FUJITSU

FUJITSU Software Enterprise Service Catalog Manager V17.6

Broker's Guide

January 2018

Trademarks	LINUX is a registered trademark of Linus Torvalds. Open Service Catalog Manager is a registered trademark of FUJITSU LIMITED. The OpenStack Word Mark and OpenStack logo are registered trademarks/service marks or trademarks/ service marks of the OpenStack Foundation in the United States and other countries. Apache Tomcat, Tomcat, and Apache are trademarks of The Apache Software Foundation. Java is a registered trademark of Oracle and/or its affiliates. UNIX is a registered trademark of the Open Group in the United States and in other countries. Other company names and product names are trademarks or registered trademarks of their respective owners.
Copyright FUJITSU ENABLING SOFTWARE TECHNOLOGY GMBH 2018	All rights reserved, including those of translation into other languages. No part of this manual may be reproduced in any form whatsoever without the written permission of FUJITSU ENABLING SOFTWARE TECHNOLOGY GMBH.

High Risk Activity

The Customer acknowledges and agrees that the Product is designed, developed and manufactured as contemplated for general use, including without limitation, general office use, personal use, household use, and ordinary industrial use, but is not designed, developed and manufactured as contemplated for use accompanying fatal risks or dangers that, unless extremely high safety is secured, could lead directly to death, personal injury, severe physical damage or other loss (hereinafter "High Safety Required Use"), including without limitation, nuclear reaction control in nuclear facility, aircraft flight control, air traffic control, mass transport control, medical life support system, missile launch control in weapon system. The Customer shall not use the Product without securing the sufficient safety required for the High Safety Required Use. In addition, FUJITSU (or other affiliate's name) shall not be liable against the Customer and/or any third party for any claims or damages arising in connection with the High Safety Required Use of the Product.

Export Restrictions

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

Contents

About this Manual 4
1 Introduction 6
1.1 The Broker's Tasks in ESCM 6
1.2 Accessing ESCM 7
2 Service Management 9
2.1 Viewing Services 9
2.2 Publishing Services 9
3 Customer Management 12
3.1 Registering Customers 12
3.2 Configuring an External User Authentication System 12
3.3 Handling Subscriptions 13
4 Billing and Payment 14
5 Reporting 15
Appendix A LDAP Keys 16
Appendix B Customer Billing Data 18
Glossary 35

About this Manual

This manual describes how brokers can support service suppliers in establishing relationships to customers with) FUJITSU Software Enterprise Service Catalog Manager, hereafter referred to as ESCM.

This manual is structured as follows:

Chapter	Description
Introduction on page 6	Outlines the role of a broker in ESCM and gives an overview of the broker's tasks.
Service Management on page 9	Describes the basic concepts of services and explains how to publish services to a marketplace.
Customer Management on page 12	Describes the different tasks involved in customer management.
Billing and Payment on page 14	Describes how ESCM supports brokers in calculating and collecting their revenue shares.
Reporting on page 15	Describes the reports available for brokers in ESCM.
Menu Options and User Roles	Gives an overview of the menu options relevant for brokers.
LDAP Keys on page 16	Lists the keys to be defined for enabling access to a customer's LDAP system.
Customer Billing Data on page 18	Describes the elements of an XML file created by exporting customer billing data.

Readers of this Manual

This manual is directed to people who want to offer services as defined by suppliers to customers for subscription and use. It assumes that you are familiar with the ESCM concepts as explained in the *Overview* manual.

Notational Conventions

This manual uses the following notational conventions:

Add	Names of graphical user interface elements.
`init`	System names, for example command names and text that is entered from the keyboard.
`<variable>`	Variables for which values must be entered.
`[option]`	Optional items, for example optional command parameters.
`one | two`	Alternative entries.
`{one | two}`	Mandatory entries with alternatives.

Abbreviations

This manual uses the following abbreviations:

ESCM	Enterprise Service Catalog Manager
IaaS	Infrastructure as a Service
LDAP	Lightweight Directory Access Protocol
PaaS	Platform as a Service
SaaS	Software as a Service

Available Documentation

The following documentation on ESCM is available:

- *Overview:* A PDF manual introducing ESCM. It is written for everybody interested in ESCM and does not require any special knowledge.
- *Operator's Guide:* A PDF manual for operators describing how to administrate and maintain ESCM.
- *Technology Provider's Guide:* A PDF manual for technology providers describing how to prepare applications for usage in a SaaS model and how to integrate them with ESCM.
- *Supplier's Guide:* A PDF manual for suppliers describing how to define and manage service offerings for applications that have been integrated with ESCM.
- *Reseller's Guide:* A PDF manual for resellers describing how to prepare, offer, and sell services defined by suppliers.
- *Broker's Guide:* A PDF manual for brokers describing how to support suppliers in establishing relationships to customers by offering their services on a marketplace.
- *Marketplace Owner's Guide:* A PDF manual for marketplace owners describing how to administrate and customize marketplaces in ESCM.
- *OpenStack Integration:* A PDF manual for operators describing how to offer and use virtual systems controlled by OpenStack through services in ESCM.
- *Amazon Web Services Integration:* A PDF manual for operators describing how to offer and use virtual servers controlled by the Amazon Elastic Compute Cloud Web service through services in ESCM.
- *Online Help:* Online help pages describing how to work with the administration portal of ESCM. The online help is intended for and available to everybody working with the administration portal.

1 Introduction

Enterprise Service Catalog Manager (ESCM) is a set of services which provide all business-related functions and features required for turning on-premise applications and tools into "as a Service" (aaS) offerings and using them in the Cloud. This includes ready-to-use account and subscription management, online service provisioning, billing and payment services, and reporting facilities.

With its components, ESCM supports software vendors as well as their customers in leveraging the advantages of Cloud Computing.

The basic scenario of deploying and using applications as services in the ESCM framework involves the following organizations:

- **Technology providers** (e.g. independent software vendors) technically prepare their applications for usage in the Cloud and integrate them with ESCM. They register the applications as technical services in ESCM.
- **Suppliers** (e.g. independent software vendors or sales organizations) define service offerings, so-called marketable services, for the technical services in ESCM. They publish the services to a marketplace.
- **Customers** register themselves or are registered by an authorized organization in ESCM and subscribe to one or more services. Users appointed by the customers work with the underlying applications under the conditions of the corresponding subscriptions.
- **Marketplace owners** are responsible for administrating and customizing the marketplaces to which services are published.
- **Operators** are responsible for deploying and maintaining ESCM.

  ESCM is provided in Docker containers and deployed in a container environment such as OpenStack or a Kubernetes cluster providing for optimum performance, scalability, failover, and non-stop operation. The applications integrated with ESCM and their data may be hosted in the same Virtual Machine (VM) as ESCM or in different locations.

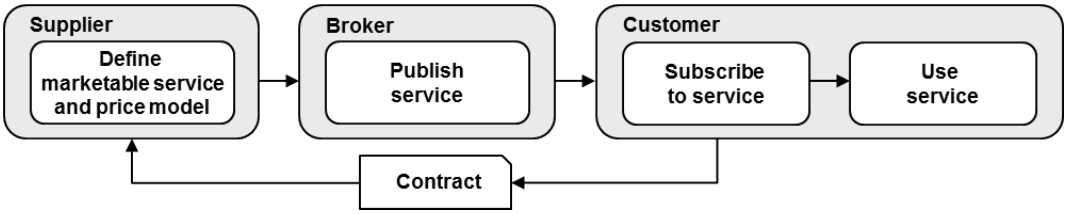
In extended usage scenarios, the suppliers who define marketable services may involve additional users and organizations in offering and selling these services:

- **Brokers** support suppliers in establishing relationships to customers by offering the suppliers' services on a marketplace. A service subscription is a contract between the customer and the supplier.
- **Resellers** offer services defined by suppliers to customers applying their own terms and conditions. A service subscription establishes a contract between the customer and the reseller.

1.1 The Broker's Tasks in ESCM

Your main task as a broker is the promotion of marketable services defined by one or more suppliers. To this end, you publish the services to marketplaces to which you have been granted

access. A subscription to a service establishes a contract between the customer and the supplier of the service.



In order to offer a service on a marketplace, you need an explicit permission by the supplier. The supplier defines the price model for the service as well as the terms and conditions. As a broker, you cannot change any of these.

