Genesys Interactive Insights User's Guide

Example - Product Line and Product

4/1/2025

Example - Product Line and Product

This customization example adds two dimensions to the Info Mart database that are derived from string-based attached data that might exist in your environment. These dimensions (Product Line and Product) form a Product hierarchy within the GI2 universe that you can drill. You can add these dimensions to the GI2 reports to provide results by product, by product line, or by any other dimension that you choose to substitute in this example.

The general steps for customization are the following:

1. Create and populate one or more user data tables in the Info Mart database.	**[+] Show Steps** ## Creating User Data Dimension Tables Within the Info Mart database, create and populate a custom user data dimension table—for example, USER_DATA_CUST_DIM_10. The USER_DATA_CUST_DIM_x tables store information about changes in data that accompany telephony events that are recorded by Interaction Concentrator (ICON) and further processed by Genesys Info Mart ETL runtime processes. Genesys Info Mart writes to these tables up to five descriptors of your business data. This example populates two fields: PRODUCT_LINE with product line data and PRODUCT_CODE with product code data. Refer to the relevant Genesys Info Mart Reference Manual (available on docs.genesys.com) for the complete data model of the USER_DATA_CUST_DIM_* tables.
2. Configure user data keys in the aggregation tables to point to your user data table(s) and populate the aggregation tables.	**[+] Show Steps** ## Mapping User Data Keys and Columns The information in this section describes how to configure user data keys and columns in the Info Mart database Mapping and Aggregation tables. ### User Data Mapping Tables in the Info Mart Deployment-specific attributes, in the form of user-defined attached data, are represented in the Genesys Info Mart model both by low-cardinality data (in string format) and high-cardinality data (in numeric, date/time, and string formats).

Low-cardinality-string user data that is associated with an interaction resource—such as automobile models and product codes—is stored in the IRF_USER_DATA_KEYS and USER_DATA_CUST_DIM_x dimension tables. High-cardinality user data that is associated with an interaction resource—such as prices, number of widgets sold, and dates—is stored in the IRF_USER_DATA_GEN_1 and IRF_USER_DATA_CUST_x fact extension tables. In addition to these tables are the CTL_UD_TO_UDE_MAPPING and CTL_UDE_KEYS_TO_DIM_MAPPING tables that you must update:

- CTL_UD_TO_UDE_MAPPING ties in user data keys that are defined in the underlying ICON application with user data columns that are defined in the previous tables.
- CTL_UDE_KEYS_TO_DIM_MAPPING maps the user data dimension tables (USER_DATA_CUST_DIM_x) to IRF_USER_DATA_KEYS.

Execute the sample script (Sample SQL Script for Creating and Mapping User Data) to set up user data mapping and recording in your environment. Also, refer to the:

- *Interaction Concentrator Deployment Guide*.
- `make_gim_UDE_template.sql` script, provided with Genesys Info Mart deployment. Note that RAA deploys similar scripts—`make_gim_UDE_template_<rdbms>.sql`. These scripts, however, hold entirely different content and are designed to configure user data for social media measures.
- Mapping User Data Worksheet in the *Genesys Info Mart Deployment Guide*. This worksheet contains several columns that you can use to record information about the specific attached data key in use in your environment. Consider adding each custom attached data table in use within your environment to this worksheet.
  - Refer to Special Note about Numeric User Data for information about configuring keys for Revenue and Satisfaction user data.
  - The instructions in step 4 (below) for adding user data dimensions to the universe and customizing the GI2 reports apply to all of the fields in this worksheet.

Predefined attached data also appears in other Info Mart database tables, including the following:

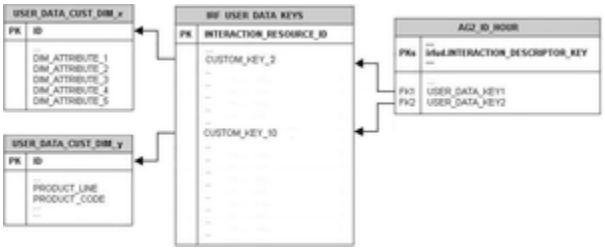
- INTERACTION_DESCRIPTOR (fields CUSTOMER_SEGMENT, SERVICE_TYPE, SERVICE_SUBTYPE, BUSINESS_RESULT)
- STRATEGY
- REQUESTED_SKILL

- ROUTING_TARGET

Using the attached data from these tables falls outside the scope of this section. Several GI2 reports, however, are provided for all of the attached data-related fields in the INTERACTION_DESCRIPTOR table.

