User's Guide

Context Services 8.0.x

3/11/2022

Table of Contents

Context Services User's Guide **3**

Prepare your Deployment **4**

Setting up the UCS Database 5

Configure DAP 7

Configure UCS Application 8

Using TLS with UCS 9

Using TLS with UCS Clients 12

Export Certificates 13

Configuration Options 14

Security and Authentication 25

UCS and Conversation Manager **27**

Conversation Manager 28

UCS in eServices and Conversation 30

UCS with Context Services **31**

Archiving and Pruning the DB 32

Contact Identification 37

Messaging, Modes, and Migration 39

Using HTTPS 40

Profile Basics 43

Service Basics 44

Set-based Archiving with MSSQL 45

Set-based Archiving with Oracle 47

Context Services User's Guide

Purpose

Provide an extensive description of the customer View:

- Features
- Product architecture
- Introduction to the API

Description

Context Services's REST APIs provide access to UCS resources (data entities) 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.

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
- Schedule callback with enhancement multimedia confirmation

Prepare your Deployment

Purpose

Describes all of the required procedures for deploying UCS and its Context Services capabilities. 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

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 or higher, from the Oracle web site:
   http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html
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:
   300px
   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:
   300px
   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).
   300px
4. On the Server Info tab, enter the host name and port number.
   300px

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.

Prerequisites

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

Procedure

1. On the General tab, enter a name.
2. On the Server Info tab, enter a host name and port number.
3. On the Start Info tab, enter an arbitrary character. The real values will be entered during installation.
4. On the Connections tab, add connections to the UCS DAP, Message Server, and Stat Server.
   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.
5. On the Options tab, in the `cview` section:
   1. Set enabled to `true`.
   2. Set port to the port on which the web services will be deployed (the default is 8080).
   3. Set tenant-id to the identifier of the tenant with which UCS will be associated.
   4. Set other configuration options as needed. All options are described on the Configuration Options page.

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.

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 Java Environment and Libraries for eServices and UCS, keytool is placed in the `\jre\bin` directory.

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 Security Guide.*
   - Use keytool with the–genkey parameter; for example:

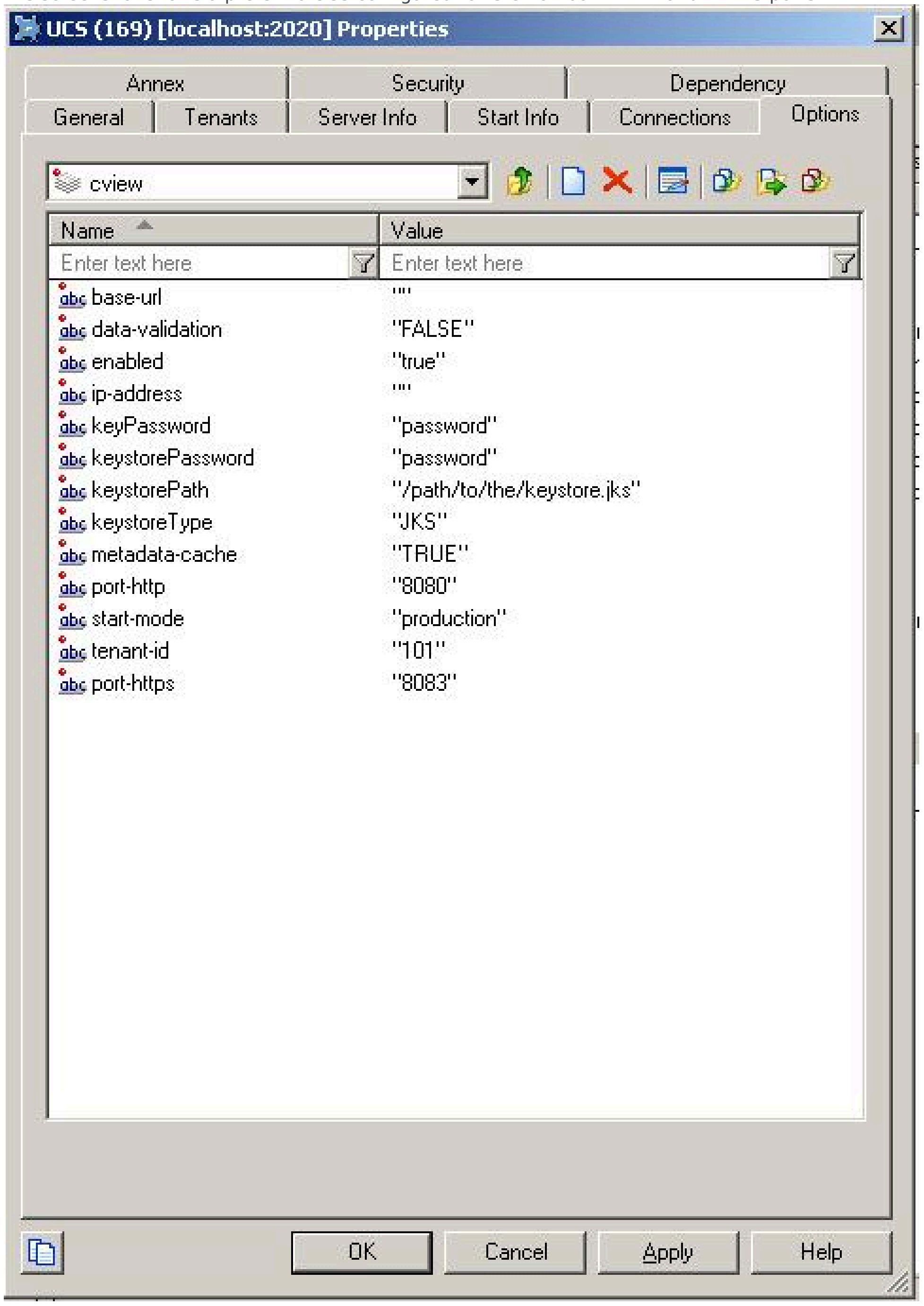
```
keytool -genkey -v -alias hostname.example.com
 -dname "CN=hostname.example.com,OU=IT,O=ourcompany,C=FR" -keypass theKeyPassword
 -keystore certificate.jks -storepass theKeystorePassword -keyalg "RSA" -sigalg
 "SHA1withRSA"
 -keysize 2048 -validity 3650
```