For subscribing to services you offer, customers need to register with ESCM, or you can register them. The customers can register or import any number of users who work with the services. As a broker, you can view the customer subscriptions, but you cannot change or terminate them.

Brokers usually receive a share of the revenue for the services they offer. This revenue share is defined by the operator. It may be different for each marketplace where you offer services or even for individual services. You can at any time retrieve billing data and information on your actual revenue depending on the usage of the services. The data can be used as the basis for collecting the shares from the suppliers or marketplace owners, depending on who is responsible for paying you according to your contracts.

If required, you can also act as a customer in ESCM. As a broker, you are automatically privileged to subscribe to services and work with the services you have subscribed to.

If you need to perform marketplace owner tasks, your organization must be assigned the corresponding role by the ESCM operator. Note that a broker organization cannot additionally be assigned the technology provider, supplier, or reseller role.

1.2 Accessing ESCM

To perform your tasks, you use the ESCM user interface. The role of your organization as a broker and your user role within the organization determine which features are available to you at the ESCM user interface.

ESCM distinguishes between the following user roles within broker organizations:

- **Administrator**: Each organization must have at least one user with this role. An administrator can manage the organization's account and subscriptions as well as its users and their roles. The first administrator of an organization is defined when the organization is created.
- **Broker**: This role allows a user to publish a supplier's marketable services without changing the terms and conditions defined by the supplier.

To access the ESCM user interface, you use the login information provided by ESCM in the email confirming the creation of your user account. If your organization uses an external authentication system, passwords are managed in this system. This means that you log in with the password as stored in this system, and the email sent by ESCM does not contain a password.

To log in to the administration portal where you will perform your tasks:

1. Click the link provided in the email, or type the access URL in your Web browser's address bar.

   The access URL has the following format:

   ```
   https://<hostname.fqdn>:<port>/oscm-portal
   ```

`<hostname.fqdn>` is the name and the fully qualified domain name of the machine where ESCM has been deployed. `<port>` is the port to address the machine (default: `8081`), `oscm-portal` is the default context root of ESCM and cannot be changed.

2. On the **Login** page, type your user ID and password.
3. Click **Login**, or press **Return**.

You are either logged in directly, or you are prompted to change your initial password when you log in for the first time. It is highly recommended to change the initial password.

If you try to log in with a wrong password, your account may be locked after the third attempt. This depends on whether your organization maintains its user data in an external authentication system. In this case, passwords can only be changed or reset in this system. If user data are maintained in the platform, contact your administrator who can reset your password. You will get a new temporary password for your next login.

If you have forgotten your password, click **Forgot your password?** on the **Login** page. This allows you to define a new password for your user ID. Defining a new password is not possible if your account is locked or if your organization maintains its user data in an external authentication system.

If you have forgotten your user ID, contact your administrator who can look up the IDs of all users registered for your organization.

If your session expires, you have to log in again.

2 Service Management

Your main task as a broker is to support suppliers in establishing relationships to customers. To this end, you promote the suppliers' services by offering them on a marketplace to which you have been granted access by the marketplace owner.

The supplier of a service is responsible for:

- Defining the service and specifying its parameters, conditions, and restrictions.
- Defining a price model for the service which specifies whether and how much customers subscribing to the service will be charged for using it.
- Defining a license agreement for the price model which is presented to customers and needs to be accepted by them.
- Granting you the permission to publish the service on a marketplace of your choice.

As a broker, you cannot change the service definition, the price model, and the terms and conditions. It is up to the supplier to change these elements, if required.

When you are granted the permission to publish a service, you can view it and publish it to a marketplace. These tasks are described in more detail in the sections below.

2.1 Viewing Services

You can view the details of all services for which you have been granted a resale permission. The following information on a service is available:

- The service name and short description which are shown in the service catalog on a marketplace when you publish the service.
- The service description which provides details of the features offered by the service. It is displayed on a marketplace when customers request details of the service.
- The license agreement which must be read and accepted by customers when they subscribe to the service.
- The details of the price model defined by the supplier of the service. The service can be free of charge or with costs. If customers are charged for using the service, you can view the charging conditions and prices.

To view the details of a service, use the **View service** menu option in the **Marketable service** menu of the ESCM administration portal.

2.2 Publishing Services

Publishing a service means making it available to customers for subscription on a marketplace.

The following prerequisites must be fulfilled for publishing a service:

- You need the permission to publish the service by the supplier.
- You must have been granted access and admitted as a seller to the marketplace by the marketplace owner.

Publishing a service involves selecting the marketplace, activating the service, and publishing the URL of the marketplace. These tasks are described in more detail in the sections below.

Specifying a Marketplace

As a prerequisite for publishing a service, you need to specify the desired marketplace.

A service can be published to exactly one marketplace. If you want to publish it to a different marketplace, you have to deactivate it before you can select the new marketplace.

Specifying a marketplace includes:

- Deciding whether the service is to be public.

  A public service is visible in the service catalog for anybody accessing the marketplace. A service that has not been marked as public is available for subscription to registered customers only.

- Selecting the marketplace.

  You can publish the service to a marketplace that is open to any seller, or to a marketplace that is not open but for which you have been admitted as a seller by the marketplace owner. The marketplace must be public or you need to be granted access by the marketplace owner.

- Assigning categories.

  Provided that categories are defined for the marketplace you select, you can assign one or multiple categories to your service. The marketplace owner is responsible for creating the categories you can choose from. Customers can use the categories for browsing the service catalog and filtering the services on the marketplace. The marketplace owner may also decide to display the services on the marketplace home page grouped by categories.

To specify the marketplace, use the **Define publishing options** menu option in the **Marketable service** menu of the ESCM administration portal.

Activating a Service

To make a service available for subscription, you must activate it. To do this, use the **Activate or deactivate services** menu option in the **Marketable service** menu of the ESCM administration portal.

The menu option can also be used for deactivating a service. This is required if you do not want to offer it any longer or if you want to change the marketplace. If you deactivate a service, customers can no longer subscribe to it. Existing subscriptions are not affected.

Publishing the URL of a Marketplace

To make your service offering known to customers, you publish the URL of the marketplace. You can do this, for example, by email or on your website or portal. You can obtain the URL from the marketplace owner, who receives it from ESCM via email when the marketplace is created.

The URL has the following format:

`https://<hostname.fqdn>:<port>/oscm-portal?marketplaceId=<mId>`

`<hostname.fqdn>` is the name and the fully qualified domain name of the machine where ESCM has been deployed. `<port>` is the port to address the machine (default: `8081`), `oscm-portal` is the default context root of ESCM and cannot be changed. `<marketplaceId>` is the ID of the marketplace.

People who use the URL for accessing a marketplace can view all services that are marked as public. When subscribing to a service, they are either redirected to the registration Web page or they can log in to ESCM directly, provided they are already registered as a customer.

You can also publish the URL of the registration Web page of ESCM. This allows customers to directly register with ESCM. Registered customers can see all services, no matter whether they are marked as public or not. A registered customer is sent an email with the login data and an initial password to be used for the first login.

The URL of the registration Web page has the following format:

`https://<hostname.fqdn>:<port>/oscm-portal/marketplace/registration.jsf?marketplaceId=<mId>`

Note: The marketplace owner may restrict the access to a marketplace to the users of specific organizations. In this case, the users need to log in to be able to see, subscribe to, and use the services published on the marketplace. New customers cannot register themselves on the marketplace but need to be registered by a supplier, broker, reseller, or the platform operator.

3 Customer Management

ESCM offers various features for managing your organization's interaction with customers. Customer management involves the following tasks:

- Registering customers.
- Configuring an external user authentication system, if required.
- Handling subscriptions.

These tasks are described in more detail in the sections below.

3.1 Registering Customers

You can register customers with ESCM so that they do not need to do this themselves. The new customer automatically receives the login information that enables him to access the given marketplace and subscribe to services.

Registering a customer includes entering the following user data for the first administrator of the new customer organization:

- **Title**, **first name**, and **last name** of the administrator.
- **Email address** that is used for notifying the customer about the registration.
- **User ID** to be used for the administrator.
- **Language** in which the user interface, available documentation, and emails are presented to the administrator.

Registering a customer includes entering the data for the new organization:

- **Name**, **address**, and **phone number**.
- **Email address**.
- **Website**.
- **Country** where the customer organization is located.
- **Description** that provides, for example, information from the company's profile.
- **Marketplace** to which the customer is directed.

