Genesys Interactive Insights User's Guide

Example - Product Line and Product

4/2/2025

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 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 8.1 Deployment Guide.
- make_gim_UDE_template.sql script, provided with Genesys Info Mart 8.1 deployment. Note that, beginning with release 8.1.1, 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.
- "Worksheet for Mapping User Data" in the *Genesys Info Mart 8.1 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 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)

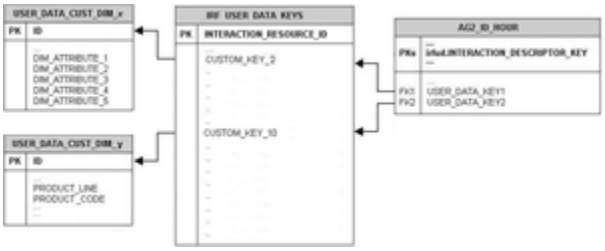
- STRATEGY

- 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 the “Configuring User Data for Aggregation” chapter of the *Reporting and Analytics Aggregates 8.1 User’s Guide*.

1. Create a file, named user-data-map.ss, 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 and place this file in Genesys Info Mart's root directory. After the Genesys Info Mart Server restarts and the aggregation process detects this file, aggregation begins. Refer to the *Reporting and Analytics Aggregates 8.1 Reference Manual* for a data model of the aggregation tables in the Info Mart database, and the relevant Genesys Info Mart 8.1 Reference Manual 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 8.1 Release Notes* • *Interaction Concentrator 8.1 Deployment Guide* • *Interaction Concentrator 8.1 Release Notes*
4. Add the attached data tables to the universe structure.	**[+] Show Steps** ## Adding Attached Data Tables

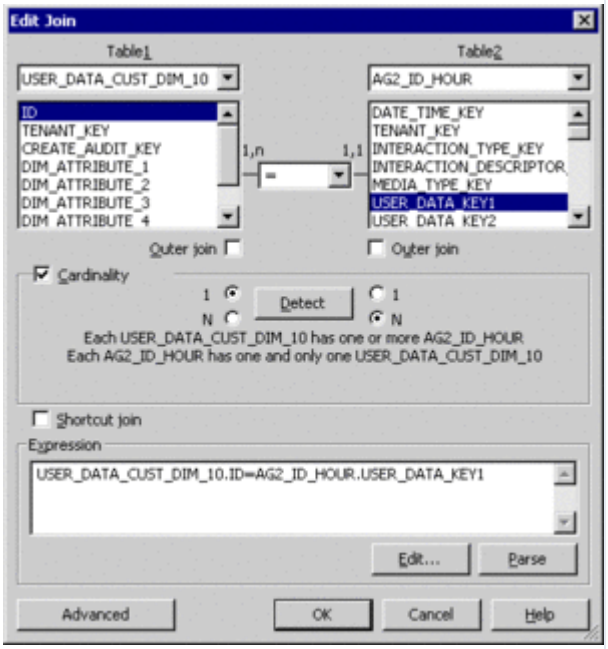
to the Universe

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 Designer. This section illustrates defining joins within Designer.

Beginning with release 8.1.001.x, 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.

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 GI2_Universe in Designer.
2. From the menu bar, select Insert > Table to open the Table Browser and locate your attached data table.
3. Select your table—USER_DATA_CUST_DIM_10, using this example—and click Insert. A symbol for the table appears in the Structure panel.



The Edit Join Dialog Box

4. From the Insert menu, select Join to display

	the Edit Join dialog box, shown in the figure to the right. Here, we will add joins between the attached data table and each variation of an aggregate table: a. From the Table1 drop-down list, select the USER_DATA_CUST_DIM_10 table and field ID immediately below the list. b. From the comparison drop-down list, select the equal sign (=). c. From the Table2 drop-down list, select the AG2_ID_HOUR table and field USER_DATA_KEY1 immediately below the list. d. Set the cardinality for one (1) to many (N). e. Click OK. 5. Reopen the Edit Join dialog box and repeat these steps 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 dimension and condition objects to the universe.	**[+] Show Steps** ## Adding Dimensions and Conditions 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 conditions to the universe. These objects are used in the report that we will create in step 8.

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.

