



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.

Developer's Guide

Profiles and Identification

12/15/2025

Profiles and Identification



Purpose: Gives guidelines for managing Customer Profiles with Context Services.

Contents

- [1 Profiles and Identification](#)
 - [1.1 Learn about the Customer Profile and Associated Resources](#)
 - [1.2 Preliminary Steps to Using Profiles](#)
 - [1.3 Managing Profiles: Creation, Identification, Extensions](#)
 - [1.4 References](#)

In versions 8.1.000.10 and higher, you need to check the privileges set according to roles prior to using the operations described on this page. See [Role-Based Access Control](#) for additional details.

Learn about the Customer Profile and Associated Resources

The [Customer Profile](#) resource associates a customer ID with:

- A list of attributes, built on top of the existing UCS<ref>UCS: Universal Contact Server</ref> Contact model.
- A list of extensions, defined at runtime through Context Services

As stated in [Profile Basics](#), the attributes correspond to Core information in UCS. Their schema fulfills defined through the list of Business Contact Attributes that you can define in Configuration Manager (see [Configuration Options](#) for additional details). The extensions are additional information that your application can create so that you can extend the profile at runtime, as explained in [Extending the Customer Profile](#). Identification Keys define which attributes should be used to identify a customer in the database. For instance, the association of [LastName, Firstname], or the e-mail address.

Attribute Values

Attributes and extension records can be either:

- "single-valued"(for instance, LastName, FirstName, identifiers, and so on);
- "multi-valued": values can be multiple (for instance, phone numbers, e-mail addresses, and so on).

The following output presents a sample of Customer profile, where "FirstName", "LastName", and "DOB" (Date Of Birth) are single-valued contact attributes, and the other fields are multi-valued extension records created at runtime.

```
{
  "FirstName": "Bruce",
  "LastName": "Banner",
  "DOB": "1962-05-10",
  "EmailAddress": [
    "bruce.banner@marvelous.com",
    "b.banner@hulk.dom"
  ],
  "Phone": [
    {
      "PhoneType": 0,
      "prefix": "+33",
      "PhoneNumber": "3145926535",
      "description": "family phone",
      "start_availability": "2009-12-18T18:30:00.000Z",
      "end_availability": "2009-12-18T21:40:00.000Z"
    },
    {
      "PhoneType": 2,
      "prefix": "+33",
      "PhoneNumber": "6543210",

```

```
    "description": "business calls only, no sales",
    "start_availability": "2009-12-18T09:30:00.000Z",
    "end_availability": "2009-12-18T17:45:00.000Z"
  },
  {
    "PhoneType": 5,
    "prefix": "+33",
    "PhoneNumber": "951357456",
    "description": ""
  }
]
```

Profile Content

The content of the Customer Profile follows a schema (a translation of the Business Contact Attributes), which describes its content with a list of [Attribute Schema](#), apart from extension content, as shown in the following output example:

```
{
  "encrypt": false, "name": "PIN", "length": 256, "type": "string"},
  {"encrypt": false, "name": "Title", "length": 256, "type": "string"},
  {"encrypt": false, "name": "CustomerSegment", "length": 256, "type": "string"},
  {"encrypt": false, "name": "LastName", "length": 256, "type": "string"},
  {"encrypt": false, "name": "AccountNumber", "length": 256, "type": "string"},
  {"encrypt": false, "name": "FirstName", "length": 256, "type": "string"},
  {"encrypt": false, "name": "PhoneNumber", "length": 256, "type": "string"},
  {"encrypt": false, "name": "ContactId", "length": 256, "type": "string"},
  {"encrypt": false, "name": "EmailAddress", "length": 256, "type": "string"},
}
```

The profile schema does not contain information related to extensions.

At runtime, your application can retrieve this schema through the [Query Profile Schema](#) operation. Your application cannot modify the profile schema through Context Services:

- If you wish to modify the profile schema, make modifications to in Configuration Manager via the Business Attribute "ContactAttributes".
- If you wish to add additional information to the profile at runtime, use the profile extensions.

Extensions

Extensions allow you to extend the content of a profile with additional records. Your application is fully responsible for the creation and management of this optional content. As stated in the [Preliminary Steps to Using Profiles](#) section, your application must define the extensions' content by creating the extension. Once the schema is defined, your application can use the extensions by specifying extension information to the profile submitted with the operations' body, or by using dedicated operations, as presented below.

Identification Key

The identification key is a combination of attributes used to identify a customer. These attributes (one or more) belong to the customer profile and/or its extensions.

