multiple ports
- Set Secured = Yes, in which case UCS starts in TLS mode
- Specify the connection protocol as ESP or HTTP

Note these limitations:

- Only one certificate per protocol can be configured for one UCS.
- There must be a default port that uses ESP and is associated with a valid certificate.
  - This is the port marked `default` on the Server Info tab (Configuration Manager) or the Server Info section of the Configuration tab (Genesys Administrator).
  - You can leave its connection protocol unspecified, in which case it uses ESP. What you must not do is specify any other protocol for it.
  - If the server is not able to start listening on this port, then an exception is raised and the server exits.

Using TLS with UCS Clients

Purpose: Set up clients of UCS to use TLS.

Overview

Procedures differ according to whether the client is integrated into the Genesys system.

Integrated Applications

To connect the client in a secured mode, execute the "Configuring a secure client connection" procedure in the "Genesys TLS Configuration" chapter of the Genesys Security Deployment Guide.

Non-Integrated Applications

Applications that are not integrated into the Genesys system must verify the public key. One way to do this is to import the public key using keytool, as in the following example for a Java client:

1. Export the certificate. The following is an example command line:

   ```
   keytool -export -v -alias hostname.example.com -file certificate.cer
    -keystore certificate.jks -storepass theKeystorePassword
   ```

2. Import the certificate on all clients of UCS. The following is an example command line:

   ```
   keytool -import -alias hostname.example.com -file certificate.cer
    -keystore .keystore -storepass anotherPassword
   ```

3. Copy this certificate (public key) to a location on the client host.
4. Configure the client to point to this imported certificate. The way to do this depends on the client. As one example, with a Java application, you can start the application with a command line like the following:

   ```
    java -Djavax.net.ssl.trustStore="<CERTIFICATE_DIRECTORY>\<CERTIFICATE_FILE>"
   <application_name>
   ```

Configuration Options

Purpose: Lists the configuration options that your application can read in UCS.

> **Important**
> This configuration options do not apply to GMS. Refer to the main options reference for details.

Description

The tables in the following sections present the UCS configuration options that Context Services can read from the Configuration Layer:

- `cview` section—Options and values specific to Context Services.
- `archiving` section—Activates set-based archiving.
- `authentication` section—Enables and configures authentication control.
- `log-filter` section—Implements security log filtering.
- `log-filter-data` section—Also relates to security log filtering.

You can modify all these option values in the Configuration Manager. However, the change may not be effective immediately. For some options you have to restart UCS for changes to take effect (see the tables below).

> **Important**
> In Configuration Manager and Genesys Administrator, you can see many options other than those described here. They relate to UCS's functioning in eServices (short description **here**), and are described in the eServices Reference Manual.
>
> Also, this page describes options that are displayed on the Options tab of the UCS Application object in Configuration Manager and Genesys Administrator. Some options that can be added to the Annex Tab in Configuration Manager, or to Advanced View (Annex) in Genesys Administrator, are described in Using Configuration Options to Schedule Service Pruning and UCS Options for TLS.

[cview] Section

This section adjusts the overall configuration of Context Services.

`cview` Section

Name	[+] Restart UCS This column indicates whether you must restart UCS for changes in the option value to take effect.	Description
enabled	Yes	• `true` to enable Context Services functionality in Universal Contact Server (UCS). • `false` (default) to disable Context Services.
base-url	Yes	The base URL used to deploy Context Services. Based on this configuration, the services are available at the following URL: http://${ip-address}:${port}/${base-url}/${operation} Where: • ${ip-address} is the IP address configured below. • ${port} is the port on which the web services are deployed (see above). • ${base-url} is the base URL used to deploy Context Services. For example, if the ip-address is `192.168.1.1`, the port `8080`, and the base URL `cms`, the Set Server Mode operation would be available at the following URL: http://192.168.1.1:8080/cms/server/mode
ip-address	Yes	IP address used to deploy Context Services (`localhost` by default).
data-validation	No	• `true` to enable data validation, which enforces additional checks on data provided by the connected

Name	[+] Restart UCS This column indicates whether you must restart UCS for changes in the option value to take effect.	Description
		clients. • `false` to disable data validation.
start-mode	Yes	Start-mode of the server mode: • `maintenance` • production
tenant-id	Yes	Defaults to 101. Specifies the numeric tenant ID associated with Context Services: subsequent customer/contact records created through your application are associated with this tenant.
metadata-cache	No	• `true` to enable the caching of metadata in the memory. • `false` to disable the metadata caching. In that case, each access to the metadata triggers a DB query. Important The cache contains metadata for contact attributes, identification keys, profiles, services, states and tasks extensions.

[archiving] Section

This section activates and deactivates set-based archiving. It is not present on the UCS template; you must create it. It contains just one option:

`archiving` Section

Name	[+] Restart UCS This column indicates whether you must restart UCS for changes in the option value to take effect.	Description
use-np	Yes	• `true` to enable set-based archiving. • `false` (default) to disable set-based archiving.

[authentication] Section

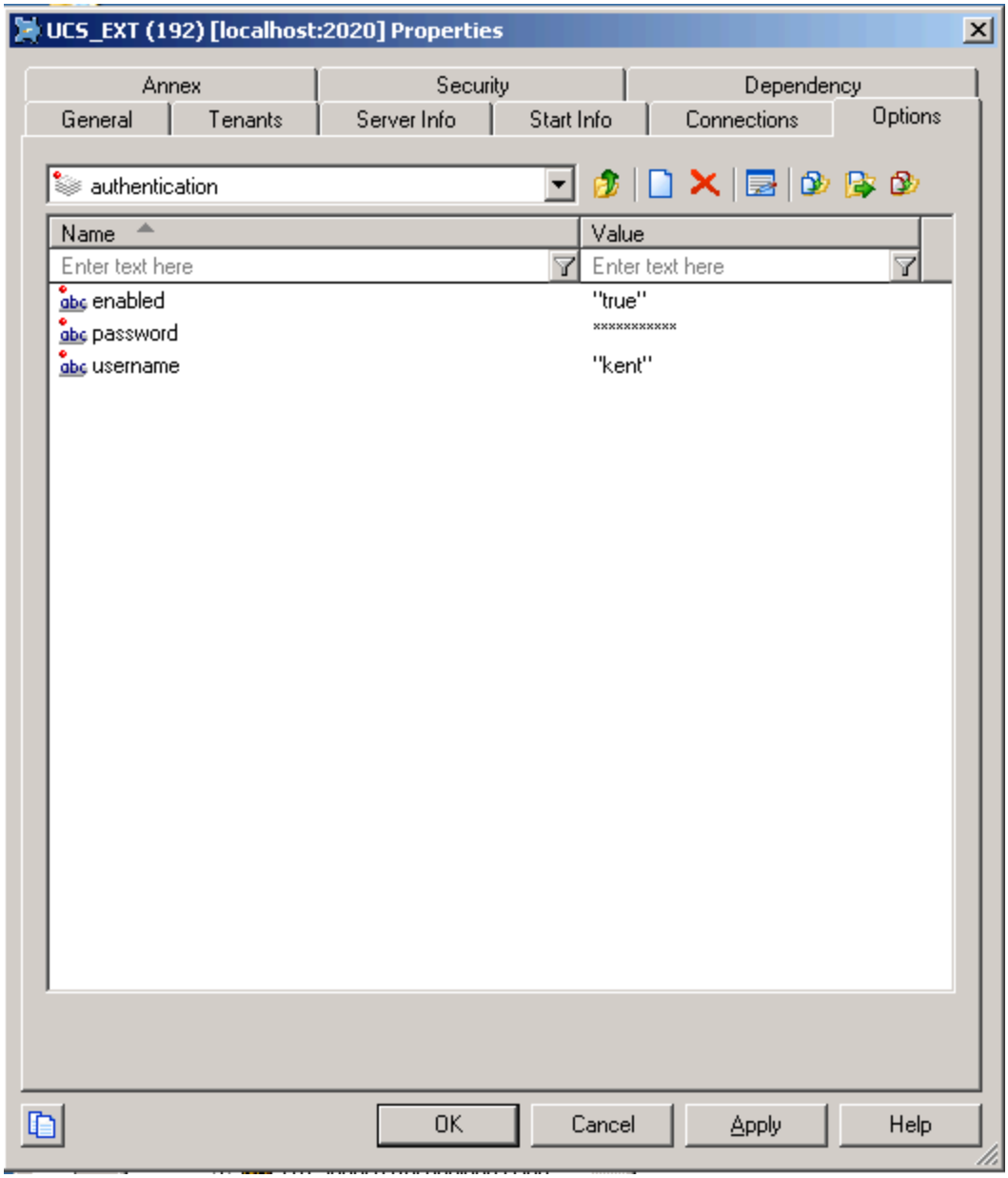
This section configures authentication for clients connecting to UCS. Authentication, available since release 8.0.300.02, applies to UCS/CS only. See also:

- The directly-related Basic Access Authentication page.
- Authentication on the Security and Authentication page.

authentication Section

Name	[+] Restart UCS This column indicates whether you must restart UCS for changes in the option value to take effect.	Description
enabled	Yes	• `false` to disable authentication (default). • `true` to enable authentication.
mode	Yes	Authentication mode: • `single-user` to authenticate using UCS options (default). • `multi-users` to authenticate using Persons from Configuration Server.
password	Yes	Password to check the identity of the specified user. Effective only if `mode` is set to `single-user`.
username	Yes	User name allowed to connect to the Context Services API. Effective only if `mode` is set to `single-user`.

Name	[+] Restart UCS This column indicates whether you must restart UCS for changes in the option value to take effect.	Description
use-role	Yes	• `false` to disable the use of roles in authentication (default). • `true` to enable the use of roles in authentication.



[log-filter] Section

This section contains general settings for how or whether user data keys appear in the logs. Its settings can be overridden for specified keys by options in the `log-filter-data` section.

`log-filter` Section

Name	[+] Restart UCS This column indicates whether you must restart UCS for changes in the option value to take effect.	Description
default-filter-type	No	Sets the default for filtering the output of user data keys to the UCS server log. Possible values: • `skip`—Does not output key-value pairs. • `hide`—Outputs the keys but hides their values. • `copy` (default)—Outputs both the keys and their values. This default filter applies to all user data keys, except that is is overridden by any settings for individual keys in the `log-filter-data` section.
filter-depth	No	Depth used while filtering nested key-value pairs. The default is `99`. Any value greater than this is not checked. Using a high value can result in lower performance in the case of deeply nested key-values.

[log-filter-data] Section

This section enables you to override the `log-filter` section's setting for one or more specific data keys. You do this by creating options with the name `<keyname>` and the value `<filtering mode>`, where:

- <keyname> is the user data key affected.
- <filtering mode> is `skip`, `hide`, or `copy`, the same as the possible values of `default-filter-type` in the `log-filter` section.

UCS Options for TLS

Purpose: Configure UCS, in releases 8.1.0 and later, to use secure connections with TLS.

Starting in release 8.1.0, you can configure UCS to use TLS for secure connections by adding sections to the Annex Tab in Configuration Manager or the Advanced View (Annex) in Genesys Administrator. The sections that you add, which specify certificate options for TLS support with the HTTP or ESP protocols, have names of the following forms:

- `<protocol>.tls.keystore`
- <protocol>.tls.key

Here <protocol> can be either `esp` or `http`, making a total of four possible sections.

<protocol>.tls.keystore section

Option Name	Default Value	Valid Values	Value Changes	Description
type	JKS	JKS, PKCS12	Take effect upon restart	The type of keystore that the `path` option points to. Not required.
path	./certificate.jks	Any valid pathname	Take effect upon restart	The path to the keystore holding the certificate key-pair information. Required if an HTTPS port or ESP TLS port is set. Examples: `c:\mypath\to\mykeystore` (Windows), `/opt/keystore/mykeystoreJava` (Unix), `./folder/keystore` (Unix, relative path).
password		Any string	Take effect upon restart	The password that secures the keystore that the `path` option points to. Required if the `path` option is present. Note that the default value is an empty string. Not to be confused with the option of the same name in the `<protocol>.tls.key`

Option Name	Default Value	Valid Values	Value Changes	Description
				section.

<protocol>.tls.key section

Option Name	Default Value	Valid Values	Value Changes	Description
password		Any string	Take effect upon restart	The password used to secure the private key from the keystore that the `<protocol>.tls.keystore` section points to. The default value is an empty string. Not to be confused with the option of the same name in the `<protocol>.tls.keystore` section.

UCS Role Privileges

Purpose: Describes the role privileges that are specific to UCS.

Description

Roles determine what actions a specified user may perform on a specified object. In UCS/CS, the user is most commonly an application; for further explanation, see Role-Based Access Control in the Context Services Developer's Guide. Privileges are assigned as configuration options in the `Role Privileges` tab of the Role object in Genesys Administrator.

To enable the use of roles, the `use-role` option must have the value `true`.

Framework 8.0 Genesys Administrator Help and the *Genesys Security Guide* provide general information on how to use Genesys Administrator and Management Framework to configure access permissions.

Privilege Groups

The following tables place privileges in related groups, as they appear in Genesys Administrator.

Customer related

Privilege
Create Customer Profile
Create Customer Profile Extension
Delete Customer Profile Extension
Read Customer Profile
Read Customer Profile Extension
Update Customer Profile
Update Customer Profile Extension

Service related

Privilege
Create Service Extension
Delete Service Extension
Read Service
Read Service Extension
Start Service
Stop Service

Privilege
Update Service Extension

State related

Privilege
Create State Extension
Delete State Extension
Read State
Read State Extension
Start State
Stop State
Update State Extension

Task related

Privilege
Create State Extension
Delete State Extension
Read State
Read State Extension
Start State
Stop State
Update State Extension
Create Task Extension
Delete Task Extension
Read Task
Read Task Extension
Start Task
Stop Task
Update Task Extension

Schema management related

Privilege
Create Id Keys
Create Profile Extension Schema
Create Service Extension Schema
Create States Extension Schema
Create Tasks Extension Schema
Read Business Attributes
Read Genesys Administrator Roles
Read Id Keys

Privilege
Read Profile Extension Schema
Read Service Extension Schema
Read States Extension Schema
Read Tasks Extension Schema

System management related

Privilege
Change server mode
Get content from interaction
Read server information

Security and Authentication

Purpose: Gathers together topics relating to security, encryption, authentication, and the like.

Database Encryption

For database encryption, Genesys recommends using Transparent Data Encryption (TDE):

- Oracle 11—Tablespace-level; see http://www.oracle-base.com/articles/11g/TablespaceEncryption_11gR1.php.
- MSSQL Server 2008—Database-level; see http://msdn.microsoft.com/en-us/library/cc278098(SQL.100).aspx.

Do not use column-level encryption.

Security Log Filtering

You can use configuration options in the `log-filter` and `log-filter-data` sections to control how or whether user data keys appear in the logs.

TLS

UCS/CS supports Transport Layer Security (TLS) in various ways:

- For UCS, see Using TLS with UCS and related pages. The procedures described also apply to E-mail Server.
- For clients of UCS, see Using TLS with UCS Clients.
- UCS/CS also supports secure connections to Configuration Server.

Authentication

When clients connect to UCS, there are two possible modes of authentication, specified by configuration options in the `authentication` section.

- Single-user—Clients connect using the user name and password specified by the UCS options `username` and `password`. This means all UCS clients must use the same credentials. To enable single-user authentication, give the `mode` option a value of `single-user`.

- Multi-User—Clients are configured as Persons in the Configuration Layer, and connect to UCS using the user name and password specified by their Person object. This means that each client can have its own credentials. To enable multi-user authentication, give the `mode` option a value of `multi-user`.

These and all other UCS/CS options are described on the Configuration Options page.

Role-Based Access Control

Role-based access control is available in UCS/CS starting in release 8.1.0. See

- UCS Role Privileges in this User's Guide.
- Role-Based Access Control in the Context Services Developer's Guide.

Multiple UCS Instances in Single Tenant

Purpose: describe a solution that enables multiple independent UCS instances to be deployed in a single tenant, based on the Access Group mechanism available in Configuration Server.

Overview

This solution uses access rights to restrict UCS instances from seeing objects that do not belong to them. Genesys components that access the Configuration Server database typically use the `system` account to access Configuration objects, granting the components global visibility. However, it is possible to use another account simply by changing the `logon as` option on the Security tab of the relevant Application object. One reason to use this solution relates to standard responses: If you have multiple UCS instances in a single tenant without restricted access, each instance will have access to the standard response library managed by the other instances. And if a UCS instance has access to a standard response library that it does not manage, it will keep deleting the standard responses from the Configuration Server database. The result will be that all instances will be repeatedly deleting other standard response libraries and re-creating their own.

> **Important**
>
> There is no need to use this configuration if you do not use standard responses and don't mind the UCS instances sharing each other's Contact and Interaction attributes. And of course these issues do not arise when the UCS instances are in different tenants.

Different access groups represent different lines of business (LOB). This example uses two LOBs, email and chat.