To register a customer, use the **Register customer** menu option in the **Customer** menu of the ESCM administration portal.

Once you have registered a customer, the organization data can only be edited by the customer's administrators. If you want to update the organization data, you must be registered as an administrator of your customer.

As a broker, you can view the organization data that is defined for your customers. To view the data, use the **View customer** menu option in the **Customer** menu of the ESCM administration portal.

3.2 Configuring an External User Authentication System

Instead of maintaining user IDs and passwords in ESCM, your customer may want to connect to his organization's own identity management system for user authentication. ESCM supports user authentication against remote LDAP systems via secure VPN connections.

When using an LDAP system, your customer does not need to register his users manually with ESCM. The administrator of the customer organization can import the users, thus automatically registering them.

To enable access to the customer's LDAP system, you need to perform the following steps:

1. Create an LDAP configuration file on your file system.
2. Define the LDAP connection settings in the file. For a list of the relevant keys, refer to *LDAP Keys* on page 16.
3. Activate the LDAP-based authentication when registering the customer, and use the **LDAP settings** option to store the connection settings specified in the configuration file in ESCM.

   If you do not store any organization-specific connection settings in ESCM, ESCM will use default settings for LDAP connections.

Access to the LDAP system is enabled for the customer as soon as the customer is registered with ESCM. The administrator of the new customer organization can then import the required users from the LDAP system. The administrator can choose to import all users from the LDAP system or apply filter criteria to select specific users only. The imported users are sent an email with the URL for accessing the provided services. If required, the administrator of the customer organization can change the LDAP settings you defined for the organization.

User data maintained in the LDAP system such as the password, first name, or last name, can only be changed in the LDAP system. If the data is changed, ESCM automatically synchronizes its own data with it. An automatic synchronization is not possible in case a user ID is changed in the LDAP system. Such a user has to be imported again.

3.3 Handling Subscriptions

You can view the details of the subscriptions customers have created for services you promote. The following information on a subscription is available:

- Activation date and number of users assigned to the subscription.
- Details of the marketable service to which the customer subscribed.
- Details of the underlying technical service.
- Details of the price model defined for the marketable service.

You cannot change subscriptions or terminate them. To view the information, use the **View subscription** menu option in the **Customer** menu of the ESCM administration portal.

The suppliers of your services can explicitly terminate a subscription. This may be required, for example, if a customer does not comply with a license agreement or does not pay for a subscription. Terminating a subscription is possible even if users are currently using the underlying service. As soon as a supplier terminates a subscription, your customer is sent an email informing him about the termination and the reason the supplier has entered for it.

To offer support to customers, ESCM allows the suppliers of your services to enter a support email address for their organization. This allows administrators and subscription managers of customer organizations to directly contact a supplier in case they have questions or want to report an incident on a service they have subscribed to.

4 Billing and Payment

A subscription by a customer to a service you offer establishes a contract between the customer and the supplier of the service. Therefore, it is the supplier's responsibility to prepare billing and payment, and collect payments from the customer. As users work with the services you promote, the usage charges are calculated based on the fees defined in the related price models. The resulting billing data can be used by the supplier for generating invoices and collecting payments.

As a broker, you usually receive a share of the revenue from your customers' service usage. The operator is responsible for defining the revenue share you can claim. You are not authorized to change it.

ESCM offers the following options that support you in collecting your revenue share:

- You can view the revenue share that is defined.
- You can retrieve and export information on the actual revenue.

For details, refer to the following sections. You can also use reports for retrieving information on your actual revenue. For details, refer to *Reporting* on page 15.

Viewing the Revenue Share

You can view the percentage you receive from the total revenue generated by a service.

For this purpose, use the **Define publishing options** menu option in the **Marketable service** menu of the ESCM administration portal.

Exporting Revenue Share Data

At the end of each calendar month (plus an offset defined by the operator), ESCM calculates the revenue share data for a marketplace.

To retrieve information on your actual revenue for a specific time frame, ESCM allows you to export the corresponding data. To do this, use the **Export billing data** menu option in the **Account** menu of the ESCM administration portal.

The data can be saved to an XML file or opened in an editor of your choice. You can edit it and work with it as required. The data can, for example, be forwarded to an accounting system. You can use it to collect your share from the suppliers or marketplace owners, depending on who is responsible for paying you according to your contracts.

Refer to *Customer Billing Data* on page 18 for a detailed description of the elements and attributes that may occur in the generated XML file.

5 Reporting

ESCM offers comprehensive reports for different purposes and at different levels of detail.

As a broker, you can generate a broker revenue share report which shows your actual revenue depending on the usage of the services you promote. The report breaks down the data to the individual suppliers who provide the services.

You can use the report data as the basis for collecting the shares from the suppliers or from the owners of the marketplaces to which you published the services, depending on who is responsible for paying you according to your contracts.

To create a report, use the **Create report** menu option in the **Account** menu of the ESCM administration portal and choose **Broker revenue share report**. The menu option allows you to specify the month for which to generate the report.

The generated report is instantly displayed at the ESCM administration portal. You can choose to print the report or save it in several formats.

Note:	Contact your platform operator if the reporting functionality is not available. He is responsible for defining the respective configuration parameters.

Appendix A: LDAP Keys

The following keys must be defined in a configuration file for enabling access to an organization's LDAP system:

Key	Description
LDAP_URL	Mandatory. Provider URL of the LDAP server. This LDAP server is used for user authentication. Example: `LDAP_URL=ldap://myldapserver.lan.est.company.de:389`
LDAP_BASE_DN	Mandatory. Position in the LDAP directory tree at which to start looking for users. Example: `LDAP_BASE_DN=ou=people,dc=est,dc=mycompany,dc=de`
LDAP_PRINCIPAL	Optional. Name of the user who is allowed to query the LDAP server. Example: `LDAP_PRINCIPAL=uid=admin,ou=system`
LDAP_CREDENTIALS	Optional. Password of the user who is allowed to query the LDAP server. Example: `LDAP_CREDENTIALS=secret`
LDAP_ATTR_UID	Mandatory. LDAP attribute from which a user ID is read. The default used when an organization is created is `uid` unless the operator has defined a different value in the platform LDAP settings. Example: `LDAP_ATTR_UID=uid`
LDAP_ATTR_EMAIL	Optional. LDAP attribute from which the email address of a user is read. Example: `LDAP_ATTR_EMAIL=scalixEmailAddress`
LDAP_ATTR_FIRST_NAME	Optional. LDAP attribute from which the first name of a user is read. Example: `LDAP_ATTR_FIRST_NAME=givenName`
LDAP_ATTR_LAST_NAME	Optional. LDAP attribute from which the last name of a user is read. Example: `LDAP_ATTR_LAST_NAME=sn`
LDAP_ATTR_ADDITIONAL_NAME	Currently not used.
LDAP_ATTR_LOCALE	Optional. LDAP attribute from which the default language to be stored for a user is read. Example: `LDAP_ATTR_LOCALE=locale`

LDAP_CONTEXT_FACTORY	Mandatory. Context factory which provides the API to query the LDAP server. The default used when an organization is created is `com.sun.jndi.ldap.LdapCtxFactory` unless the operator has defined a different value in the platform LDAP settings. Example: `LDAP_CONTEXT_FACTORY=com.sun.jndi.ldap.LdapCtxFactory`
LDAP_ATTR_REFERRAL	Optional. Property defining how LDAP referrals are to be processed. If an organization uses an Active Directory with sub-domains from which users are to be imported into ESCM, the sub-domains can be modeled as referrals. In this sense, a referral is a reference to another directory partition or sub-domain. By default, values from referrals are not retrieved. This property can take on the following values: `follow`: Referrals are followed, i.e. users are imported from all referenced directory partitions or sub-domains. `ignore`: Referrals are ignored (default), i.e. users are imported from the current domain directory only. Example: `LDAP_ATTR_REFERRAL=ignore`

Appendix B: Customer Billing Data

The charges for the usage of a service in ESCM are calculated based on the price model defined for the service, customer, or subscription.

A supplier or reseller can export the billing data for one or several of his customers for a specific time. Suppliers also have access to the billing data of customers of broker organizations that sell their services. The exported data can be forwarded, for example, to an accounting system for further processing.