Configuring User Data Keys in the Aggregation Tables

The AG2_AGENT, AG2_AGENT_CAMPAIGN, AG2_AGENT_QUEUE, AG2_CAMPAIGN, and AG2_ID aggregate tables provide two key columns each that you can configure to join to two user data dimension tables of your choice. (Recall that the user data dimension tables store low cardinality, string data only.) The AG2_AGENT_GRP aggregate tables also provide two such columns, but their values are inherited from the AG2_AGENT tables. The USER_DATA_KEY fields are not available in the agent session, agent states, agent interval, and queue-only aggregate tables.



Mapping User Data Keys in the Aggregate Tables/Views to User Data Dimensions

These columns are:

- USER_DATA_KEY1—A key that points to one dimension table, such as USER_DATA_CUST_DIM_10, storing five dimensions
- USER_DATA_KEY2—A key that points to a second dimension table, storing another five dimensions

These two fields provide access to a total of 10 attached data dimensions—or two hierarchies—for each aggregate table and view, as shown in the figure to the right. You must configure the aggregation job to aggregate and populate these fields.

Our product-line example uses the business attribute aggregate set, AG2_ID_*, which consists of four tables and three views. We must configure the USER_DATA_KEY1 column in each to point to the custom user data dimension table, USER_DATA_CUST_DIM_10. For more information about how to map the USER_DATA_KEY2 field, see How Do I Configure User Data for Aggregation? in the *Reporting and Analytics Aggregates User's Guide*.

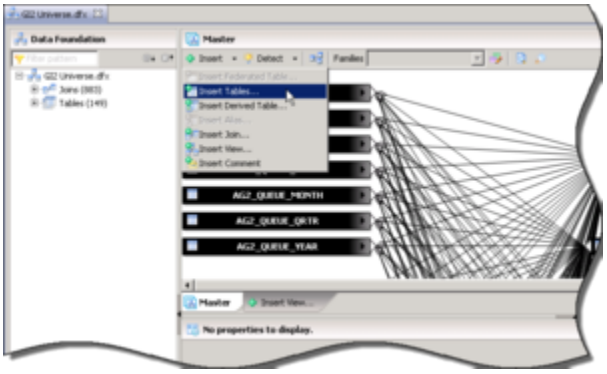
1. Create a text file having the following content on a single line:
   `(map-user-data-key (hierarchy: H_ID)`

	`(dimension: USER_DATA_KEY1) (expression: irfud.CUSTOM_KEY_10))` 2. Save the file in the Genesys Info Mart root directory with the name `user-data-map.ss`. The next time Genesys Info Mart Server restarts, the aggregation process detects this file, and aggregation begins. Refer to the *Reporting and Analytics Aggregates Reference Manual* for a data model of the aggregation tables in the Info Mart database, and the relevant Genesys Info Mart Reference Manual (available on docs.genesys.com for the structure of the USER_DATA_CUST_DIM_* tables.
3. Set Genesys Info Mart and Interaction Concentrator configuration options for collection of user data.	**[+] Show Steps** ## Setting Configuration Options Several options are available that you can use to configure what data is written to the Info Mart database, and how long data is retained. In particular, you can configure storage of user data as follows: • On Interaction Concentrator, by means of the attached data specification file (`adata_spec.xml`) and ICON configuration options, such as **EventData**, for event-based user data. • On Genesys Info Mart, by means of customizable SQL scripts to create mapping and storage tables in the Info Mart database. Some of these options apply specifically to user data. Interim releases of Genesys Info Mart and Interaction Concentrator might also introduce new configuration options that affect results. Review the following documents for a listing and description of these options: • *Genesys Info Mart Deployment Guide* • *Genesys Info Mart Release Notes* • *Interaction Concentrator Deployment Guide* • *Interaction Concentrator Release Notes*
4. Add the attached data tables to the universe structure.	**[+] Show Steps**

Adding Attached Data Tables to the Universe

After you have created and populated a user data table, you must add it to the universe and define joins between the user data table and those aggregate tables from which you plan to dimension data in the Interactive Insights reports. Note that you can use your RDBMS to define joins between tables or you can define these joins within the Information Design Tool. This section illustrates defining joins within the Information Design Tool.