Configuration Procedure

1. Create an access group for each Line of Business (LOB).

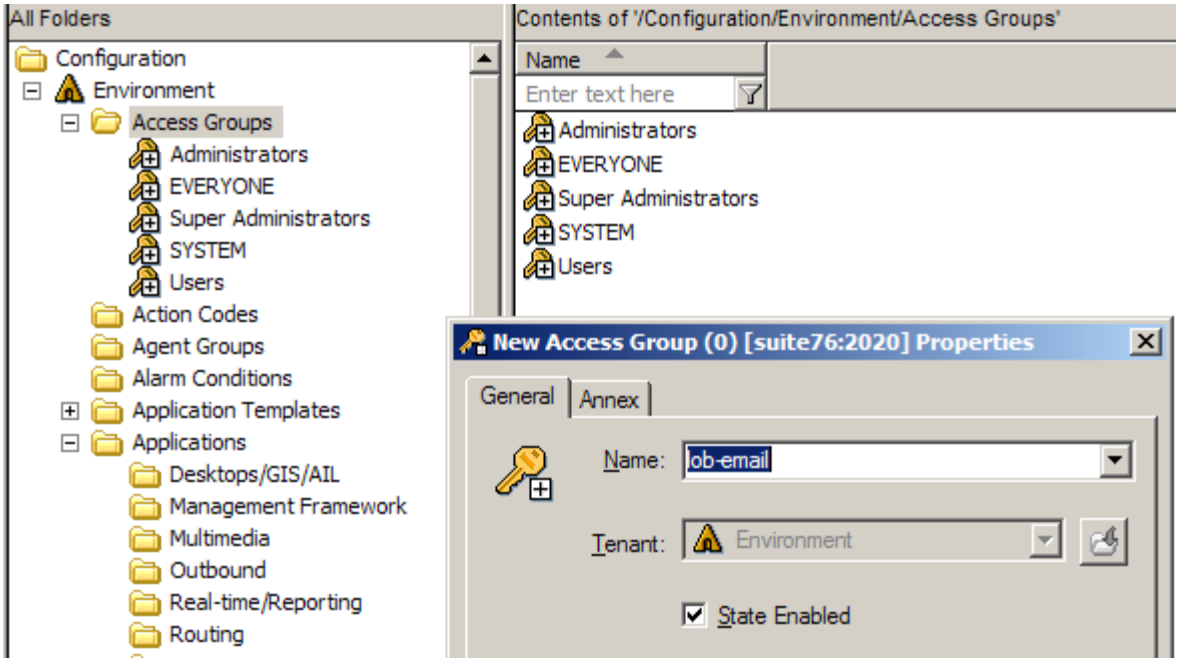
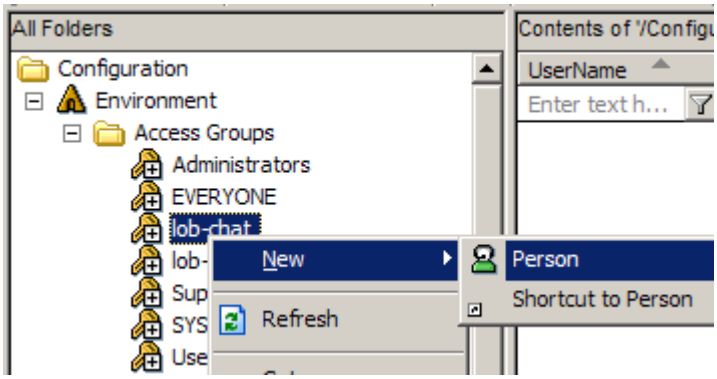


Figure 2: Creating Access Groups

2. Create a user for each of the access groups: right-click the access group, then select `New > Person`.



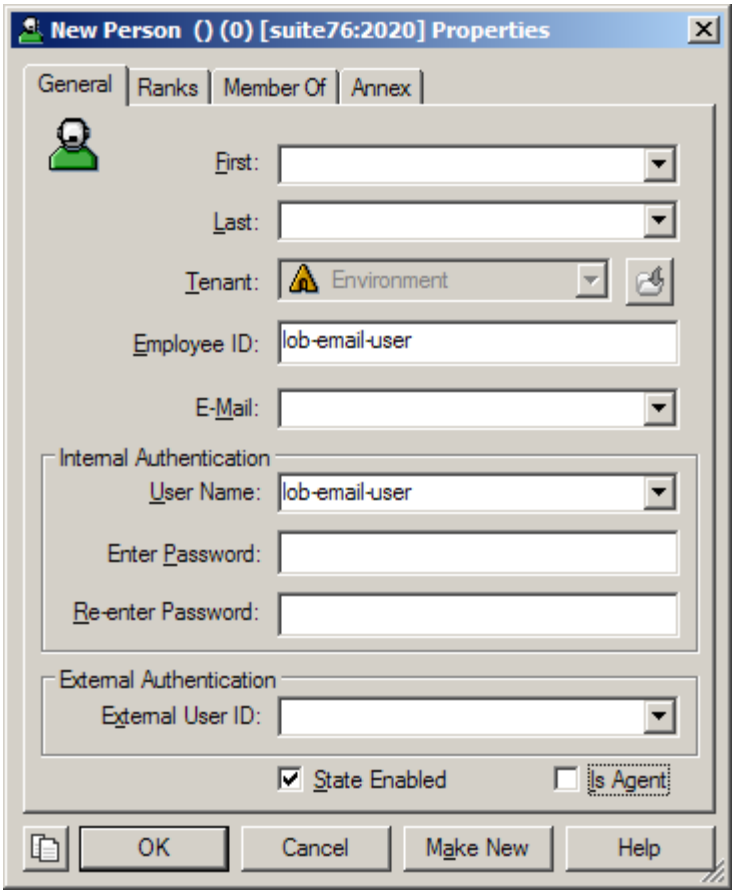


Figure 3: Creating a Person

3. Configure each access group's permission to access configuration objects:
   a. Right-click the tenant.
   b. On the `Security` tab, click `Permissions`.
   c. In the resulting Object Permission dialog, click `Add...`
   d. In the resulting Add dialog, click `Add` and select `Full Control`.

Do this for Environment and all defined Tenants (multi-tenant environment), or for Environment and Resources (single-tenant). The figure below shows the process for the Environment tenant.

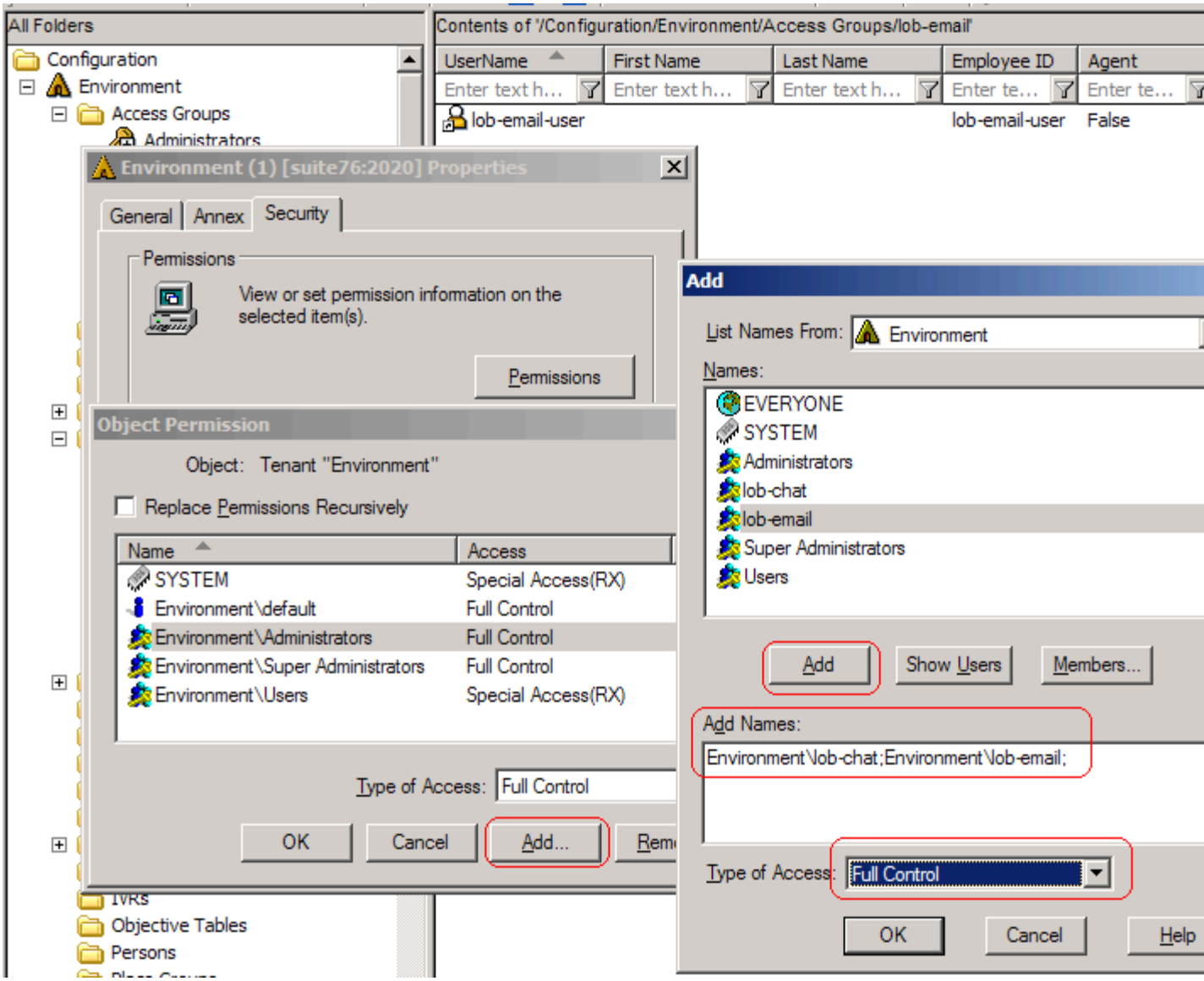


Figure 4: Setting Tenant Permissions

4. On the `Security` tab of the UCS Application, set the `Log On As` option to associate this UCS with one of the created users, and hence with an access group.

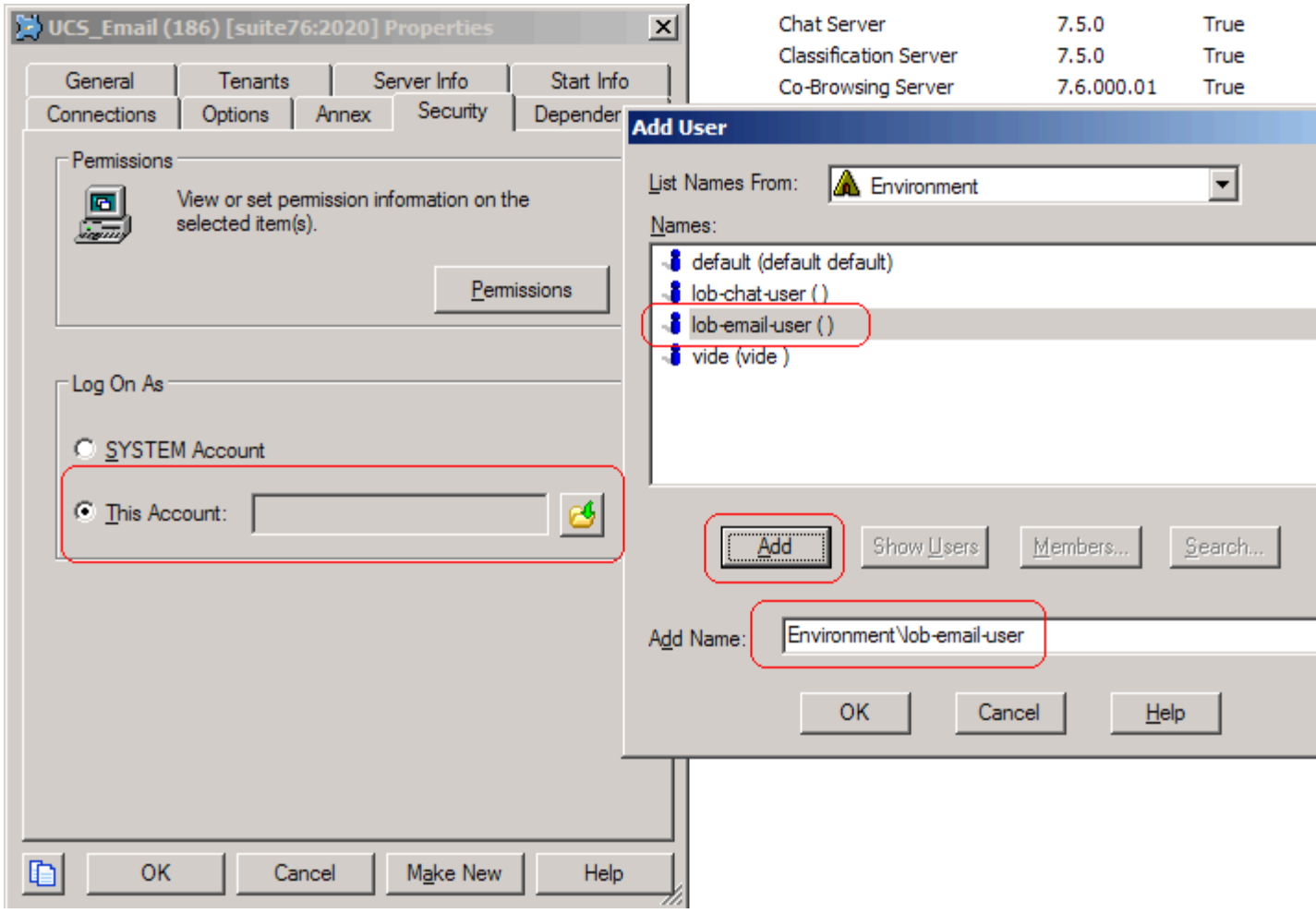


Figure 5: Setting the Log On As Account

The UCS is now able to access only the objects for which the access group has permissions.

5. Set permissions for attributes: Contact and Interaction Attributes are created in the Configuration Server database before being propagated to UCS. Therefore, in order to restrict a given attribute to one of the LOBs, you must specify permissions manually in the Configuration Server database.

   a. Right-click the desired `Attribute Value`.

   b. On the `Security` tab, click `Permissions`.

   c. In the resulting Object Permission dialog, click the various LOB groups and select the desired permissions.

The figure below shows the `customerId` contact attribute being restricted to the Chat LOB.

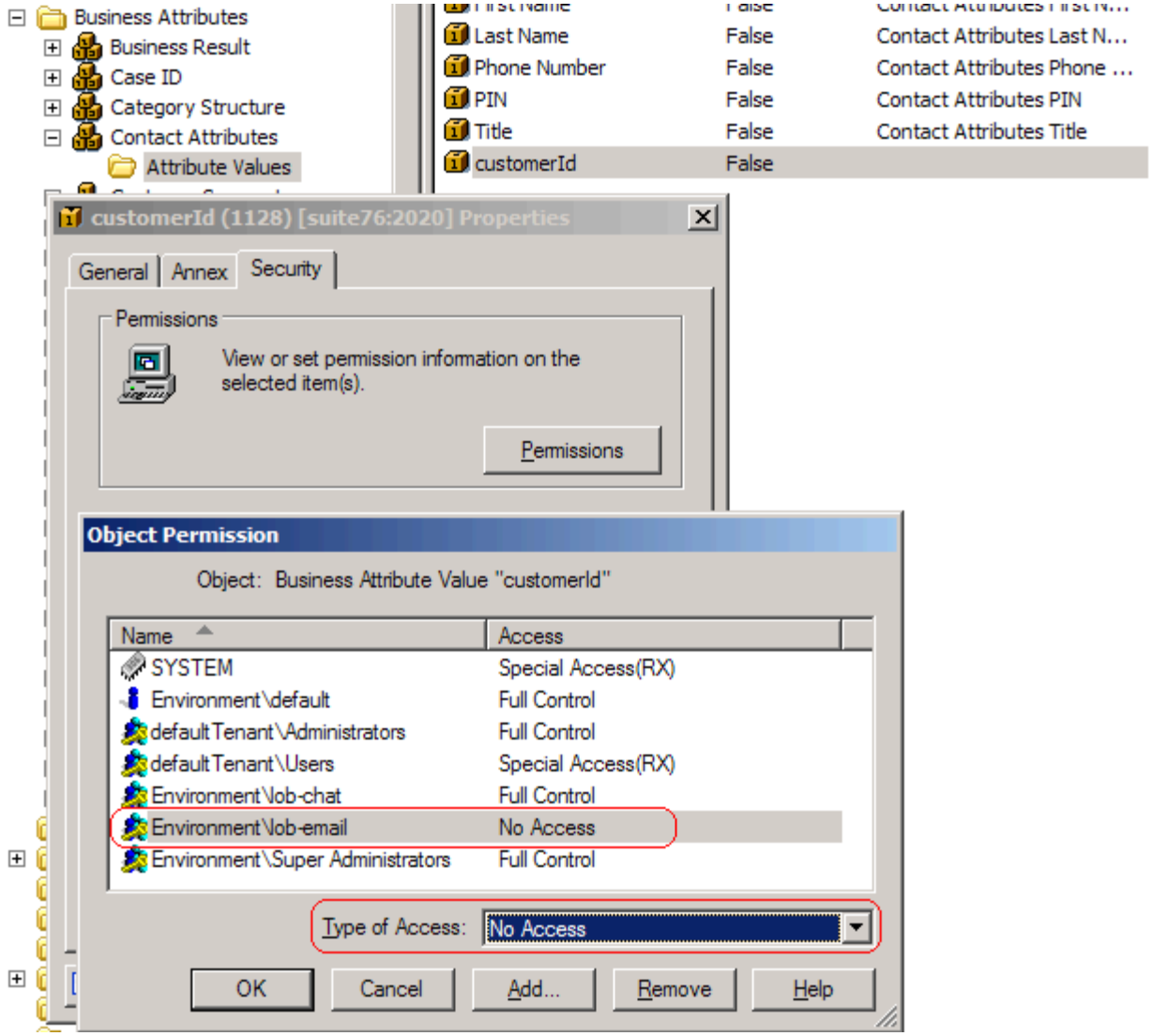


Figure 6: LOB-Specific Contact Attributes

The email UCS will now behave as if `customerId` does not exist.

> **Important**
>
> Genesys recommends doing this before starting the email UCS, to keep attribute metadata from being prematurely propagated.

6. To avoid having to perform the task in the previous step multiple times, you can group attributes in a folder and set permissions on the folder, as shown in the figure below. When an attribute is moved to the folder, it inherits the permissions.