The result of the export is stored in an XML file, the customer billing data file. The billing data file conforms to the XML schema `BillingResult.xsd`, which can be found in the ESCM integration package.

The billing data file is named `<date>BillingData.xml`, where `<date>` represents the creation date.

This appendix describes the meaning of the elements and attributes that may occur in a billing data file.

BillingDetails

Top-level container element of a billing data file. For each subscription, a `BillingDetails` element is added to the billing data file.

A `BillingDetails` element contains the following subelements:

- `Period` (see *Period* on page 18)
- `OrganizationDetails` (see *OrganizationDetails* on page 19)
- `Subscriptions` (see *Subscriptions* on page 19)
- `OverallCosts` (see *OverallCosts* on page 32)

A `BillingDetails` element has the following attributes:

`key` - (optional, data type `long`) Unique identifier allowing, for example, accounting systems to relate billing data to an invoice. The billing data key is printed on the invoice. Suppliers and customers may use the billing data key to create a detailed billing report for an existing invoice or subscription. A supplier can retrieve the key from a billing report, a customer gets the billing data key from the corresponding invoice.

`timezone` - (required, data type `string`) Time zone based on the UTC time standard. It reflects the standard server time without daylight saving time. For example, 18:00:00 o'clock on June 1st in Berlin (UTC+1) will be output as follows in the XML file:

```
<BillingDetails timezone="UTC+01:00" key="31122">
  <Period startDate="1370106000000"
  startDateIsoFormat="2013-06-01T16:00:00.000Z" ..."/>
```

The time zone is relevant for price models with costs (see *PriceModel* on page 21).

Period

Specifies the billing period for which the data is exported. The start and end time of the billing period are output according to the start day of the billing period which was defined by the supplier or reseller.

A `Period` element has the following attributes:

- **`startDate`** - (data type `long`) Start time of the period. The start time is specified in milliseconds, the starting point for the calculation is `1970-01-01, 00:00`.

- **startDateIsoFormat** - (optional, data type dateTime) Same as startDate, but specified in ISO 8601 format (YYYY-MM-DDThh:mm:ss.fffZ) .
- **endDate** - (data type long) End time of the period. The end time is specified in milliseconds, the starting point for the calculation is 1970-01-01, 00:00.
- **endDateIsoFormat** - (optional, data type dateTime) Same as endDate, but specified in ISO 8601 format (YYYY-MM-DDThh:mm:ss.fffZ).

Example:

```
<Period startDateIsoFormat="2012-08-31T22:00:00.000Z"
  startDate="1346450400000"
  endDateIsoFormat="2012-09-30T22:00:00.000Z"
  endDate="1349042400000"/>
```

OrganizationDetails

Provides details of the customer for which the billing data have been exported. The details may include a Udas element with custom attributes that store additional information on the customer organization.

An OrganizationDetails element contains the following subelements:

Email

Specifies the email address of the organization (data type string).

Name

Specifies the name of the organization (data type string).

Address

Specifies the address of the organization (data type string).

Paymenttype

Specifies the payment type used for subscriptions of the organization (data type string).

Example:

```
<BillingDetails key="10002" timezone="UTC+01:00">
...
  <OrganizationDetails>
    <Email>info@company.com</Email>
    <Name>company</Name>
    <Address>Street</Address>
    <Paymenttype>INVOICE</Paymenttype>
  </OrganizationDetails>
...
</BillingDetails>
```

Subscriptions

Contains the billing data for the subscriptions of the customer which are relevant for the current billing period.

For every subscription of an organization, the Subscriptions element contains a Subscription element. In this element, the costs of the affected subscription are specified.

A Subscription element has the following attributes:

- **id** - (required, data type string) Unique subscription name.

- **purchaseOrderNumber** - (data type string) Optional reference number as specified by the customer when subscribing to a service.

A Subscription element contains a PriceModel element with the billing data for the price model of the subscription (see *PriceModel* on page 21). A Udas element with custom attributes that store additional information on the subscription may also be included (see *Udas* on page 20).

If a subscription is assigned to an organizational unit at the end of the billing period, the Subscription element also contains an OrganizationalUnit element with the following attributes:

- **name** - (required, data type string) Required name of the organizational unit to which the subscription is assigned.
- **referenceID** - (optional, data type string) Optional reference ID of the organizational unit to which the subscription is assigned.

Be aware that the organizational units to which the subscription may have been assigned before the generation of the billing data are not shown.

Example:

```
<BillingDetails key="10002" timezone="UTC+01:00">
...
  <Subscriptions>
    <Subscription id="Mega Office Basic" purchaseOrderNumber="12345">
      <OrganizationalUnit name="ProjectTeam" referenceID="123abc"/>
      <PriceModels>
        <PriceModel calculationMode="PRO_RATA" id="14001">
...
        </PriceModel>
      </PriceModels>
    </Subscription>
  </Subscriptions>
...
</BillingDetails>
```

Udas

Contains custom attributes that store additional information on an organization or subscription. This could be, for example, the profit center to which a customer's revenue is to be accounted.

A Udas element may be included in an OrganizationDetails or a Subscription element.

For every custom attribute, a Uda element is included in the Udas element.

A Uda element has the following attributes:

- **id** - (required, data type string) ID of the custom attribute.
- **value** - (required, data type string) Value of the custom attribute.

Example:

```
<BillingDetails key="10002" timezone="UTC+01:00">
...
  <Subscriptions>
    <Subscription id="Mega Office Basic" purchaseOrderNumber="12345">
...
      <Udas>
        <Uda id="Profit Center" value="My Company"/>
      </Udas>
    </Subscription>
  </Subscriptions>
```

```
...
</BillingDetails>
```

PriceModel

Contains the billing data for a price model used to calculate the utilization charges for a subscription.

A `PriceModel` element is included in every subscription element. It contains the following subelements:

- `UsagePeriod` (see *UsagePeriod* on page 21)
- `GatheredEvents` (see *GatheredEvents* on page 22)
- `PeriodFee` (see *PeriodFee* on page 23)
- `UserAssignmentCosts` (see *UserAssignmentCosts* on page 24)
- `OneTimeFee` (see *OneTimeFee* on page 25)
- `PriceModelCosts` (see *PriceModelCosts* on page 25)
- `Parameters` (see *Parameters* on page 26)

A `PriceModel` element has the following attributes:

`id` - (required, data type `string`) Unique name identifying the price model.

`calculationMode` - (required, data type `string`) Cost calculation option of the price model. Can be set to one of the following values: `FREE_OF_CHARGE` (the service is free of charge), `PRO_RATA` (the costs are calculated exactly for the time a service is used, based on milliseconds), `PER_UNIT` (the costs are calculated based on fixed time units).

UsagePeriod

Specifies the actual period in which a price model is used for calculating the charges of a subscription.

A usage period usually begins when a customer subscribes to a service and ends when the subscription is terminated. In case a free trial period is defined for the service, the usage period begins when the free trial period has ended. When the customer upgrades or downgrades the subscription, a new usage period is started in which the price model of the new service is applied. If the customer changes elements that determine how the charges for the service are calculated (e.g. the number of assigned users, the service roles of the assigned users, or the parameter values), a new usage period is started in which the updated elements are applied.

A `UsagePeriod` element is contained in a `PriceModel` element.

A `UsagePeriod` element has the following attributes:

- **`startDate`** - (data type `long`) Start time of the period. The start time is specified in milliseconds, the starting point for the calculation is `1970-01-01, 00:00`.
- **`startDateIsoFormat`** - (optional, data type `dateTime`) Same as `startDate`, but specified in ISO 8601 format (`YYYY-MM-DDThh:mm:ss.fffZ`) .
- **`endDate`** - (data type `long`) End time of the period. The end time is specified in milliseconds, the starting point for the calculation is `1970-01-01, 00:00`.
- **`endDateIsoFormat`** - (optional, data type `dateTime`) Same as `endDate`, but specified in ISO 8601 format (`YYYY-MM-DDThh:mm:ss.fffZ`).

Example:

```
<PriceModel calculationMode="PRO_RATA" id="14001">
```

```
  <UsagePeriod endDate="1306879200000"
   endDateIsoFormat="2011-05-31T22:00:00.000Z"
   startDate="1304755088065"
   startDateIsoFormat="2011-05-07T07:58:08.065Z"/>
...
</PriceModel>
```

GatheredEvents

Specifies the costs for all chargeable events that occurred in the current usage period of the subscription. These include, for example, login and logout by users to the underlying application, the completion of specific transactions, or the creation or deletion of specific data. It depends on the implementation and integration of the underlying application which events are available.

