



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

Profile Extensions

12/14/2025

Contents

- [1 Profile Extensions](#)
 - [1.1 About Profile Extensions](#)
 - [1.2 Managing Extension Schema](#)
 - [1.3 Managing Extensions](#)
 - [1.4 Read More](#)

Profile Extensions

Important

Prerequisites: You need to [enable](#) profiles in UCS.

This page provides guidelines for managing Profile Extensions in UCS.

About Profile Extensions

Profile extensions are stored in the UCS database and are not JSON values as service extensions. Their management is similar to former extension management in Context Services. You must define database schemas before you can use them in your application.

Profile extensions are additional information which extend the standard contents of resources such as [Customer Profile](#). A [Extension](#) is a record—a list of attributes—or an array of records, associated with a resource ID.

- You can define as many extension types as you need by creating an [Extension Schema](#) for each of them.
- Extension schema are created through Context Services (see [List of Schema Operations](#)), not through the Configuration Layer (Configuration Manager).

Extension records can be either:

- "single-valued": The extension contains a single record across the resource (for instance, LastName, FirstName, identifiers, etc.)
- "multi-valued": The extension can contain several values (for instance, phone numbers, e-mail addresses, etc.)

Extensions are provided at the same time and at the same level than the attributes of the resource. For instance, the following output presents a profile containing the attributes FirstName, LastName, DOB<ref>DOB: Date Of Birth</ref> and one multi-valued extension *EmailAddress*:

```
{
  "FirstName": "Bruce",
  "LastName": "Banner",
  "DOB": "1962-05-10",
  "EmailAddress": [
    "bruce.banner@marvelous.com",
    "b.banner@hulk.dom"
  ]
}
```

<references />

Unique Attributes

In the case of multi-valued extensions, the attributes which are part of the 'unique' list (specified in the [Extension Schema](#)) are used to identify records. The combination of these attributes' values must be unique across the related resource, and this enables UCS to identify a given record in the given extension. For example, consider a 'Bill' extension which includes the attribute *bill_id*. To ensure that a given service does not have two 'Bill' extensions with the same *bill_id*, set the following unique array in the extension schema:

```
unique = ["bill_id"]
```

The attributes of the unique list are mandatory at the extension record's creation. You need to provide values for the 'unique' attributes:

- At the creation of an extension record.
- In operations which update or delete a specific record, such as [Update Record In Profile Extension](#) or [Delete Record From Profile Extension](#).

Warning

Operations which manage extension records are part of the related resource operations. For instance, the operations which manage records of profile extensions are part of the [List of Profile Operations](#).

Limitations

- Once created, you cannot update the schema.
- When you are dealing with extensions or extension schema, make sure that you do not use one of the

[Unauthorized Strings](#) as an attribute name or value.

Managing Extension Schema

Operations and resources in this section are part of

Before you can start using extensions, you must create their schema.

Important

Once created, you cannot update or remove them.

You can create schema with the following operations:

- [Create Profile Extension Schema](#)

Then, you can retrieve extension schema.

- [Query Profile Schema](#)
- [Query Profile Extension Schema](#)

Example: Retrieving the schema for profile extensions:

```
GET /metadata/profiles/extensions
```

Result

```
200 OK
[
  {
    "name": "Phone",
    "type": "multi-valued",
    "attributes": [
      {"name": "PhoneType", "type": "integer", "default": 0, "mandatory": "true"},
      {"name": "prefix", "type": "string", "length": "3", "default": "555", },
      {"name": "PhoneNumber", "type": "integer", "length": 15, "mandatory": "true"},
      {"name": "description", "type": "string", "length": 32, "mandatory": "true"},
      {"name": "start_availability", "type": "datetime"},
      {"name": "end_availability", "type": "datetime", "mandatory": "false"}
    ]
  },
  {
    "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}
    ]
  }
]
```

Managing Extensions

Adding Extensions to a given Resource

You can add extensions when managing the resources with related operations which authorize the `<extension n>` attribute in the operation's body. In that case, if a former value of the extension exists for the given resource, this former extension value is replaced with the new extension value specified in the body.

Let's consider the following multi-valued extension record named 'Satisfaction'. The unique field which identifies records is "place" (the name of the proposed place for the booking).

Example: Records for a 'Satisfaction' extension

```
PUT /profiles/00027a52JCGY000M
{
  "FirstName": "Bruce",
  "LastName": "Banner",
  "DOB": "1962-05-10",
  "EmailAddress": [
    "bruce.banner@marvelous.com",
    "b.banner@hulk.dom"
  ],
  "Address": { "Type":1, "Address":"21 JumpStreet", "City":"Hollywood",
    "County":"Santa Barbara", "PostCode":"555", "Country":"United States" }
}
POST /profiles/00027a52JCGY000M/extensions
{
  "customer_id":"00027a52JCGY000M",
  "Satisfaction": [
    {
      "rating":2,
      "pertinence":8,
      "usefull":true,
      "place":"Terranova mexico resort"
    },
    {
      "rating":8,
      "pertinence":4,
      "usefull":false,
      "place":"Fancy resort Paris"
    }
  ]
}
```

Example: Operation which updates the 'Satisfaction' extension

```
POST /profiles/00027a52JCGY000M/extensions
{
  "customer_id":"00027a52JCGY000M",
  "Feedback":
  {
    "FeedbackType":"survey",
    "rating":7,
    "notes":"warm welcome at frontdesk, thanks for the nice trip"
  },
  "Satisfaction": [
    {
      "rating":2,
      "pertinence":6,
      "usefull":true,
      "place":"Marina Porto Vecchio"
    }
  ]
}
```

As a result, the previous records 'Fancy resort Paris ' and 'Terranova mexico resort ' are lost. In this case, to add a new record to the extension, you must specify the whole extension content. For instance, note the following:

Example: Operation which updates the 'Satisfaction' extension without losing records

```
POST /profiles/00027a52JCGY000M/extensions
{
  "customer_id": "00027a52JCGY000M",
  "Feedback": {
    "FeedbackType": "survey",
    "rating": 7,
    "notes": "warm welcome at frontdesk, thanks for the nice trip"
  },
  "Satisfaction": [
    {
      "rating": 2,
      "pertinence": 6,
      "usefull": true,
      "place": "Marina Porto Vecchio"
    },
    {
      "rating": 2,
      "pertinence": 8,
      "usefull": true,
      "place": "Terranova mexico resort"
    },
    {
      "rating": 8,
      "pertinence": 4,
      "usefull": false,
      "place": "Fancy resort Paris"
    }
  ]
}
```

Retrieving Extensions

GET operations which enable to retrieve resources include the "extensions" parameter to specify a list of extensions to retrieve. By default, extensions are not returned. The following list is not exhaustive:

- [Query Customer Profile](#)

Deleting an Extension

To delete the extension of a given resource, use the related *Update XXX Extension* operation with no attributes in the operation's body.

- [Update Customer Profile](#) and [Update Record In Profile Extension](#)

Example: Deleting the *relatedOffers* multi-valued extension of the customer *00027a52JCGY000M*

```
PUT
POST /profiles/00027a52JCGY000M/extensions

[]
```

As explained in [Role-Based Access Control](#), you need update privileges to clear the extension, as follows:

- Clear Profile Extension—UCS.Customer.updateProfileExtension

Read More

- [Profiles and Identification](#)