

# **FUJITSU Software Enterprise Service Catalog Manager V17.5 alpha**

A horizontal band featuring a red abstract graphic with flowing, curved lines and a bright light source, creating a sense of motion and energy.

## **Operator's Guide**

December 2017

## Trademarks

LINUX is a registered trademark of Linus Torvalds.

Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Open Service Catalog Manager is a registered trademark of FUJITSU LIMITED.

Oracle, Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle Corporation and/or its affiliates.

Apache Ant, Ant, Apache Tomcat, Tomcat, and Apache are trademarks of The Apache Software Foundation.

UNIX is a registered trademark of the Open Group in the United States and in other countries.

VMware vSphere is a registered trademark of VMware 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  
LIMITED 2017

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 LIMITED.

## 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

	<b>About this Manual.....</b>	<b>6</b>
<b>1</b>	<b>Introduction.....</b>	<b>10</b>
1.1	ESCM Container Communication.....	10
1.2	ESCM Architecture.....	11
1.3	Organizations and User Roles.....	12
<b>2</b>	<b>Getting Started.....</b>	<b>15</b>
<b>3</b>	<b>Configuring ESCM.....</b>	<b>17</b>
3.1	Updating Configuration Settings.....	17
3.2	Adding a Currency to ESCM.....	18
3.3	Adding a Language to ESCM and Customizing Texts.....	19
3.4	Configuring Timers.....	19
3.5	Managing LDAP Settings.....	21
<b>4</b>	<b>###Monitoring, Backup and Recovery###? How does this work now?###.....</b>	<b>23</b>
4.1	Monitoring ESCM.....	23
4.1.1	Log Types and Location of Log Files.....	23
4.1.2	Configuring the Log Level.....	23
4.1.3	Log Entry Format.....	25
4.1.4	Maintaining the Log Files.....	25
4.2	Exporting Audit Log Data.....	25
4.3	Backup and Recovery.....	26
<b>5</b>	<b>Managing Organizations and Users.....</b>	<b>28</b>
5.1	Creating an Organization.....	28
5.2	Changing the Address Data of an Organization.....	28
5.3	Adding a Role to an Organization.....	28
5.4	Managing User Accounts.....	28
5.5	Registering Multiple Users for Organizations.....	29

---

<b>6</b>	<b>Reporting.....</b>	<b>30</b>
<b>7</b>	<b>Managing Billing and Payment.....</b>	<b>32</b>
7.1	Billing Runs.....	32
7.2	Defining Revenue Shares.....	32
7.3	Integrating PSP Support.....	34
7.3.1	Prerequisites.....	35
7.3.2	Registering a Payment Service Provider With ESCM.....	36
7.3.3	Configuring PSP Communication and Payment Types.....	36
7.3.4	Setting the PSP Identifier for a Supplier or Reseller.....	36
7.4	Assigning Payment Types to Suppliers and Resellers.....	37
7.5	Handling Billing Data.....	37
7.5.1	Start Billing Run.....	37
7.5.2	Start Payment Processing.....	37
7.5.3	Retry Failed Payment Processes.....	37
7.5.4	Preview Billing Data.....	38
7.5.5	Export Revenue Share Data.....	38
<b>8</b>	<b>Integrating Certificates for Trusted Communication.....</b>	<b>39</b>
8.1	Introduction.....	39
8.2	Requirements for Web Service Calls from ESCM.....	40
8.3	Requirements for Web Service Calls to ESCM.....	40
8.4	Certificate Integration Procedures.....	40
8.4.1	Creating a CA Root Certificate for ESCM.....	42
8.4.2	Creating a Self-Signed Domain Certificate.....	42
8.4.3	Exporting the Self-Signed Domain Certificate.....	42
8.4.4	Importing the Certificates.....	42
8.4.5	Importing the GlassFish Instance Certificate.....	43
8.4.6	Signing an External Organization's Certificate.....	43
8.4.7	Setting the Distinguished Name for an Organization.....	43
<b>9</b>	<b>Managing Marketplaces.....</b>	<b>45</b>
9.1	Creating a Marketplace.....	45
9.2	Changing the Owner of a Marketplace.....	45
9.3	Deleting a Marketplace.....	45

---

---

9.4	Configuring Marketplaces.....	46
9.5	Updating the Search Index.....	46
<b>Appendix A Configuration Settings.....</b>		<b>47</b>
A.1	Database Configuration Settings.....	47
A.2	ESCM Configuration Settings.....	48
<b>Appendix B LDAP Keys.....</b>		<b>58</b>
<b>Appendix C Audit Log.....</b>		<b>60</b>
C.1	User Operations.....	60
C.2	Administrator Operations.....	61
C.3	Service Manager Operations.....	67
<b>Appendix D Language Resource Bundles.....</b>		<b>84</b>
D.1	User Interface Resources.....	84
D.2	Online Help and FAQs.....	85
D.3	Report Labels.....	88
D.4	Supported Language Codes.....	89
<b>Appendix E User Data File for Multiple User Import.....</b>		<b>92</b>
<b>Appendix F Menu Options and Required Roles.....</b>		<b>94</b>
<b>Glossary .....</b>		<b>98</b>

## About this Manual

This manual describes the basic tasks involved in the operation and maintenance of ) FUJITSU Software Enterprise Service Catalog Manager, hereafter referred to as ESCM.

The manual is structured as follows:

Chapter	Description
<i>Introduction</i> on page 10	Describes ESCM and its architecture, its organizations and users.
<i>Getting Started</i> on page 15	Provides an overview of the tasks involved in the operation and maintenance of ESCM, shows the initial steps to be performed after installing ESCM, and introduces the tools that support you in performing operational tasks.
<i>Configuring ESCM</i> on page 17	Describes how to maintain the ESCM configuration settings, how to tune system performance, how to add a currency and a language, and how to use and configure timers for automatic task processing.
<i>###Monitoring, Backup and Recovery###? How does this work now? ###</i> on page 23	Describes the ESCM log files and log entry types, how to configure the log level, as well as which data is subject to regular backups. In addition, it describes how to export the audit log data and how to monitor the JMS queues.
<i>Managing Organizations and Users</i> on page 28	Describes how to create and manage organizations and user accounts.
<i>Reporting</i> on page 30	Describes the reports available for operators in ESCM.
<i>Managing Billing and Payment</i> on page 32	Describes how to define revenue shares, how to set up the payment types allowed for suppliers and resellers, and how to integrate ESCM with a payment service provider (PSP). In addition, it describes how to handle billing data: how to restart payment processes and billing runs, preview and export billing data.
<i>Integrating Certificates for Trusted Communication</i> on page 39	Describes the usage of certificates for secure communication between ESCM and applications integrated with it.
<i>Managing Marketplaces</i> on page 45	Describes how to create, update, and delete marketplaces.
<i>Application Server Resources</i>	Describes the resources required for ESCM on the application server.
<i>Configuration Settings</i> on page 47	Describes the ESCM configuration settings.

Chapter	Description
<i>LDAP Keys</i> on page 58	Describes the keys to be defined for enabling access to an organization's LDAP system.
<i>Audit Log</i> on page 60	Describes the elements of the audit log.
<i>Language Resource Bundles</i> on page 84	Describes the resources that can be provided and customized in different languages.
<i>User Data File for Multiple User Import</i> on page 92	Describes the content and format of a user data file for importing multiple users in one operation.
<i>Menu Options and Required Roles</i> on page 94	Gives an overview of the ESCM administration portal menu options and the roles required for using them.

## Readers of this Manual

This manual is directed to operators who maintain and operate ESCM in their environment.

It assumes that you are familiar with the following:

- Administration of the operating systems in use, including the adaption and execution of batch files or shell scripts.
- Java EE technology, particularly as to the deployment on application servers.
- Relational databases and their administration, in particular the PostgreSQL database.
- ESCM concepts as explained in the *Overview* manual.
- Web services concepts.
- Installation and administration of Web servers.
- Installation and administration of the GlassFish application server.
- Certificate-based authentication and communication.

