User's Guide

Multiple UCS Instances in Single Tenant

5/4/2025

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.

Contents

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).

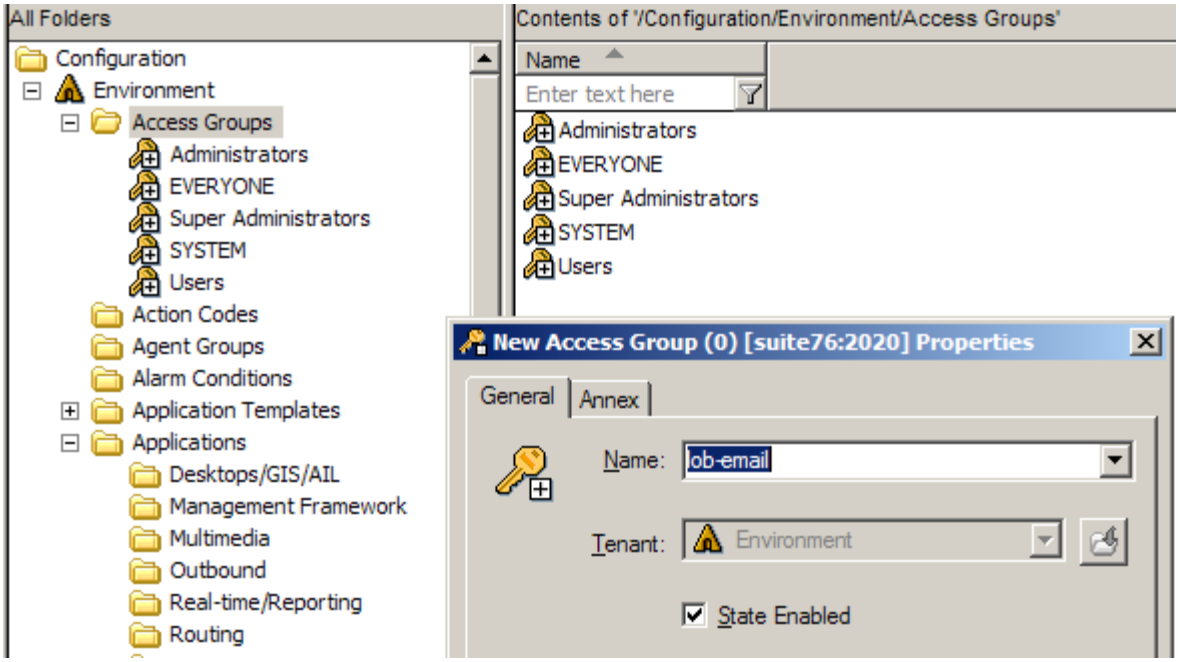
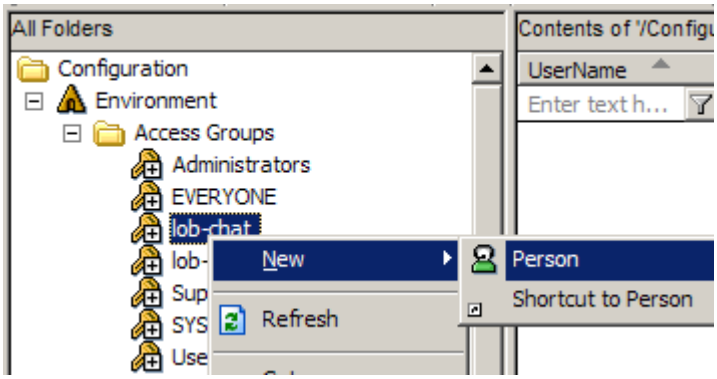