   - 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 Security 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.
   - Prior to release 8.1.0, add the following options to the `cview` section. See the Configuration Options page for full descriptions.
     - `keyPassword, keystorePassword, keystorePath, keystoreType`
       - If you used keystore, set values for these options according to the values used in the command line (as in the example in Step 1).
       - If you used Microsoft Management Console, `keystoreType` must be `PCKS12`, and `keystorePassword` and `keyPassword` must both be equal to the password that you defined

in the export procedure.

- `port-https`—Choose a value that is appropriate for your environment. You can also configure the `port-http` option, but it is not required.

This screenshot shows a pre-8.1.0 UCS configured to listen on both HTTP and HTTPS ports.



Name	Value
base-url	""
data-validation	"FALSE"
enabled	"true"
ip-address	""
keyPassword	"password"
keystorePassword	"password"
keystorePath	"/path/to/the/keystore.jks"
keystoreType	"JKS"
metadata-cache	"TRUE"
port-http	"8080"
start-mode	"production"
tenant-id	"101"
port-https	"8083"

Next Steps

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

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 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>
   ```

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.

Configuration Options

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

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.
- `business-attributes` section—Maps Context Service keys to Business Attribute key-value pairs.
- `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). 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<ref name="restart">This column indicates whether you must restart UCS for changes in the option value to take effect.</ref>	Description
enabled	Yes	• `true` to enable Context Services functionality in UCS<ref name= "ucs">Universal Contact Server</ref>. • `false` (default) to disable Context Services.
port-http	Yes	The HTTP port used to deploy UCS/CS (defaults to 8080).
port-https	Yes	The HTTPS port used to deploy UCS/CS (defaults to 8083).
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

Name	Restart UCS<ref name="restart">This column indicates whether you must restart UCS for changes in the option value to take effect.</ref>	Description