## Notational Conventions

This manual uses the following notational conventions:

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

## Abbreviations

This manual uses the following abbreviations:

**API**                      Application Programming Interface

<b>ESCM</b>	Enterprise Service Catalog Manager
<b>CA</b>	Certification authority
<b>JAX-WS</b>	Java API for XML - Web Services
<b>JMS</b>	Java Message Service
<b>JSP</b>	Java Server Pages
<b>LDAP</b>	Lightweight Directory Access Protocol
<b>PaaS</b>	Platform as a Service
<b>PSP</b>	Payment service provider
<b>SaaS</b>	Software as a Service
<b>SAML</b>	Security Assertion Markup Language
<b>SOAP</b>	Simple Object Access Protocol
<b>STS</b>	Security Token Service
<b>WSDL</b>	Web Services Description Language
<b>WSIT</b>	Web Services Interoperability Technologies
<b>XSD</b>	XML Schema Definition

## 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.
- *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.
- *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.
- *Developer's Guide*: A PDF manual for application developers describing the public Web services and application programming interfaces of ESCM and how to integrate applications and external systems with 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.
- *OpenStack Integration*: A PDF manual for operators describing how to offer and use virtual systems controlled by OpenStack through services in ESCM.
- Javadoc and YAML documentation for the public Web services and application programming interfaces of ESCM and additional resources and utilities for application developers.

# 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.

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

The following ESCM images are available:

- `oscm-core`: The ESCM core application.
- `oscm-birt`: The report engine that ESCM uses for generating reports.
- `oscm-branding`: A static Web server providing an empty directory structure for customizing the layout and branding of ESCM marketplaces.
- `oscm-help`: A static Web server providing the online help for the ESCM administration portal.
- `oscm-app`: The Asynchronous Provisioning Platform (APP) together with the OpenStack and the Amazon Web Services (AWS) service controllers.
- `oscm-db`: Database SQL server providing for the database schema required for running ESCM and APP.
- `oscm-proxy`: A static Web server providing reverse proxy functionality.
- `oscm-initdb`: Temporary database container for initializing the databases required for ESCM and APP.

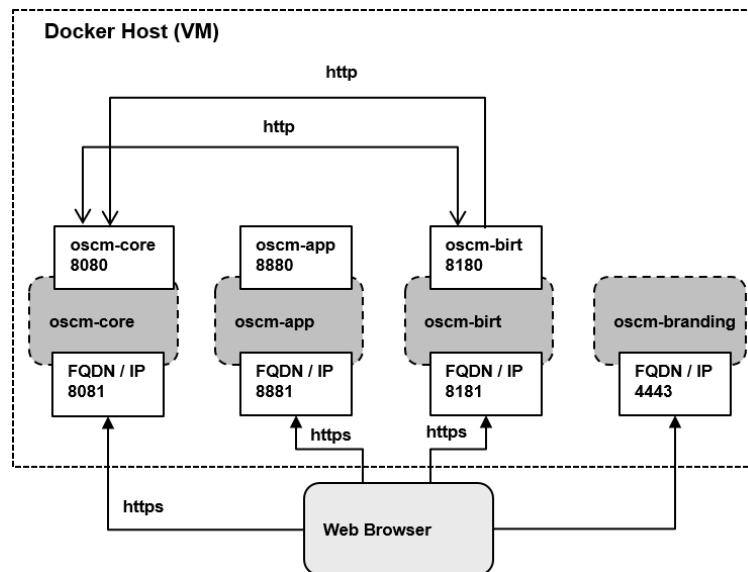
The Docker images and the deployment of corresponding containers on a Docker host is described in detail on [####DockerHub/GitHub link to oscm-core####](#).

For each container, a storage is mounted so that data is kept persistent when starting or stopping the containers.

## 1.1 ESCM Container Communication

The following figure provides an overview of the container communication on a Docker Host (a Virtual Machine (VM)):

### TODO: Add help container and Cinder volumes ###

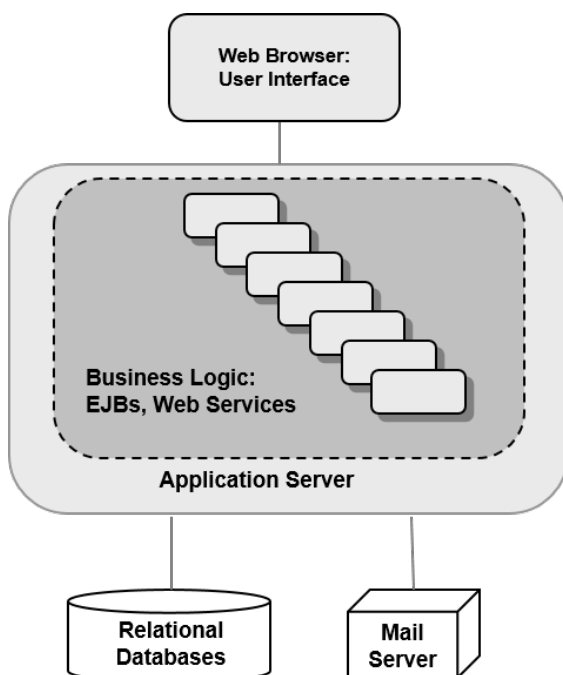


The host-internal communication always relies on the HTTP protocol, whereas calls from outside of the host are secured via HTTPS.

## 1.2 ESCM Architecture

ESCM is implemented in Java, using Java Platform, Enterprise Edition (Java EE ) technology. It is deployed on an application server supporting this technology.

The following figure provides an overview of the architecture:



ESCM has a three-tier architecture:

- The **presentation layer** in the application server's Web container includes the **user interface** (administration portal and marketplaces), realized as JavaServer Faces. Users access the user interface in Web browsers.
- The **business logic** is implemented in Enterprise JavaBeans (EJB). Both the Enterprise JavaBeans and the **public Web services** are available in the application server's EJB container. The public Web services and their interfaces are mainly used for integrating applications and external systems with ESCM. However, they can also be employed for accessing ESCM functionality from a Web service client. HTTPS must be used for communication with the public Web services.
- ESCM **persists** its data through the Java Persistence API in **relational databases**.

For informing users about relevant issues (e.g. their registration or assignment to a subscription), ESCM must have access to a mail server.

## 1.3 Organizations and User Roles

Each user working in ESCM is a member of a specific organization. An organization typically represents a company, but it may also stand for a department of a company or a single person. Each organization in ESCM has a unique account and ID as well as one or more of the following roles: **operator**, **technology provider**, **supplier**, **reseller**, **broker**, **marketplace owner**, **customer**.

Customers can register themselves with ESCM or be registered by a supplier, reseller, broker, or operator. In any case, an organization with the customer role is created. Organizations with

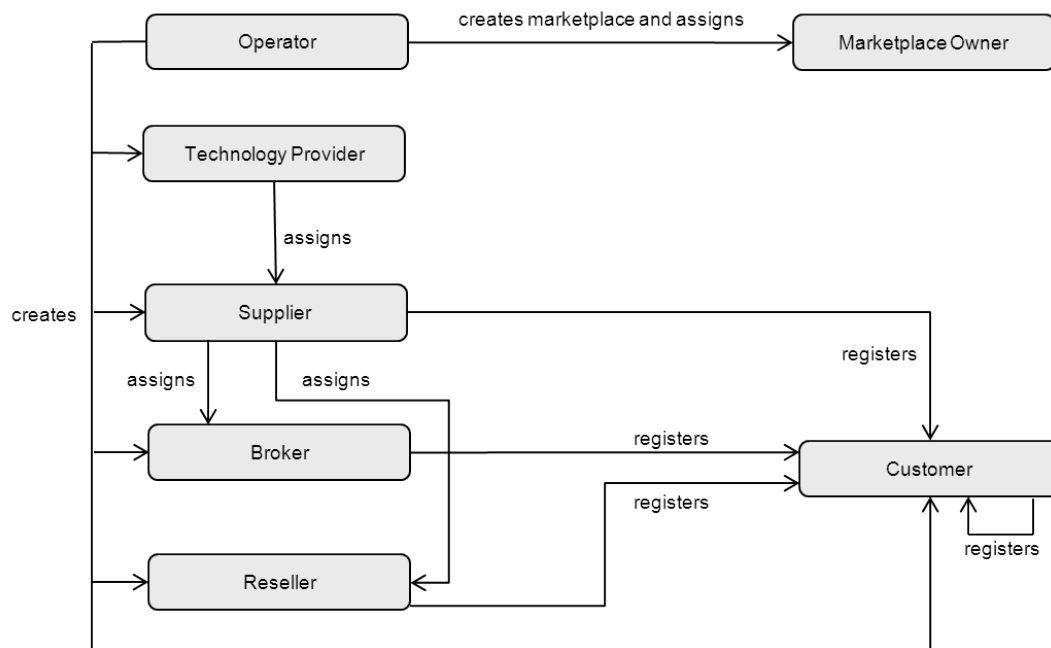
other roles can also act as customers, i.e. they are implicitly assigned the customer role. These organizations are created and assigned their roles as follows:

- When ESCM is installed, an organization with the operator role is created.
- Operators can assign the supplier, reseller, broker, and technology provider role to any existing organization or create new organizations with these roles. An organization can have both the supplier and the technology provider role. The reseller and broker roles, however, cannot be combined with each other or with the supplier or technology provider role.
- When operators create a marketplace, they specify an existing organization as its owner. In this way, the organization is assigned the marketplace owner role.

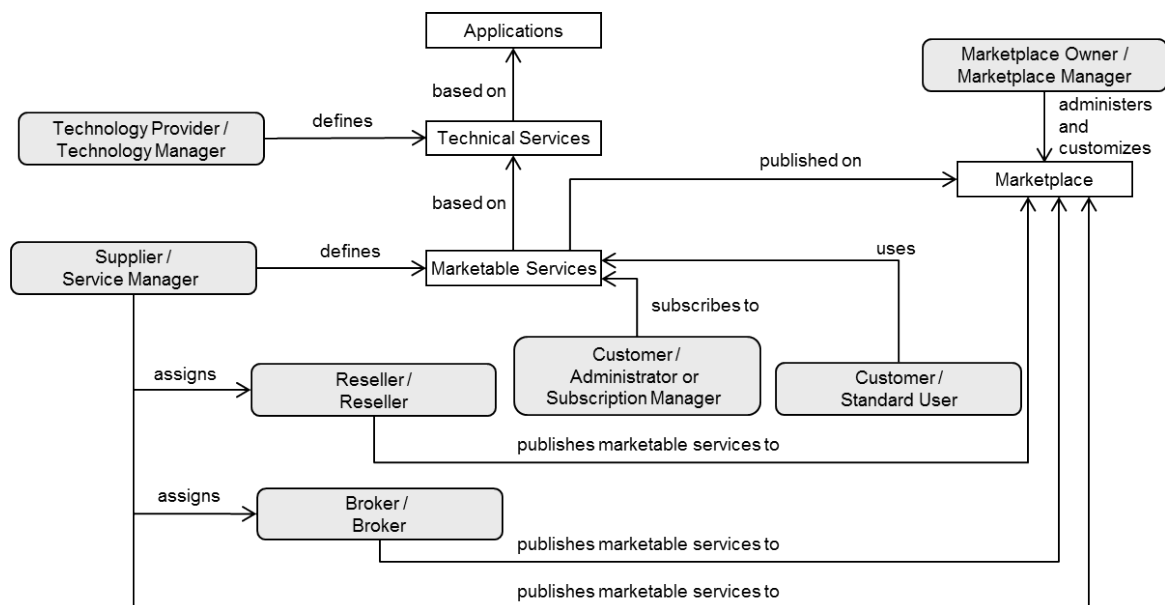
The roles of an organization determine which features are available to its users at the ESCM interfaces and which roles the users can be assigned. These user roles control the actions an individual user is allowed to carry out:

- **Standard user:** Users with this non-privileged role can work with services their organization has subscribed to. Every user registered in ESCM automatically is a standard user. Additional user roles must be assigned explicitly by an administrator.
- **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.
- **OU administrator:** The users of an organization can be grouped in organizational units (OUs). The OU administrator role allows a user to manage the organizational units for which he has been appointed as an administrator, to create, modify, and terminate subscriptions for these units, as well as generate reports for cost-controlling purposes.
- **Subscription manager:** This role allows a user to subscribe to services and manage his own subscriptions. Unlike administrators, subscription managers are not permitted to work on subscriptions belonging to others or on subscription data related to billing and payment.
- **Technology manager:** This role allows a user to define technical services. It can be assigned to users of technology provider organizations.
- **Service manager:** This role allows a user to define marketable services and price models as well as publish marketable services. It can be assigned to users of supplier organizations.
- **Reseller:** This role allows a user to publish a supplier's marketable services applying different terms and conditions. It can be assigned to users of reseller organizations.
- **Broker:** This role allows a user to publish a supplier's marketable services without changing the terms and conditions defined by the supplier. It can be assigned to users of broker organizations.
- **Marketplace manager:** This role allows a user to define the organizations who are permitted to access a marketplace and publish services to it as well as update and customize a marketplace. This role can be assigned to users of marketplace owner organizations. It is automatically assigned to all administrators of the marketplace owner organization when a marketplace is created.
- **Operator:** This role allows a user to carry out configuration and maintenance tasks, manage organizations, and create marketplaces. The first operator is created together with its operator organization when ESCM is installed.

The following illustration provides an overview of how organizations with the different roles are created and related with each other:



The following illustration provides an overview of the user roles and the main tasks of users with these roles:



## 2 Getting Started

After having successfully deployed ESCM, you can follow the instructions on the [Getting Started](#) page for creating and publishing your first service to an ESCM marketplace.

### Accessing ESCM

After the deployment, you have access to the ESCM administration portal where you can specify configuration settings that do not match your environment:

You can access the ESCM administration portal in a Web browser using an URL in the following format:

```
https://<server>:<port>/<context-root>
```

**<server>** is the machine where ESCM has been deployed. **<port>** is by default 8080 for HTTP, 8081 for HTTPS. **<context-root>** is the context root of ESCM (default: `oscm-portal`).

You are prompted for the user ID and password. The initial credentials are as follows:

User ID: `administrator`

Password: `admin123`

It is recommended that you change the initial password in the ESCM administration portal (**Change Password** page in the **Account** menu).

After login, the operator functionality is available in the **Operation** menu.

### Next Steps

In order for a supplier to be able to define price models and for the rating and billing engine of ESCM to be able to calculate usage costs, you need to add one or several currencies to the system. For details, refer to *Adding a Currency to ESCM* on page 18.

In the next step, you can create technology provider and seller organizations (suppliers, resellers, brokers), and set up marketplaces so that services can be offered to customers. For details, refer to *Managing Organizations and Users* on page 28 and *Managing Marketplaces* on page 45.

### Startup and Shutdown

The ESCM containers can be started, stopped, removed, and restarted using the standard utilities of Docker.

To **stop** a container, use the following command on your Docker host:

```
stop <container-name>
```

For example:

```
stop oscm-core
```

To **delete** a container:

```
docker-compose -f docker-compose-oscm.yml rm <container-name>
```

To **create and start** a container:

```
docker-compose up -d docker-compose-oscm.yml <container-name>
```

To **restart** a container:

```
start <container-name>
```

When shutting down the containers, the `oscm-db` container must be stopped as last one. When starting or restarting the containers, make sure that the `oscm-db` container is first.



## 3 Configuring ESCM

This chapter describes:

- How to update configuration settings for ESCM.
- How to add a currency definition to ESCM.
- How to add an additional language to ESCM and how to customize existing texts.
- How to use and configure timers in ESCM.
- How to define and manage system-wide LDAP connection settings.

### 3.1 Updating Configuration Settings

The ESCM software and setup utilities require a number of settings. In the installation, you adapted the settings to your environment, in particular server names, ports, paths, and user IDs.