- ⚠ Identification keys cannot be issued from both the profile and its extensions.

Some identification key samples:

- An identification key consisting of the "name" and "birthdate" attributes from the profile.
- A key based on the "pin" attribute of a single-valued extension used to hold customer pass codes.
- A key based on the "number" attribute of a multi-valued extension used to record the phone numbers we have seen a customer call from.

When UCS receives a new customer profile and the associated extension data, it builds the indexing structures according to the specified identification keys, in order to ensure efficient customer identification.

 Read [Identification Key](#) for a resource example.

Preliminary Steps to Using Profiles

Before your application can start interacting with customers and recording their information by using [Customer Profile](#) resources, you must create certain types of information that enable you to manage these profiles. This includes the profile schema, extension schemas, and identification keys.

Creating Extension Schema (Optional)

Operations and resources in this section are part of

If your application only needs to use the predefined profile attributes, you can use them as provided. But if you need to work with information that is not contained in the existing attributes -- for instance, for use as identification keys -- you can create extensions that contain this extra information by creating an extension schema for each additional piece of information.

 Your application cannot use an extension if the associated schema does not exist.

As for the profile schema, the profile extension schema is composed of [Attribute Schema](#) which define the schema content. The following output shows the definition of the single-valued "Address" extension.

```
"name": "Address",
"type": "single-valued",
"attributes": [
  {"name": "AddressType", "type": "integer", "default": 0},
  {"name": "Address", "type": "string", "length": 256},
  {"name": "City", "type": "string", "length": 32},
  {"name": "County", "type": "string", "length": 32},
  {"name": "PostCode", "type": "string", "length": 10},
  {"name": "Country", "type": "string", "length": 32}
]
```

The creation of the profile extension is possible through the [Create Profile Extension Schema](#) operation. See [this example section](#) for detailed information.

Creating Identification Keys (Mandatory)

Operations and resources cited in this section are part of **Schema Operations**

If your application does not specify identification keys, the only way to access customer information is to **retrieve the customer profile** with the customer ID. However, you can use the **Create Identification Key** operation to set up an **Identification Key**. You create these keys by using profile attributes. For example, you can use a single attribute, or a combination of attributes ("LastName" and "FirstName" for instance), or a combination of extension attributes if you have created a schema for a profile extension. You can define as many identification keys as you need, but you should consider that creating too many identification keys will slow down creation, update, and removal operations. Once identification keys have been created, your application can continue to work with customer data, accessing it by means of the **Get Identification Keys** operation.

Your application only needs to register these schemas and identification keys once.

Managing Profiles: Creation, Identification, Extensions

The Customer Profile and its extensions should only be created for the storage of customer-level information. You should use the results of interactions and dialogs with the customer to fill in service, state, and task information.

Managing a Profile

The **Create Customer Profile** operation creates an entry for the Customer Profile in UCS Database. This step must be completed prior to using the profile data in other operations such as:

- **Query Customer Profile**
- **Update Customer Profile**

Identify a Customer

Operations and resources in this section are part of **Schema Operations**

The operation **Identify Customer** enables your application to retrieve customer profiles based on a few attribute values passed in as parameters, without specifying the customer ID, as shown in this example:

```
GET /profiles/  
contacts.phone_number=408-888-3214&extensions=contacts,purchases&exclude_profile=yes  
&exclude_extensions=unique
```

If no customer profile is returned, your application can create a new profile based on the current set of information available.

This enables your application to determine the customer's identity without having to gather as much information. For instance, if your application deals with

calls by using a customer phone number, the customer is easy to identify if he or she calls back.

Extending the Customer Profile

As stated in the [introduction sections](#), your application can add new types of information to the basic customer profile by using extensions. If your application needs to record a specific set of data (for instance, e-mail addresses), your application can create a schema for this extension. Once an [extension schema has been created in UCS](#), your application can use the new extension and create associated records. Your application can either:

1. [Insert Extension Records](#) for a given customer, or
2. [Create Customer Profile](#) or [Update Customer Profile](#) with extension records.

Importing Customer Profiles

Available since: 8.1.000.10

The [Bulk Profile Import](#) operation enables to import a .csv file which contains a wide set of contacts. The .csv file must be compliant with [RFC4180](#) and already available on the UCS local file system. Profiles must match metadata and can include extensions. An identification key can be set to avoid ambiguities with former profiles when updating the profile database. See [Bulk Profile Import](#) for examples and details.

References

<references />