		additional checks on data provided by the connected clients. • `false` to disable data validation.
keyPassword Deprecated since: 8.1.000.xx	Yes	Password for the certificate. No longer available in 8.1. See the Security and Authentication page.
keystorePassword Deprecated since: 8.1.000.xx	Yes	Password for the keystore file. No longer available in 8.1. See the Security and Authentication page.
keystorePath Deprecated since: 8.1.000.xx	Yes	Path to standard JKS file. No longer available in 8.1. See the Security and Authentication page.
keystoreType Deprecated since: 8.1.000.xx	Yes	JKS or other JSSE-supported type. No longer available in 8.1. See the Security and Authentication page.
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 available since: 8.0.3	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. The cache contains metadata for

Name	Restart UCS<ref name="restart">This column indicates whether you must restart UCS for changes in the option value to take effect.</ref>	Description
		contact attributes, identification keys, profiles, services, states and tasks extensions.

<references />

[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<ref name="restart">This column indicates whether you must restart UCS for changes in the option value to take effect.</ref>	Description
use-np	Yes	• `true` to enable set-based archiving. • `false` (default) to disable set-based archiving.

<references />

[authentication] Section

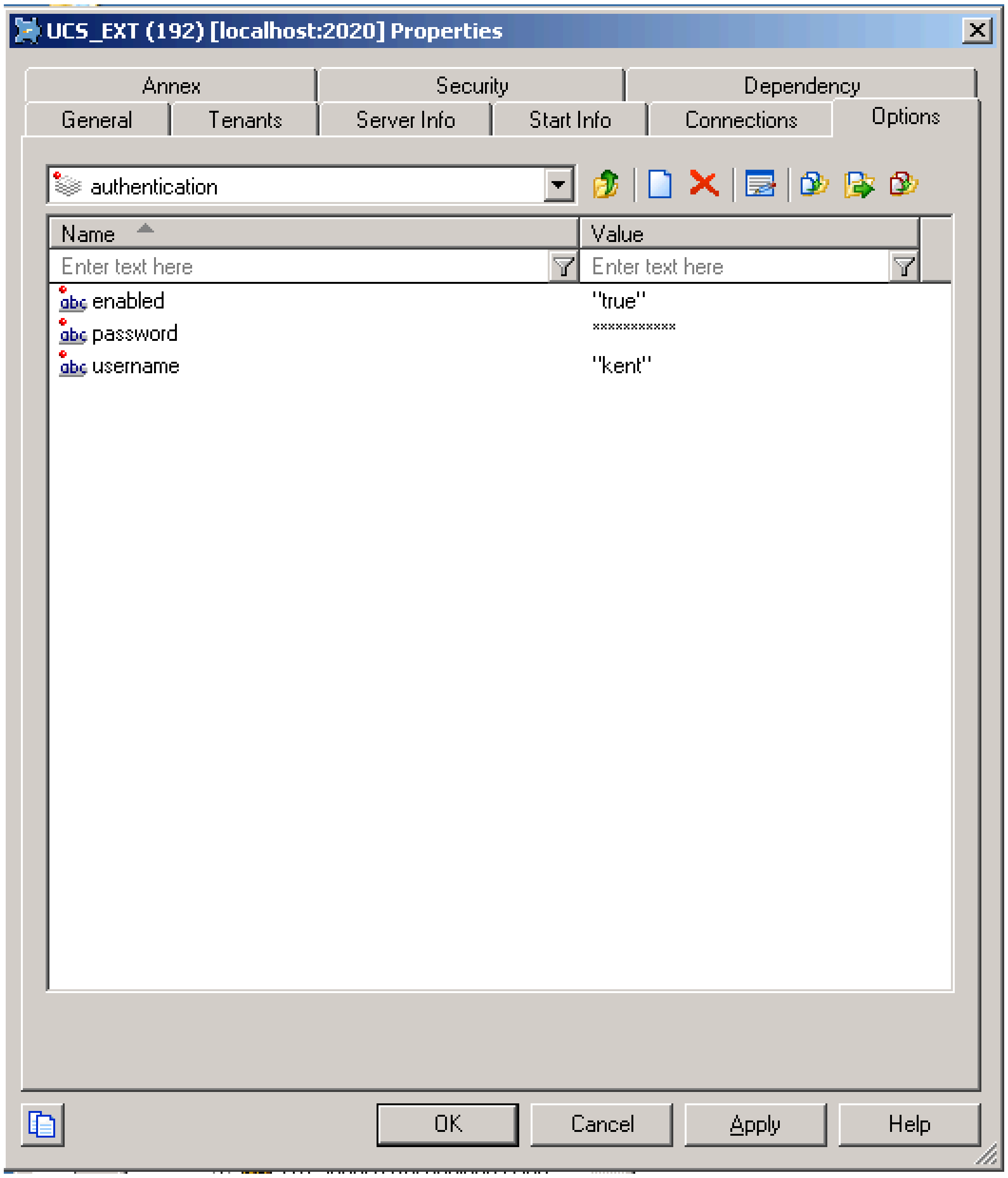
This section configures authentication for clients connecting to UCS. Authentication, available since release 8.0.300.02, applies to UCS/CS only. See also:

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

authentication Section

Name	Restart UCS<ref name="restart">This column indicates whether you must restart UCS for changes in the option value to take effect.</ref>	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.</tt>`
username	Yes	User name allowed to connect to the Context Services API. Effective only if `mode` is set to `single-user.</tt>`
use-role	Yes	• `false` to disable the use of roles in authentication (default). • `true` to enable the use of roles in authentication.

<references />


UCS_EXT (192) [localhost:2020] Properties

authentication

Name	Value