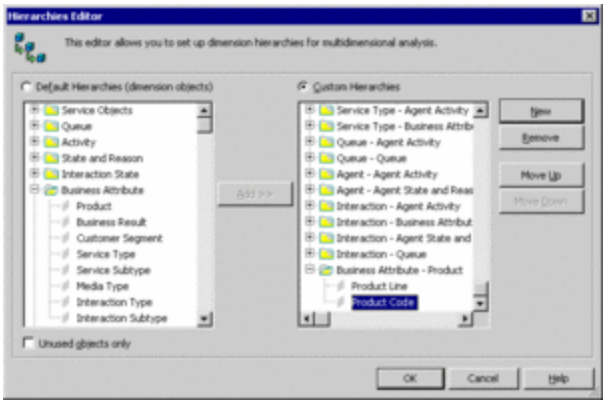
1. Within Designer, in the Classes and Objects pane, click the Business Attribute class.
2. On the menu bar, click Insert Object. The Edit Properties of Object1 dialog box opens.
3. Rename the Object1 to Product Line, and set other properties of this object as follows:
   a. On the Definition tab, set the data type to Character and the Select statement to USER_DATA_CUST_DIM_10.PRODUCT_LINE.
   b. On the Properties tab, click Dimension, and select the following checkboxes:
      - Associate a List of Values
      - Export with universe
   c. In the List Name text box, type an appropriate name for the product line list of values, such as lov_product_line.
4. Click OK to save your changes and close the Edit Properties dialog box.
5. Repeat these steps 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.
6. Save the universe.

Creating Conditions Based on User Data

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 condition universe objects that populate two product-oriented user prompts in our report:

1. Within Designer, in the Classes and Objects panel, click the Conditions radio button and open the Business Attribute class.
2. On the menu bar, click Insert Condition. The Edit Properties of Condition1 dialog box

	opens. 3. Rename Condition1 to Product Line, and set its Where clause to the following: `@Select(Business Attribute\Product Line) IN @Prompt('Product Line:', 'A', 'Business Attribute\Product Line',MULTI, CONSTRAINED,Persistent, {'ALL'},user:13) or 'ALL' in @Prompt('Product Line:','A', 'Business Attribute\Product Line',MULTI, CONSTRAINED,Persistent,{'ALL'}, user:13)` Refer to BO documentation for more information about how to define conditions. 4. Click OK to save your changes and close the Edit Properties dialog box. 5. On the menu bar, click Insert Condition. The Edit Properties of Condition1 dialog box opens. 6. Rename Condition1 to Product Code, and set its Where clause to the following: `@Select(Business Attribute\Product Code) IN @Prompt('Product Code:', 'A', 'Business Attribute\Product Code',MULTI, CONSTRAINED,Persistent, {'ALL'},user:14) or 'ALL' in @Prompt('Product Code:','A', 'Business Attribute\Product Code',MULTI, CONSTRAINED,Persistent,{'ALL'}, user:14)` Refer to BO documentation for more information about how to define conditions. 7. Click OK to save your changes and close the Edit Properties dialog box. 8. Save the universe.
6. Define a hierarchy within the universe for attached data that has parentchild 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 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. From the `Tools` menu in Designer, select `Hierarchies` to open the `Hierarchies Editor`, which is shown in the figure to the right.
2. Click New to add a custom hierarchy and name it appropriately—for example, `Business Attribute - Product`. Keep this hierarchy highlighted.
3. From the `Default Hierarchies` frame, in the `Business Attribute` class, select the `Product Line` dimension and click Add.
4. Select the `Product Code` dimension and click Add.
5. Click 0K to save this new hierarchy and close the editor.
6. Save the universe and export it to the repository.

7. Save the universe and export it to the repository.

[+] Show Steps

Creating a Product-Line Business Attribute Report

8. Customize the Interactive Insights reports to include your attached data dimensions.

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 InfoView, copy Interaction Volume Service Type Report to a working folder.

	2. Open the report's 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. 4. Open and edit the report's query within Web Intelligence. 1. From the Data Manager, drag the Product Line dimension to the Result Objects window. (Do not yet remove the Service Type dimension.) 2. 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. Remove the Service Type section, and replace it with a Product Line section. 3. In the Query Filters window, replace the Service Type and Service Subtype dimensions with Product Line and Product Code, respectively. 4. Save and test your changes. 5. When you are satisfied with the report's results, edit the report query to remove the Service Type dimension from the Result Objects window, 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
```