Developer's Guide

Grouping Customer Profiles

12/14/2025

Contents

Grouping Customer Profiles

> **Important**
> **Prerequisites:** You need to enable profiles in UCS.

This page helps you to learn how you can group customer profiles.

Overview

In the following example, the requirement is to enable an "account" or "family" view of multiple customer profiles. If a new object is created, the profile entity and attribute `Group` is set to `true`. If a multi-valued profile extension points to a group member profile, id-keys can be used. It then becomes possible to either attach services to the group, or to the member depending on the scenario.

Examples

The following examples are similar with the difference being the way the profiles are created. Relationship between profiles and group are either "direct link" from entity account owner to other profiles, or "indirect link" where each profile belongs to a family profile.

Account

There are many Profiles. Some are **Admin** for one or more other Profiles. For example, a telephony provider has a single account (billing account) for several persons. One of the persons is an *admin* for the account and has rights to change options for each cellular.

Example data:

- Profile with Id=**"XXXXXXXXXXX-JOHN":**

```
{
 CustomerId="XXXXXXXXXXX-JOHN",
 LastName:"Doe",
 FirstName:"John",
 Cellular:"555-123456",
 EmailAddress:"john@doe.net",
 GroupAdmin: [
   {AdminForProfile:"XXXXXXXXXXXX-JANE"},
   {AdminForProfile:"XXXXXXXXXXXX-PETER"}
 ],
 ProviderOptions: { AccountInfo:"account number", MemberLevel:"High" }
}
```

- Profile with Id=**"XXXXXXXXXXX-JANE":**

```
{
 CustomerId="XXXXXXXXXXX-JANE",
 LastName:"Doe",
 FirstName:"Jane",
 Cellular:"555-987654",
 EmailAddress:"jane@doe.net",
 ProviderOptions: { AccountInfo:"account number", MemberLevel:"Untouchable" }
}
```

- Profile with Id=**"XXXXXXXXXX-PETER":**

```
{
 CustomerId="XXXXXXXXXX-PETER",
 LastName:"Doe",
 FirstName:"Peter",
 Cellular:"555-654321",
 EmailAddress:"peter@doe.net",
 ProviderOptions: { AccountInfo:"account number", MemberLevel:"Untouchable" }
}
```

- Two extensions for the example:
  - Single-valued extension **ProviderOptions** with any type of attributes used for the example to customize Cellular options.
  - Multi-valued extension **GroupAdmin** with attribute **AdminForProfile**
  - Identification keys :
    - on attribute **EmailAddress** from Core Profile
    - on attribute **Cellular** from Core Profile
    - on attribute **LastName+FirstName** from Core Profile
    - on attribute **AdminForProfile** of extension **GroupAdmin**
- Example scenario:
  - John calls to update his cellular account.
  - He is identified by his Cellular="555-123456".
  - The system knows that he his and admin for Jane and Peter because of the **GroupAdmin** extension.
  - He is asked "Do you want to change settings for your account #1, for Jane's account #2 or for Peter's account #3?".
  - If he enters #2 or #3, the system picks the correct Profile by the Id and can retrieve specific cellular options.
  - ...
  - Peter calls to change some options.
  - He is identified by Cellular="555-654321".
  - The system knows he is not Admin.
  - Depending on the requests, the system may "identify" who is admin for the cellular by the id-key on **GroupAdmin.AdminForProfile="XXXXXXXXXX-PETER"**.

- The system might fail the request stating "need admin rights".

Family

Each member of a family has its own profile. They belong to the Family Group (same house hold). This result in slightly different from the previous example because the Family itself is identified as a profile.
Note: The Account example can also be implemented this way.

Example data:

- Profile with Id=**"XXXXXXXXXXX-JOHN"**:

```
{
 CustomerId="XXXXXXXXXXX-JOHN",
 LastName:"Doe",
 FirstName:"John",
 Cellular:"555-123456",
 EmailAddress:"john@doe.net",
 GroupFamily: { ProfileFamily:"XXXXXXXXXXX-DOE" }
}
```

- Profile with Id=**"XXXXXXXXXXX-JANE"**:

```
{
 CustomerId="XXXXXXXXXXX-JANE",
 LastName:"Doe",
 FirstName:"Jane",
 Cellular:"555-987654",
 EmailAddress:"jane@doe.net",
 GroupFamily: { ProfileFamily:"XXXXXXXXXXX-DOE" }
}
```

- **Family = Profile 'XXXXXXXXXXX-DOE'**:

```
{
 CustomerId="XXXXXXXXXXX-DOE",
 LastName:"Doe",
 PhoneNumber:"555-1592648",
 EmailAddress:"family@doe.net",
 PostalAddress: { Address:"5, This Road", ZipCode:65536 }
}
```

- Extensions:
  - Single-valued extension **PostalAddress** with attributes like 'Zip Code', 'State', etc. This extension may have values only for the **Family** since all profiles are to live at the same place.
  - Single-valued extension **GroupFamily** with attribute **ProfileFamily** pointing to the main family profile.
- Identification keys:
  - on attribute **EmailAddress** from Core Profile.
  - on attribute **Cellular** from Core Profile.
  - on attribute **PhoneNumber** from Core Profile.

    - on attributes **LastName+FirstName** from Core Profile.
    - on attribute **ProfileFamily** from extension **GroupFamily**.

- Example scenario:
    - Assuming John sends the request from e-mail or cellular:
        - He is identified by id-key **Profile.Cellular="555-123456"** or **Profile.EmailAddress= "john@doe.net"**.
        - Then his family information is gathered from querying Profile with Id **GroupFamily.ProfileFamily="XXXXXXXXXXXX-DOE"**.
    - Assuming John calls from Home:
        - His **Family** information is matched by id-key **Profile.PhoneNumber="555-1592648"**.
        - Members of the family can be identified by id-key on **GroupFamily.ProfileFamily= "XXXXXXXXXXXX-DOE"**.
        - The IVR might question "Who are you? Jane or John?".