enabled	"true"
password	***********
username	"kent"

Section 'authentication' in Configuration Manager.

[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.

`business-attributes` Section

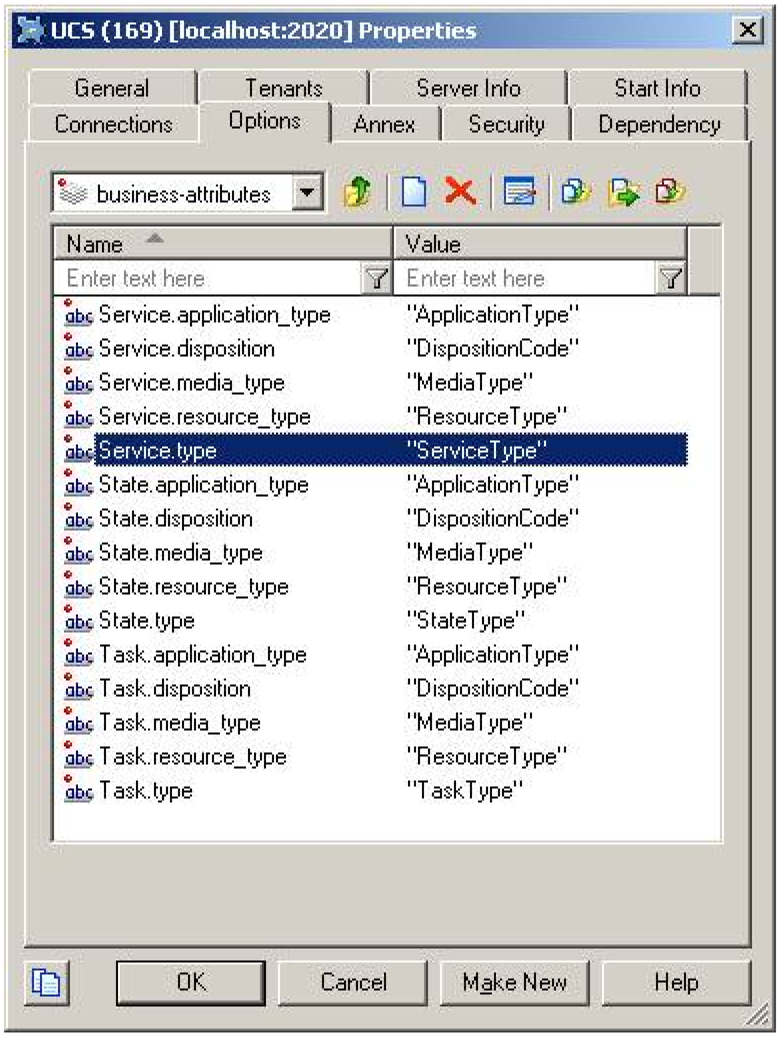
Name	Restart UCS<ref name="restart">This column indicates whether you must restart UCS for changes in the option value to take effect.</ref>	Description
${resource name}.${field name}	No	Associates a Business Attribute key with the name of the Business Attribute configured in the proper tenant. The option key name follows this syntax: ${resource name}.${field name} 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. Notes: • 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 "MediaType" Business Attributes.
map-names	No	Possible values are: • `true` to return the Names of Business Attribute Values

Name	Restart UCS<ref name="restart">This column indicates whether you must restart UCS for changes in the option value to take effect.</ref>	Description
		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.

<references />

Mapping Example

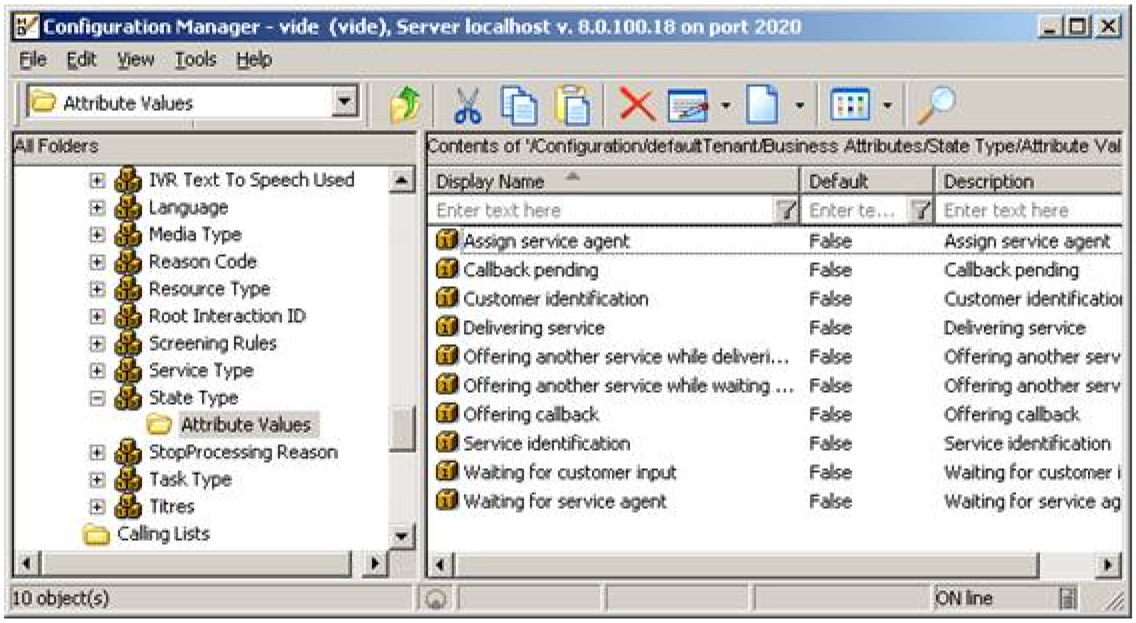
This first screenshot shows the section in UCS options, with the list of mapped keys, such as, for instance: Service.service_type, Task.disposition, State.media_type.



Name	Value
Service.application_type	"ApplicationType"
Service.disposition	"DispositionCode"
Service.media_type	"MediaType"
Service.resource_type	"ResourceType"
Service.type	"ServiceType"
State.application_type	"ApplicationType"
State.disposition	"DispositionCode"
State.media_type	"MediaType"
State.resource_type	"ResourceType"
State.type	"StateType"
Task.application_type	"ApplicationType"
Task.disposition	"DispositionCode"
Task.media_type	"MediaType"
Task.resource_type	"ResourceType"
Task.type	"TaskType"

Business Attribute Configuration in UCS Options

The following screenshot shows one of the mapped business attributes, the key and the associated values, which your application can retrieve in the result of GET operations by setting to `true` the map-names UCS options, as stated above.


Configuration Manager - vide (vide), Server localhost v. 8.0.100.18 on port 2020
File Edit View Tools Help
Attribute Values
All Folders
ON line

IVR Text To Speech Used
Language
Media Type
Reason Code
Resource Type
Root Interaction ID
Screening Rules