A `GatheredEvents` element is contained in a `PriceModel` element.

A `GatheredEvents` element contains the following subelements:

- `Event`
- `GatheredEventsCosts`

Event

For every event, an `Event` element is included in the `GatheredEvents` element.

An `Event` element has the following attribute:

`id` - (required, data type `string`) Event ID as specified in the technical service definition.

An `Event` element contains the following subelements:

- `Description`
- `SingleCost`
- `NumberOfOccurence`
- `CostForEventType`

Description

Contains the description of the event.

SingleCost

Specifies the price for the event as defined in the price model. If an event has stepped prices, this element is omitted. A `SteppedPrices` element is included instead (see *SteppedPrices* on page 31).

A `SingleCost` element has the following attribute:

`amount` - (required, data type `decimal`) Price for a single event.

NumberOfOccurence

Specifies how often the event occurred.

A `NumberOfOccurence` element has the following attribute:

`amount` - (required, data type `long`) Number of times the event occurred.

CostForEventType

Specifies the total costs for the event in the billing period.

A `CostForEventType` element has the following attribute:

`amount` - (required, data type `decimal`) Total costs for the event. The total costs for an event are calculated from the singular costs (`SingleCost`) multiplied with the number of occurrences (`NumberOfOccurence`). If role-based costs and/or stepped prices are specified for events, these

costs are added (see *RoleCosts* on page 30 and *SteppedPrices* on page 31). The value is rounded to two decimal places.

GatheredEventsCosts

Specifies the total costs for all events in the current `GatheredEvents` element.

A `GatheredEventsCosts` element has the following attribute:

`amount` - (required, data type `decimal`) Total costs for events. The value is rounded to two decimal places.

Example:

```
<PriceModel calculationMode="PRO_RATA" id="14001">
...
  <GatheredEvents>
    <Event id="USER_LOGOUT_FROM_SERVICE">
      <Description xml:lang="en">Logout from the service.</Description>
      <SingleCost amount="100.00"/>
      <NumberOfOccurrence amount="3"/>
      <CostForEventType amount="300.00"/>
    </Event>
...
    <GatheredEventsCosts amount="1200.00"/>
  </GatheredEvents>
...
</PriceModel>
```

PeriodFee

Specifies the costs for using the subscription in the given usage period.

For each subscription, a charge can be defined that a customer has to pay on a recurring basis. Monthly, weekly, daily, or hourly periods are supported. The recurring charge for a subscription is independent of the amount of users, events, or other usage data.

The calculation of the charges depends on the cost calculation option which was chosen for the price model (see *PriceModel* on page 21 for details).

A `PeriodFee` element is contained in a `PriceModel` element.

A `PeriodFee` element has the following attributes:

- **`basePeriod`** - (required, data type `string`) Period on which the charges are based. Can be set to one of the following values: `MONTH`, `WEEK`, `DAY`, `HOUR`.
- **`basePrice`** - (required, data type `decimal`) Recurring charge per base period according to the price model.
- **`factor`** - (required, data type `decimal`) Factor used to calculate the period fee for the subscription. The factor is calculated from the usage period of the subscription divided by the base period (`basePeriod`). The recurring charge is multiplied with this factor to calculate the total costs (`price`).
- **`price`** - (required, data type `decimal`) Total period fee for the subscription. This value is calculated from the recurring charge (`basePrice`) multiplied with the factor (`factor`). The value is rounded to two decimal places.

Example:

```
<PriceModel calculationMode="PRO_RATA" id="14001">
...
  <PeriodFee basePeriod="MONTH" basePrice="10.00"
 factor="0.4020212567204301" price="4.02"/>
```

```
...
</PriceModel>
```

UserAssignmentCosts

Specifies the costs for the user assignments to the subscription.

For the users assigned to a subscription, a charge can be defined that a customer has to pay on a recurring basis. Monthly, weekly, daily, or hourly periods are supported. The charge depends on the amount of time units one or more users are assigned to the subscription. This type of charge can only be defined for services with the login or user access type.

The recurring charge for users is independent of the recurring charge per subscription or other usage data.

For this type of charge, stepped prices can be applied: Recurring charges can be defined that depend on the sum of the time units of all user assignments.

The calculation of the charges depends on the cost calculation option which was chosen for the price model (see *PriceModel* on page 21 for details).

With per time unit calculation, the costs for a time unit in which a user is assigned to a subscription are always fully charged. There is no difference in the costs between a user who is assigned from the start until the end of the time unit and a user who is assigned for a part of the time unit only. A time unit is charged only once if a user is deassigned from and re-assigned to a subscription within the same time unit. Yet, canceling an assignment, deleting the user, and then creating a new user with the same user ID is treated as if two different users are assigned to the subscription. The time unit is charged twice, accordingly.

A `UserAssignmentCosts` element is contained in a `PriceModel` element.

A `UserAssignmentCosts` element has the following attributes:

- **`basePeriod`** - (optional, data type `string`) Period on which the charges are based. Can be set to one of the following values: `MONTH`, `WEEK`, `DAY`, `HOUR`.
- **`basePrice`** - (optional, data type `decimal`) Recurring charge for users per base period according to the price model. If the charge for users has stepped prices, this attribute is omitted.
- **`factor`** - (optional, data type `decimal`) Factor used to calculate the costs for the user assignments. The factor is calculated by summing up the factors for each user specified in the `UserAssignmentCostsByUser` element. The recurring charge (`basePrice`) is multiplied with this factor to calculate the costs (`price`).
- **`numberOfUsersTotal`** - (optional, data type `long`) Number of users assigned to the subscription in the usage period.
- **`total`** - (data type `decimal`) Total costs for the user assignments including role-based costs and stepped prices. The value is rounded to two decimal places. For details on role-based costs and stepped prices, refer to *RoleCosts* on page 30 and *SteppedPrices* on page 31.
- **`price`** - (optional, data type `decimal`) Costs for the user assignments. This value is calculated from the recurring charge (`basePrice`) multiplied with the factor (`factor`). The value is rounded to two decimal places.

A `UserAssignmentCosts` element contains the following subelement. If role-based costs and/or stepped prices are specified, a `RoleCosts` element and/or `SteppedPrices` element is also present (see *RoleCosts* on page 30 and *SteppedPrices* on page 31).

UserAssignmentCostsByUser

Specifies the fraction of the usage period a user was assigned to the subscription.

A `UserAssignmentCostsByUser` element has the following attributes:

- **`factor`** - (required, data type `decimal`) Fraction of the usage period the given user was assigned to the subscription. The factors of the single user assignments are summed up to calculate the total costs for the user assignments.
- **`userId`** - (required, data type `string`) User ID.

Example:

```
<PriceModel calculationMode="PRO_RATA" id="18000">
...
  <UserAssignmentCosts basePeriod="MONTH" basePrice="19.00"
   factor="0.5337726052867383" numberOfUsersTotal="2"
   total ="50.00" price="10.14">
     <UserAssignmentCostsByUser factor="1.0499215949820788E-4"
     userId="admin"/>
     <UserAssignmentCostsByUser factor="0.5336676131272401"
     userId="miller"/>
   </UserAssignmentCosts>
...
</PriceModel>
```

OneTimeFee

Specifies the one-time fee for the subscription.

A one-time fee defines the amount a customer has to pay for a subscription in the first billing period of the subscription. It is added to the total charges for the first billing period. It is independent of the number of users, events, or other usage data.

If a one-time fee is defined for a service to which a customer upgrades or downgrades a subscription, it is added to the total charges for the customer, even if the service from which the customer migrates also defines a one-time fee.

A `OneTimeFee` element is contained in a `PriceModel` element.

A `OneTimeFee` element has the following attributes:

- **`amount`** - (required, data type `decimal`) Total costs for the one-time fee. The value is rounded to two decimal places.
- **`baseAmount`** - (required, data type `decimal`) One-time fee as defined in the price model.
- **`factor`** - (required, data type `long`) Factor used for calculating the one-time fee. Since this charge occurs only once, the factor is 1 for the first billing period, and 0 in case the one-time fee has already been charged in a previous billing period.