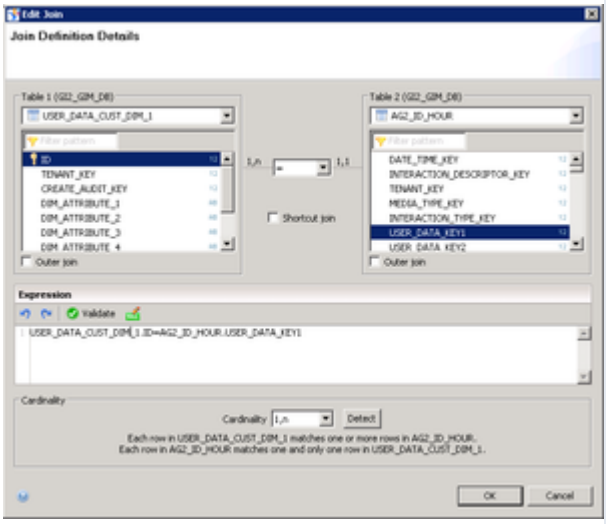
The GI2_universe schema includes custom user data tables: USER_DATA_CUST_DIM_1, USER_DATA_CUST_DIM_2, IRF_USER_DATA_CUST_1, and IRF_USER_DATA_CUST_2 with the appropriate joins and contexts already defined. If, however, you choose to include user data tables that are named or structured differently, you must add the tables to the universe schema yourself.



Inserting a table

Our product-line example relies on data that is stored in the USER_DATA_CUST_DIM_10 table, so we must perform the following steps:

1. Open the GI2_Universe Data Foundation in the Information Design Tool.
2. From the menu bar, select **Insert Object** > **Insert Tables** to open the **Insert Tables** Browser and locate your attached data table.
3. Select your table—USER_DATA_CUST_DIM_10, using this example—and click **Finish**. A symbol for the table appears in the Structure panel.



The Edit Join Dialog Box

4. From the **Insert** menu, select **Insert Join** to display the **Edit Join** dialog box, shown in the figure *The Edit Join Dialog Box*. Here, we will add joins between the attached data table and each variation of an aggregate table:

   a. From the **Table 1** drop-down list, select the **USER_DATA_CUST_DIM_10** table, and select **ID** from the list.

   b. From the comparison drop-down list, select the equal sign (=).

   c. From the **Table 2** drop-down list, select the **AG2_ID_HOUR** table, and select **USER_DATA_KEY1** from the list.

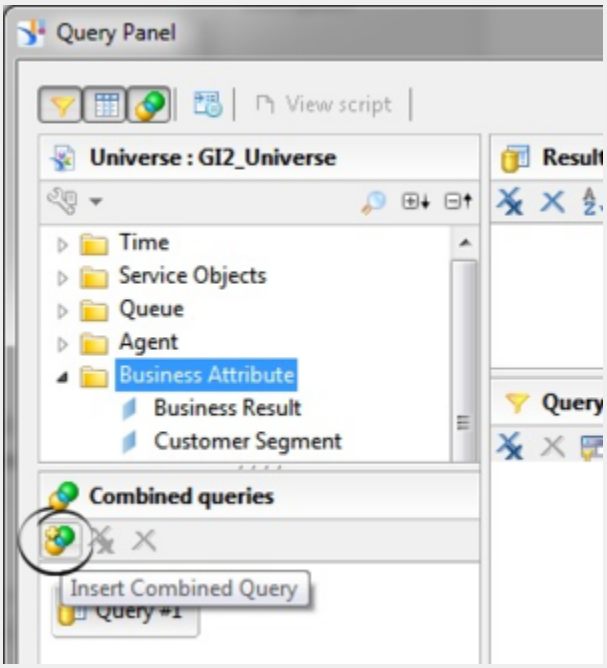
   d. Set the cardinality for one (1) to many (N).

   e. Click **OK**.