Service Type
State Type
Attribute Values
StopProcessing Reason
Task Type
Titres
Calling Lists

Contents of '/Configuration/defaultTenant/Business Attributes/State Type/Attribute Val

Display Name	Default	Description
Enter text here	Enter te...	Enter text here
Assign service agent	False	Assign service agent
Callback pending	False	Callback pending
Customer identification	False	Customer identificatio
Delivering service	False	Delivering service
Offering another service while deliveri...	False	Offering another serv
Offering another service while waiting ...	False	Offering another serv
Offering callback	False	Offering callback
Service identification	False	Service identification
Waiting for customer input	False	Waiting for customer i
Waiting for service agent	False	Waiting for service ag

10 object(s)

Related State Type Business Attribute

[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<ref name="restart">This column indicates whether you must restart UCS for changes in the option value to take effect.</ref>	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-`

Name	Restart UCS<ref name="restart">This column indicates whether you must restart UCS for changes in the option value to take effect.</ref>	Description
		`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.

<references />

[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.

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.

UCS and Conversation Manager

Purpose

Provides general descriptions of the Conversation Manager solution and the role that Universal Contact Server (UCS) plays within it. Contrasts UCS/CS with classic UCS.

Conversation Manager

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 e-mail congratulating him on the approval of his application.

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

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.

Service

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 the diagram in Service Basics).

file:important.png This term *state* does not have the same meaning as "SCXML state."

Architecture

file: ConvMgrArchitec.jpg

UCS in eServices and Conversation

This page contrasts the role of UCS in eServices with its role in Conversation Manager

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, e-mail 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.*