Figure 2: Creating Access Groups

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



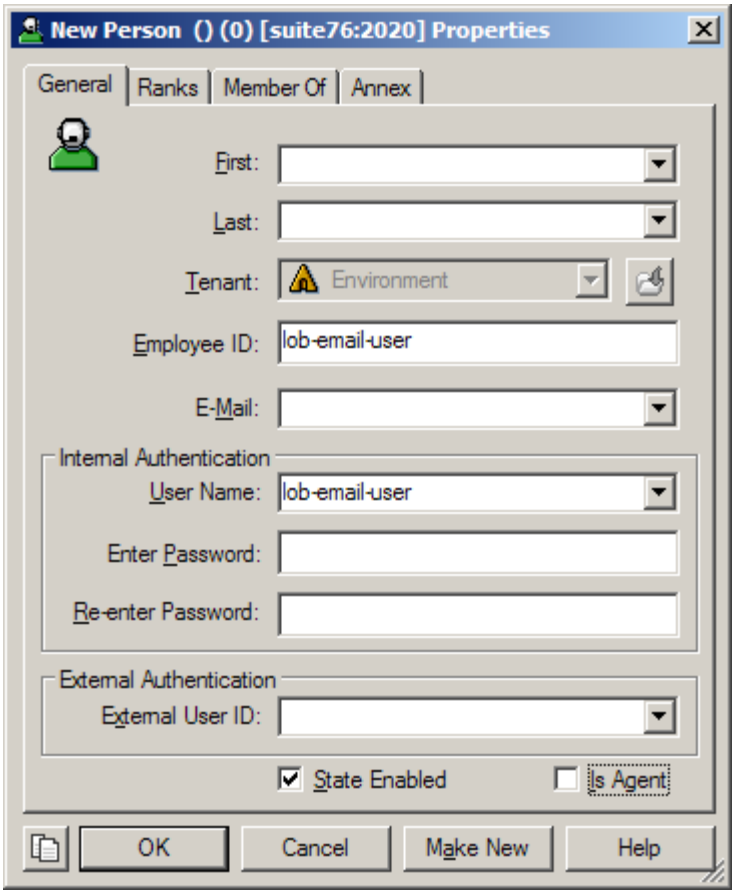


Figure 3: Creating a Person

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.