Example:

```
<PriceModel calculationMode="PRO_RATA" id="14001">
...
  <OneTimeFee amount="10.00" baseAmount="10.00" factor="1"/>
...
</PriceModel>
```

PriceModelCosts

Specifies the total costs for the subscription in the current usage period.

A `PriceModelCosts` element is contained in a `PriceModel` element.

A `PriceModelCosts` element has the following attributes:

- **`currency`** - (required, data type `string`) ISO code of the currency in which the costs are calculated.
- **`amount`** - (required, data type `decimal`) Total amount of the costs for the subscription. The value is rounded to two decimal places.

Example:

```
<PriceModel calculationMode="PRO_RATA" id="14001">
...
  <PriceModelCosts currency="EUR" amount="990.00"/>
</PriceModel>
```

Parameters

Specifies the costs for parameters defined for the service underlying the subscription.

A price model can define prices for service parameters and options. It depends on the implementation and integration of the underlying application whether and which parameters and options are available.

A price can be defined for every parameter and option, and the price can be charged per subscription or per user assigned to the subscription. Numeric parameters are a multiplier for the price. For boolean parameters, the multiplier is `1` if the value is `true`. In all other cases, the multiplier is `0`.

The calculation of charges for parameters and options depends on the cost calculation option which was chosen for the price model (see *PriceModel* on page 21 for details).

If the charges for a subscription are calculated per time unit and a customer changes a parameter value within a time unit, the affected time unit is charged pro rata. This means that the customer is charged exactly for the time each parameter value is set.

For numeric parameters, stepped prices can be applied per subscription: Different prices can be defined depending on the parameter values.

The prices for parameters and options are independent of other price model elements.

A `Parameters` element is contained in a `PriceModel` element.

A `Parameters` element contains the following subelements:

- `Parameter`
- `ParametersCosts`

Parameter

For every parameter, a `Parameter` element is included in the `Parameters` element.

A `Parameter` element has the following attribute:

`id` - (required, data type `string`) Parameter ID.

A `Parameter` element contains the following subelements:

- `ParameterUsagePeriod`
- `ParameterValue`
- `PeriodFee`
- `UserAssignmentCosts`
- `ParameterCosts`
- `Options`

ParameterUsagePeriod

Specifies the actual usage period for the parameter.

The usage period for a parameter begins when a customer subscribes to the service with the given parameter definition in the price model and ends when the subscription is terminated.

In case a free trial period is defined for the service, the usage period for the subscription begins when the free trial period has ended. If a parameter value is changed, a new usage period is started in which the updated value is applied for calculating the costs.

A `ParameterUsagePeriod` element has the following attributes:

- **`startDate`** - (data type `long`) Start time of the period. The start time is specified in milliseconds, the starting point for the calculation is `1970-01-01, 00:00`.
- **`startDateIsoFormat`** - (optional, data type `dateTime`) Same as `startDate`, but specified in ISO 8601 format (`YYYY-MM-DDThh:mm:ss.fffZ`) .
- **`endDate`** - (data type `long`) End time of the period. The end time is specified in milliseconds, the starting point for the calculation is `1970-01-01, 00:00`.
- **`endDateIsoFormat`** - (optional, data type `dateTime`) Same as `endDate`, but specified in ISO 8601 format (`YYYY-MM-DDThh:mm:ss.fffZ`).

ParameterValue

Specifies the costs and data type for the parameter.

A `ParameterValue` element has the following attributes:

- **`amount`** - (required, data type `string`) Costs for the parameter as defined in the price model.
- **`type`** - (required, data type `string`) Data type of the parameter. Can be set to one of the following values: `BOOLEAN`, `INTEGER`, `LONG`, `STRING`, `ENUMERATION`, `DURATION`.

PeriodFee

Specifies the costs for using the parameter in the given usage period. If a parameter has stepped prices, a `SteppedPrices` element is included in the `PeriodFee` element.

A `PeriodFee` element has the following attributes:

- **`basePeriod`** - (required, data type `string`) Period on which the charges are based. Can be set to one of the following values: `MONTH`, `WEEK`, `DAY`, `HOUR`.
- **`basePrice`** - (optional, data type `decimal`) Recurring charge per base period according to the price model. If a parameter has stepped prices, this attribute is omitted.
- **`factor`** - (required, data type `decimal`) Factor used to calculate the costs for using the parameter. The factor is calculated from the usage period divided by the base period (`basePeriod`). The recurring charge (`basePrice`) is multiplied with this factor to calculate the costs (`price`).
- **`price`** - (required, data type `decimal`) Costs for using the parameter. This value is calculated from the recurring charge (`basePrice`) multiplied with the factors (`factor` and `valueFactor`). The value is rounded to two decimal places.
- **`valueFactor`** - (required, data type `float`) Factor to calculate the total costs for using the parameter depending on the parameter value. The recurring charge (`basePrice`) is multiplied with this factor to calculate the costs (`price`). This factor is set depending on the data type of the parameter. For numeric parameters it is set to the value of the parameter. For boolean parameters, the factor is set to `1` if the value is `true`. In all other cases, the factor is set to `0`.

UserAssignmentCosts

Specifies the costs for the parameter related to the user assignments of the subscription based on the price per user for the parameter as defined in the price model. If costs for service roles are

defined, a `RoleCosts` element is included in the `UserAssignmentCosts` element (see *RoleCosts* on page 30 for details).

A `UserAssignmentCosts` element has the following attributes:

- **`basePeriod`** - (required, data type `string`) Period on which the charges are based. Can be set to one of the following values: `MONTH`, `WEEK`, `DAY`, `HOUR`.
- **`basePrice`** - (required, data type `decimal`) Recurring charge for users per base period for the parameter according to the price model.
- **`factor`** - (required, data type `decimal`) Factor used to calculate the costs for using the parameter. The factor is calculated from the parameter usage period divided by the base period (`basePeriod`) multiplied with the number of users. The recurring charge (`basePrice`) is multiplied with this factor to calculate the costs (`price`).
- **`price`** - (required, data type `decimal`) Costs for using the parameter. This value is calculated from the recurring charge (`basePrice`) multiplied with the factors (`factor` and `valueFactor`). The value is rounded to two decimal places. If stepped prices are defined for user assignments, the costs are given in the `price` attribute.
- **`total`** - (data type `decimal`) Total costs for using the parameter including role-based costs. The value is rounded to two decimal places. For details on role-based costs, refer to *RoleCosts* on page 30.
- **`valueFactor`** - (required, data type `float`) Factor to calculate the total costs for using the parameter depending on the parameter value. The recurring charge (`basePrice`) is multiplied with this factor to calculate the costs (`price`). This factor is set depending on the data type of the parameter. For numeric parameters it is set to the value of the parameter. For boolean parameters, the factor is set to `1` if the value is `true`. In all other cases, the factor is set to `0`.

ParameterCosts

Specifies the total costs for using the parameter.

A `ParameterCosts` element has the following attribute:

`amount` - (required, data type `decimal`) Total costs for the parameter calculated by summing up the costs specified in the `PeriodFee` and the `UserAssignmentCosts` element for the parameter and its options. If role-based costs and/or stepped prices are specified for the parameter, these are added (see *RoleCosts* on page 30 and *SteppedPrices* on page 31). The value is rounded to two decimal places.

ParametersCosts

Specifies the total costs for all parameters.

A `ParametersCosts` element has the following attribute:

`amount` - (required, data type `decimal`) Total costs for the parameters calculated by summing up the costs of the individual parameters as specified in the `ParameterCosts` elements. The value is rounded to two decimal places.

Example:

```
<Parameters>
...
  <Parameter id="MAX_FOLDER_NUMBER2">
    <ParameterUsagePeriod endDate="1306879200000"
     endDateIsoFormat="2011-05-31T22:00:00.000Z"
     startDate="1304755088065"
     startDateIsoFormat="2011-05-07T07:58:08.065Z"/>
    <ParameterValue amount="200" type="INTEGER"/>
    <PeriodFee basePeriod="MONTH" basePrice="0.00"
     factor="0.5337789669205496" price="0.00" valueFactor="200.0"/>
```

```
    <UserAssignmentCosts basePeriod="MONTH" basePrice="0.00"
     factor="0.5337726052867383" total ="0.00"
     price="0.00" valueFactor="200.0"/>
    <ParameterCosts amount="0.00"/>
  </Parameter>

...
  <ParametersCosts amount="600.00"/>
</Parameters>
```

Options

Specifies the costs for parameter options.