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 Service Basics.
- Clients communicate with UCS/CS 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 Profile Basics.

UCS with Context Services

Purpose

Provides a general description of how UCS works.

Archiving and Pruning the DB

	Purpose: 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. 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;

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.

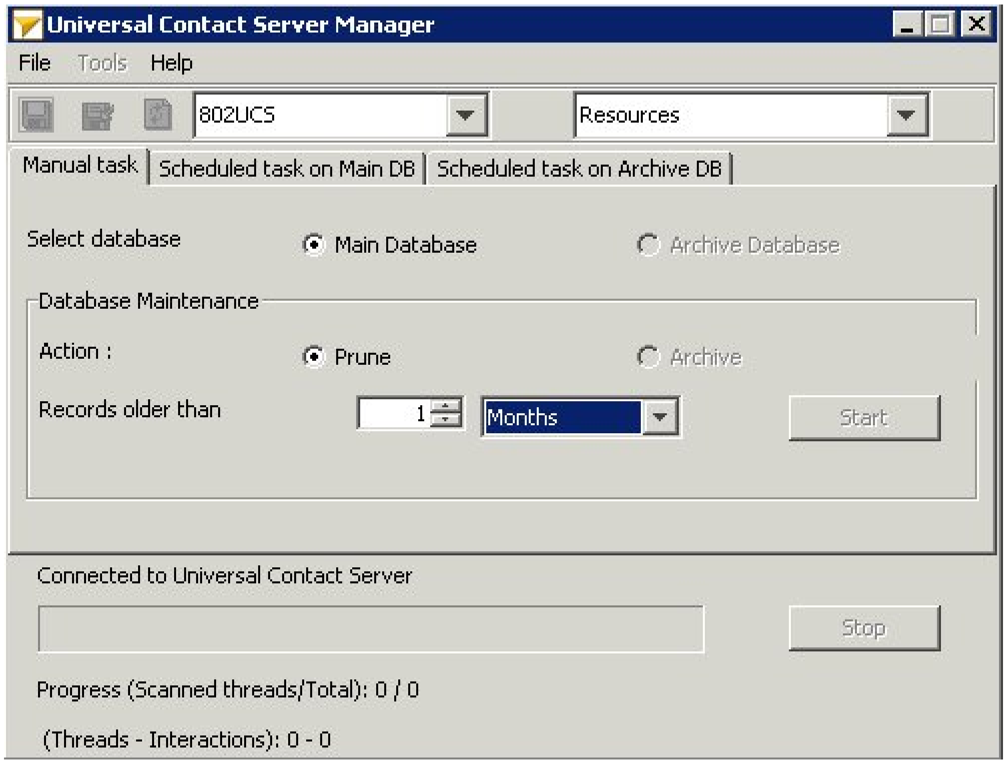
Configuration

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.


Universal Contact Server Manager
File
Tools
Help
802UCS
Resources
Manual task
Scheduled task on Main DB
Scheduled task on Archive DB
Start
Stop

Select database [✓ Main Database] Archive Database

Database Maintenance

Action : [✓ Prune] Archive

Records older than 1 Months

Connected to Universal Contact Server

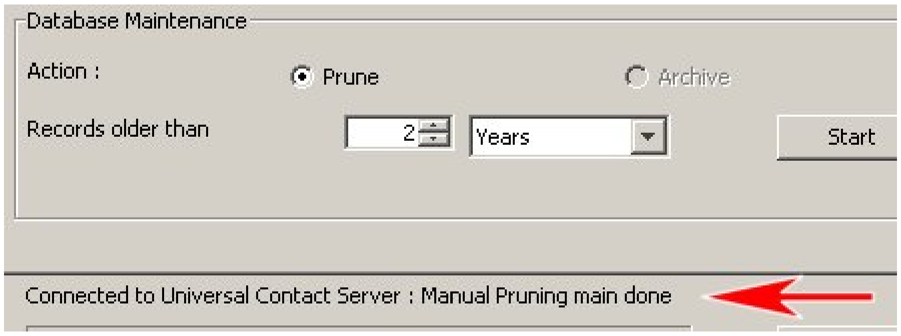
Progress (Scanned threads/Total): 0 / 0

(Threads - Interactions): 0 - 0

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

   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.



Action : [✓ Prune] Archive

Records older than 2 Years

Connected to Universal Contact Server : Manual Pruning main done

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 e-mail address, last_name + first_name, and so on). If you attempt to identify by e-mail 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.

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.*

Using HTTPS

	Purpose: To configure UCS to use secure HTTPS connections.

Overview

This page describes using configiration options and the keytool utility to configure UCS to use secure HTTPS connections.

- Keytool is a key and certificate management utility included in JDK or JRE installations. For instance, when you install Java Environment and Libraries for eServices and UCS, keytool is placed in the `\jre\bin` directory.
- Prior to release 8.1.0, you add the options to the `cview` section.

Procedure

This procedure makes use of keytool, which is a key and certificate management utility included in JDK or JRE installations. For instance, when you install Java Environment and Libraries for eServices and UCS, keytool is placed in the `\jre\bin` directory.

1. Generate a key-pair for the certificate. The following is an example command line:

