



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

## Composer Help

Identify Customer Block

# Identify Customer Block

## Contents

- [1 Identify Customer Block](#)
  - [1.1 Name Property](#)
  - [1.2 Block Notes Property](#)
  - [1.3 Extensions Property](#)
  - [1.4 Get Attributes Property](#)
  - [1.5 Exceptions Property](#)
  - [1.6 Suppress Customer Not Found Exception Property](#)
  - [1.7 Customer Attributes Property](#)
  - [1.8 Identification Key Property](#)
  - [1.9 Customer Count Property](#)
  - [1.10 Customer Data Property](#)
  - [1.11 Variables Mapping Property](#)
  - [1.12 Condition Property](#)
  - [1.13 Logging Details Property](#)
  - [1.14 Log Level Property](#)
  - [1.15 Enable Status Property](#)
  - [1.16 ORS Extensions Property](#)
  - [1.17 Using the Identify Contact and Identify Customer Blocks](#)

## Identify Customer Block

---

Use this block to identify a customer in the database based on search criteria, which can be [customer profile data](#) or [customer extension](#) data. If the customer is found, then [Context Services](#) can provide data that can be used to personalize offer or to resume/modify a [service](#) in process.

### Tip

If using Context Services 8.5, the database for service/state information is the Genesys Mobile Services database. If using Context Services 8.1, the database for service/state information is the Universal Contact Server database. For more information, see [General Guidelines for Context Services Preferences](#).

Prior to using this block, set [Context Services Preferences](#). For detailed information on how Universal Contact Server identifies customers, see the [Context Services User's Guide](#). For an example of how to use this block, see the [Getting and Using E-mail Addresses](#) topic.

The Identify Customer block has the following properties. The behavior of some properties can vary depending on whether you are in [offline or online mode](#).

### Name Property

Find this property's details under [Common Properties for Workflow Blocks](#).

### Block Notes Property

Find this property's details under [Common Properties for Workflow Blocks](#).

### Extensions Property

Find this property's details under [Common Properties Context Services](#).

### Get Attributes Property

Use this property to control whether only matching Customer IDs are returned (No) or whether all profile attributes are returned (Yes).

## Exceptions Property

Find this property's details under [Common Properties for Workflow Blocks](#).

You can also define [custom events](#).

## Suppress Customer Not Found Exception Property

- If set to true, no exception will be raised when no customer is found. The Customer Data array will be empty. See the [Customer Data](#) property below.
- If set to false (default), the `error.com.genesyslab.composer.customernotfound` exception is raised when no matching customer is found.

## Customer Attributes Property

Use this property to specify a list of attributes which will be used to search for the customer. To specify customer attributes:

1. Click under Value to display the  button.
2. Click the  button to open the Customer Attributes dialog box.
3. Click **Add** in the dialog box to open the Customer Attribute dialog box.
4. Opposite **Extension**, click the down arrow and select either **Core** (for customer profile [core](#) data) or a customer profile [extension data](#).
  - If you select **Core**, select a core attribute from the **Attribute** dropdown menu. For example, you might select Core and then CustomerSegment.
  - If you select an extension, select the extension attribute name from the **Attribute** dropdown menu.

Note: In offline mode, for both core and extension data, there is an additional field "Attribute Type" where you must choose between string, Boolean, integer, long, double, date, datetime, or currency depending on the customer profile attribute definition.

5. Click the down arrow opposite Type and select **Literal** or **Variable**.
  - If you select **Literal**, then for **Value**, enter the value of the attribute. For example, if you are looking for a customer having LastName=Rosen, you would key in Rosen.
  - If you select **Variable**, select the variable under **Value**.
6. Click OK to close the Customer Attribute dialog box. The Customer Attributes dialog box reflects your entries. You can also use this dialog box to edit and remove entries.

### Identification Key Property

Use this property to specify the name of the key to be used for lookups. If specified, speeds the lookup. The key can be contained in a variable.

1. Click under **Value** to display the  button.
2. Click the  button to open the Identification Key dialog box.
3. Opposite **Type**, click the down arrow and select one of the following:
  - **Context Services**. Then click the *Value* down arrow and select the key. If Context Services attributes have been mapped to Configuration Server Business Attributes, you can select an Identification Key name for Value.
  - **Variable**. Select the name of the variable.
  - **Literal**. Enter the name of the key.

### Customer Count Property

Click the down arrow under Value to select a variable whose value is the number of customer records returned by Universal Contact Server. This feature is for your convenience. It also serves the purpose of retaining the original number of records returned in case the returned data is modified through other blocks.

### Customer Data Property

Click the down arrow under Value to select a variable whose value will be the JSON array containing data returned.

If no matching customers are found, an empty array is returned and an exception will be thrown unless the Suppress Customer Not Found Exception property is set to true.

### Variables Mapping Property

Variables Mapping Property Use this property to map the JSON data returned by this block to variables. See the [Variables Mapping](#) topic for details.

### Condition Property

Find this property's details under [Common Properties](#).

## Logging Details Property

Find this property's details under [Common Properties](#).

## Log Level Property

Find this property's details under [Common Properties](#).

## Enable Status Property

Find this property's details under [Common Properties](#).

## ORS Extensions Property

Starting with 8.1.4, Composer blocks used to build routing applications (with the exception of the Disconnect and EndParallel blocks) add a new [ORS Extensions](#) property.

## Using the Identify Contact and Identify Customer Blocks

- You may use the blocks Identify Contact (UCS – ESP protocol) or Identify Customer (UCS Context Services, if Context Services is enabled – REST protocol).
- In either case, the customer record is fetched into the Composer SCXML application.
- To have the customer data also available in [Interaction Workspace](#), you need to have that customer data stored into the interaction's user data, so Interaction Workspace can access it.
- If using the Identify Customer block, you can then use a User Data block to update the interaction's user data according to your needs.
- If using the Identify Contact block, the generated SCXML code automatically copies the user data part of the UCS response to the interaction's user data.
- If the Update Interaction User Data property value is true, it also automatically copies the parameters part of the UCS response to the interaction's user data.