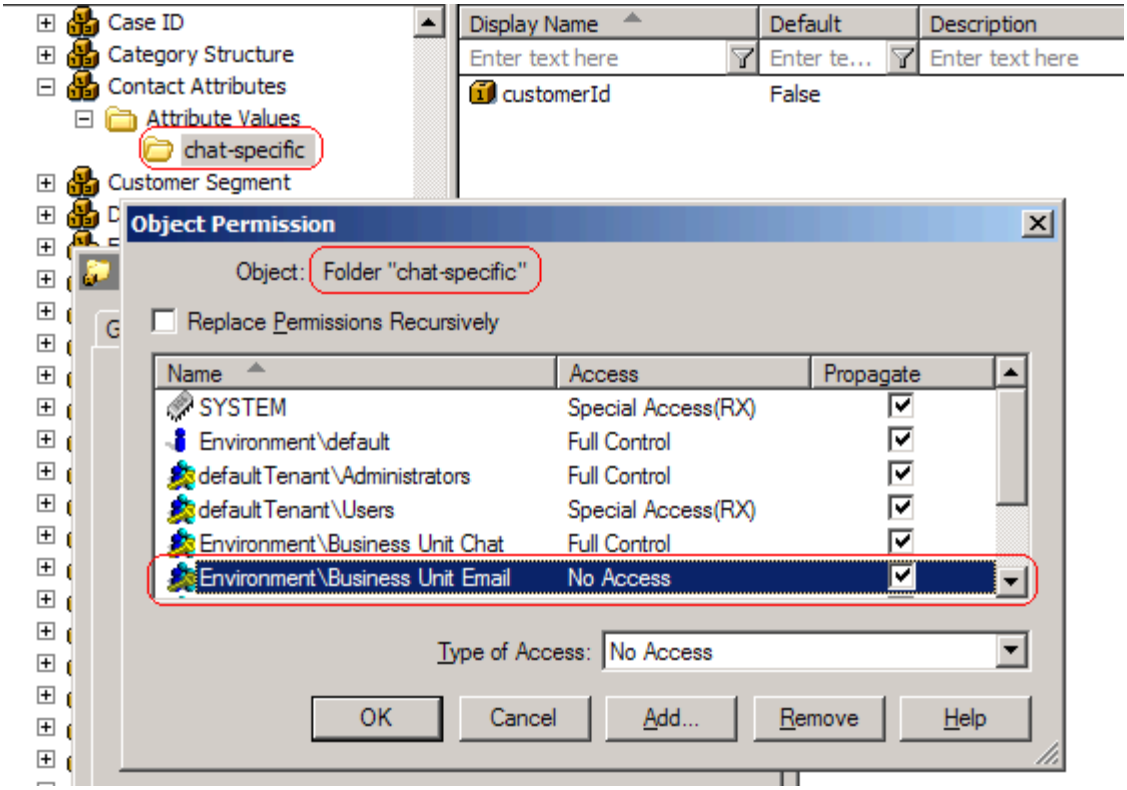


Figure 7: Chat-Specific Contact Attributes

7. Further configuration of UCS Application objects.

> **Important**
>
> Both Primary and Backup UCS must have the same configuration options and permission settings.

   a. Set `No Access` permissions on the UCS application for all LOBs other than the one that this UCS is dedicated to. These permissions will be copied to all new objects created by this UCS in the Configuration Server database.

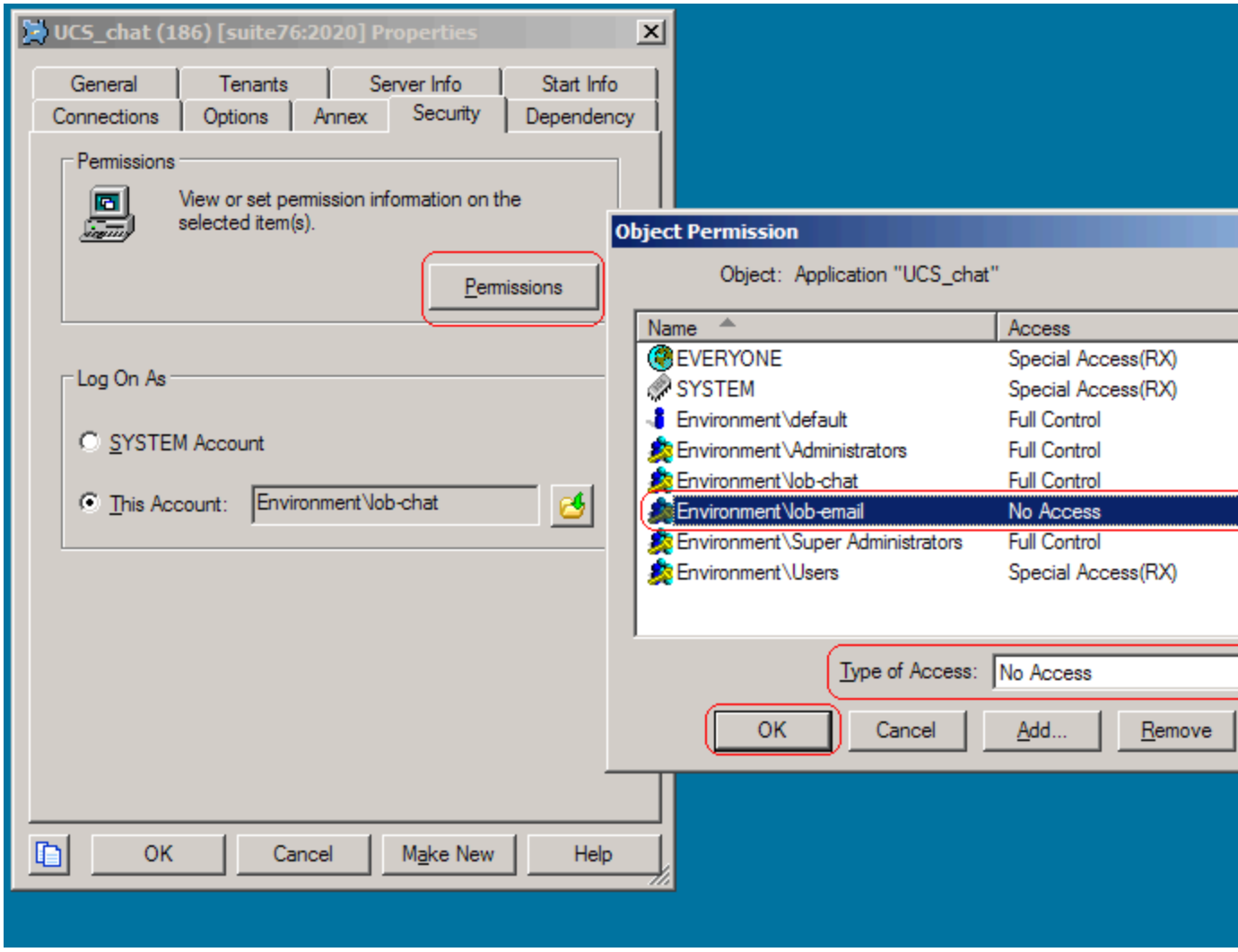


Figure 8: UCS Application Permissions

b. In the UCS `settings` section, set the `auto-propagate-rights` option to `true`.

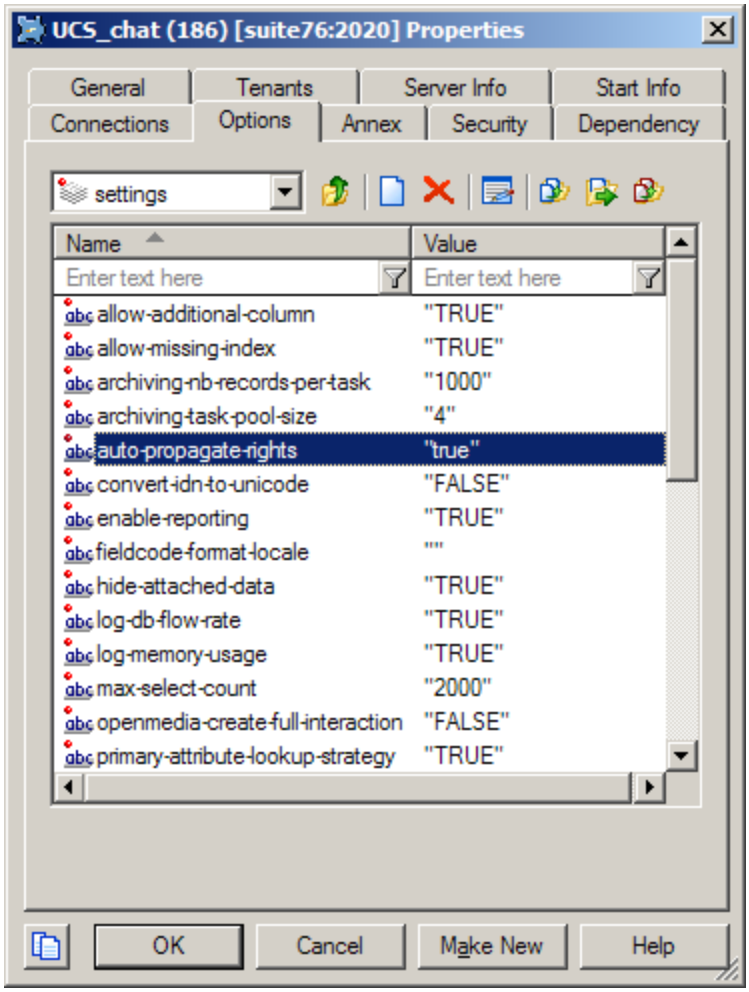


Figure 9: UCS Options

From this point on, any new root category (as well as child categories or standard response) and screening rules will inherit the access permissions of the UCS application that created them.

Adjustments

This section describes some adjustments that may be required for Knowledge Manager objects (categories, standard responses, field codes, screening rules, training objects, and models).

Migration

If Knowledge Manager objects already exist in the Configuration Server database, you must use the following migration procedure:

1. Back up the UCS and Configuration Server databases.
2. Use Knowledge Manager to export all objects for each LOB to a file. Importing and exporting is described in "Importing and Exporting" in the "Knowledge Management: Basics" chapter of the

eServices User's Guide.

3. Use Knowledge Manager to delete all Knowledge Manager objects for each LOB.
4. Check that all Knowledge Manager objects have been removed from the Configuration Server database.
5. Upgrade all UCS database instances and specify permissions as outlined in Configuration Procedure.
6. Use Knowledge Manager to import all objects for each LOB, being sure to not select the option to generate new IDs for any LOB that previously synchronized with the Configuration Server database (since these IDs may already be used in strategies).
7. Wait for Knowledge Manager data to be synchronized.

> **Important**
>
> Categories, standard responses, and field codes can have the same names in both LOBs, but not the same IDs. However, root categories and screening rules must have different names. IDs of these objects must be different as they are used as Configuration Server database object names.

Copying Knowledge Manager Data from One LOB to Another

Copying Knowledge Manager objects from one LOB to another can give rise to an issue with duplicated IDs. To avoid this you must rename the root category. This cannot be done in Knowledge Manager; instead you must manually edit the exported file, as in the following procedure. To copy Knowledge Manager data from one LOB to another:

1. Back up the UCS and Configuration Server databases.
2. Ensure that the target UCS is not able to write to the Configuration Server database.
3. Export the desired Knowledge Manager objects from the source UCS.
4. Rename the exported `.kme` file to a `.zip` file and extract the content, preserving the folder structure.
5. Open the `category-sre` folder and rename the folders that it contains. These folders are the root categories.
6. Compress the `category-sre, field-codes and screening-rules` files back to `.zip` files.
7. Rename the `.zip` file to `.kme` file.
8. Import the data into the target UCS, being sure to select `Preserve uniqueness of objects by creation of new UCS IDs.`
9. If you imported screening rules, you must now rename them.
10. Stop UCS, then set options and access rights as described in Configuration Procedure.
11. Start UCS and wait for Knowledge Manager to be synchronized to the Configuration Server database.

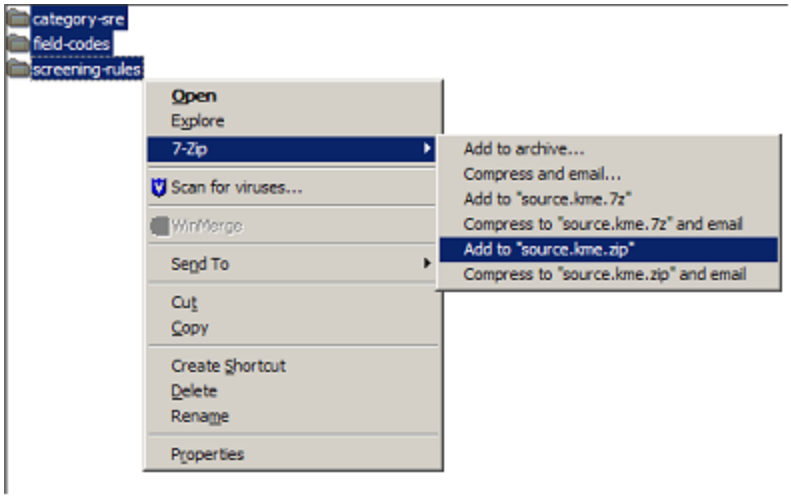


Figure 10: Knowledge Manager Folders Must be at the Root of the Zip File

Use with Other Genesys Applications

Access groups offer a generic ability to restrict access by other Genesys applications to the Configuration Server database. Consider, for example, Interaction Routing Designer (IRD). If IRD uses the default `system` account for logging into the Configuration Server database, it will have access to categories and screening rules for all LOBs. In this situation the strategy developer must keep track of which objects belong to which LOB. Otherwise he or she runs the risk of creating strategies that request rendering of a standard response that does not exist in that UCS. This is why Genesys has recommended the use of a LOB-specific naming convention on root categories and screening rules. However, if IRD logs in using the `lob-email-user` account, it will only have access to objects relevant to the email LOB.

Limitations

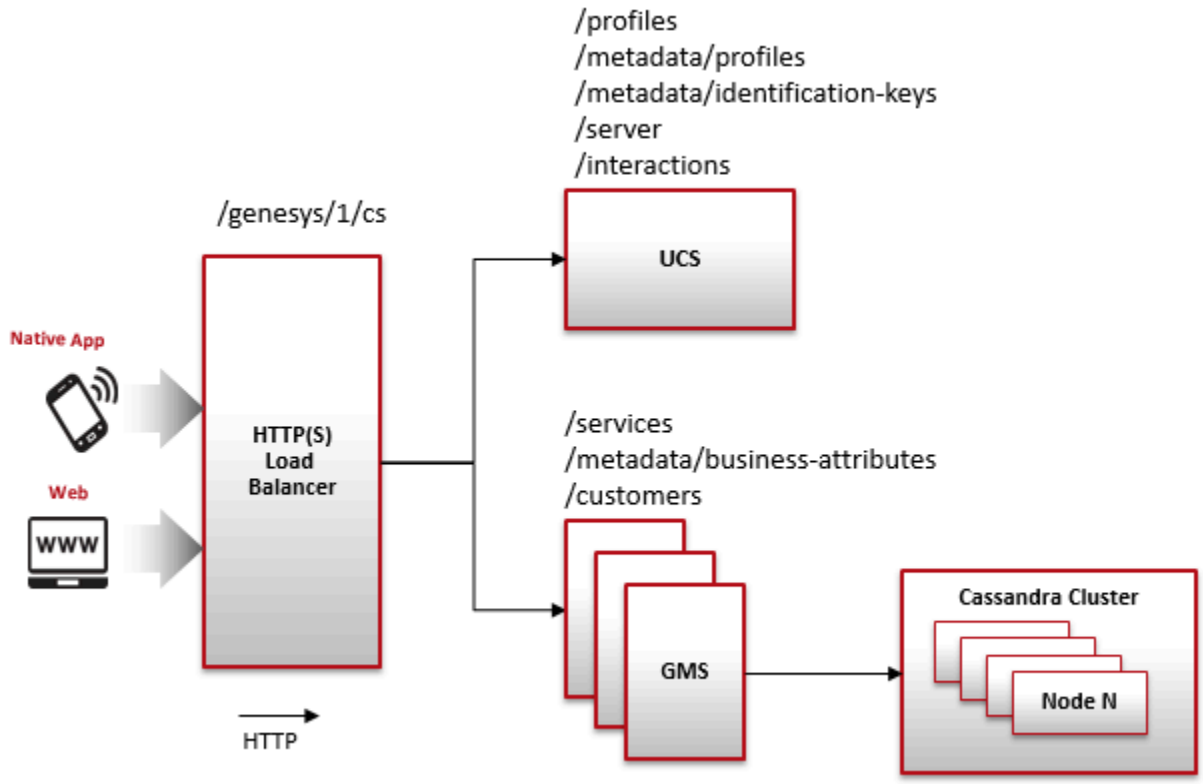
- If IRD does not log in with a limited user, it will have access to standard responses that belong to other UCS, which makes it easy to create invalid strategies, as described in the preceding section.
- It is preferable to use one Universal Routing Server and one Interaction Server per business unit in order to prevent interactions from switching from one LOB to another.
- The solution described here makes it difficult to have multiple users to manage different objects in the Configuration Server database, such as when there is one user account per real person. It is preferable to have exactly one account for each LOB.
- All UCS instances access the same Business Attributes. This makes it difficult to define different Contact Attributes, Interaction Attributes, Media, Languages, and so on in different UCS instances. The only solution is to manually apply the access limitation to each created object.
- Applications (whether desktops or servers) that are not connected to Configuration Server using limited user(s) will see standard responses that are not usable by connected UCS instances. While Genesys applications can be configured to prevent this, that configuration must be done manually. It is

preferable to use UCS as source for standard response titles anyway.

Load Balancing for a Single UCS Database

Purpose: To configure Apache HTTP load balancing for the UCS database.

This solution provides a deployment example of multiple Context Services instances, running with a single UCS database. The requests to Context Services are balanced between the servers using the Apache HTTP load balancing server. This type of configuration is beneficial for those environments experiencing high traffic to the Context Services server by providing high availability or redirecting requests to another site based on bandwidth consumption.



Important

For examples of proxy settings, refer to the additional configuration instructions to set for UCS here.

UCS Configuration

UCS/Context Services Configuration

The UCS application must be configured to run in Context Services mode. All other services must be disabled in the Configuration Server.

Apache HTTP Server Configuration

The Apache HTTP server uses the mod_proxy module for load balancing configuration. This `mod_proxy` module is directly maintained by Apache and allows more features for better performance compared to other modules (for example, `mod_jk`). The Apache server must load the modules. Requests to Context Services are forwarded to a cluster member, depending on the load factor. If the cluster member fails, requests are sent to the hot standby members. Apache see these members as hot stand-by; however, the Genesys configuration has them configured as Primary.