```
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
```

2. Set configuration options in your UCS Application object.
   - Add the following options to the `cview` section. See the Configuration Options page for full descriptions.
     - `keyPassword`, `keystorePassword`, `keystorePath`, `keystoreType`—Set values according to the values used in the command line.
     - `port-https`—Choose a value that is appropriate for your environment. You can also configure the `port-http` option, but it is not required.
3. Export the certificate from UCS. Following is an example command line:

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

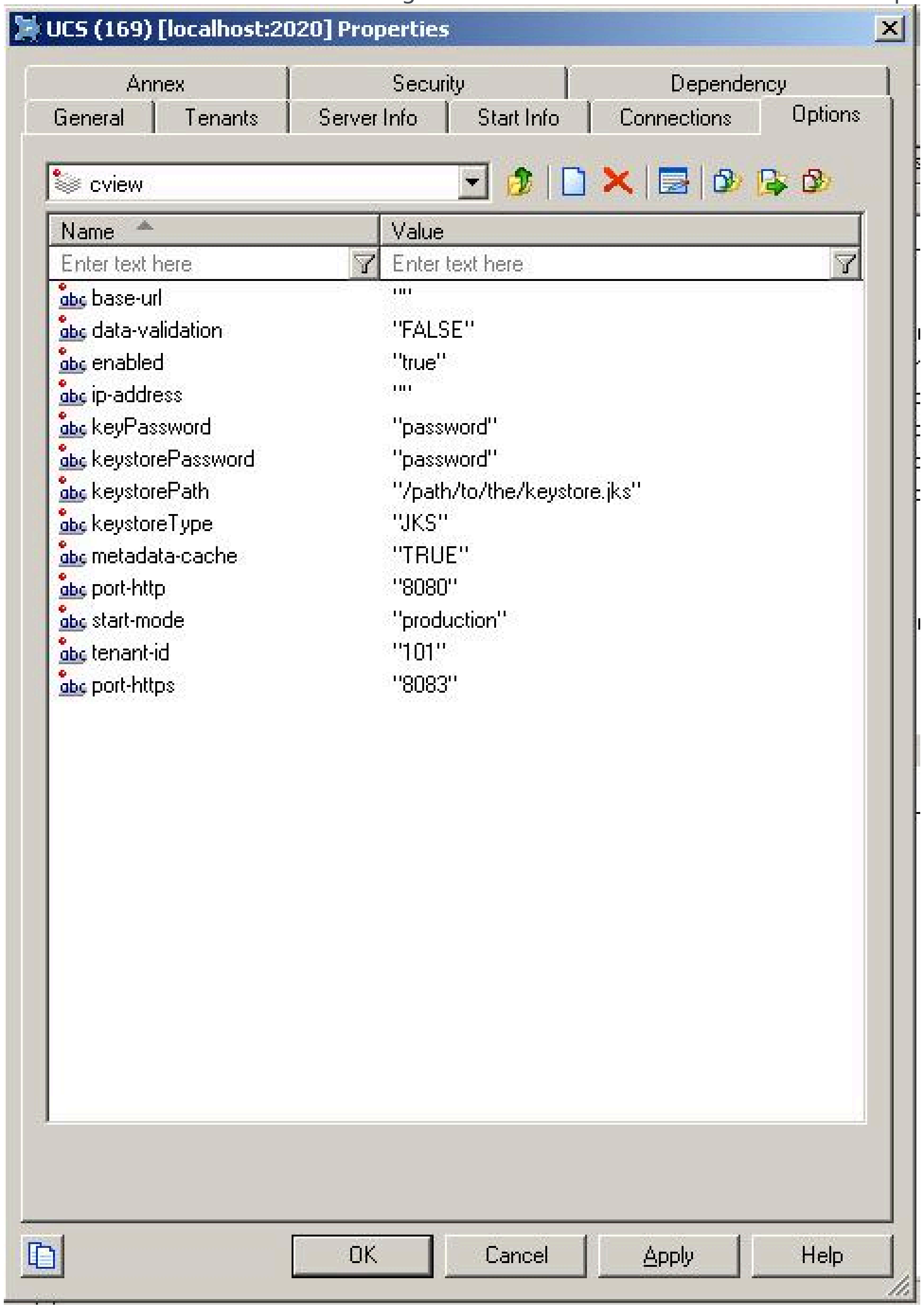
4. Import the certificate on all clients of UCS. Following is an example command line:

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