###The ESCM configuration settings are provided in the following subdirectories of the directory to which you extracted the `oscm-install-pack.zip` file (`<install_pack_dir>`) when you installed ESCM:###

- **databases/bes\_db**
  - `db.properties`: Settings for the database setup and access.
  - `configsettings.properties`: Configuration settings for the ESCM services.  
The initial installation stores these settings in the `bss` database, where you can change them later, if required. An update installation only adds new settings to the database but does not overwrite existing ones. In the case that mandatory settings are missing, an error is thrown, and you need to add these settings manually before executing the installation scripts again.
- **domains/bes\_domain**  
The configuration settings for setting up the application server domain to which the ESCM applications will be deployed are provided in the following files:
  - `glassfish.properties`: Configuration settings for the application server.
  - `glassfishJMSBroker.properties`: Configuration settings for the Java Message Service (JMS) in the application server.
- **domains/indexer\_domain**  
The configuration settings for setting up the application server domain to which the search indexer application will be deployed are provided in the following files:
  - `glassfish.properties`: Configuration settings for the application server.
  - `glassfishJMSBroker.properties`: Configuration settings for the Java Message Service (JMS) in the application server.

For details on the individual configuration settings, refer to *Configuration Settings* on page 47.

**To update the configuration settings for ###JMS### or the database, proceed as follows:**

1. Log in to the administration console of the application server.
2. Adapt the settings as required.

**To update the configuration settings for the ###application server:###**

1. Open the `glassfish.properties` file located in `<install_pack_dir>/domains/bes_domain` with an editor.

2. Check the settings in the file and adapt them to your environment, if required.
3. Save the file to its original location in `<install_pack_dir>/domains/bes_domain`.
4. Update the settings and resources in the application server by executing the `build-glassfish.xml` file in `<install_pack_dir>/install` as follows:

```
<ANT_HOME>/bin/ant -f build-glassfish.xml
    SETUP.configureDomains
```

**Note:** Be aware that existing entries in the JMS queues are deleted by executing the above command.

**To update existing settings for the ESCM services, choose one of the following options:**

- In the ESCM administration portal, choose **Update configuration settings** in the **Operation** menu.

The current configuration settings as stored in the database are retrieved. You can check all settings and update them as required by entering a new value for the corresponding configuration setting.

For detailed step-by-step instructions, refer to the online help.

- Edit the content of the `configsettings.properties` file as required.

Then execute the `build-db.xml` file in `<install_pack_dir>/install` as follows:

```
<ANT_HOME>/bin/ant -f build-db.xml
    UPDATE.configSettings
```

`<ANT_HOME>` is the installation directory of Apache Ant.

**Note:** If you execute the above command while ESCM is running, you need to restart ESCM for viewing the updated configuration settings in the administration portal, because the previous settings still exist in the cache.

## Retrieving the Current Configuration Settings

You can at any time view the current configuration settings for the ESCM services.

**To retrieve the current configuration settings:**

In the ESCM administration portal, choose **Update configuration settings** in the **Operation** menu. For detailed step-by-step instructions, refer to the online help.

For details on the individual configuration settings, refer to *Configuration Settings* on page 47.

## 3.2 Adding a Currency to ESCM

After installation, you need to add the currency or currencies to be supported by ESCM. These currencies will be available to suppliers when defining the price models for marketable services. The selected currencies are used by the integrated rating and billing engine of ESCM when calculating subscription usage charges.

**To add a currency:**

In the ESCM administration portal, choose **Manage currencies** in the **Operation** menu. For detailed step-by-step instructions, refer to the online help.

### 3.3 Adding a Language to ESCM and Customizing Texts

ESCM supports multiple languages in which users can work. After installation, English, German, and Japanese language bundles are available. The operator can add additional languages and customize the texts provided in the English, German, and Japanese language bundles. Refer to *Supported Language Codes* on page 88 for the list of languages that can be added.

Adding a language or customizing existing texts comprises the following steps:

1. Translating all resources of a language bundle:
  - The user interface resources are translated or updated in a Microsoft Excel file.  
Refer to *User Interface Resources* on page 84 for details.
  - The online help and FAQ files are translated or updated directly in the HTML files.  
Refer to *Online Help and FAQs* on page 85 for details.
  - The report labels are translated or updated in properties files.  
Refer to *Report Labels* on page 88 for details.
2. For a new language: Registering the language with ESCM.
3. Importing the translated or updated Microsoft Excel file.
4. Deploying the translated or updated online help and FAQ files.
5. Deploying the translated or updated report labels.
6. For a new language: Activating the language in ESCM. All activated languages are available to users for selection when they edit their user profile.

#### To add a language:

In the ESCM administration portal, choose **Manage languages** in the **Operation** menu. For detailed step-by-step instructions, refer to the online help.

### 3.4 Configuring Timers

Timers are used to handle background tasks, for example to check for expired subscriptions that are to be deleted. Each timer has a time interval specifying when it is executed periodically. The values are indicated in milliseconds.

Time intervals are defined and can be changed in the ESCM configuration settings.

**Note:** Be aware that several functions in ESCM will not work if you do not enable the timers. For example, if a customer specifies that a subscription is to expire in 10 days, and you did not configure the timer `RESTRICTED_SUBSCRIPTION_USAGE_PERIOD`, the subscription will not expire after 10 days.

The initial expiration time of a timer for which a time interval is defined is calculated based on January 1st, 00:00:00.000, of the current year. For example, if you specify an interval of one week for a timer on January 5th, 14:30:00, the timer will expire for the first time on January 8th at 00:00:00.000, next on the 15th, the 22nd, etc. Or, if you specify an interval of one month for a timer on August 4th, 17:00:00, the timer will expire for the first time on September 1st at 00:00:00.000, next on October 1st, November 1st, etc. To avoid the expiration of several timers at the same time, which would result in heavy load on the system, there is an additional setting: An **offset** for each timer. The offset is added to the expiration time.

## Example

For the timer used to remove customer accounts that have not been confirmed, the following configuration settings are defined:

```
PERMITTED_PERIOD_UNCONFIRMED_ORGANIZATIONS=604800000
```

```
TIMER_INTERVAL_ORGANIZATION=86400000
```

```
TIMER_INTERVAL_ORGANIZATION_OFFSET=300000
```

The first setting indicates how long an organization account is allowed to remain unconfirmed: 7 days. The second setting indicates the time interval at which the check for unconfirmed accounts is executed: every 24 hours. The third setting, the offset, is set to 5 minutes. The timer will expire at 0:05 a.m. every day. The offset is not accumulated, but stays the same every day.

## Available Timers

There are the following timers:

- **ORGANIZATION\_UNCONFIRMED:** Timer to check for organization accounts that have not been confirmed by a login of the initial administrator within a certain period of time. When this timer expires, the respective organization accounts are removed.  
This timer requires the following configuration settings:
  - `TIMER_INTERVAL_ORGANIZATION`
  - `TIMER_INTERVAL_ORGANIZATION_OFFSET`
  - `PERMITTED_PERIOD_UNCONFIRMED_ORGANIZATIONS`
- **USER\_NUM\_CHECK:** Timer to check for the current amount of users registered with the platform.  
This timer requires the following configuration setting:
  - `TIMER_INTERVAL_USER_COUNT`
- **RESTRICTED\_SUBSCRIPTION\_USAGE\_PERIOD:** Timer used to ensure that subscriptions can only be used for the time specified in the underlying service's parameters. If this period is exceeded, the timer-related operations must be executed to make sure that the subscription cannot be used anymore unless the supplier upgrades or downgrades the underlying service.  
This timer requires the following configuration settings:
  - `TIMER_INTERVAL_SUBSCRIPTION_EXPIRATION`
  - `TIMER_INTERVAL_SUBSCRIPTION_EXPIRATION_OFFSET`
- **TENANT\_PROVISIONING\_TIMEOUT:** Timer used to check pending subscriptions. When the timeout time is reached, an email is sent to the administrators and the relevant OU administrators and subscription managers of the organizations who created the subscriptions, informing them about the timeout.  
This timer requires the following configuration settings:
  - `TIMER_INTERVAL_TENANT_PROVISIONING_TIMEOUT`
  - `TIMER_INTERVAL_TENANT_PROVISIONING_TIMEOUT_OFFSET`
- **BILLING\_INVOCATION:** Timer for billing runs calculating subscription usage costs (customer billing data) or revenue share data. The interval for this timer is one day and cannot be changed.  
This timer requires the following configuration setting:
  - `TIMER_INTERVAL_BILLING_OFFSET`

- **DISCOUNT\_END\_CHECK**: Timer used to check whether the end date for discounts granted to customers has been reached. The timer interval is one day and cannot be changed.  
This timer requires the following configuration settings:
  - `TIMER_INTERVAL_DISCOUNT_END_NOTIFICATION_OFFSET`
- **INACTIVE\_ON\_BEHALF\_USERS\_REMOVAL**: Timer used to remove non-existing users from the database that were created because an organization acted on behalf of another organization.  
The timer for database cleanup requires the following configuration settings:
  - `TIMER_INTERVAL_INACTIVE_ON_BEHALF_USERS`
  - `TIMER_INTERVAL_INACTIVE_ON_BEHALF_USERS_OFFSET`

For a detailed description of the timers, refer to *ESCM Configuration Settings* on page 48.