5. Reopen the **Edit Join** dialog box and repeat steps a through e to establish joins between the attached data table and each of the remaining AG2_ID_* tables and views:

   - AG2_ID_DAY
   - AG2_ID_MONTH
   - AG2_ID_SUBHR
   - AG2_ID_WEEK
   - AG2_ID_QRTR
   - AG2_ID_YEAR

Next, we add these new joins to the contexts that already exist in the universe so that any join paths will automatically be

	resolved when report queries are run against the Info Mart database.
5. Add LOV, dimensions, and filters.	**[+] Show Steps** ## Adding Dimensions and Filters to the Universe With the user data tables added to the universe, joins established between them, and the aggregate tables and contexts defined, we can now add two new dimensions and filters to the universe. These objects are used in the report that we will create. ### Adding LOV Before we create the dimensions, we must add LOV that will be associated with the dimensions. Complete the following steps to add and define Product and Product Line LOV.  Insert Combined Query 1. In the Information Design Tool, in the Business Layer list, click **Parameters and Lists of Values**. 2. On the menu bar for List of Values, click **Insert Object -> List of Values based on business layer objects**.

The properties for the new LOV appears.

3. In the **Name** field, enter `lov_product_line`.
4. Click **Edit Query**.
   The **Query Panel** appears.
5. In the **Query Panel**, select the **Business Attribute\Product Line** dimension.
6. If the **Combined Queries** panel is not visible, click **Show/Hide Combined Queries Panel**, then click **Insert Combined Query**, and create a combined query, selecting the dimensions **Service Objects** and **All**.
7. Save the universe.
8. Repeat steps 1 through 7 for the Product code (in the **Name** field, enter `lov_product`).
9. Publish the universe to a repository.

Creating Dimensions Based on User Data

Complete the following steps to add and define the **Product** and **Product Line** dimensions to the **Business Attribute** class.

1. Within the Information Design Tool, in the **Business Layer** list, click the **Business Attribute** class.
2. On the menu bar, click **Insert** > **Dimension**.
   The properties for the new dimension appear.
3. In the name field, enter the name `Product Line`, and set other properties of this object as follows:
   a. Set the data type to `String` and the **Select** statement to `USER_DATA_CUST_DIM_10.PRODUCT_LINE`.
   b. On the **Advanced** tab, select the **Associate a List of Values** checkbox, and select an appropriate name for the product line list of values, such as `lov_product_line`.
4. Repeat steps 1 through 3 to create the **Product Code** dimension, setting the **Select** statement to `USER_DATA_CUST_DIM_10.PRODUCT_CODE` and the list of values to `lov_product`.
5. Save the universe and export it to the repository.

The sample report that we create offers report users the

opportunity to select one or more products or product lines in which to generate results. One way to utilize this capability is to prepare two filter universe objects that populate two product-oriented user prompts in our report. To do so, first create necessary parameters, and then create fileters based on them:

Creating Parameters

Complete the following steps to add and define the Parameters required for filtering.

1. In the Information Design Tool, click the **Parameters and Lists of Values** class.
2. On the menu bar for Parameters, click **Insert Parameter**.
   The properties for the new Parameter appear.
3. In the **Name** field, enter `Product Line`.
4. In the **Prompt Text** field, enter `Product Line:`.
5. Next to **Associate a List Of Values**, click the **...** button.
   The **Select a List of Values** dialog box appears.
6. Select **List of Values defined in the business layer**, and select **product_line_lov** from the list.
7. Click the **Set default Value** radio button, and from the list, select **ALL**.
8. Save the universe.
9. Publish the universe to a repository.

Creating Filters

Complete the following steps to add and define Filters based on the LOV we previously created.