```
    -keystore .keystore -storepass anotherPassword
```

Example

This screenshot shows UCS configured to listen on both HTTP and HTTPS ports.



Name	Value
base-url	""
data-validation	"FALSE"
enabled	"true"
ip-address	""
keyPassword	"password"
keystorePassword	"password"
keystorePath	"/path/to/the/keystore.jks"
keystoreType	"JKS"
metadata-cache	"TRUE"
port-http	"8080"
start-mode	"production"
tenant-id	"101"
port-https	"8083"

Profile Basics

	Purpose: Basic information on profiles

Description

UCS stores contact information in Profiles. Profile data is separated into the following types:

- *Core information* consists of one or more typed attributes, which are defined by a Schema.
- *Extensions* consist of one or more typed attributes. Users configure these as needed for their particular organization or customer service application.
- *Identification Keys* are an attribute or attributes that you specify as the way to identify a customer. These can be attributes of the core information or of associated extensions.

One of the primary features of the Context Services API is the ability to identify customers based on one or more attributes of the customer, known as Identification Keys. Each identification key consists of one or more attributes of the core customer profile, or of any defined extension. An attribute must be specified as an Identification Key to be usable in customer identification.

Service Basics

	Purpose: Basic information on Services

UCS makes use of a model in which customers are associated with any number of Services. Services are composed of any number of States, and States can in turn be composed of any number of Tasks. This three-level structure provides a flexible vocabulary by which organizations store the history of the services that they provide to customers. A Service may also be divided directly into tasks: file:serviceStateTask.jpg Services are defined by association to Service Types that you create as Business Attributes in the Configuration Layer. States may be used to represent components of customer service, such as:

- Customer identification
- Assigning a service agent (automated or live)
- Service identification
- Waiting for a service agent
- Offering another service while waiting for an agent
- Offering callback
- Waiting for customer input

For more information on this point, see this page in the Context Services Developer's Guide. Services, States and Tasks exist over some application-defined lifecycle. Upon completion, applications may specify a Disposition. For example, the offering of a new product or service might be recorded as a State of type `Offer another service.` The Disposition might be set to show whether the customer accepted or declined the offer. Information on past declined or accepted offers could then be used to calculate the likelihood that the customer might be interested in the offer at some point in the future. **Note:** This Service Model can be used by any component that can access UCS/CS’s HTTP interface. It is not limited to use in Conversation Manager.

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'
   ```

   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. ⚠ 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;
```

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.