**To configure a timer:**

In the ESCM administration portal, choose **Update configuration settings** in the **Operation** menu. For detailed step-by-step instructions, refer to the online help.

### Retrieving Expiration Times

You can check when the currently registered timers expire.

**To retrieve the expiration times:**

In the ESCM administration portal, choose **Manage timers** in the **Operation** menu. For detailed step-by-step instructions, refer to the online help.

### Re-Initializing Timers

When you set a timer or update the settings for a timer, you need to re-initialize the timers in order to start them.

**To re-initialize the timers:**

In the ESCM administration portal, choose **Manage timers** in the **Operation** menu. For detailed step-by-step instructions, refer to the online help.

## 3.5 Managing LDAP Settings

User IDs and passwords of an organization can be created and maintained in the platform or in an existing LDAP system of an organization.

When maintained in the platform, the user data is stored in the platform's database. An organization's administrator can register new users, and, if required, request passwords to be reset by the operator.

When using an LDAP system, an organization does not need to register its users manually with the platform. The organization's administrator can import the users from the LDAP system, thus automatically registering them with the platform. The users are managed in the LDAP system. The platform continuously synchronizes its information on the users. Connection settings have to be defined in a configuration file so that the platform can connect to the LDAP system.

Whether an organization uses an LDAP system for user management is determined when the organization is created in the platform. An organization can be created in several ways:

- A customer registers himself. In this way, an organization with the customer role is created. Users are always managed in the platform, and an LDAP system cannot be used.

- A seller (supplier, reseller, or broker) registers a customer. In this way, an organization with the customer role is created. The seller can specify whether user management in an external LDAP system is to be used.
- You as the platform operator create an organization of any role. For any organization, you can specify whether user management in an external LDAP system is to be used. For details, refer to *Creating an Organization* on page 28.

A mixture of maintaining users in the platform and in the LDAP system is not supported. In addition, the type of user management can no longer be changed once an organization has been created.

The operator organization cannot use an external LDAP system. So you can always log in to the platform and change connection settings, for example if the LDAP system of an organization is not available and thus authentication against it is not possible.

As an operator, you can define default LDAP configuration settings for the entire platform. These settings apply as long as no organization-specific LDAP settings are specified. You can also check an organization's connection to its LDAP system.

## Defining Default LDAP Settings

Defining default LDAP settings for the entire platform is useful if settings are to be reused for several organizations. If organization-specific LDAP settings exist, they overrule the default LDAP settings.

### To define default LDAP settings:

In the ESCM administration portal, choose **Manage LDAP settings** in the **Operation** menu. For detailed step-by-step instructions, refer to the online help. For a list of LDAP keys, refer to *LDAP Keys* on page 58.

## Checking the Connection to an LDAP System

If for some reason an organization's LDAP system cannot be reached, you can check and restore the connection so that user authentication is possible again.

### To check an LDAP connection:

In the ESCM administration portal, choose **Manage LDAP settings** in the **Operation** menu. For detailed step-by-step instructions, refer to the online help. For a list of LDAP keys, refer to *LDAP Keys* on page 58.

## 4 #####Monitoring, Backup and Recovery####? How does this work now?####

Regular system operation and maintenance includes the monitoring of the system and its processes as well as the backup of the databases, configuration settings, and log files.

This chapter describes:

- The monitoring of ESCM, including the types and locations of log files.
- How to configure the log level.
- The format of the log entries.
- How to maintain the log files.
- How to export audit log data.
- The backup of the databases, configuration settings, encryption key file, and log files.
- The monitoring of the JMS queues.

### 4.1 Monitoring ESCM

In addition to entries in the standard application server log file, ESCM provides log files of its own that help you detect problems and identify, for example, illegal access to the system.

The logging of ESCM is based on the `log4j` tool.

#### 4.1.1 Log Types and Location of Log Files

ESCM writes its logging information into the following log files:

- `system.log`: System status, errors, warnings, and debug information.
- `access.log`: User logins.
- `audit.log`: System security status.

The log files are stored in the location defined by the `LOG_FILE_PATH` configuration setting. By default, they are located in the following directory:

```
<GLASSFISH_HOME>/glassfish/domains/bes-domain/logs
```

In addition to the ESCM-specific log files, the application server log file (`server.log`) is another source for tracking system usage and possible errors.

#### 4.1.2 Configuring the Log Level

ESCM supports the following types of log information with their corresponding log level:

- **ERROR**: Problems that do not allow to continue working with ESCM in the current transaction or that indicate an issue which must be solved. With the `ERROR` log level set, the log file contains all exceptions that occurred at runtime with a complete stack trace.
- **WARN**: Problems that allow for the completion of an operation, irrespective of whether the operation is completed fully or only partially. For example, an email could not be sent. With the `WARN` log level set, the log file comprises all messages of type `WARN` and `ERROR`.
- **INFO**: Basic information such as the state of the server, whether it was started or stopped, whether a user logged in. With the `INFO` log level set, the log file comprises all messages of type `INFO`, `WARN`, and `ERROR`.

- **DEBUG:** Detailed information with references to the ESCM implementation, mainly start and exit of methods as well as proposals of how to solve the problem. With the `DEBUG` log level set, the log file comprises all messages.

The `log4j` logging framework optionally uses the `log4j.properties` configuration file. The `log4j.properties` file contains the settings for the log level and a list of configuration settings that determine the behavior of the framework at runtime. For details on `log4j` settings, refer to the relevant Apache documentation

The log level defines which type of information is written to the log files. You can change the log level by adapting the file (see below). By default, the `INFO` log level is used.

The `LOG_CONFIG_FILE` configuration setting stores the absolute path to the `log4j.properties` configuration file. By default, ESCM looks for this file in the `config` subdirectory of your application server domain:

```
<GLASSFISH_HOME>/glassfish/domains/bes-domain/config
```

When the `log4j.properties` file is specified for the first time, ESCM needs to be restarted. Afterwards, if the `log4j.properties` file is found, its content is checked for changes every 60 seconds. If there are changes to the log level, they are applied to all loggers at runtime; ESCM need not be restarted.

If the `log4j.properties` file does not exist or the value for the `LOG_CONFIG_FILE` configuration setting is not set or has been changed, the log level is the one stored in the `LOG_LEVEL` configuration setting. If you change this setting, you need to restart ESCM for the new level to become effective.