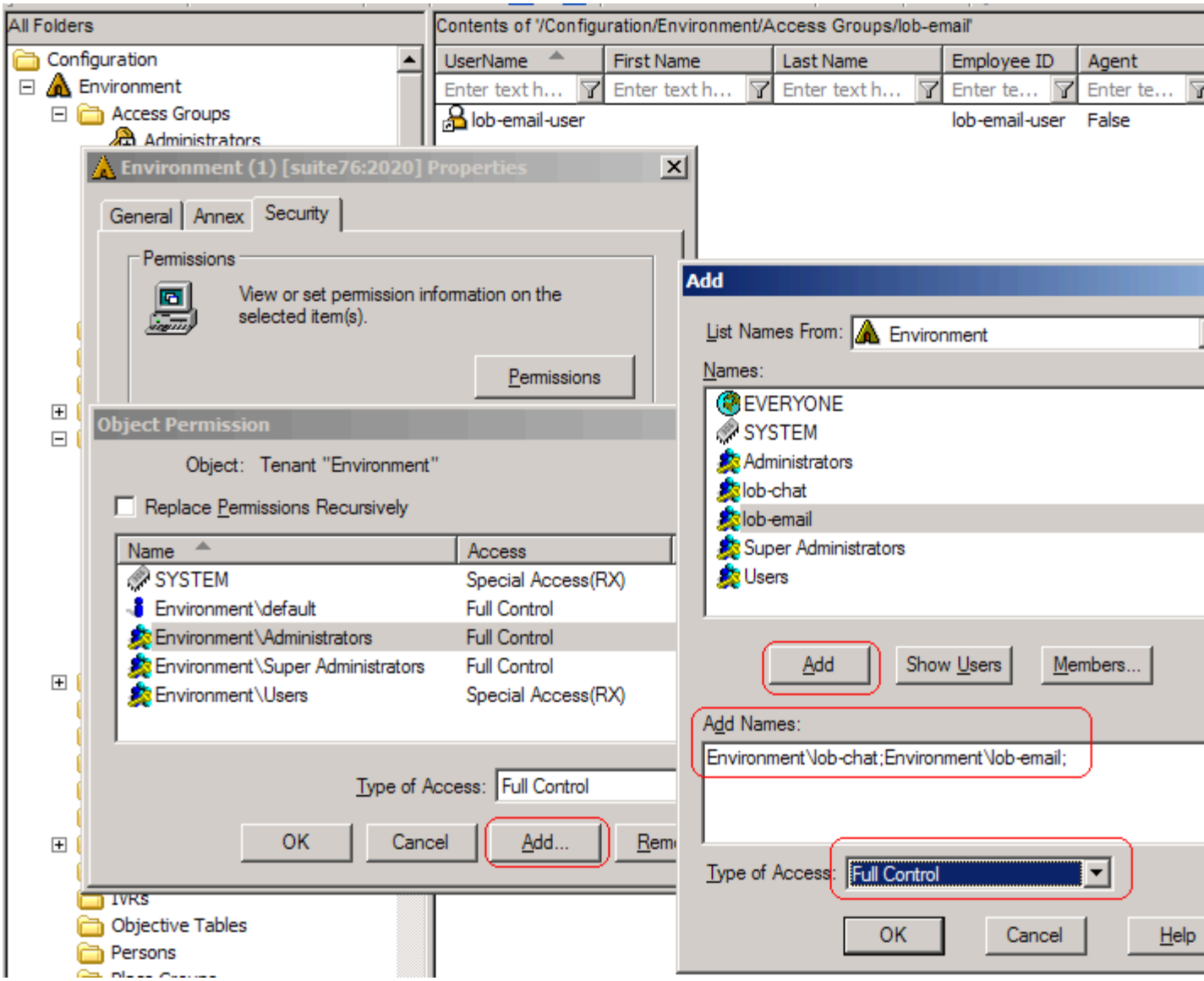


Figure 4: Setting Tenant Permissions

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.