The mod_proxy module uses the `lbmethod` load balancing scheduler. It has three algorithms:

- `byrequests>`—performs weighted request counting.
- `bytraffic`—performs weighted traffic byte count balancing.
- `bybusyness` (Apache HTTP Server 2.2.10 and later)—performs pending request balancing.

The default configuration uses the `byrequests` algorithm. For more information, see the Apache documentation for mod_proxy.

Limitation

If the schema changes after creating the extension on one Context Services server instance, you must refresh other instances internal caches by calling the /metadata/cache URI.

Configure Context Services

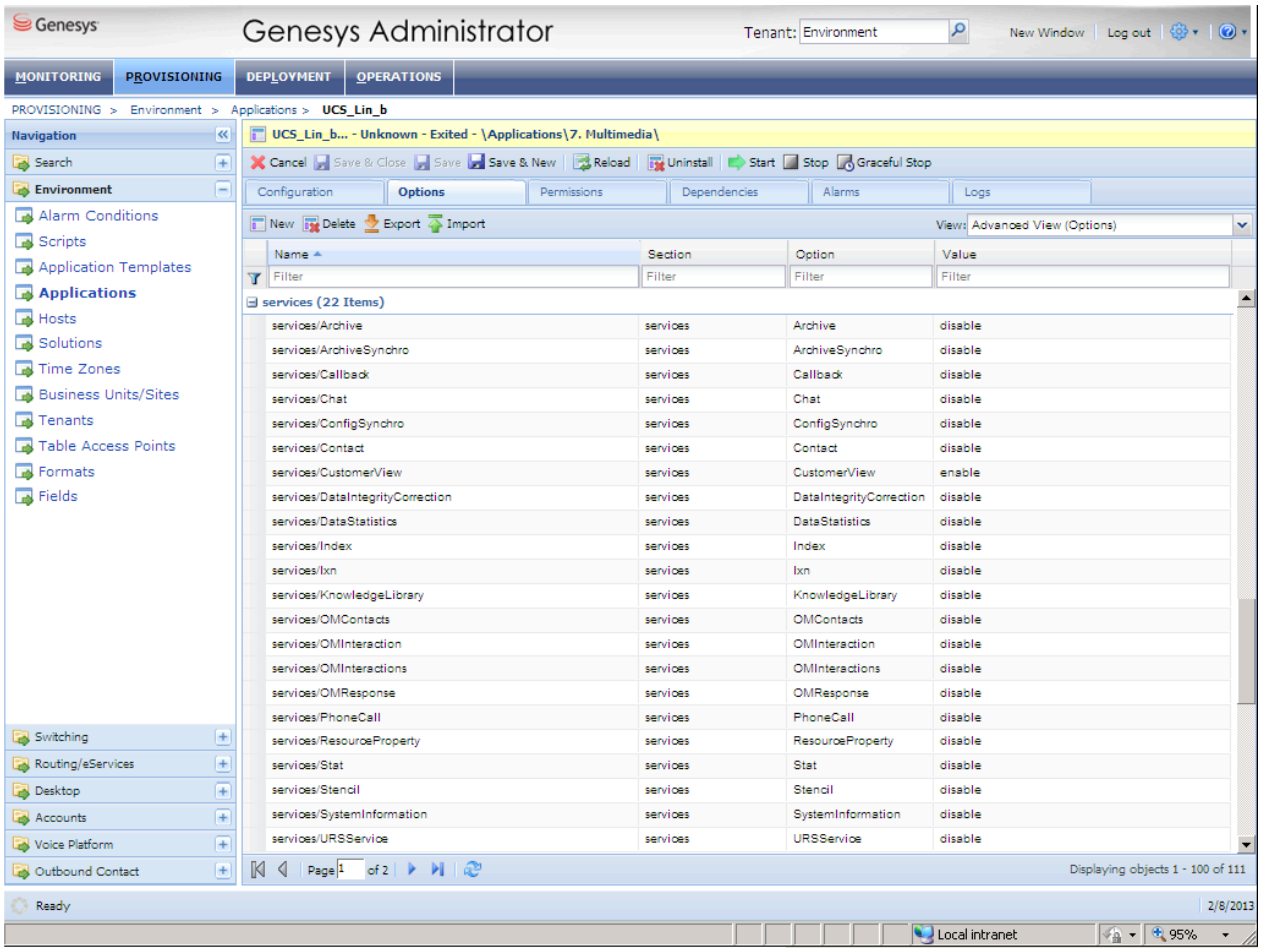
Purpose: To configure Context Services in the Configuration Layer.

Procedure

1. To disable the other services, set the following parameters in the `services` section of the `Options` tab of the Context Services UCS application:

```
SystemInformation=DISABLE
CustomerView=ENABLE
ConfigSynchro=DISABLE
Contact=DISABLE
OMContacts=DISABLE
Archive=DISABLE
ArchiveSynchro=DISABLE
Callback=DISABLE
Chat=DISABLE
DataIntegrityCorrection=DISABLE
DataStatistics=DISABLE
Index=DISABLE
Ixn=DISABLE
KnowledgeLibrary=DISABLE
OMInteraction=DISABLE
OMInteractions=DISABLE
OMResponse=DISABLE
PhoneCall=DISABLE
ResourceProperty=DISABLE
Stat=DISABLE
Stencil=DISABLE
URSService=DISABLE
```

   The following screen shot displays this configuration in Genesys Administrator.



> **Important**
>
> `CustomerView` is the only enabled parameter.

2. To disable Remote Method Invocation (RMI), set the `enable-rmi` parameter in the `settings` section on the `Options` tab of the of the Context Services UCS application to `false`.
3. To assign the RMI port, set the `ucsapi` parameter in the `ports` section on the `Options` tab of the Context Services UCS application to a correct port number. For example, `7550`.

> **Important**
>
> Even if RMI is not used, this port number must be assigned.

Multiple Context Services on a Single Database

When configuring multiple Context Services on the same database:

1. Start one Context Services with the option `ConfigSynchro` set to `ENABLE` and all other options set to `DISABLE`.
2. Configure all other Context Services instances as described previously, with the listed options set to `DISABLE`.

When in the customizing phase, you may have to change the configuration of the core Profile attributes in the Configuration Layer. In this case:

- One (and only one) instance must have the option `ConfigSynchro` set to `ENABLE`. Use this instance to update Profiles or Services metadata.
- After each change, reload the cache on each of other instances using /metadata/cache URI.

Configure the Apache Server

Purpose: To configure Apache HTTP load balancing for the UCS database.

Procedure

Start

1. In the <path to Apache>\conf\httpd.conf file, configure the Apache capacity:

```
<IfModule mpm_winnt_module>
    ThreadsPerChild      3350
    ThreadLimit 4000
    MaxRequestsPerChild    0
</IfModule>
```

> **Important**
>
> The module mpm_winnt_module is available for Windows only. For more information on the mpm_winnt_module or other modules for Unix/Linux, see your Apache's documentation.

2. In the <path to Apache>>\conf\httpd.conf file, enable (uncomment) the following lines:

```
LoadModule proxy_http_module /usr/lib/apache2/modules/mod_proxy_http.so
LoadModule proxy_module /usr/lib/apache2/modules/mod_proxy.so
LoadModule proxy_balancer_module /usr/lib/apache2/modules/mod_proxy_balancer.so
LoadModule proxy_ajp_module /usr/lib/apache2/modules/mod_proxy_ajp.so
LoadModule jk_module /usr/lib/apache2/modules/mod_jk.so
```

3. At the end of the <path to Apache>\conf\httpd.conf file, specify the path to mod_proxy.conf:
   include "<path to Apache>\conf\mod_proxy.conf"
4. In the <path to Apache>\conf\mod_proxy.conf file, enter the following:

```
<Location /someUrl/>
  # Turn on Proxy status reporting at /status
  # This should be better protected than: Allow from all
  ProxyStatus On
 <Location /status>
   SetHandler server-status
   Order Deny,Allow
   Allow from all
 </Location>
 ######### Proxy HTTP #########
 ProxyPass /cs/ balancer://cscluster/
 <Proxy balancer://cscluster>
 BalancerMember <nowiki>http://Context Services Server1:8182 loadfactor=1</nowiki>
 BalancerMember <nowiki>http://Context Services Server2:8485 loadfactor=1</nowiki>
 BalancerMember <nowiki>http://Context Services Server3:8283 status=+H</nowiki>
 BalancerMember <nowiki>http://Context Services Server4:8384 status=+H</nowiki>
 # status 'H' is hot standby
```

```
ProxySet lbmethod=byrequests
</Proxy>
</Location>
```

> **Important**
>
> Replace the above server names with your environment server values.

End

UCS with Context Services

Provides a general description of how UCS works with Context Services.

Archiving and Pruning the DB

This page describes maintenance of the UCS database.

Overview

To prevent your UCS database from expanding to an unmanageable size, you may wish to perform archiving and pruning.

- Archiving is the process of removing selected threads from the main database and storing them in the archive database.
- Pruning (sometimes also called purging) is the process of removing threads from either the main or the archive database.
- Maintenance is a cover term for pruning and archiving.

For both archiving and pruning you have a choice of two processes, as laid out in the following table.

Comparison of Maintenance Processes

Process	Configuration options (section/ option)	Speed	Complexity	Availability	Objects accessible
UCS Manager archiving	`cview/enabled = false`	Slower	Simple, can stop midway	All releases of UCS	Interactions only
UCS Manager pruning	`cview/enabled = false`	Slower	Simple, can stop midway	All releases of UCS	All
Set-based archiving	`cview/enabled = true archiving/ use-np = true`	Very fast	Several steps, cannot stop midway	UCS 8.0.2 and later	Interactions only
Prune using options	`cview/enabled = true`	Fast	Cannot stop midway	UCS 8.1.0 and later	All

Using UCS Manager Only

For all releases of UCS, you can use UCS Manager to configure and run the complete process of maintaining the UCS database, as described in the online Help that is delivered with UCS Manager. UCS Manager can also:

1. correct certain problems that may exist with data integrity, and
2. display statistics about the UCS database.

Set-Based Archiving

Beginning with release 8.0.2 of UCS, you can also use set-based archiving. One way to characterize the difference between the new set-based archiving and the existing archiving via UCS Manager only is that the former moves data table by table while the latter moves it interaction by interaction.

> **Important**
> Set-based archiving requires expertise in database management. Therefore it should be performed only by a qualified database administrator.

Prerequisites

Disk Space

Set-based archiving requires temporary space in the main database constituting about 90% of the space occupied by the archivable interactions. For example, if one million interactions, including 350,000 attachments, take up 10.2 GB in the main database, the temporary space needed is 9 GB.

User rights

UCS must create and drop tables during the archiving process. These rights must be granted to the UCS user in the main DB during the archiving process. Once this process is completed these rights can be revoked for normal operation of UCS. Consult your RDBMS documentation for directions on granting and revoking these rights.

The user will have to execute special queries to transfer data from temporary table to archive DB. These queries are particular to MS SQL Server and Oracle.

For Oracle, the user must be able to create and drop database links using the following queries:

```
create database link arch using 'ucsarch';

drop database link arch;
```

For MS SQL Server, the user must be able to execute the following stored procedure:

```
EXEC sp_addlinkedserver @server = N'suite801',
@srvproduct=N'SQL Server'
```

```
EXEC sp_dropserver 'suite801', null;
```

> **Important**
>
> The queries presented here describe the minimum needed to create the database link between the main and archive UCS databases. Depending on the configuration of your database, you may need to pass more parameters, such as `usernames`, `password`, `schemas`, `tablespaces`, and so on. Consult your RDBMS documentation for guidance.

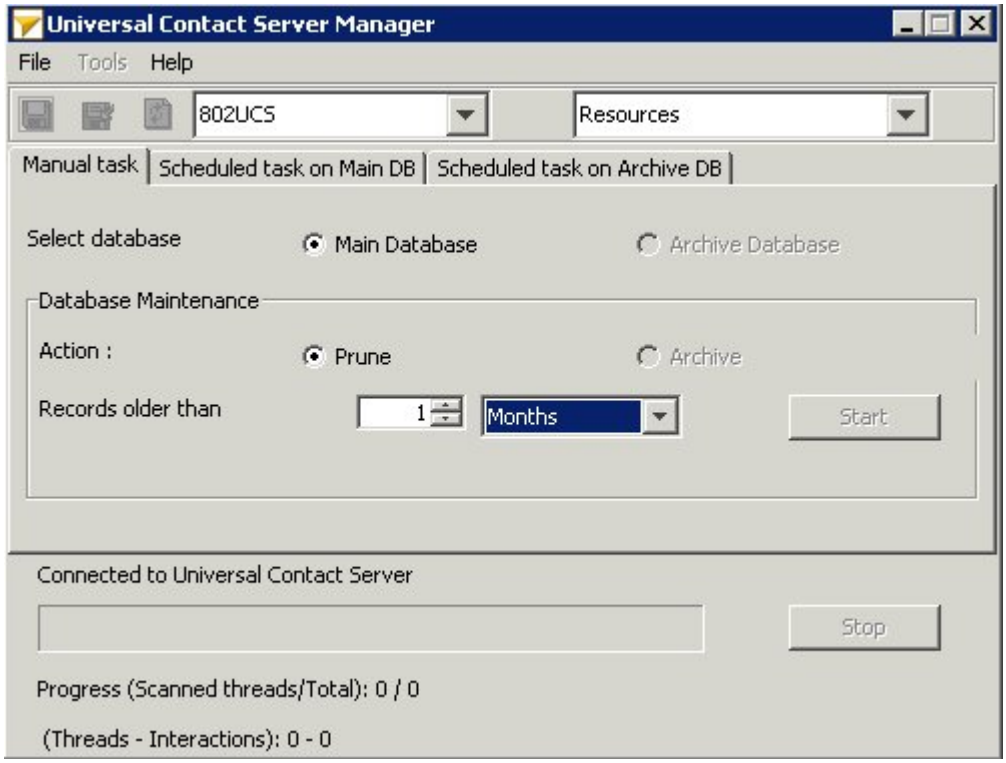
Configuration

Open your UCS Application object and:

- In the `cview` section, set the `enabled` option to `true`.
- Create a section called `archiving`. In it create an option called `use-np` with the value `true`.

Start the Archiving from UCS Manager

1. Open UCS Manager.

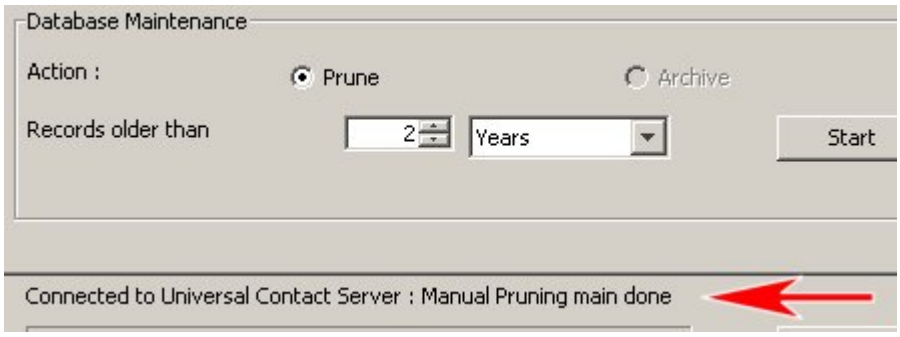


2. Select one of the following tabs:
   - `Manual task` for performing one-time maintenance

- `Scheduled task on Main DB` for scheduling periodic maintenance on the main database
- `Scheduled task on Archive DB` for scheduling periodic maintenance on the archive database

  Note: If you select either of the scheduled tasks, you must also be able to schedule the execution of the SQL queries described lower down on this page.

3. Click Start.
4. When UCS Manager displays `<task> <database> done`, in the area indicated by the red arrow in the screenshot below,
   - For pruning, stop here.
   - For archiving, proceed to the next section below.



Continue The Process

Continue by issuing the SQL queries described for the two supported RDBMs on these pages:

- Oracle databases
- MicroSoft SQL databases

Failure Recovery

The steps to recover from a failure depend on whether the failure occurred during the archiving process, or when data was being moved. Both possibilities are described below. **Failure During Archiving** If a failure occurs during archiving, the archiving process must be restarted from the beginning. To do so,

1. Stop UCS Manager and UCS.
2. Execute the following queries:

```
drop table docid_temp;
drop table ixnid_temp;
drop table interaction_arch;
drop table emailin_arch;
drop table emailout_arch;
drop table phonecall_arch;
drop table callback_arch;
drop table cobrowseurl_arch;
drop table chat_arch;
drop table attachment_arch;
```

```
drop table ixncontent_arch;
drop table ixnContentSentReceived_arch;
drop table document_arch;
```

3. Restart both UCS Manager and UCS, and restart the archiving process.

Failure During Data Movement If a failure occurs during data movement, roll back all movement operations. The archiving procedure does not need to be executed again. Just restart the "Transferring Data into UCS DB Archive" procedure, described here for Oracle and here for MS SQL.

Multiple Attachment of a Single Document

In order to save space, UCS re-uses the same document object in the database if it is attached multiple times to an interaction. This is, for example, the case when using Standard Responses with attachments, either for agent use or for automatic replies. Like archiving using UCS Manager alone, set-based archiving does not remove unused documents from the main database because it would require an SQL operation that could take several hours to execute on large databases. For the same reason, the archiving mechanism cannot check if a document has already been inserted into the archive database. If a certain document is used multiple times, insertion of the document object in the archive database will fail with a Primary Key Constraint Violation during the execution of the following query: Oracle:

```
insert into document@arch select * from document_arch;
```

Microsoft SQL Server:

```
insert into bsgenucsdb.UCSArch.dbo.document select * from document_arch;
```

There are two possible workarounds:

- Skip this operation and avoid copying documents into the archive database.
- Use database-specific commands to merge the data into the archive database. Consult your database documentation for instructions on executing this operation.