The `log4j.properties` file may contain entries for defining different log levels for the different ESCM services. For example, if the following is defined in the `log4j.properties` file, all events caused by the Identity Service are logged on `DEBUG` level, all others on `INFO` level in the `system.log` file.

```
log4j.rootLogger=INFO
log4j.logger.sys.org.oscm.identityservice=DEBUG
```

You can define different log levels for the following services:

Service	Package name
Account management service	org.oscm.accountservice
Billing service	org.oscm.billingservice
Categorization service	org.oscm.categorizationservice
Discount service	org.oscm.discountservice
Event management service	org.oscm.eventservice
Identification service	org.oscm.identityservice
Marketplace management service	org.oscm.marketplace
Reporting service	org.oscm.reportingservice
Review service	org.oscm.reviewservice
Search service	org.oscm.searchservice
Service provisioning service	org.oscm.serviceprovisioningservice



Service	Package name
Session service	org.oscm.sessionservice
Subscription management service	org.oscm.subscription-service
Tag service	org.oscm.tag-service
Trigger service	org.oscm.trigger-service
Trigger definition service	org.oscm.trigger-definition-service
VAT service	org.oscm.vat-service

To log every operation in ESCM, you can use the following line in the `log4j.properties` file:

```
log4j.logger.sys.org.oscm=DEBUG
```

### 4.1.3 Log Entry Format

Every entry in an ESCM log file has the following format:

```
<YYYY-MM-DD'T'HH:MM:SS z> LogLabel:<label> LogType:<level> MessageID:
<MessageID> Message: <Message>
```

`LogLabel` identifies the product that created the log entry. For ESCM, this is `FSP_INTS-BSS`. `LogType` is one of the log levels, `MessageID` is a numeric identifier of the entry.

Detailed information on the logged issue is contained in the `Message` text. The text contains the ID of the thread handling a transaction, which can be a useful indicator of relations between the activities on the server itself. Knowing the ID of the thread handling a transaction, you can completely track all steps involved in the transaction and thus easily investigate an error.

### 4.1.4 Maintaining the Log Files

New messages are continually appended to the log files at runtime. When a file reaches a size of 10 MB, a new one is created; the number 1 is appended to the old one. When the second log file reaches 10 MB in size, the number 2 is appended to it, and a third log file is created. When the maximum number of 5 log files is reached, the first one is overwritten.

It is recommended to check the log files and their sizes from time to time and to create a backup when you see that the maximum number of files will be reached.

## 4.2 Exporting Audit Log Data

You can export information on all kinds of user operations related to subscriptions and marketable services including their price model. The data represents an audit log. Exporting this data may be useful, for example to check when and by whom a price model was changed or when subscriptions were created.

You can export information on subscription-related operations performed by administrators or standard users, as well as information on service and price model-related operations performed by service managers of supplier organizations. Administrator operations include actions performed by OU administrators and subscription managers. Service manager operations include actions performed by resellers and brokers.

**Note:** The system records user operations only if the `AUDIT_LOG_ENABLED` configuration setting is set to `true`.

You can specify the start and end date of the time period for which you want to export the log data. The dates must be specified in the format `YYYY-MM-DD`.

#### To export the audit log:

In the ESCM administration portal, choose **Export audit log** in the **Operation** menu. For detailed step-by-step instructions, refer to the online help.

You can choose to view the exported data or save it to a file. For details on the content, refer to *Audit Log* on page 60.

## 4.3 Backup and Recovery

ESCM does not offer integrated backup and recovery mechanisms. Use the standard file system, application server, and database mechanisms instead.

### Backup

It is recommended to create a regular backup of the following data according to the general guidelines of the data center:

- ESCM database (`bss`). The frequency of database backups depends on the amount of changes and on the availability of time slots with low load. PostgreSQL supports database backups without previous shutdown. For details, refer to the PostgreSQL documentation.

Make sure to also make a backup of the file containing the key required for encryption and decryption of service parameters with data type `PWD` and custom attributes marked for encryption. By default, this file is named `key` and located in the following directory:

```
<GLASSFISH_HOME>/glassfish/domains/bes-domain/config
```

The location of this file can be changed using the `KEY_FILE_PATH` configuration setting.

- Certificates contained in the `bes-domain` keystore and truststore (`keystore.jks` and `cacerts.jks` files).
- ESCM log files located in the configured directory.
- Configuration files.
- Customizations made to marketplaces, especially style sheets and localized texts presented at the user interface.

**Note:** When preparing for an update installation of your current ESCM release, always create a backup of all data mentioned above.

### Recovery

If you need to recover your ESCM installation, the recommended procedure is as follows:

1. Make sure that all JMS messages have been processed. They are stored in the `bssjms` database. For example, check the JMS broker as follows:

```
<GLASSFISH_HOME>/mq/bin/imqcmd.exe -b localhost:8076
-u admin query bkr
```

where `8076` is the port where the JMS broker is running.

2. Stop the ESCM domain and the master indexer domain.
3. Restore the `bss` database from the backup using the relevant PostgreSQL commands.

4. Make sure that the file containing the key required for encryption and decryption of service parameters with data type `PWD` and custom attributes marked for encryption exists in the location specified in the `KEY_FILE_PATH` configuration setting. By default, this file is named `key` and located in the following directory:

`<GLASSFISH_HOME>/glassfish/domains/bes-domain/config`

If the file is missing, copy it from your backup to the correct location.

5. Start the master indexer domain.
6. Start the ESCM domain.  
The search index is rebuilt automatically.

## 5 Managing Organizations and Users

This chapter describes how to:

- Create an organization.
- Maintain the address data of organizations.
- Add a role to an organization.
- Manage user accounts: lock and unlock user accounts, and reset the password for a user.
- Register users for organizations.

### 5.1 Creating an Organization

You can create an organization and specify its roles (technology provider, supplier, reseller, broker, or - implicitly - customer). For every new organization, you must define a user who is to become its first administrator. This user can later register new users and assign roles to them.

**To create an organization:**

In the ESCM administration portal, choose **Create organization** in the **Operation** menu. For detailed step-by-step instructions, refer to the online help.

### 5.2 Changing the Address Data of an Organization

As an operator, you can at any time update the address data of an organization.

**To update the address data of an organization:**

In the ESCM administration portal, choose **Manage organization** in the **Operation** menu. For detailed step-by-step instructions, refer to the online help.

### 5.3 Adding a Role to an Organization

As an operator, you can add a role to an organization (technology provider, supplier, reseller and/or broker).

**To do so:**

In the ESCM administration portal, choose **Manage organization** in the **Operation** menu. For detailed step-by-step instructions, refer to the online help.

### 5.4 Managing User Accounts

As an operator, you can lock the account of a specific user, for example when you assume unauthorized access. You can also unlock an account, for example in case it was locked because the user tried to log in using a wrong password for the configured number of times (default: 3). In addition, you can initiate the generation of a new password for a user. In all cases, the user is notified by email.

You can, at any time, view the number of users registered with ESCM as well as the configured maximum number of registered users (default: 10). At regular intervals (default: 12 hours), the system checks the number of registered users. You are informed by email when this check returns that the allowed number of registered users is exceeded.

**Note:** If an organization uses an external LDAP system for user management, only those fields that are not mapped to settings in the LDAP system can be edited. For example, passwords can only be changed in the LDAP system.

**To manage user accounts:**

In the ESCM administration portal, choose **Manage users** in the **Operation** menu and click the appropriate button. For detailed step-by-step instructions, refer to the online help.

## 5.5 Registering Multiple Users for Organizations

Users that are to work with an organization's subscriptions are usually registered by an organization's administrator.

As an operator, you can also register multiple users on behalf of organizations that do not use an external system for user authentication. The data of the users must be specified in a user data file which can then be imported into ESCM.

The user data file must be in `csv` (comma-separated values) format. For details, refer to *User Data File for Multiple User Import* on page 92.

**To register multiple users for an organization:**

In the ESCM administration portal, choose **Manage users** in the **Operation** menu. For detailed step-by-step instructions, refer to the online help.

## 6 Reporting

ESCM offers comprehensive reports for different purposes and at different levels of detail. You can choose from various predefined reports.