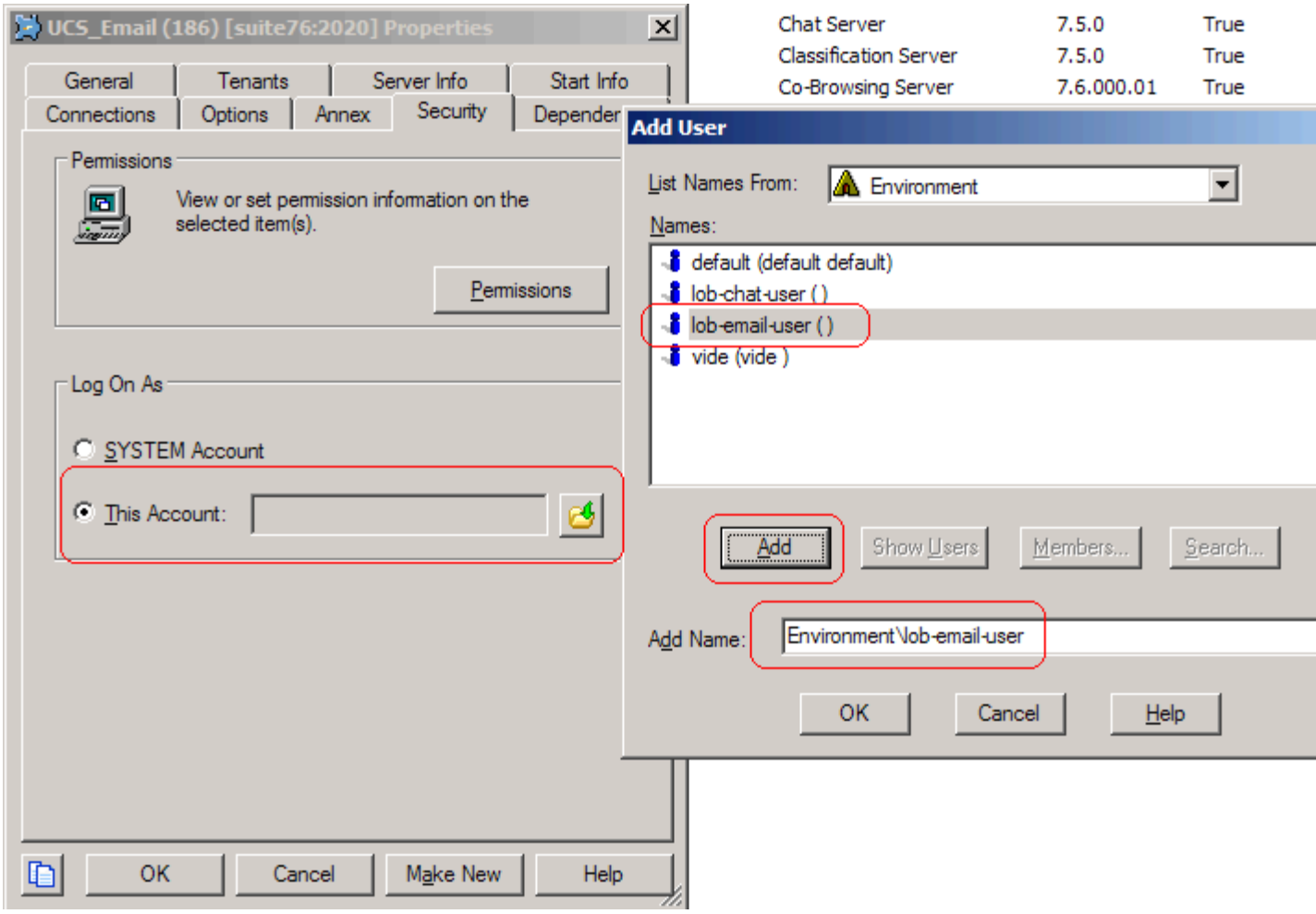


Figure 5: Setting the Log On As Account

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.