Limitations

Set-based archiving has the following limitations:

- DB2 is not supported.
- The progress indicators in UCS Manager do not function.
- As with archiving using UCS Manager only, set-based archiving does not remove documents from the main database. Use UCS Manager's Data Integrity Correction tool to remove the orphan documents.
- If you stop the archiving from UCS Manager, processing will stop only when the current operation is finished. Depending on the size of the database, this can take from minutes to hours.
- If you stop the archiving process, you must restart the process from the beginning, first ensuring that no temporary tables are left (see Failure During Archiving). Unlike the existing archiving using UCS Manager only, set-based archiving does not support resuming the process from the point that it stopped.
- If an error occurs at any level during archiving process, you must restart the process from the beginning, first removing any temporary tables.

- You must ensure that enough space is available in the main database before starting the process. If there is insufficient space the process will fail and must be started over.
- Pruning is supported on the main database only, not on the archive database.

Contact Identification

Purpose: Provides a high-level description of Context Services method of identifying customers, and contrasts it with the way that UCS (without CS) does so.

If either method produces a unique match for the incoming customer data, there is of course no problem. The differences become relevant when there are multiple matches or when there is no match.

Multiple Matches Found

If UCS tries to identify a customer, and receives more than one match in return:

- In UCS, there are various possibilities depending on the entity that requested the identification. For example, UCS selects the first customer in the returned list if it is responding to E-mail Server. A description of all possible scenarios can be found in the "Contact Identification and Creation" chapter of the eServices 8.0 User's Guide.
- In UCS/CS, you define arbitrary identification keys (such as email address, last_name + first_name, and so on). If you attempt to identify by email address, for example, and this maps to more than one customer, the application receives complete profiles for all matched customers. This gives the application the opportunity to disambiguate.

For example, the SCXML application may send the matched profiles to the IVR, which might prompt for the customer's name (with the grammar formed by taking the names from the matched profiles). More generally, the application will prompt for additional information and use other identification keys to further isolate the customer's identity. Once a given identity is assumed, the application will often use additional information (such as the customer's ZIP code) to validate the customer's identity. In this sense, UCS/ allows for the application to distinguish between assumed and validated customer identities.

No Matches Found

- In UCS, if a customer is not found on lookup, a new contact record is created. Again, this may or may not be correct.
- In UCS/CS, the application again has the opportunity to collect additional information and attempt to identify the customer using some other identification key. In the end, the application or the agent may separately decide to create a new customer/contact profile, but the decision to do this is completely application-specific.

> **Important**
> The preceding statements about how UCS (without Context Services) identifies and

creates contacts apply only to the default behavior of UCS. The "Contact Identification and Creation" chapter of the eServices 8.0 User’s Guide describes ways that you can customize this default behavior. However, what you can customize is limited to 1) the contact attributes that UCS checks and the order it checks them in, and 2) whether UCS creates a new contact in the event of no match, or if it does, a minimum set of attributes that must match. In neither case does it allow the application to expand the attributes that it checks, unlike UCS/CS.

Messaging, Modes, and Migration

Purpose: More on the basic operation of UCS

Messaging

Clients connect to UCS and send requests, to which UCS responds. Clients communicate with UCS via RESTful web services, using HTTP request methods that are based on the GET, POST, PUT, and DELETE methods. Clients of UCS/CS may include Orchestration Server, Genesys Voice Portal (GVP), agent desktops, or any third party application that makes use of real-time customer service information.

Modes

UCS has two modes of operation. Each message can be sent in only one mode.

- Production—The normal operating mode. UCS accepts incoming requests for querying/updating customer profiles and service-related data.
- Maintenance—For configuring the database and other operations; normally to be used only at times of low traffic. Use this mode to create extensions to the customer profile model, or to define identification keys. While in maintenance mode, the system does not process incoming requests for querying or updating customer profiles or service history.

Migration and Transition

For migration from versions 7.0 through 8.0.0 of UCS, see the *Genesys Migration Guide.* For versions previous to 7.0, there is no complete migration, but you can convert most of the UCS (then called Contact Server) database. The procedure is described in the "Transitioning to eServices from ICS 6.x" chapter in the *eServices 8.0 User's Guide.*

Set-based Archiving with MSSQL

Purpose: This page presents the SQL queries used for set-based maintenance of the UCS database on an MS SQL RDBMS.

Prerequisites

See Archiving and Pruning the DB for prerequisites. In particular, before using these queries you must first **run archiving from UCS Manager.**

Creating the Database Link

1. Be sure that the DNS name resolves properly to the archive database server. If not, you can add it to the host file; on Windows, for example, this is located at `C:\WINDOWS\system32\drivers\etc\hosts`.
2. To create the DB link execute the following command. Note that the command will return no error, even if a parameter is wrong or the destination host does not resolve correctly.

   ```
   EXEC sp_addlinkedserver @server = N'bsgenucsdbarch', @srvproduct=N'SQL Server'
   ```

   > **Important**
   >
   > The queries presented here describe the minimum needed to create the database link between the main and archive UCS databases. Depending on the configuration of your database, you may need to pass more parameters, such as `usernames`, `password`, `schemas`, `tablespaces`, and so on. Consult your RDBMS documentation for guidance.

3. To test if creation was successful, execute the following command:

   ```
   select count(*) from bsgenucsdbarch.UCSARCH.dbo.interaction;
   ```

   In this example,

   - `bsgenucsdbarch` is the destination host.
   - `UCSARCH` is the database.
   - dbo is the schema.

   Edit these names to match your configuration. Do the same in the queries provided in Moving the Data to the Archive Database below.
4. To drop the link, execute the following command:

   ```
   EXEC sp_dropserver 'bsgenucsdbarch', null;
   ```

   The link can be kept permanently and will not affect UCS operations. But when the link is no longer used, you may wish to drop it for security concerns.

Moving the Data to the Archive Database

1. Use the following commands:

```
insert into bsgenucsdb.UCSArch.dbo.interaction select * from interaction_arch;
 insert into bsgenucsdb.UCSArch.dbo.emailin select * from emailin_arch;
 insert into bsgenucsdb.UCSArch.dbo.emailout select * from emailout_arch;
 insert into bsgenucsdb.UCSArch.dbo.phonecall select * from phonecall_arch;
 insert into bsgenucsdb.UCSArch.dbo.callback select * from callback_arch;
 insert into bsgenucsdb.UCSArch.dbo.chat select * from chat_arch;
 insert into bsgenucsdb.UCSArch.dbo.ixncontent select * from ixncontent_arch;
 insert into bsgenucsdb.UCSArch.dbo.ixnContentSentReceived select * from
 ixnContentSentReceived_arch;
 insert into bsgenucsdb.UCSArch.dbo.document select * from document_arch;
 insert into bsgenucsdb.UCSArch.dbo.cobrowseurl select * from cobrowseurl_arch;
 insert into bsgenucsdb.UCSArch.dbo.attachment select * from attachment_arch;
```

2. If the move is successful, the temporary tables can be dropped:

```
drop table interaction_arch;
 drop table emailin_arch;
 drop table emailout_arch;
 drop table phonecall_arch;
 drop table callback_arch;
 drop table cobrowseurl_arch;
 drop table chat_arch;
 drop table attachment_arch;
 drop table ixncontent_arch;
 drop table ixnContentSentReceived_arch;
 drop table document_arch;
```

End

Archiving is now complete. Return to Archiving and Pruning the DB for descriptions of limitations and failure recovery methods.

Set-based Archiving with Oracle

Purpose: This page presents the SQL queries used for set-based maintenance of the UCS database on an Oracle RDBMS.

Prerequisites

See Archiving and Pruning the DB for prerequisites. In particular, before using these queries you must first **run archiving from UCS Manager.**

Creating the Database Link

1. The `tnsname.ora` file must refer to the destination database host in order to enable database link creation. The file must do this even if the destination database is on the same server as the main database. Below is an example `tnsname` file:

```
# tnsnames.ora Network Configuration File: D:\app\Administrator\product\11.1.0\db_1\
network\admin\tnsnames.ora
# Generated by Oracle configuration tools.
UCS =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = bsgenuscdb.emea.lucent.com)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = UCS)
    )
  )
UCSARCH =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = bsgenuscdbarch.emea.lucent.com)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = UCSArch)
    )
  )
```

   In this example, the UCS entry is for the main database and UCSARCH is for the archive database. Note the following:

   (1) These names must match the names of the databases used. (2) Because is no need for a link from archive to main, you do not have to modify the `tnsnames.ora` on the archive database machine.

2. Ensure that the destination host is reachable from the main machine by pinging the destination host from the main machine.
3. Once the `tnsname` file is properly configured, the execution of the following SQL command will create the DB Link. Note that you will receive no error message even if the `tnsnames.ora` file or the parameters of `ucsarch` are incorrect.

```
create database link arch using 'ucsarch';
```

   Replace `ucsarch` with the name that you configured in the `tnsnames.ora` file.

> **Important**
>
> The queries presented here describe the minimum needed to create the database link between the main and archive UCS databases. Depending on the configuration of your database, you may need to pass more parameters, such as `usernames, password, schemas, tablespaces`, and so on. Consult your RDBMS documentation for guidance.

4. To test the link, execute the following command, which lists the structure of the interaction table in the archive database:

```
desc interaction@arch;
```

5. To drop the link, execute the following command:

```
drop database link arch;
```

> **Important**
>
> Database links persist through restarts.

Moving the Data to the Archive Database

1. Use the following commands:

```
create database link arch using 'ucsarch';
 insert into interaction@arch select * from interaction_arch;
 insert into emailin@arch select * from emailin_arch;
 insert into emailout@arch select * from emailout_arch;
 insert into phonecall@arch select * from phonecall_arch;
 insert into callback@arch select * from callback_arch;
 insert into chat@arch select * from chat_arch;
 insert into ixncontent@arch select * from ixncontent_arch;
 insert into ixnContentSentReceived@arch select * from ixnContentSentReceived_arch;
 insert into document@arch select * from document_arch;
 insert into cobrowseurl@arch select * from cobrowseurl_arch;
 insert into attachment@arch select * from attachment_arch;
 drop database link arch;
```

2. If the move is successful, the temporary tables can be dropped:

```
drop table interaction_arch;
 drop table emailin_arch;
 drop table emailout_arch;
 drop table phonecall_arch;
 drop table callback_arch;
 drop table cobrowseurl_arch;
 drop table chat_arch;
 drop table attachment_arch;
 drop table ixncontent_arch;
 drop table ixnContentSentReceived_arch;
 drop table document_arch;
```

End

Archiving is now complete. Return to Archiving and Pruning the DB for descriptions of limitations and failure recovery methods.

Configuring CS for UCS Compatibility

> **Important**
> Your application can keep using 8.1 queries, even if you upgrade to 8.5, but you should not use deprecated methods. Make sure to read the developer page about the 8.5 changes.

Configuring the new Context Services URL for UCS

Purpose

To enable your UCS application to run concurrently with your GMS application.

1. Open the Configuration Manager or the Genesys Administrator, and edit your UCS application.
2. Set the following option values for the `cview` section:
   - Set `base-url` to `/genesys/1/cs`
   - Set `data-validation` to `false`
   - Set `enabled` to `true`
   - Set `metadata-cache` to `true`
   - Set `start-mode` to `production`
   - Set `tenant-id` to the same tenant ID than your GMS application.

Setting the Proxy for UCS Profiles

If you install a Load Balancer for UCS and GMS applications, you must set up redirections:

- All `/genesys/1/cs/profiles` URLs should be redirected to UCS.

```
/profiles
/metadata/profiles
/metadata/identification-keys
/server
/interactions
```

- Other `/genesys/1/cs/*` URLs should be redirected to GMS.

```
/services
/metadata/business-attributes
```

`/customers`

Additionally, you must:

- Configure Genesys Composer to point to the Load Balancer in **Window > Preferences > Composer > Context Services**.
- Configure the Genesys Orchestration application to point to the Load Balancer in **Genesys Administrator > Provisioning > Routing/eServices > Orchestration > <your application> > Context Manager Parameters**.

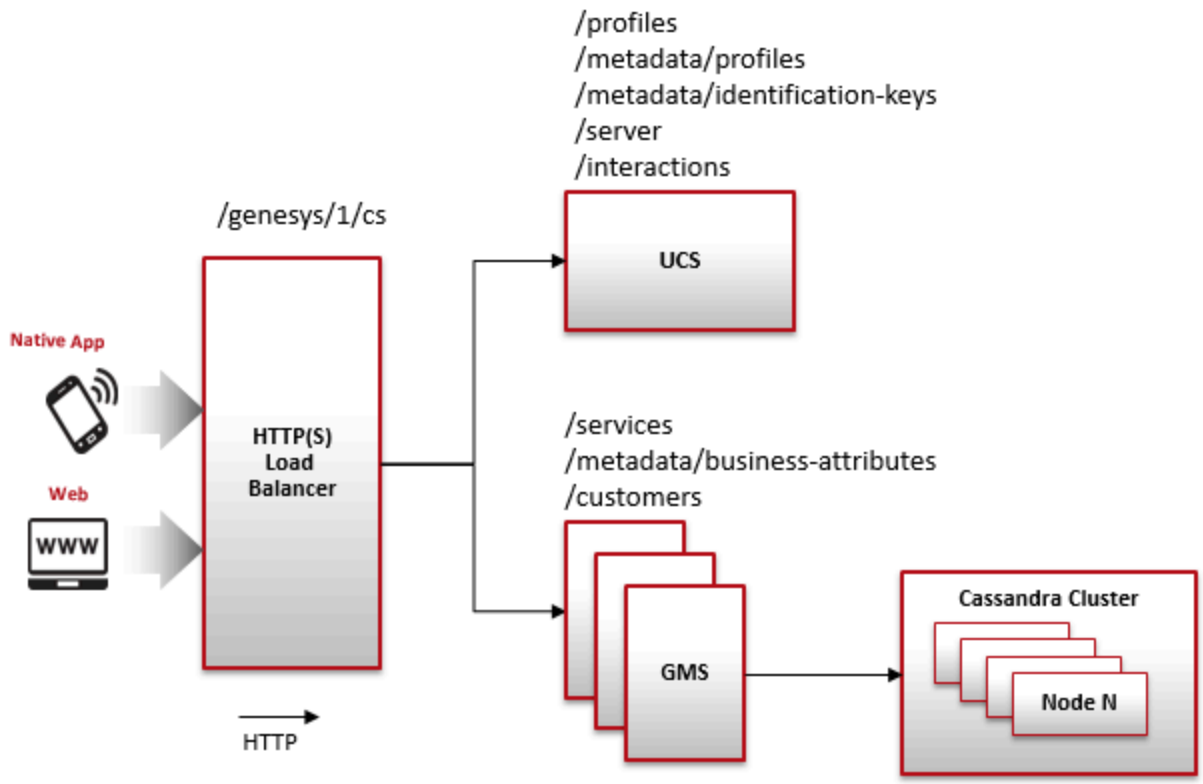
The following configuration examples should help you to manage URLs and redirections.

> **Important**
> Do not forget to restart your proxy after you saved your changes.

<tabber> NGINX =

NGINX Example

If your application uses NGINX, edit the NGINX configuration file, available in the `<NGINX INSTALLATION DIR>/conf` directory and add the `/genesys/1/cs` as the new base URL.



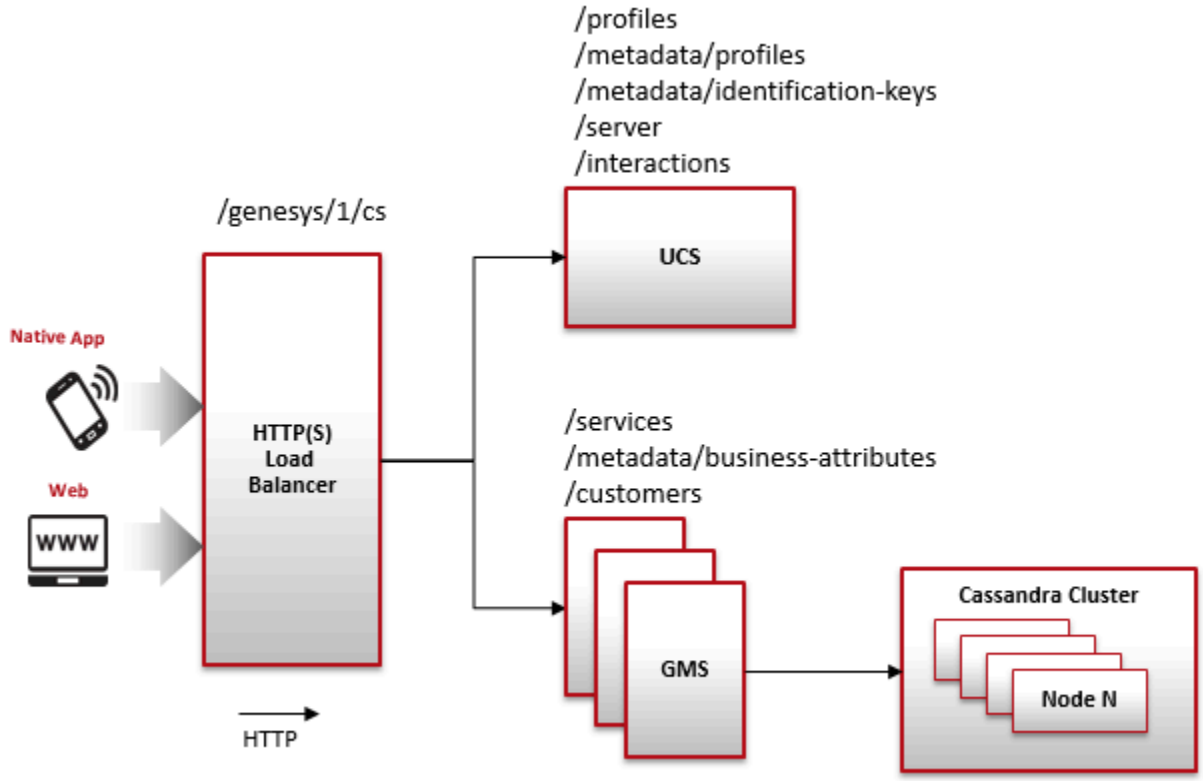

```
worker_processes  1;

events {
    worker_connections  1024;
}

http {
    include       mime.types;
    default_type  application/octet-stream;
    sendfile      on;
    keepalive_timeout  65;

    server {
        listen       3080;
        server_name  localhost;

                # let's gms_user header in request to pass
                underscores_in_headers on;

                ######################################
                # If profiles, interactions or /server -> goto UCS
                # /profiles
                # /metadata/profiles
                # /metadata/identification-keys
                # /server
                # /interactions
                ######################################
                location ~ (interactions|profiles|identification-keys|server) {
                        proxy_pass http://localhost:7580;
                }
```

```
                ####################################
                # If base_url (default) -> goto GMS
                # /services
                # /metadata/business-attributes
                # /customers/${customer id}/services
                ####################################
                location /genesys {
                        proxy_pass  http://localhost:8080/genesys;
                }
    }
}
```

|-| Apache=

Apache

If your application uses Apache, edit the `httpd.conf` file (or alternate file) and implement the new base URL for Context Service, as follows:




```
LoadModule headers_module modules/mod_headers.so
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule rewrite_module modules/mod_rewrite.so

# CORS headers
Header set Access-Control-Allow-Origin *
```