The following reports are available for operators:

- **Supplier revenue report:** Shows the accumulated revenues of all suppliers and resellers registered with the platform.  
A time frame can be specified. Dates are to be entered in the format `YYYY-MM-DD`.  
The report shows the name, ID, and revenue of each supplier and reseller, and one accumulated value per currency. A supplier's revenue includes all revenues generated by his authorized brokers.
- **External services report:** Shows all sellers (suppliers, brokers, resellers) who have published services with the external access type to a marketplace on the platform.  
For each seller, the report shows the corresponding marketable services with their activation and deactivation time. If the services were activated and deactivated several times, all timestamps are listed.
- **Supplier revenue share report:** Shows the revenue share data for all suppliers registered with the platform for a specified month.  
For each marketplace, the revenue share data is calculated from the accumulated charges for subscriptions which were due during the past calendar month, irrespective of supplier-specific billing periods. For each supplier, the data is broken down to the organizations which generated the revenue as well as to the individual services. The suppliers' liabilities to the other participating parties are calculated. Discounts granted by a supplier to his customers are deducted from the revenue shares. The operator can specify the month for which to generate the report. The month is to be entered in the format `MM`, the year in the format `YYYY`.
- **Broker/reseller revenue share report:** Shows the revenue share data for all brokers and resellers registered with the platform.  
For each marketplace, the revenue share data is calculated from the accumulated charges for subscriptions which were due during the past calendar month. For each broker and reseller, the data is broken down to the suppliers who are providing the services to the brokers and resellers. The operator can specify the month for which to generate the report. The month is to be entered in the format `MM`, the year in the format `YYYY`.
- **Service report (of a supplier):** Shows all marketable services of a supplier with their existing subscriptions. Subscriptions to services offered by brokers are also listed. The operator selects the supplier by specifying the supplier organization ID.
- **Customer report (of a supplier):** Shows all customers of a supplier and his brokers with their organization ID and the services they have subscribed to. The report outputs whether a subscription is still active or has already ended. The operator selects the supplier by specifying the supplier organization ID.
- **Billing report (of a supplier):** Shows a summary of all billing data for each customer of a supplier and his brokers.  
The operator selects the supplier by specifying the supplier organization ID. The billing data includes the billing data key of each subscription.
- **Detailed billing report for an existing invoice of a supplier's customer:** Shows the billing data of the current billing period for a selected subscription of a supplier's customer.

With pro rata cost calculation, the costs for the real service usage are calculated based on milliseconds. With per time unit calculation, the report contains the subscription costs for the time units that ended in the billing period. The operator selects the subscription by specifying a billing data key. The billing data key of each subscription is output by the **Billing report (of a supplier)**.

**To create a report:**

In the ESCM administration portal, choose **Create report** in the **Account** menu, and select the supplier revenue report. For detailed step-by-step instructions, refer to the online help.

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

**Note:** The reporting functionality is available only if you have defined the following configuration parameters: `REPORT_ENGINEURL`, `REPORT_SOAP_ENDPOINT`, `REPORT_WSDLURL`. For details, refer to *ESCM Configuration Settings* on page 48.

## 7 Managing Billing and Payment

This chapter describes concepts and tasks of the operator related to billing and payment. It explains what billing runs are and how to:

- Define revenue shares.
- Integrate the services of a PSP with ESCM.
- Assign payment types to suppliers and resellers.
- Handle billing data, including how to manually start a billing run or payment processing and retry failed payment processes, as well as how to preview or export billing data to an XML file.

### 7.1 Billing Runs

At daily intervals, the database content is checked for organizations which have produced billing-relevant data, the billing data is calculated, and the data is collected and stored in the database. These checks and calculations are called "billing runs for customer billing data".

Each supplier and reseller can define his preferred start day of the monthly billing periods. When a billing period ends, the costs for all customer subscriptions that were produced in the course of this period are calculated. This is done in the billing run that is executed on the start day of the next billing period plus the offset defined by the `TIMER_INTERVAL_BILLING_OFFSET` configuration setting. For example: A supplier defines that his billing period is to start on the 8th of a month. The `TIMER_INTERVAL_BILLING_OFFSET` is set to 5 days and 4 hours. The billing run calculating the costs for the supplier's customers is executed on the 13th of each month at 04:00:00.000.

Brokers and resellers, as well as the platform operator and the owners of the marketplaces on which services are published, usually receive a share of the revenue for the services. The revenue shares for these organizations and their effects on suppliers' revenues are calculated every month based on the results of the daily checks for customer billing data. This calculation is called "billing run for revenue share data". The billing period for this billing run cannot be changed; it is always executed for the previous month on the first day of a month plus the offset defined by the `TIMER_INTERVAL_BILLING_OFFSET` configuration setting.

When a billing run for customer billing data is finished, the billing data that must be handled by a payment service provider (PSP) is processed. This transaction is called "payment processing".

You should check at regular intervals whether billing runs or payment processing fail, and, if yes, explicitly execute them.

### 7.2 Defining Revenue Shares

Suppliers may involve brokers and resellers in selling their services. The brokers and resellers as well as the platform operator and the owners of the marketplaces on which the services are published, usually receive a share of the revenue for the services. ESCM calculates these revenue shares based on the billing data for the customers who use the services. The operator revenue share applies irrespective of whether services are sold by the supplier or by his resellers and brokers. Discounts granted by a supplier to his customers are deducted from the operator revenue share.

Suppliers, brokers, resellers, and marketplace owners can generate reports for their revenue shares and export the revenue share data for a specific month. As an operator, you can export the data for all the suppliers, brokers, resellers, or marketplace owners known to your platform installation. The exported data can be forwarded, for example, to an accounting system which continues to process it. For details, refer to *Handling Billing Data* on page 37.



As an operator, you are responsible for defining the revenue shares. You can define the following:

- Operator revenue share to be paid by each supplier for using the platform.
- Marketplace owner revenue share to be paid by a supplier for publishing services on a specific marketplace.
- Revenue shares for broker and reseller organizations to be paid by a supplier for selling his services.

The revenue shares specify the percentage of the revenue the operator, marketplace owners, brokers, or resellers are entitled to.

The values you enter for revenue share percentages are based on agreed conditions between the operator, marketplace owners, suppliers, resellers, and brokers. You as the operator are responsible for setting correct values. It is possible to set the percentages to a total of over 100%. This might be intentional. For example, a supplier who wants to strongly promote a service for a limited period of time may grant a broker or reseller a revenue share of 80%. In addition, the supplier may need to pay 30% of the revenue to the marketplace owner and 10% to the operator. This results in a total revenue share percentage of 120 and thus in a negative revenue for the supplier.

Revenue shares can be defined on the following levels:

1. **Operator revenue share.** It is defined for every supplier organization that wants to sell its services on your platform. The default operator revenue share is specified when you create a supplier organization. It is independent of the marketplace to which a service is published, and whether the supplier sells services himself or authorizes resellers and brokers to do so. Discounts granted by a supplier to his customers are deducted from the operator revenue share. The default operator revenue share applies to all services of a supplier as long as you do not define service-specific operator revenue shares. A service-specific operator revenue share applies to a specific marketable service and overrules the default operator revenue share defined for the respective supplier organization.
2. **Marketplace-specific revenue shares.** They are defined for one marketplace and comprise a percentage for the marketplace owner as well as a default for all broker and reseller organizations. Discounts granted by a supplier to his customers are deducted from the revenue share for the marketplace owner.
3. **Service-specific revenue shares.** For every marketable service a supplier offers to brokers or resellers for publishing, a service-specific revenue share can be defined. This revenue share applies irrespective of the marketplace the service is published to. It overrules any defined marketplace-specific revenue shares.
4. **Individual broker or reseller revenue shares.** An individual revenue share can be defined for each broker and reseller organization. This revenue share definition overrules the service-specific and marketplace-specific revenue share definitions.

**To define operator revenue shares, proceed as follows:**

- Default operator revenue share for a supplier organization:  
In the ESCM administration portal, choose **Create organization** in the **Operation** menu for a new supplier organization or **Manage organization** in the **Operation** menu for an existing supplier organization.
- To define a service-specific operator revenue share:  
In the ESCM administration portal, choose **Manage operator revenue share** in the **Operation** menu.  
For detailed step-by-step instructions, refer to the online help.