An `Options` element is contained in a `Parameter` element.

For every option, an `Option` element is included in the `Options` element.

An `Option` element has the following attribute:

`id` - (required, data type `string`) Option ID.

An `Option` element contains the following subelements:

- `PeriodFee`
- `UserAssignmentCosts`
- `OptionCosts`

PeriodFee

Specifies the costs for using the parameter option in the given usage period.

A `PeriodFee` element has the following attributes:

- **`basePeriod`** - (required, data type `string`) Period on which the charges are based. Can be set to one of the following values: `MONTH`, `WEEK`, `DAY`, `HOUR`.
- **`basePrice`** - (required, data type `decimal`) Recurring charge per base period according to the price model.
- **`factor`** - (required, data type `decimal`) Factor used to calculate the costs for using the parameter option. The factor is calculated from the usage period divided by the base period (`basePeriod`). The recurring charge (`basePrice`) is multiplied with this factor to calculate the total costs (`price`).
- **`price`** - (required, data type `decimal`) Costs for the parameter option. This value is calculated from the recurring charge (`basePrice`) multiplied with the factor (`factor`), and is rounded to two decimal places.

UserAssignmentCosts

Specifies the costs for the parameter option related to the user assignments of the subscription based on the price per user for the option as defined in the price model. If costs for service roles are defined, a `RoleCosts` element is included in the `UserAssignmentCosts` element (see *RoleCosts* on page 30).

A `UserAssignmentCosts` element has the following attributes:

- **`basePeriod`** - (required, data type `string`) Period on which the charges are based. Can be set to one of the following values: `MONTH`, `WEEK`, `DAY`, `HOUR`.
- **`basePrice`** - (required, data type `decimal`) Recurring charge for users per base period for the parameter option according to the price model.
- **`factor`** - (required, data type `decimal`) Factor used to calculate the costs for using the parameter option. The factor is calculated from the usage period divided by the base period

(basePeriod). The recurring charge (basePrice) is multiplied with this factor to calculate the costs (price).

- **total** - (data type decimal) Total costs for using the parameter option including role-based costs. The value is rounded to two decimal places. For details on role-based costs, refer to *RoleCosts* on page 30.
- **price** - (required, data type decimal) Costs for using the parameter option. This value is calculated from the recurring charge (basePrice) multiplied with the factor (factor). The value is rounded to two decimal places. If stepped prices are defined for user assignments, the costs are given in the price attribute.

OptionCosts

Specifies the total costs for using the parameter option.

An OptionCosts element has the following attribute:

amount - (required, data type decimal) Total costs for the parameter option calculated by summing up the costs specified in the PeriodFee and the UserAssignmentCosts element. The value is rounded to two decimal places.

Example:

```
<Parameter id="MEMORY_STORAGE">
...
  <Options>
    <Option id="2">
      <PeriodFee basePeriod="MONTH" basePrice="100.00"
       factor="1.0" price="100.00"/>
      <UserAssignmentCosts basePeriod="MONTH" basePrice="0.00"
       factor="1.0" total ="0.00" price="0.00"/>
      <OptionCosts amount="100.00"/>
    </Option>
  </Options>
...
</Parameter>
```

RoleCosts

Specifies the costs for service roles.

If defined for the underlying application, roles can be used to grant specific privileges to different users. The roles are specified in the technical service definition as service roles. Service roles can be mapped to corresponding permissions in the application.

For each role, a price can be defined. This price is added to the base price per user in the cost calculation for a billing period.

The calculation of the charges for service roles depends on the cost calculation option which was chosen for the price model (see *PriceModel* on page 21 for details).

If the charges are calculated per time unit and the role assignment of a user is changed within a time unit, the affected time unit is charged pro rata. This means that the customer is charged exactly for the time each user role is assigned.

If the charges are calculated per time unit and a user with a specific role is removed from the subscription and assigned to it again with a different role in the same time unit, the customer is also charged for the time during which the user is not assigned to the subscription. This means that he is charged with the price for the first service role until the user is assigned to the subscription with the second service role.

A `RoleCosts` element is contained in a `UserAssignmentCosts` element (as subelement of the `PriceModel`, `Parameters`, or `Option` element).

A `RoleCosts` element has the following attribute:

`total` - (required, data type `decimal`) Total amount of costs for the service roles. The value is rounded to two decimal places.

For every service role, a `RoleCost` element is included in the `RoleCosts` element.

A `RoleCost` element has the following attributes:

- **`id`** - (required, data type `string`) ID of the service role.
- **`basePrice`** - (required, data type `decimal`) Recurring charge for the service role according to the price model.
- **`factor`** - (required, data type `decimal`) Factor used to calculate the costs for the service role. The factor is calculated as a fraction of the actual usage period. The recurring charge (`basePrice`) is multiplied with this factor to calculate the costs (`price`).
- **`price`** - (required, data type `decimal`) Costs for the service role. This value is calculated from the recurring charge (`basePrice`) multiplied with the factor (`factor`). The value is rounded to two decimal places.

Example:

```
<Parameter id="MEMORY_STORAGE">
...
  <RoleCosts total="0.00">
    <RoleCost basePrice="0.00" factor="0.4020087753882915"
     id="USER" price="0.00"/>
    <RoleCost basePrice="0.00" factor="0.8040175186678614"
     id="ADMIN" price="0.00"/>
  </RoleCosts>
...
</Parameter>
```

SteppedPrices

Specifies the stepped prices for a user assignment, event, or parameter.

Stepped prices allow for the definition of ranges for which different price factors apply. Step limits, i.e. the upper limits of ranges, can be set for:

- The **sum of the time units** users are assigned and work with a subscription in a billing period. For example, up to 10 hours one user is assigned to a subscription cost 10.00 € per hour, every additional hour the user is assigned costs 8.00 €.
- The **number of events** occurring in the usage of a subscription. For example, up to 10 file downloads cost 1.00 € per download, any additional download costs 0.50 €.
- **Values of numeric parameters**. For example, uploading up to 100 files costs 1.00 € per file, any additional upload costs 0.50 € per file.

Stepped prices are independent of any other price model elements.

A `SteppedPrices` element is contained in `UserAssignmentCosts` (as subelement of the `PriceModel` element), `Event`, and `PeriodFee` (as subelement of the `Parameters` element) elements.

A `SteppedPrices` element has the following attribute:

`amount` - (required, data type `decimal`) Summed up costs for all steps including the last one.

For every price step, a `SteppedPrice` element is included in a `SteppedPrices` element.

A `SteppedPrice` element has the following attributes:

- **`additionalPrice`** - (required, data type `decimal`) Summed up costs for the previous steps. The costs are calculated from the `limit`, `freeAmount` and `basePrice` attributes of the previous step (`(limit - freeAmount) * price`). The `additionalPrice` attribute of the first step always has a value of `0`. The value is rounded to two decimal places.
- **`basePrice`** - (required, data type `decimal`) Costs for the current step according to the price model.
- **`freeAmount`** - (required, data type `long`) Amount of units for the current step that are considered as a fixed discount, for example, the number of users that are free of charge. The value corresponds to the value of the `limit` attribute in the previous step. The `freeAmount` attribute of the first step always has a value of `0`.
- **`limit`** - (required, data type `string`) Step limit as defined in the price model.
- **`stepAmount`** - (optional, data type `decimal`) Summed up costs for the current step. These costs are calculated from the `basePrice` and `stepEntityCount` attributes of the current step. The value is rounded to two decimal places.
- **`stepEntityCount`** - (optional, data type `decimal`) Factor used to calculate the costs for the current step.

Example:

```
<PriceModel calculationMode="PRO_RATA" id="350001">
...
  <UserAssignmentCosts basePeriod="MONTH" factor="2.707940780619112"
 numberOfUsersTotal="4" price="1283.18">
    <SteppedPrices amount="1283.18">
      <SteppedPrice additionalPrice="0.00" basePrice="500.00"
       freeAmount="0" limit="2" stepAmount="1000.00"
 stepEntityCount="2"/>
      <SteppedPrice additionalPrice="1000.00" basePrice="400.00"
       freeAmount="2" limit="3" stepAmount="283.18"
       stepEntityCount="0.707940780619112"/>
      <SteppedPrice additionalPrice="1400.00" basePrice="300.00"
       freeAmount="3" limit="null" stepAmount="0.00"
 stepEntityCount="0"/>
    </SteppedPrices>
  </UserAssignmentCosts>
...
</PriceModel>
```

OverallCosts

Contains the total amount of the charges to be paid by a customer for all subscriptions in the current billing period. The costs are given in the currency specified in the price model.