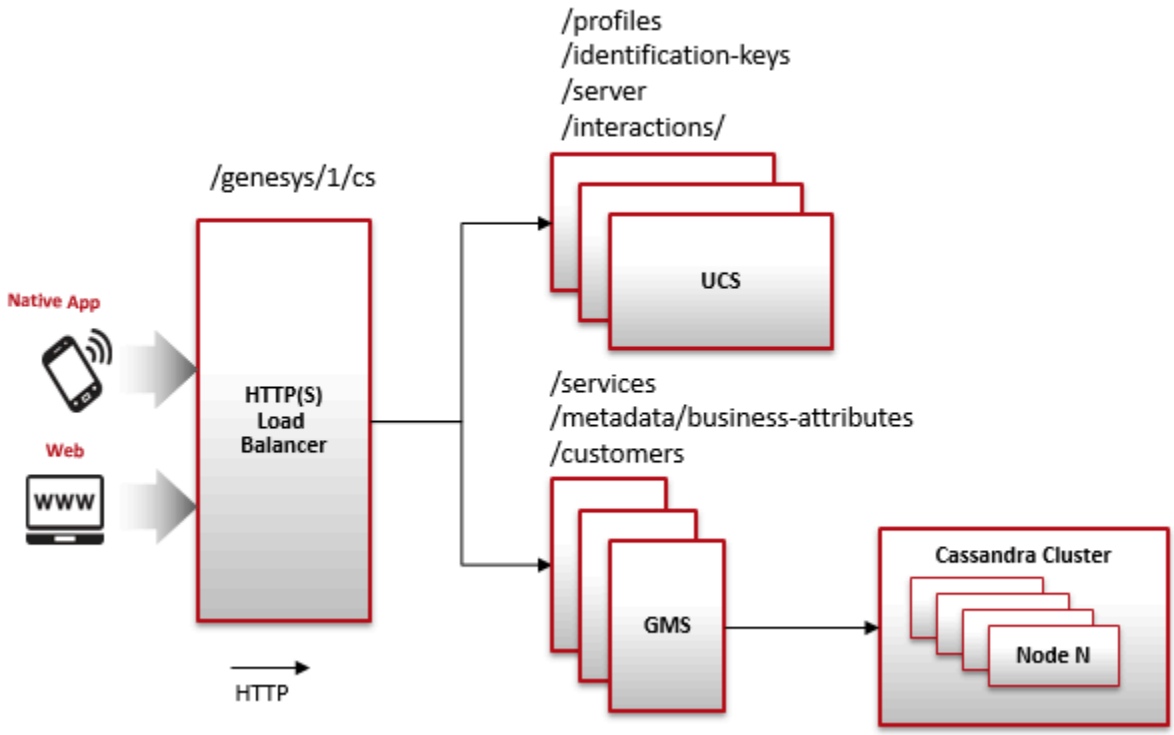
```
Header set Access-Control-Allow-Credentials true
Header set Access-Control-Allow-Headers "Origin, Content-Type, Authorization, Destination"
Header set Access-Control-Allow-Methods "GET, POST, OPTIONS, DELETE"
Header set Access-Control-Request-Headers "Origin, Content-Type"
Header set Access-Control-Max-Age 3600

# proxy to CS
ProxyPass /genesys/1/cs/profiles http://localhost:7580/genesys/1/cs/profiles
ProxyPass /genesys/1/cs/metadata/profiles http://localhost:7580/genesys/1/cs/metadata/profiles
ProxyPass /genesys/1/cs/metadata/identification-keys http://localhost:7580/genesys/1/cs/
metadata/identification-keys
ProxyPass /genesys/1/cs/server http://localhost:7580/genesys/1/cs/server
ProxyPass /genesys/1/cs/interactions http://localhost:7580/genesys/1/cs/interactions

# proxy to CS
ProxyPass /genesys http://localhost:8080/genesys
ProxyPassReverse /genesys http://localhost:8080/genesys
```

|-|

F5=

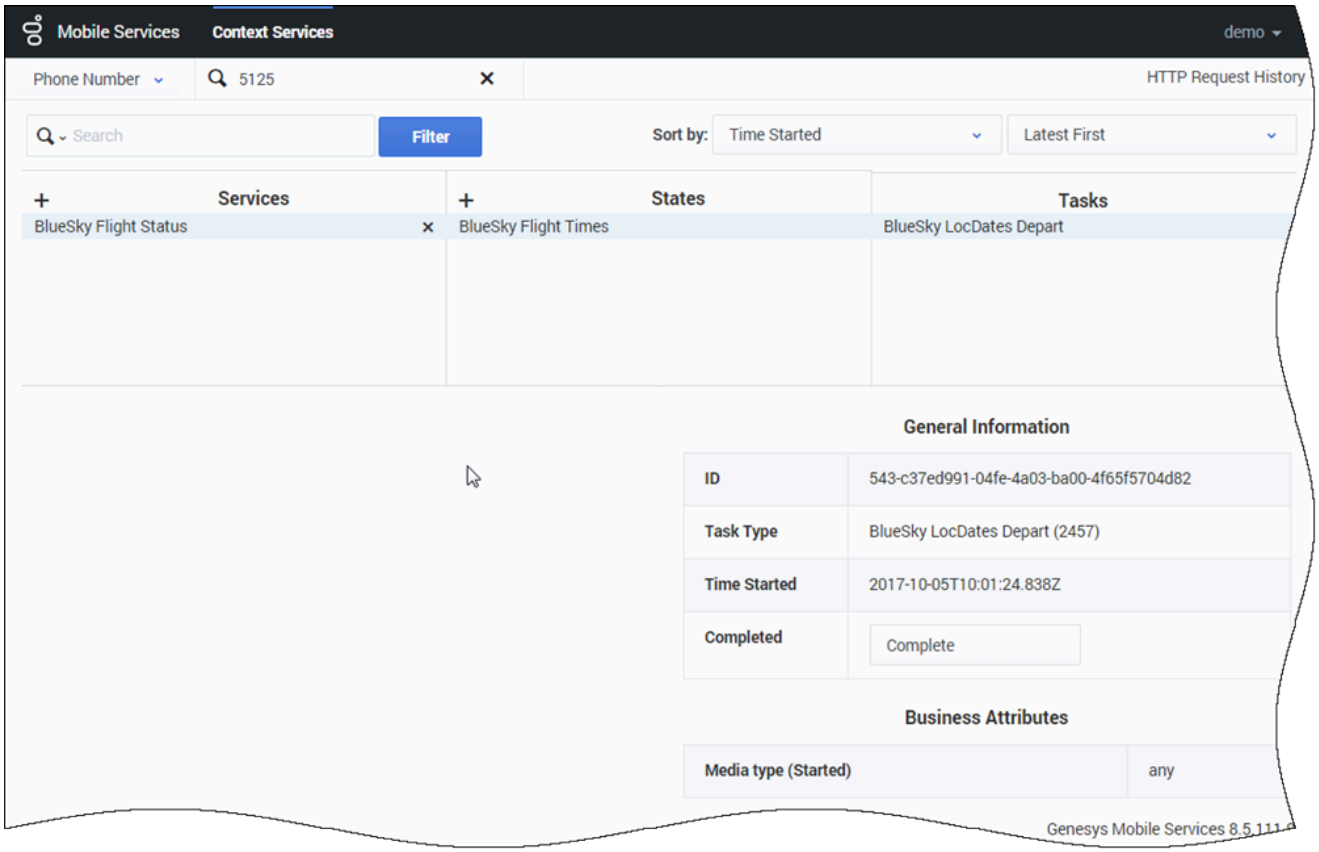
F5

Edit your configuration file and implement the new base URL for Context Service, with two pools, as follows:



```
LOAD BALANCER SITE 1:  Interface name <LB Site A Interface Name>

when HTTP_REQUEST {
    if { [HTTP::uri] starts_with "/genesys/1/cs/interactions"
      or [HTTP::uri] starts_with "/genesys/1/cs/profiles"
      or [HTTP::uri] starts_with "/genesys/1/cs/identification-keys"
      or [HTTP::uri] starts_with "/genesys/1/cs/server"}{
        pool <UCS pool A side name>
    }
    else {
        pool <GMS pool A side name>
    }
}

LOAD BALANCER SITE 2:  Interface name <LB Site A Interface Name>

when HTTP_REQUEST {
    if { [HTTP::uri] starts_with "/genesys/1/cs/interactions"
      or [HTTP::uri] starts_with "/genesys/1/cs/profiles"
      or [HTTP::uri] starts_with "/genesys/1/cs/identification-keys"
      or [HTTP::uri] starts_with "/genesys/1/cs/server"}{
        pool <UCS pool B side name>
    }
    else {
        pool <GMS pool B side name>
    }
}
```

Context Services Interface



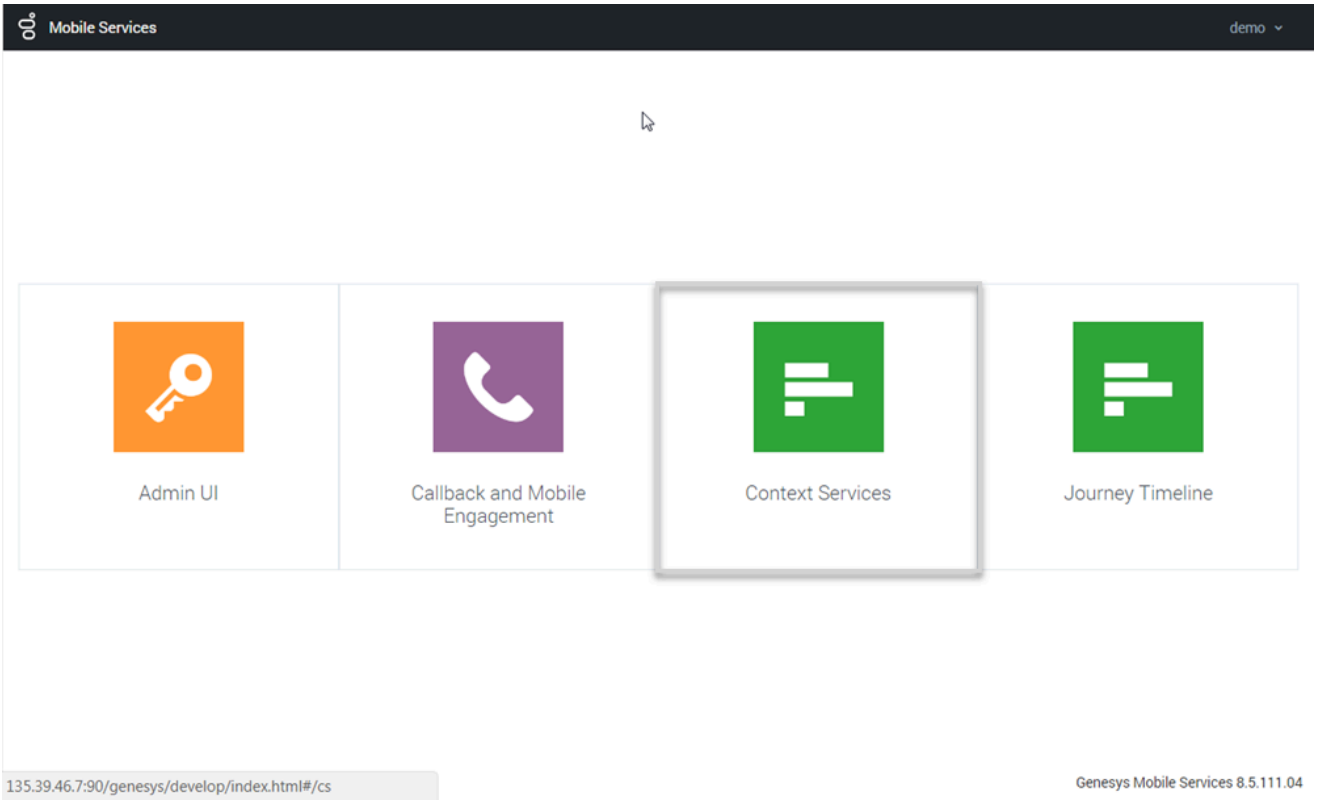
The Context Services Interface is a web-based interface that enables edition of Context Services data. This interface is intended to be used by developers and supervisors looking for detailed information about services because it is built to search for profiles, services, states, and tasks based on ID information or UCS information. It does not include all the search abilities that are available in typical agent interfaces.

This interface also enables you to modify or delete a given service.

Important

1. The Context Services Interface is available only for single-tenant installations.
2. If you change business attribute values in your configuration, users will need to refresh the Context Services Interface to see the changes taken into account.

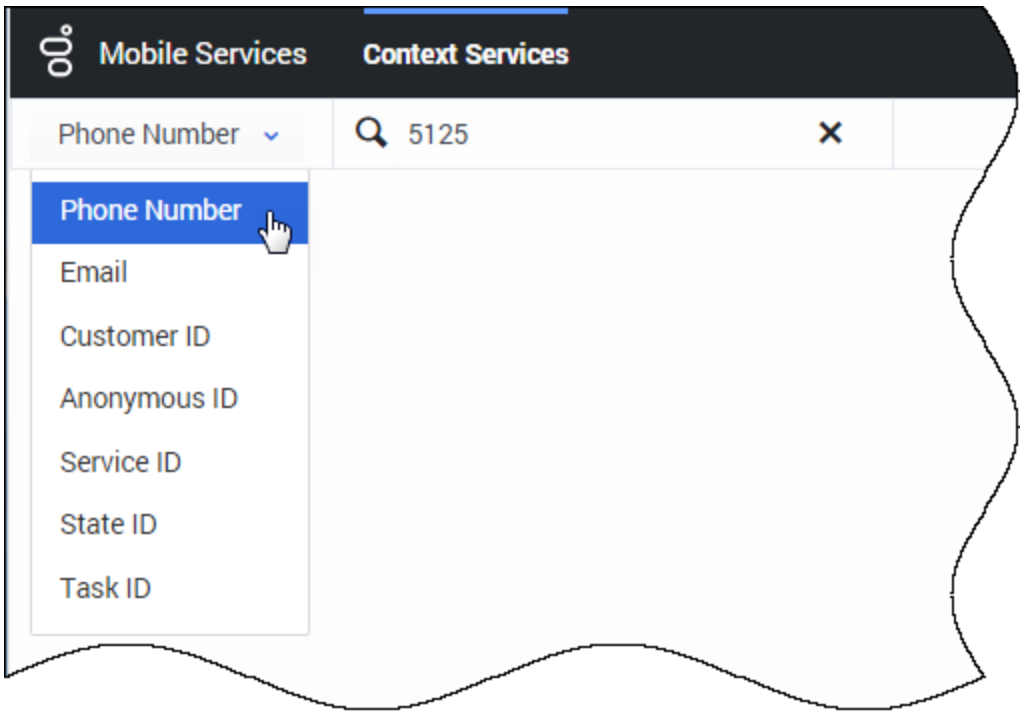
Login Panel



The Context Services Interface is available as part of the GMS Service Management User interface (detailed in the Service Management Help).

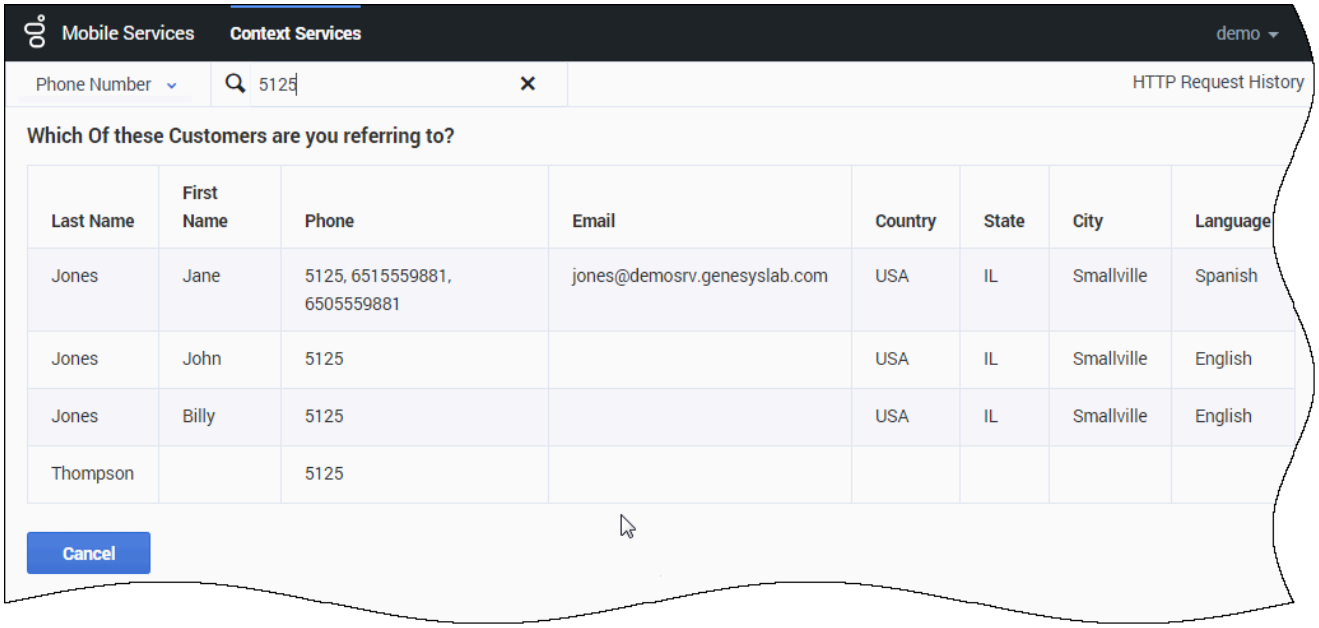
- To access this interface, you must login as a user who owns the Administrator or Supervisor role.
- Then, you can select the `Context Services` icon.