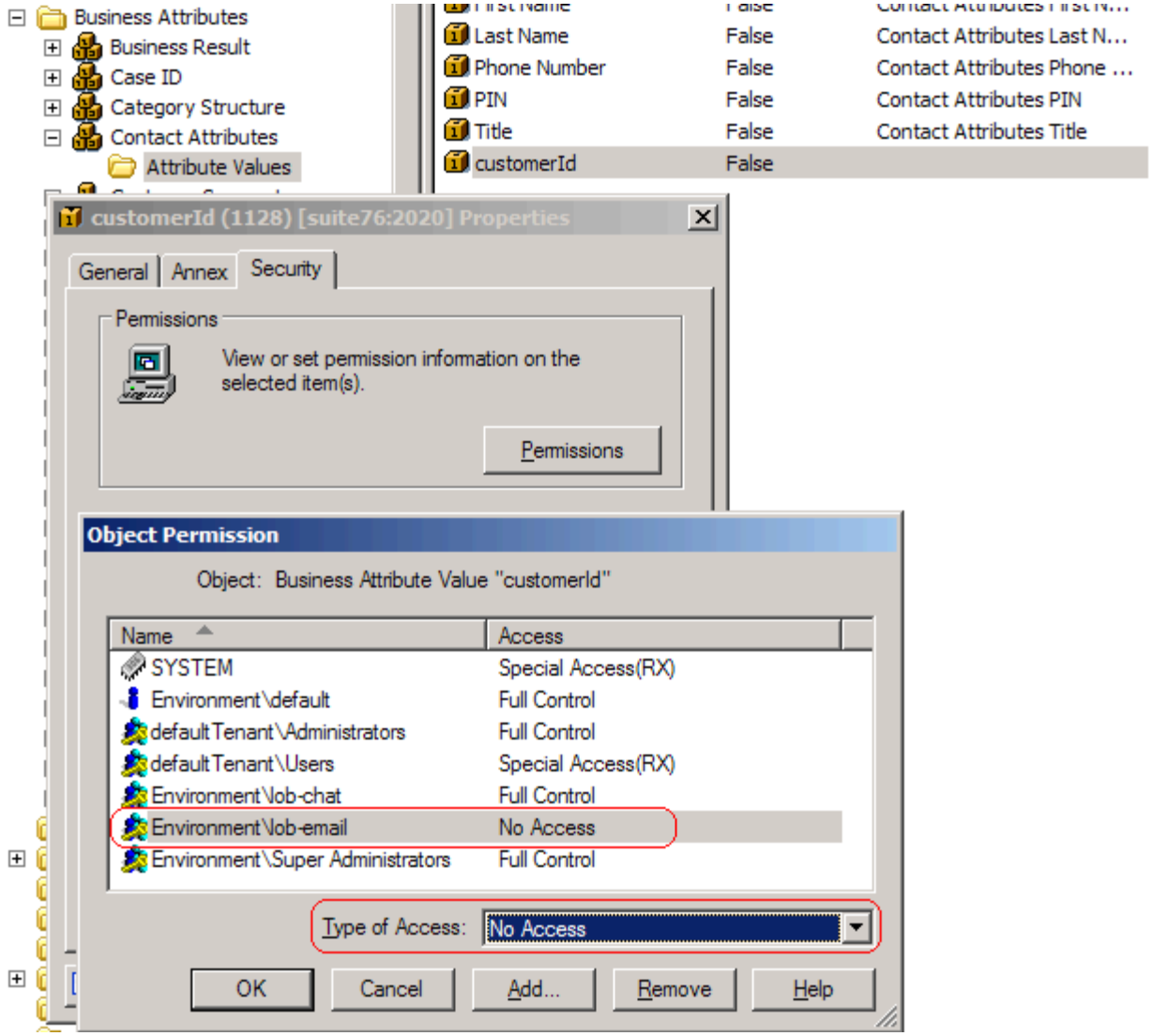


Figure 6: LOB-Specific Contact Attributes

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.

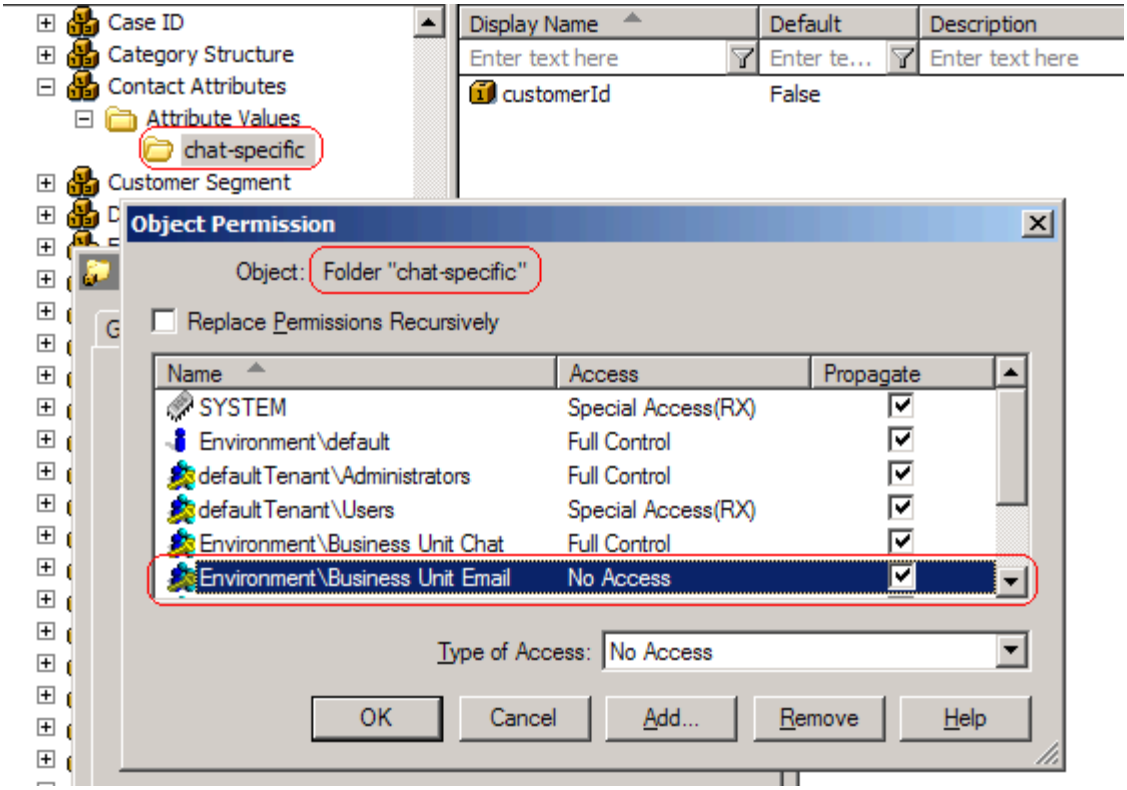


Figure 7: Chat-Specific Contact Attributes

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.