1. In the Information Design Tool, in the Business Layer list, click the **Business Attribute** class.
2. On the menu bar, click **Insert Object** > **Filter**.
   The properties for the new filter appear.
3. Rename the Filter as **Product Line**, and set its **Where** clause to the following:
   `@Select(Business Attribute\Product Line) IN @Prompt(Product Line:) OR ' ALL' in @Prompt(Product Line:)`
4. On the menu bar, click **Insert Object** > **Filter**.
   The properties for the new filter appear.
5. Rename the Filter as **Product Code**, and set its

	Where clause to the following: `@Select(Business Attribute\Product Code) IN @Prompt(Product Code:) OR ' ALL' in @Prompt(Product Code:)` Refer to BO/BI documentation for more information about how to define filters.
6. Define a hierarchy within the universe for attached data that has parent-child relationships, such as Product Line and Product.	**[+] Show Steps** ## Defining a Hierarchy to the Universe This example created two dimensions that share a parent-child relationship: a product belongs to a specific product line, and a product line consists of one or more products. The custom dimensions that you create might not share this section. We continue this example by defining a Product hierarchy, which makes drill up and drill down functionality available along product lines in the reports that you customize.  Defining the Product Hierarchy 1. Open the Business Layer by clicking the business layer name in the **Local Projects** view. 2. In the Business Layer list, click the **Navigation Path** class. The **Navigation Path** properties appear, as shown in the figure *Defining the Product Hierarchy*. 3. In the **Name** field, type a suitable name for the hierarchy—for example, `Business Attribute - Product`. Keep this hierarchy highlighted. 4. Click **Add**, and in the **Available Dimensions** list, in the **Business Attribute** class, select the **Product Line** and **Product Code** dimensions. 5. Click **OK** to save this new hierarchy and close the editor. 6. Save the universe and publish it to the repository.
7. Save the universe and export it to the repository.	

8. Customize the Interactive Insights reports to include your attached data dimensions.	**[+] Show Steps** ## Creating a Product-Line Business Attribute Report Using the dimensions that you added to the universe in step 5, you can now build one or more business attribute reports that provide the results of your contact center activity by product line and product. The easiest way to create one such report is to make a copy of the Interaction Volume Service Type Report, and tailor it to use the **Product Line** and **Product** dimensions (instead of **Service Type** and **Service Subtype** dimensions) using the following steps: 1. Within BI LaunchPad, copy Interaction Volume Service Type Report to a working folder. 2. Open the report properties, and rename the report appropriately—for example, Interaction Volume Product Line Report. Change its description, as desired, and save your changes. 3. Edit the text on the **Description** tab, as appropriate, and click **Save and Close**. 4. Right-click a report, and in the context menu, select **Modify** to open the report for editing. Complete the following steps: a. On the **Data Access** tab, **Data Providers** sub-tab, click **Edit Data**. b. In the **Query Panel**, browse in the hierarchical list on the left, and double-click the **Product Line** dimension to copy it into the **Result Objects** window. (Do *not* yet remove the **Service Type** dimension from the **Result Objects** window.) c. Close the **Query Panel**. d. Edit the report structure on both the **Summary** and **Main** tabs to replace the 'Service Type' dimension, labels, and text with 'Product Line', wherever it occurs. Do the same for 'Service Subtype', replacing it with 'Product Code'. e. Replace the **Service Type** section with a **Product Line** section: i. Right click the report and choose **Show Structure**. ii. Select the **Service Type** section and change the formula from `=[Service Type]` to `=[Product Line]`.

	f. Click **Edit Data Provider**, and in the **Query Panel**, in the **Query Filters** field, replace the **Service Type** and **Service Subtype** dimensions with **Product Line** and **Product Code**, respectively. Close the **Query Panel**. g. Save and test your changes. 5. When you are satisfied with the report results, click **Edit Data Provider**, and in the **Query Panel**, edit the report query to remove the **Service Type** dimension from the **Result Objects** window, save, and retest the report.

Sample SQL Script for Creating and Mapping User Data

The following sample script provides the SQL code that is used for this example.