Search for Services



First, you start by searching for services or customers in the Context Services panel. You can search for UCS keys or Service, State, and Task IDs. These fields must be identical to a key in the UCS database to work correctly. There is no automatic completion available.

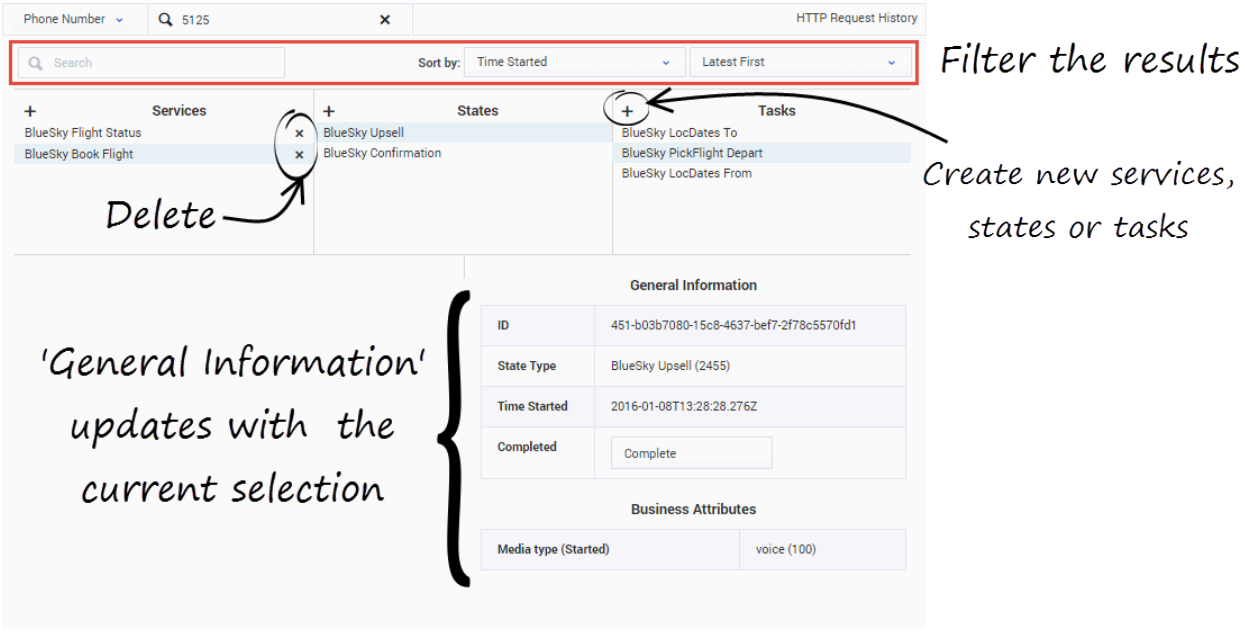
Select a key in the Search drop-down menu, then enter a value in the Search text box. The value must match a UCS entry to provide a result.



The interface displays a list of results. Select an item in the list.

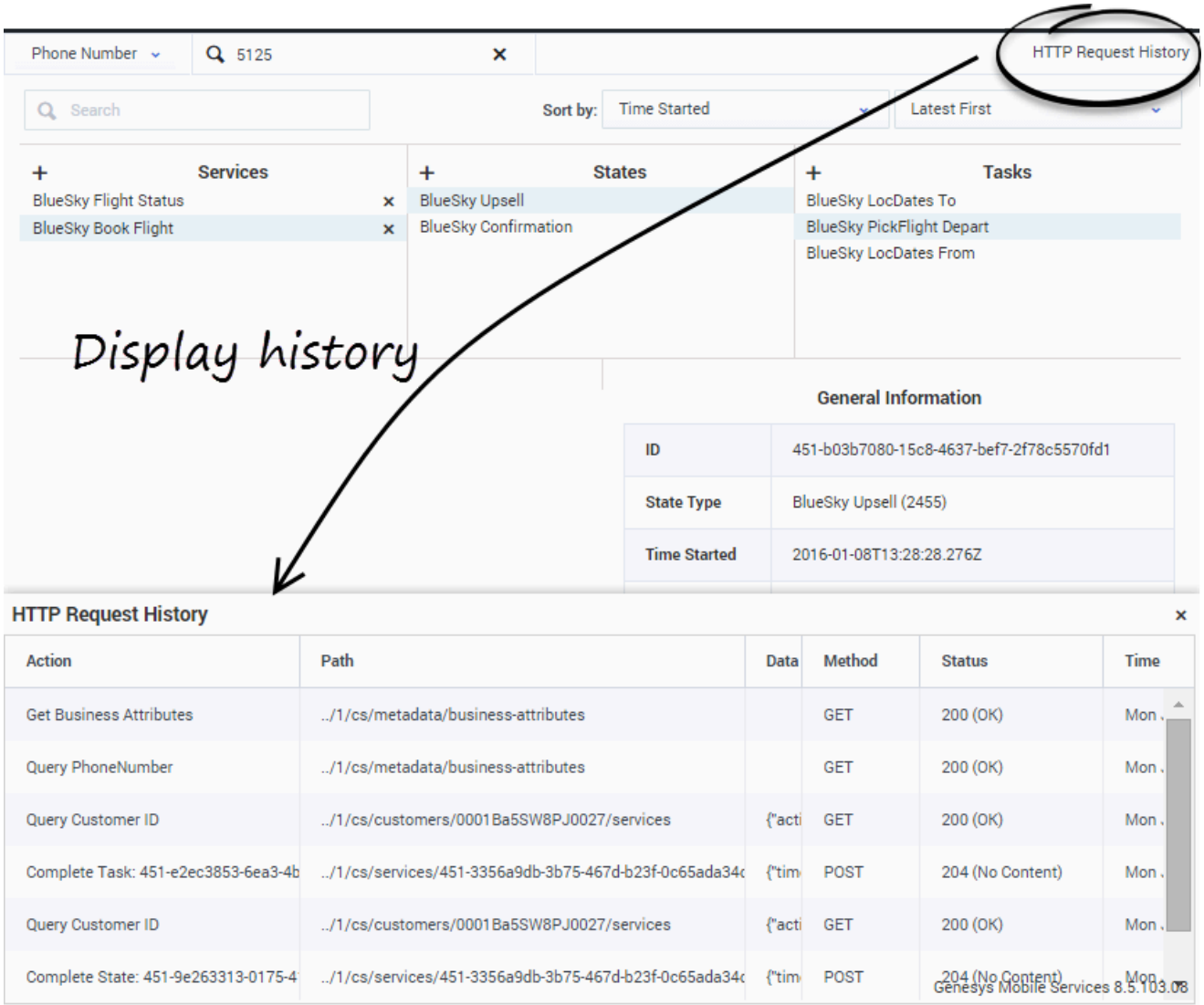
You can then use the interface to modify the service.

Manage your Services



The interface lets you manage the list of objects that you selected. If you selected a customer instance, you get the complete list of objects associated with the Customer ID.

- You can use the sorting tools to change the list displayed.
- You can select an item in the list, and get more details about the object.
- You can delete an object by clicking the 'x' icons.
- You can use the Action menu to perform more actions, such as creating new services, states, tasks.
- You can add services, states, and tasks by clicking the '+' icons.

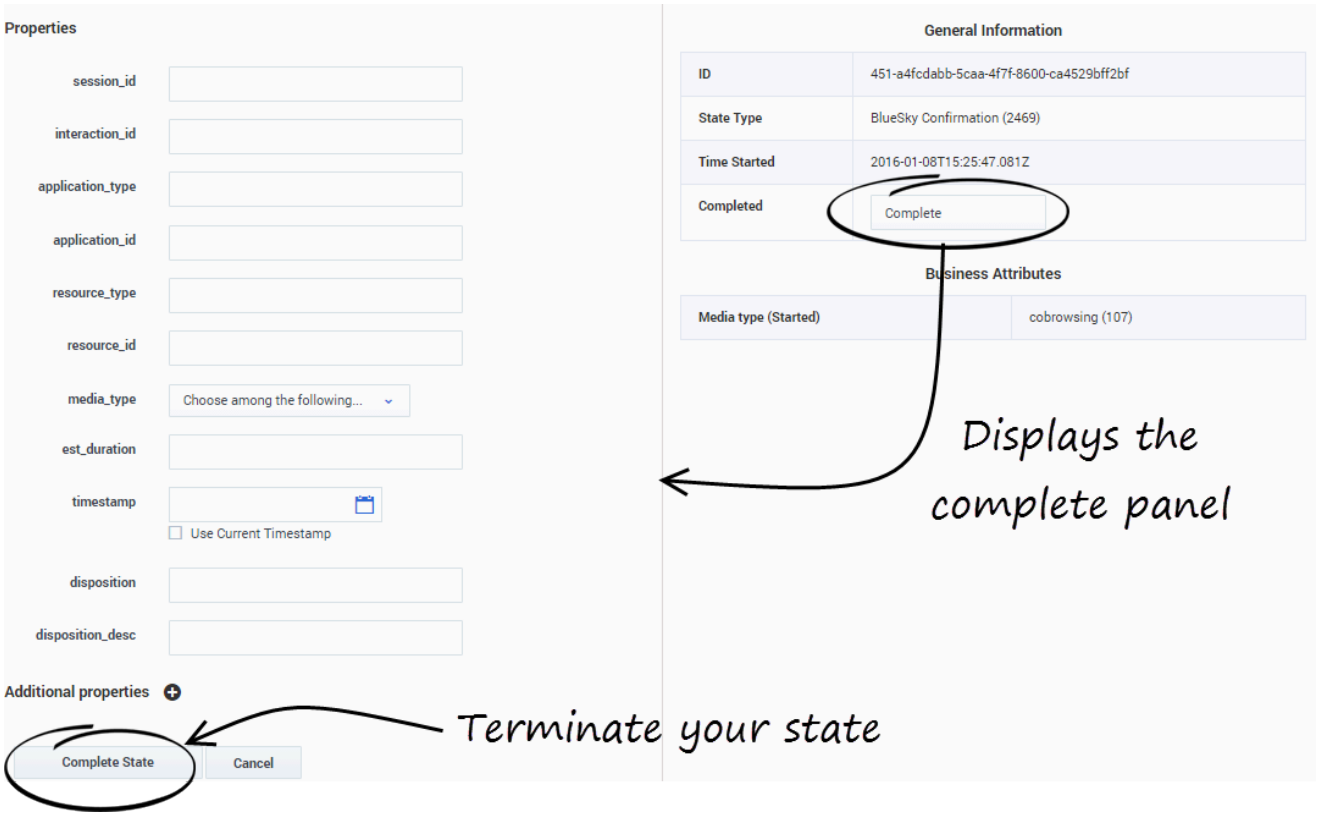


- You can get a history view of your queries by clicking **HTTP Request History**.

Create a Service, State, or Task

You can add services, states, and tasks by clicking the '+' icons. To create states or tasks, first, select a service or a state, then choose a type. When you create a new resource, you can fill it and even add some extension data.

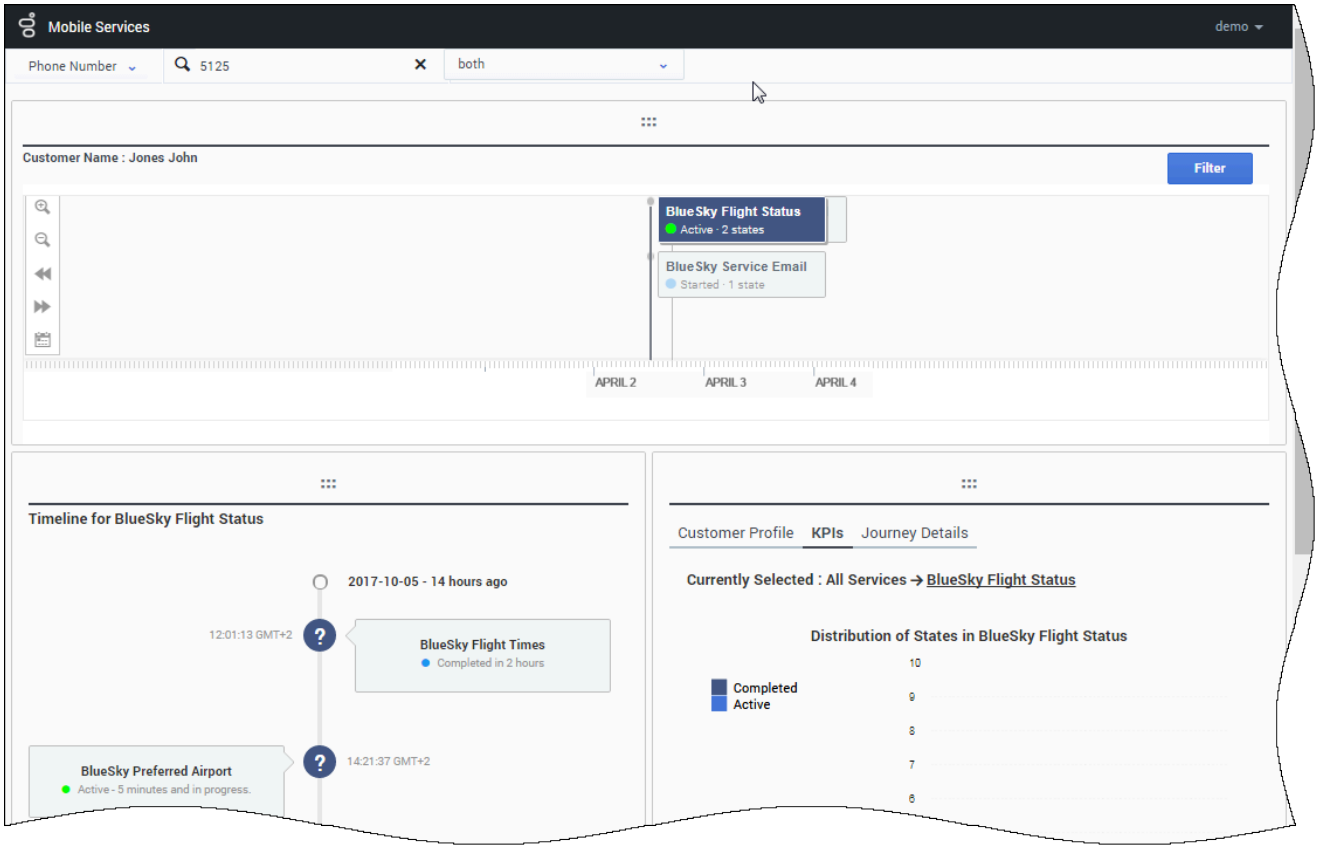
Complete a Service, State, or Task



If you can complete a Service, State, or Task, the interface displays a Complete button, that opens the Completion panel.

Journey Timeline Interface

> **Important**
> Starting in 8.5.103.16, you must enable profiles in UCS to access this interface.



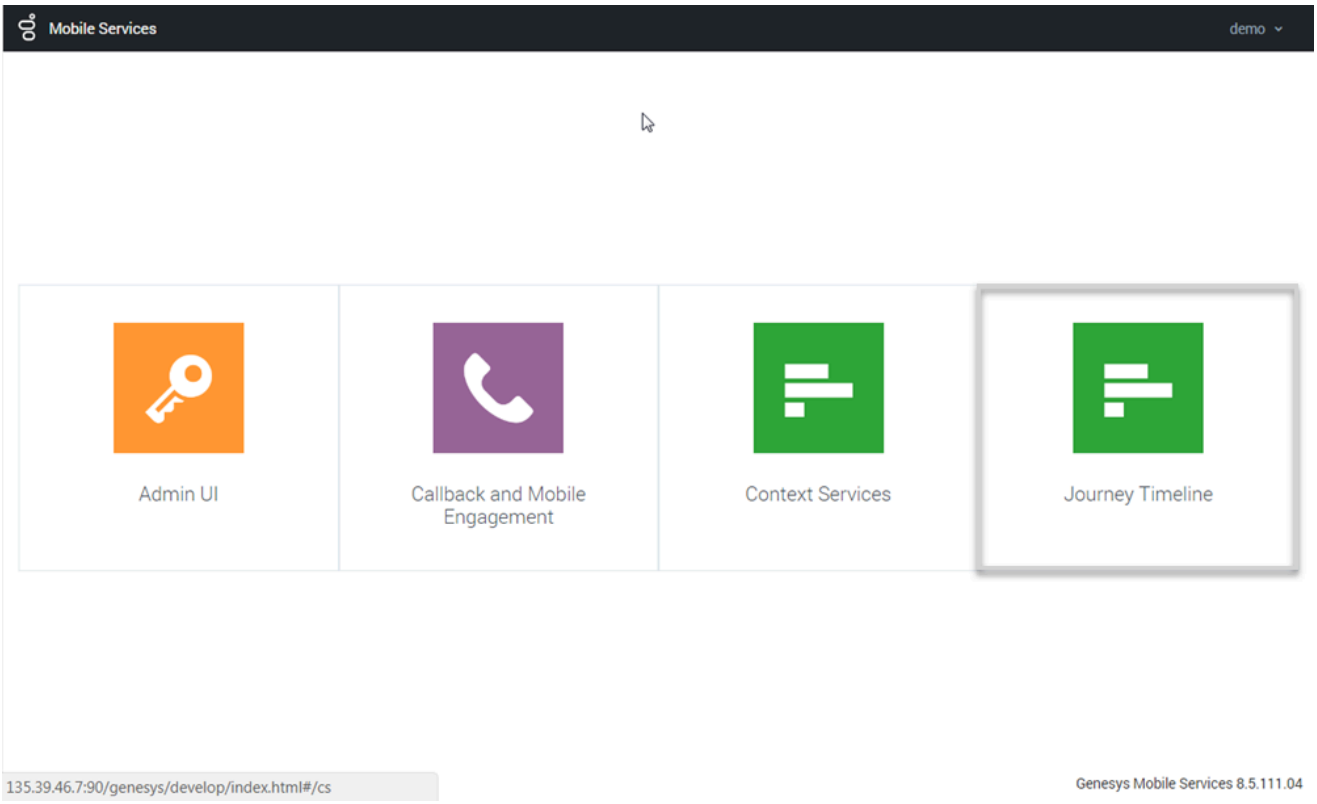
The Journey Timeline is a web-based interface that provides a visualization of Context Services data. This interface is intended to be used by developers and supervisors looking for detailed information about a specific customer because it is built to search for profiles, services, states, and tasks based on ID information or UCS information. It does not include all the search abilities that are available in typical agent interfaces.