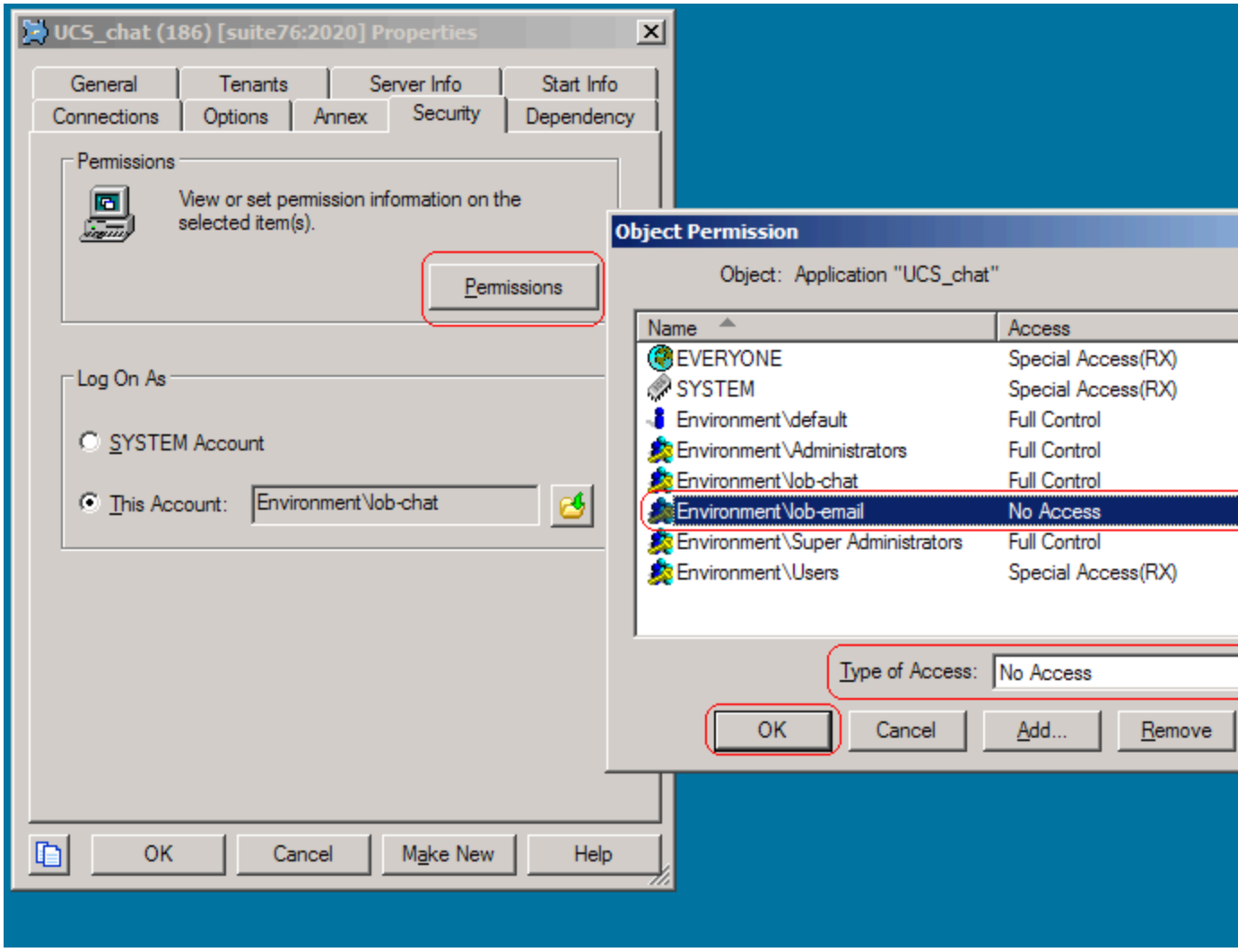


Figure 8: UCS Application Permissions

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

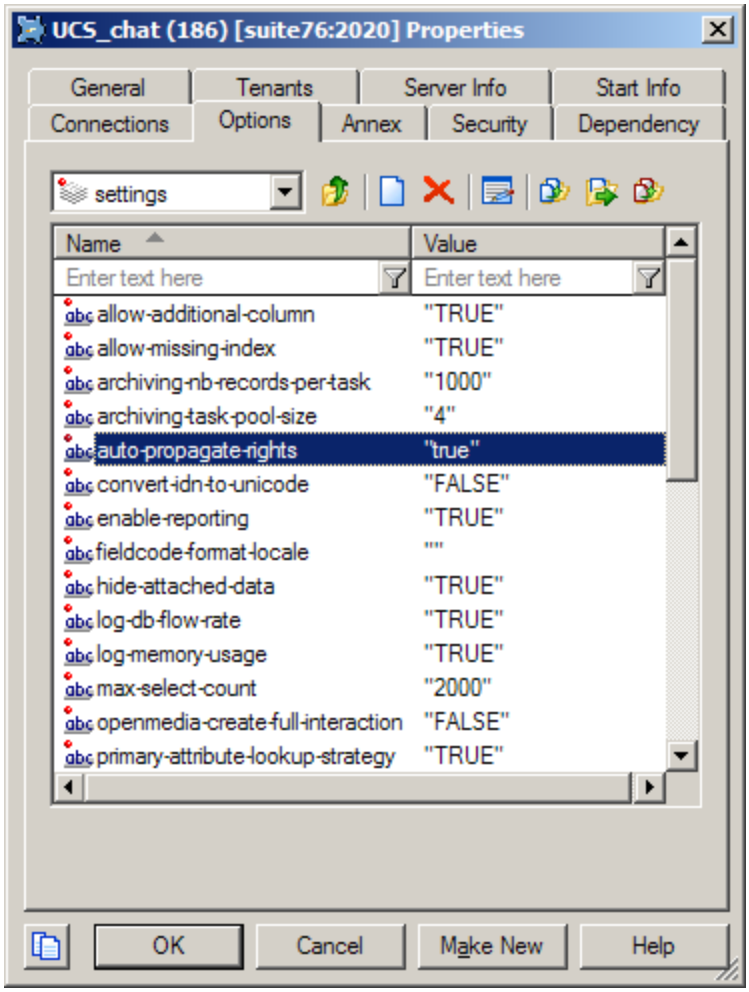


Figure 9: UCS Options