If a discount was specified, the net amount of the costs is given in the `Discount` element (see *Discount* on page 34). If a VAT rate was defined, it is given in the `VAT` element (see *VAT* on page 33).

An `OverallCosts` element has the following attributes:

- **`netAmount`** - (required, data type `decimal`) Net costs after the net discount has been deducted from the original net costs (see *Discount* on page 34). The value is rounded to two decimal places.
- **`currency`** - (required, data type `string`) ISO code of the currency in which the costs are calculated.

- **grossAmount** - (required, data type `decimal`) Gross amount of the costs, calculated from the net costs (`netAmount`) plus VAT (see *VAT* on page 33). The value is rounded to two decimal places.

Example:

```
<BillingDetails key="10002" timezone="UTC+01:00">
...
<OverallCosts netAmount="900.00" currency="EUR" grossAmount="1053.00"/>

</OverallCosts>
</BillingDetails>
```

VAT

Specifies the VAT rate to be applied.

A supplier can define a basic VAT rate that applies by default to all prices for his customers. In addition to this basic VAT rate, country-specific or even customer-specific VAT rates can be defined. You can:

- Enable VAT rate support for your organization.
- Define a default VAT rate that applies to all prices for all customers.
- Define a country-specific VAT rate for every country where you want to sell your services.
- Define a customer-specific VAT rate, for example, in case a customer organization has a subsidiary located in another country than its parent organization.

The VAT rate settings have the following effects on the cost calculation for a customer:

- If VAT rate support is disabled, prices are calculated as net prices; no VAT is added to the overall costs.
- A customer-specific VAT rate takes priority over any default or country-specific VAT rate.
- The country-specific VAT rate for the country where the customer organization is located is applied to the cost calculation when no customer-specific VAT rate is defined.
- The default VAT rate is used in all other cases.

The VAT rate does not affect any price model elements. The calculated VAT amount is added to the overall costs and results in the gross price to be paid by a customer.

A `VAT` element is contained in the `OverallCosts` element.

A `VAT` element has the following attributes:

- **percent** - (required, data type `float`) VAT rate in percent, specified as a decimal number.
- **amount** - (required, data type `decimal`) Net amount of VAT to be added to the net costs (`netAmount` attribute of the `OverallCosts` element). The value is rounded to two decimal places.

Example:

```
<BillingDetails key="10002" timezone="UTC+01:00">
...
  <OverallCosts netAmount="900.00" currency="EUR" grossAmount="1053.00">
    <VAT percent="17.0" amount="153.00"/>
  </OverallCosts>
</BillingDetails>
```

Discount

Specifies the discount granted to the customer.

A discount can be defined for a customer which applies to all subscriptions of the customer to services. A discount may be valid as of the current or a future month. It can be restricted to a certain period of time. Before the time expires, the customer is notified by email so that he can react and contact the supplier.

The discount is defined as a percentage that is subtracted from the regular total price for a subscription. It is granted for all costs of a customer that incur in a billing period in which the discount is valid. It does not matter whether the discount is valid for the whole billing period or only a part of it.

A discount is completely independent of what a customer might purchase. If a discount is changed, the new discount is valid the next time the billing data is generated. Usually, a discount is only changed in agreement with the relevant customer.

A `Discount` element is contained in the `OverallCosts` element.

A `Discount` element has the following attributes:

- **`percent`** - (required, data type `float`) Percentage of costs to be deducted from the net costs, specified as a decimal number.
- **`discountNetAmount`** - (required, data type `decimal`) Net discount to be deducted from the original net costs (`netAmountBeforeDiscount`). The value is rounded to two decimal places.
- **`netAmountAfterDiscount`** - (required, data type `decimal`) Net costs after the net discount (`discountNetAmount`) has been deducted from the original net costs (`netAmountBeforeDiscount`). The value is rounded to two decimal places.
- **`netAmountBeforeDiscount`** - (required, data type `decimal`) Net costs before the net discount (`discountNetAmount`) has been deducted. The value is rounded to two decimal places.

Example:

```
<BillingDetails key="10002" timezone="UTC+01:00">
...
  <OverallCosts netAmount="900.00" currency="EUR" grossAmount="1053.00">
    <Discount percent="10.00" discountNetAmount="100.00"
     netAmountAfterDiscount="900.00"
     netAmountBeforeDiscount="1000.00" />
    <VAT percent="17.0" amount="153.00"/>
  </OverallCosts>
</BillingDetails>
```

Glossary

Administrator

A privileged user role within an organization with the permission to manage the organization's account and subscriptions as well as its users and their roles. Each organization has at least one administrator.

Application

A software, including procedures and documentation, which performs productive tasks for users.

Billing System

A system responsible for calculating the charges for using a service.

Broker

An organization which supports suppliers in establishing relationships to customers by offering the suppliers' services on a marketplace, as well as a privileged user role within such an organization.

Cloud

A metaphor for the Internet and an abstraction of the underlying infrastructure it conceals.

Cloud Computing

The provisioning of dynamically scalable and often virtualized resources as a service over the Internet on a utility basis.

Customer

An organization which subscribes to one or more marketable services in ESCM in order to use the underlying applications in the Cloud.

Infrastructure as a Service (IaaS)

The delivery of computer infrastructure (typically a platform virtualization environment) as a service.

Marketable Service

A service offering to customers in ESCM, based on a technical service. A marketable service defines prices, conditions, and restrictions for using the underlying application.

Marketplace

A virtual platform for suppliers, brokers, and resellers in ESCM to provide their services to customers.

Marketplace Owner

An organization which holds a marketplace in ESCM, where one or more suppliers, brokers, or resellers can offer their marketable services.

Marketplace Manager

A privileged user role within a marketplace owner organization.

Operator

An organization or person responsible for maintaining and operating ESCM.

Organization

An organization typically represents a company, but it may also stand for a department of a company or a single person. An organization has a unique account and ID, and is assigned one or more of the following roles: technology provider, supplier, customer, broker, reseller, marketplace owner, operator.

Organizational Unit

A set of one or more users within an organization representing, for example, a department in a company, an individual project, a cost center, or a single person. A user may be assigned to one or more organizational units.

OU Administrator

A privileged user role within an organization allowing a user to manage the organizational units for which he has been appointed as an administrator, and to create, modify, and terminate subscriptions for these units.

Payment Type

A specification of how a customer may pay for the usage of his subscriptions. The operator defines the payment types available in ESCM; the supplier or reseller determines which payment types are offered to his customers, for example payment on receipt of invoice, direct debit, or credit card.

Platform as a Service (PaaS)

The delivery of a computing platform and solution stack as a service.

Price Model

A specification for a marketable service defining whether and how much customers subscribing to the service will be charged for the subscription as such, each user assigned to the subscription, specific events, or parameters and their options.

Reseller

An organization which offers services defined by suppliers to customers applying its own terms and conditions, as well as a privileged user role within such an organization.

Role

A collection of authorities that control which actions can be carried out by an organization or user to whom the role is assigned.

Seller

Collective term for supplier, broker, and reseller organizations.

Service

Generally, a discretely defined set of contiguous or autonomous business or technical functionality, for example an infrastructure or Web service. ESCM distinguishes between technical services and marketable services, and uses the term "service" as a synonym for "marketable service".

Service Manager

A privileged user role within a supplier organization.

Standard User

A non-privileged user role within an organization.

Software as a Service (SaaS)

A model of software deployment where a provider licenses an application to customers for use as a service on demand.

Subscription

An agreement registered by a customer for a marketable service in ESCM. By subscribing to a service, the customer is given access to the underlying application under the conditions defined in the marketable service.

Subscription Manager

A privileged user role within an organization with the permission to create and manage his own subscriptions.

Supplier

An organization which defines marketable services in ESCM for offering applications provisioned by technology providers to customers.

Technical Service

The representation of an application in ESCM. A technical service describes parameters and interfaces of the underlying application and is the basis for one or more marketable services.

Technology Manager

A privileged user role within a technology provider organization.

Technology Provider

An organization which provisions applications as technical services in ESCM.