[+] Show Sample Code

Note that this is a sample script only. You should validate it for use within your environment.

```
IF EXISTS ( SELECT 1
    FROM sysobjects
    WHERE id = object_id('USER_DATA_CUST_DIM_10') AND type = 'U' )
    DROP TABLE USER_DATA_CUST_DIM_10
GO

CREATE TABLE USER_DATA_CUST_DIM_10 (
    ID               INT identity     ,
    TENANT_KEY       INT NOT NULL     ,
    CREATE_AUDIT_KEY INT NOT NULL     ,
    PRODUCT_LINE     VARCHAR(170) NOT NULL DEFAULT 'none',
    PRODUCT_CODE     VARCHAR(170) NOT NULL DEFAULT 'none',
    DIM_ATTRIBUTE_3  VARCHAR(170) NOT NULL DEFAULT 'none',
    DIM_ATTRIBUTE_4  VARCHAR(170) NOT NULL DEFAULT 'none',
    DIM_ATTRIBUTE_5  VARCHAR(170) NOT NULL DEFAULT 'none',
    CONSTRAINT PK_USER_DATA_CUST_DIM_10 PRIMARY KEY(ID) )
GO

SET IDENTITY_INSERT USER_DATA_CUST_DIM_10 ON;

-- This row is for the predefined key 'UNKNOWN'. It is
-- mandatory. Do not remove it!
INSERT INTO USER_DATA_CUST_DIM_10 (
    ID,
    TENANT_KEY,
    CREATE_AUDIT_KEY )
VALUES ( -1, -1, -1 ) ;
GO

-- This row is for the predefined key 'NO_VALUE'. It is
-- mandatory. Do not remove it!
```

```
INSERT INTO USER_DATA_CUST_DIM_10 (
    ID,
    TENANT_KEY,
    CREATE_AUDIT_KEY )
VALUES ( -2, -2, -1 ) ;
GO

SET IDENTITY_INSERT USER_DATA_CUST_DIM_10 OFF;

-- Add a foreign key reference column from IRF_USER_DATA_KEYS
-- to the user data dimension table.
--
-- Note: Adding columns to a sizeable IRF_USER_DATA_KEYS table
-- could consume significant DBMS resources and time. Consider the
-- tradeoff between:
-- (1) adding redundant columns initially and adding/activating
--     mapping later and
-- (2) adding columns later.

ALTER TABLE IRF_USER_DATA_KEYS
   ADD CUSTOM_KEY_10 INT NOT NULL DEFAULT -2
GO

-- Add mapping between user data dimension table and
-- IRF_USER_DATA_KEYS to CTL_UDE_KEYS_TO_DIM_MAPPING

INSERT INTO CTL_UDE_KEYS_TO_DIM_MAPPING (
    DIM_TABLE_NAME,
    DIM_TABLE_PK_NAME,
    UDE_KEY_NAME )
VALUES (
    'USER_DATA_CUST_DIM_10',
    'ID',
    'CUSTOM_KEY_10' )
GO

-- Add mapping between user data keys and user data tables to
-- CTL_UD_TO_UDE_MAPPING.
--
-- Note: ICON should be configured to record these user data keys.

INSERT INTO CTL_UD_TO_UDE_MAPPING (
    ID              ,
    UD_KEY_NAME     ,
    UDE_TABLE_NAME  ,
    UDE_COLUMN_NAME ,
    PROPAGATION_RULE,
    DEFAULT_VALUE   ,
    ACTIVE_FLAG )
VALUES (
    103                     ,
    'CustomProductLine'     ,
    'USER_DATA_CUST_DIM_10',
    'PRODUCT_LINE'          ,
    'CALL'                  ,
    ''                      ,
    1 )
GO

INSERT INTO CTL_UD_TO_UDE_MAPPING (
    ID              ,
    UD_KEY_NAME     ,
    UDE_TABLE_NAME  ,
```

```
    UDE_COLUMN_NAME ,
    PROPAGATION_RULE,
    DEFAULT_VALUE   ,
    ACTIVE_FLAG  )
VALUES (
    104                     ,
    'CustomProductCode'     ,
    'USER_DATA_CUST_DIM_10',
    'PRODUCT_CODE'          ,
    'CALL'                  ,
    ''                      ,
    1  )
GO
```