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.

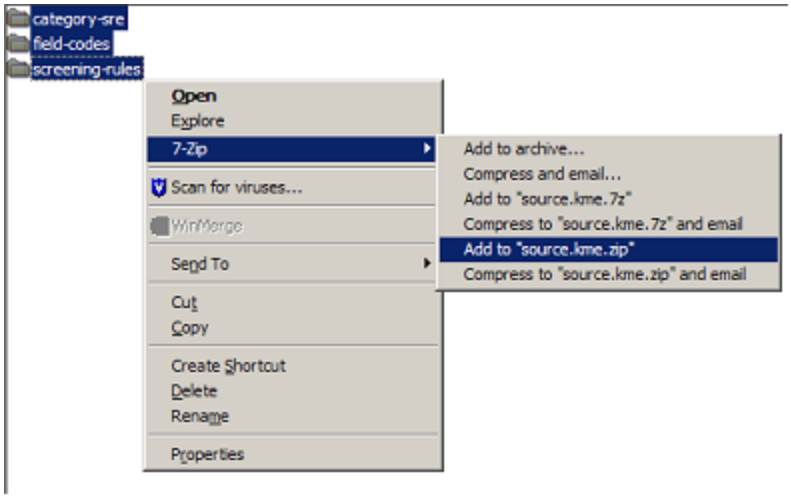


Figure 10: Knowledge Manager Folders Must be at the Root of the Zip File

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.