Starting in 8.5.103, you can customize this interface.

> **Important**

> The Journey Timeline is available only for single-tenant installations.

Login



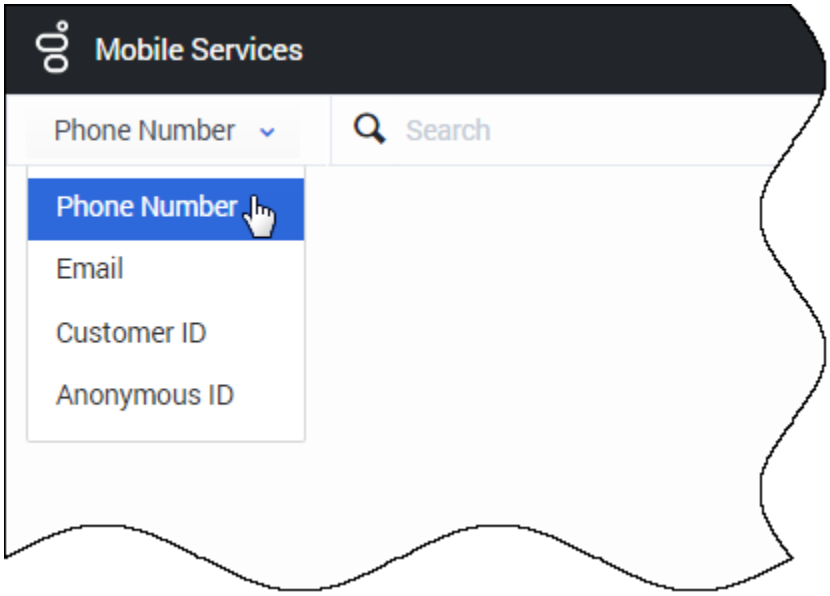
The Customer Journey Timeline is available as part of the GMS Service Management User interface (you can read help information here). To access this interface, you must login as a user who owns the Administrator or Supervisor privilege.

Then, you can select the `Journey Timeline` item and:

1. Search and select a customer.
2. Visualize the customer's Journey Timeline.

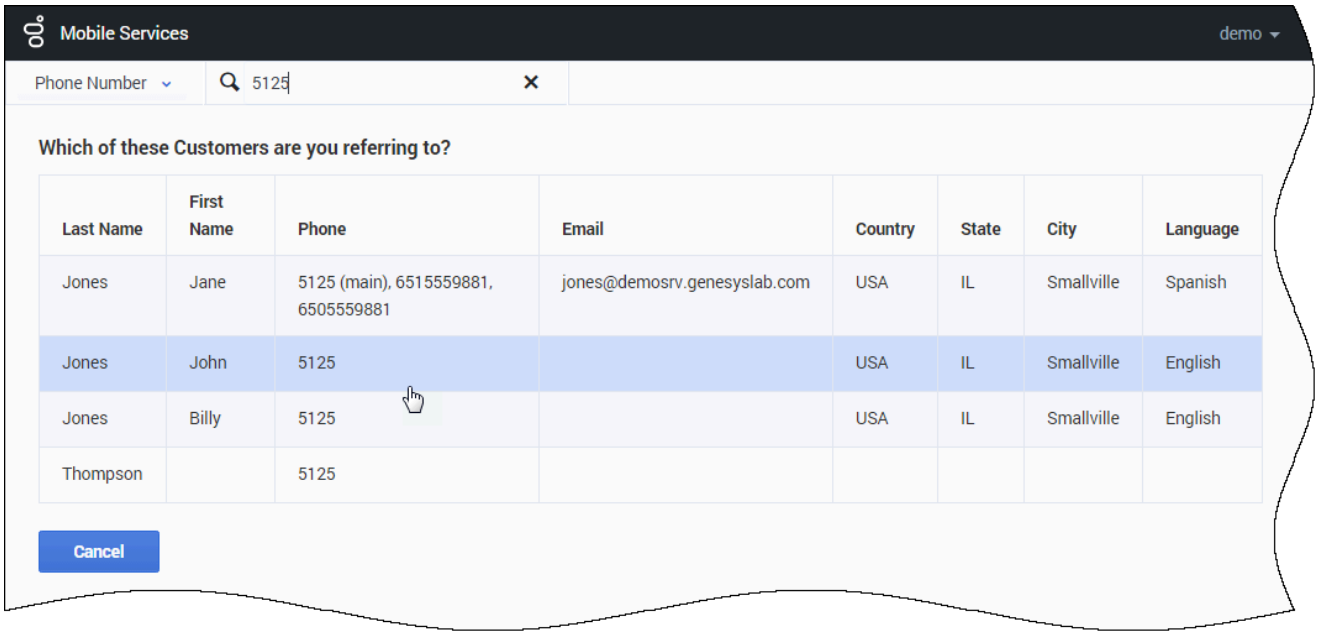
> ### Important
> If you do not see the Journey Timeline item, it means that you did not enable Context Services properly.

Search a Customer with Customer Journey



You can query a user based on the email address, phone number, and name fields. These fields must match a value in the UCS database to work correctly. There is no automatic completion available.

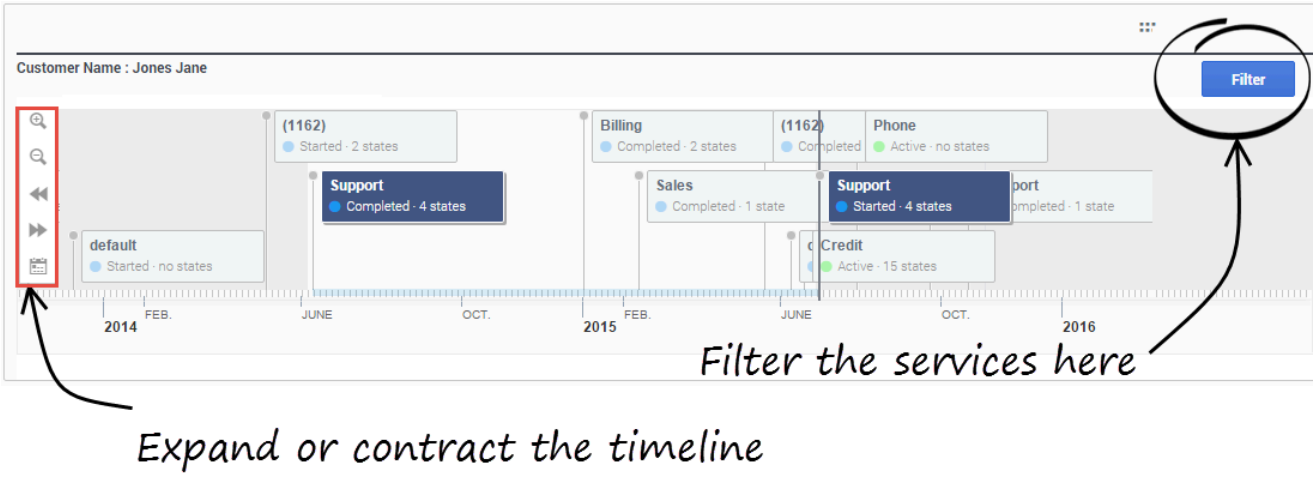
Select a key in the search drop-down menu, then enter a value in the Search textbox. The value must match a UCS entry to get a result.



The interface displays a list of results. Select a customer in the list to display the customer's timeline.

> **Important**
>
> You can customize the searched items through JSON configuration. See Customizing Profiles.

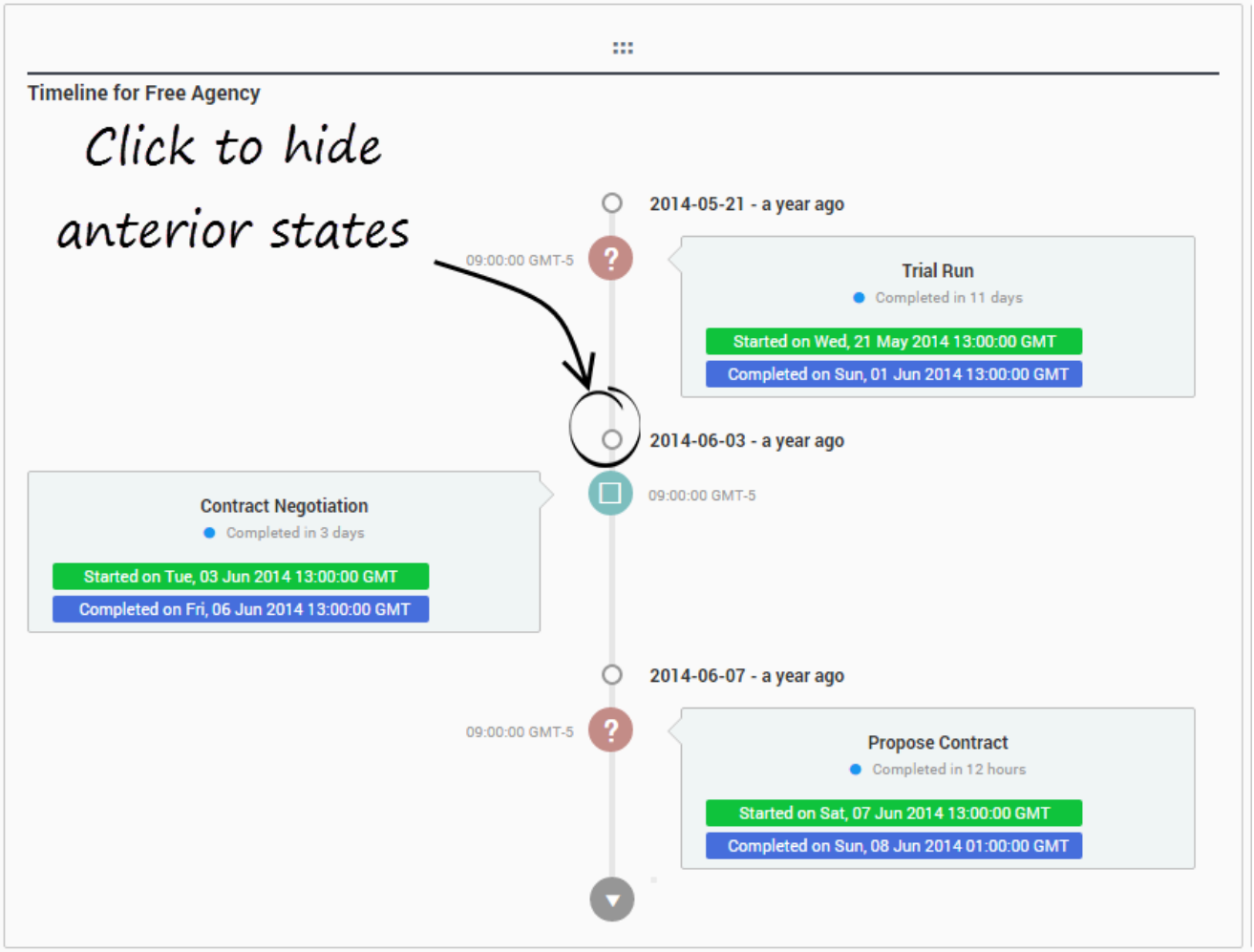
Manage the Timeline



The timeline shows all the customer's services and their current status (active, inactive). If you select a service, Customer Journey displays the list of states for the given selection.

- You can manage the timeline (expand or contract) by using the icons in the left menu sidebar.
- You can zoom or navigate to services by using the icons in the left menu sidebar, or you can simply left-click in the timeline to move it.
- You can also filter the displayed services.

Display States and Tasks

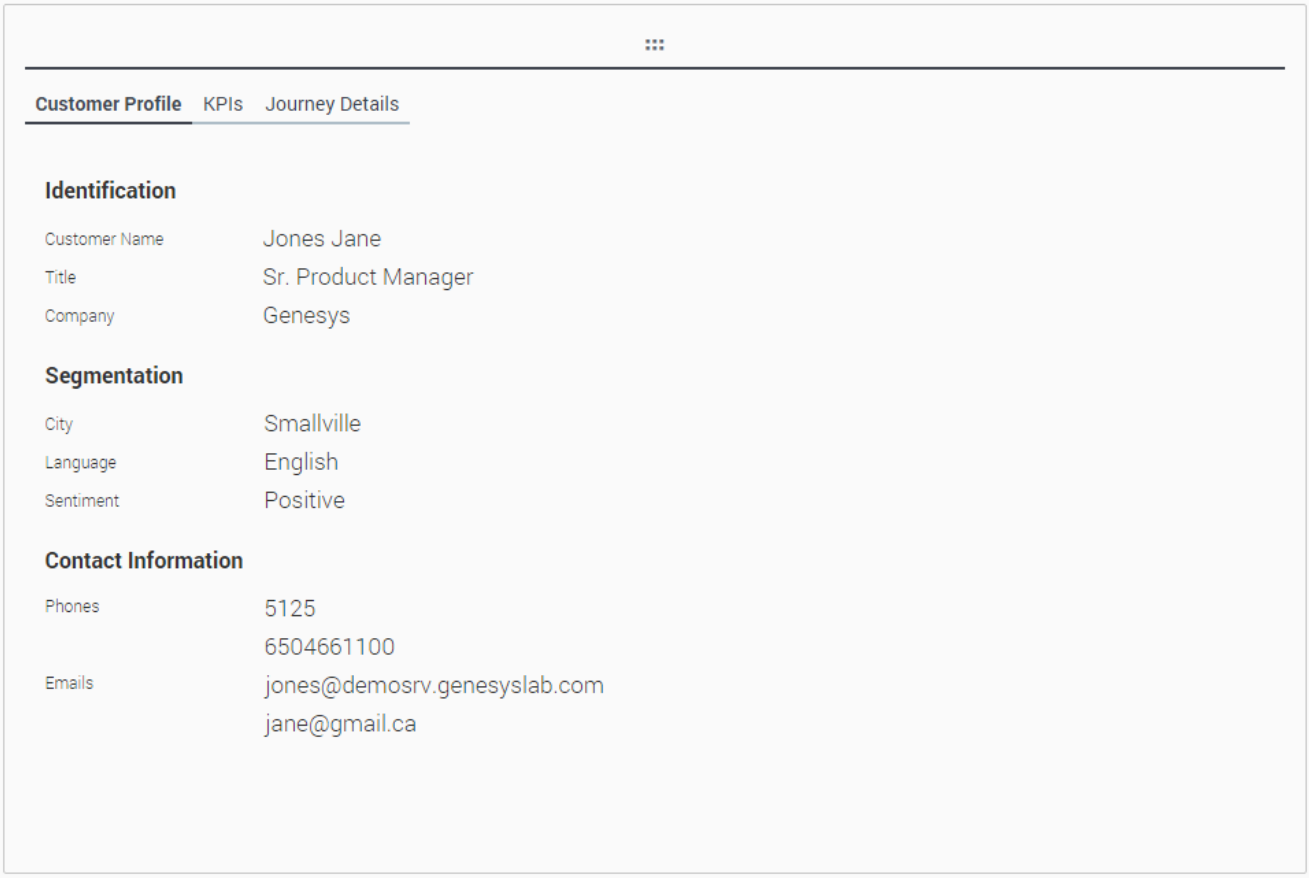


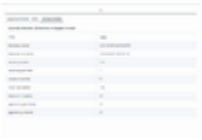
If you select a service in the timeline, for instance, Agency, its nested states and tasks are displayed in the Vertical Timeline. You can then select one of them to get KPIs, customer or journey details.

Important

You can customize the display through templates. See the guidelines in the Developer's Guide.

Display Details related to Service and State Selection





The selection in the horizontal and vertical timelines automatically updates the information area.

> **Important**
> You can customize the display through templates. See the guidelines in the Developer's Guide.

Integrate with UCS Profiles

If you enabled Customer profiles in UCS, you can integrate Journey Timeline with UCS profiles as follows:

- Install and set up NGINX on your local machine
- Ensure that the NGINX configuration includes the following information:

```
location /genesys/1/cs/profiles { proxy_pass http://<location of UCS>/profiles; }
location /genesys { proxy_pass http://localhost:8080/genesys; }
```

You can read more details here.

Frequently Asked Questions

GMS stands for Genesys Mobile Services.

Do I have to install GMS to run Context Services?

Yes, as detailed in the installation page, the installation of Genesys Mobile Services is mandatory.

Do I have to license GMS to run Context Services?

Yes.

If I install Context Services, can I use GMS?

Yes. GMS and Context Services are two distinct products which can be used within the same GMS instance, assuming that licensing for Context Services is correctly setup.

What is the difference between PUT and POST queries?

Both can be used to create and modify a resource, however:

- PUT should be used to create or overwrite a resource.
- POST should be used to modify and update a resource.

Can I manage profiles in Context Services with the GMS installation?

No. Context Services does not include the Customer Profile API by default. The Customer Profile API is part of the UCS product and will remain there, along with Contact related information and interactions history. See the Enabling Customer Profiles in UCS for further details. The Service API which is migrated to GMS keeps backward compatibility between APIs and no longer requires the schema provisioning for extensions.