**To define marketplace-specific revenue shares:**

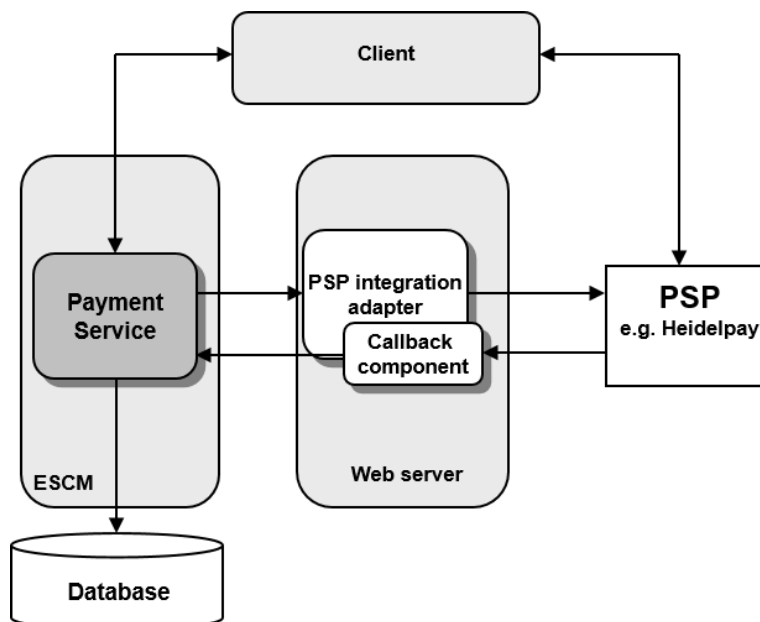
In the ESCM administration portal, choose **Update marketplace** in the **Marketplace** menu. For detailed step-by-step instructions, refer to the online help.

**To define revenue shares for specific services, brokers, or resellers:**

In the ESCM administration portal, choose **Manage broker revenue share** or **Manage reseller revenue share** in the **Marketplace** menu, respectively. For detailed step-by-step instructions, refer to the online help.

## 7.3 Integrating PSP Support

For payment collection, a supplier or reseller organization can use the facilities of a commercial payment service provider (PSP). ESCM provides a Web service interface (payment service) to enable operator organizations to integrate payment service providers (PSPs) with ESCM. The following figure illustrates the communication paths between the components involved in PSP integration:



The process of integrating PSP support for a supplier or reseller organization and the tasks to be performed by an operator are as follows:



The subsequent sections describe the prerequisites and the operator tasks in detail.

### 7.3.1 Prerequisites

Before a supplier or reseller can use the services of a commercial payment service provider (PSP), several steps need to be performed:

1. Your organization, i.e. the operator organization, must be registered in the PSP system.
2. The suppliers or resellers who are to use the services of the PSP need to have a contractual relationship with the PSP. The supplier or reseller organizations must also be registered in the PSP system.
3. The PSP issues an identifier for the supplier or reseller so that payments can be immediately linked to the supplier's or reseller's account.
4. Your own organization and the PSP organization must have exchanged certificates so that trusted and secure communication between ESCM and the PSP system is ensured.

For details, refer to *Integrating Certificates for Trusted Communication* on page 39.

5. A PSP integration adapter must have been implemented as a Web service and deployed on an application server. This adapter is a Web service providing the methods for integrating the PSP with ESCM. You need the URL pointing to the WSDL file of this service.

The PSP must have informed you about the types of payment he offers and which configuration values to set for using his services.

You find a detailed description of the implementation of a PSP integration adapter in the documentation shipped with the `oscm-integration-psp-pack.zip` file.

6. You need to make the following configuration setting for ESCM:

`PSP_USAGE_ENABLED`: Set the value to `true` in order to enable PSP integration (default: `false`).

For details, refer to *Updating Configuration Settings* on page 17.

### 7.3.2 Registering a Payment Service Provider With ESCM

Before an organization can use the services of a commercial PSP, you need to register the PSP with ESCM.

#### To register a PSP:

In the ESCM administration portal, choose **Create payment service provider** in the **Operation** menu. For detailed step-by-step instructions, refer to the online help.

You need to provide the following information:

- Identifier of the PSP. This ID uniquely identifies the PSP in ESCM and is visible to the operator only.
- WSDL URL: Link to the WSDL file containing the PSP integration adapter.
- Distinguished name retrieved from the certificate of the Web server on which the PSP integration adapter has been deployed. The integration of a PSP with ESCM requires the usage of certificates for trusted communication between the platform and the PSP. For details, refer to *Integrating Certificates for Trusted Communication* on page 39.

### 7.3.3 Configuring PSP Communication and Payment Types

Before an organization can use the services of a commercial PSP, you need to specify the parameter values required for communicating with the PSP and define the payment types offered by the PSP.

#### To configure a PSP and define the payment types offered by the PSP:

In the ESCM administration portal, choose **Manage payment service provider** in the **Operation** menu. For detailed step-by-step instructions, refer to the online help.

### 7.3.4 Setting the PSP Identifier for a Supplier or Reseller

When defining the payment types offered by a PSP for a supplier or reseller organization, you need to set the ID provided by the PSP for the organization. This ID may be, for example, the supplier's or reseller's account number at the PSP.

An organization needs to have a contractual relationship with the PSP. The PSP configures the channel for a supplier or reseller organization and issues an identifier so that payments can be immediately linked to the supplier's or reseller's account.

#### To set the PSP identifier for an organization:

In the ESCM administration portal, choose **Manage payment service provider** in the **Operation** menu. For detailed step-by-step instructions, refer to the online help.

## 7.4 Assigning Payment Types to Suppliers and Resellers

You are responsible for setting up the payment types that suppliers and resellers can offer to their customers. By default, suppliers and resellers can allow customers to pay on invoice so that they can subscribe to a service that is not free of charge.

If a supplier or reseller requests a payment type other than on invoice, he must use the services of a commercial PSP.

### To set the payment types for a supplier or reseller:

In the ESCM administration portal, choose **Manage organization** in the **Operation** menu. For detailed step-by-step instructions, refer to the online help.

## 7.5 Handling Billing Data

This section describes how you can check whether billing runs have failed, re-invoke payment processes, or explicitly start a billing run. In addition, you can preview customer billing data for a specific period or export revenue share data to an XML file.

Use the ESCM reporting facilities to retrieve detailed information on all billing-relevant data of the suppliers, resellers, brokers, and marketplace owners managed on your platform. For details, refer to *Reporting* on page 30.

The definition of revenue shares is described in detail in *Defining Revenue Shares* on page 32.

### 7.5.1 Start Billing Run

You can explicitly start a billing run for calculating the customer billing data for a billing period. The billing period for which the data is generated depends on the day when you start the billing run and on the `TIMER_INTERVAL_BILLING_OFFSET` configuration setting. Explicitly starting a billing run is useful, for example, if you want to check the payment processing of billing data by a PSP test system, and you do not want to wait for the automatic calculation of the data.

If you start a billing run on the first day of a month plus the day(s) defined in the `TIMER_INTERVAL_BILLING_OFFSET` setting, the revenue share data is also calculated and stored.

### To start a billing run:

In the ESCM administration portal, choose **Execute billing tasks** in the **Operation** menu, and click **Execute** in the respective section of the Web page. For detailed step-by-step instructions, refer to the online help.

### 7.5.2 Start Payment Processing

You can manually start the payment processing of customer billing data that must be handled by a PSP.

### To start the payment processing:

In the ESCM administration portal, choose **Execute billing tasks** in the **Operation** menu, and click **Execute** in the respective section of the Web page. For detailed step-by-step instructions, refer to the online help.

### 7.5.3 Retry Failed Payment Processes

When communication problems caused the automatic payment processing for an organization to fail, you can re-invoke these payment processes manually.

### To retry failed payment processes:

In the ESCM administration portal, choose **Execute billing tasks** in the **Operation** menu, and click **Execute** in the respective section of the Web page. For detailed step-by-step instructions, refer to the online help.

### 7.5.4 Preview Billing Data

You can preview the billing data for a customer's subscriptions for a specified time frame. The billing data can be saved to an XML file or opened in an editor of your choice. You can edit the billing data and work with it as required, for example when forwarding the data to an accounting system.

The billing data preview collects the billing-relevant data for the specified customer and accumulates it for every day within the specified time frame. The data is not stored in the database; the result is just a cost projection for the customer organization.

**To preview billing data:**

In the ESCM administration portal, choose **Billing data preview** in the **Operation** menu. For detailed step-by-step instructions, refer to the online help.

The billing data is saved to an XML file (<date>BillingData.xml).

You find a detailed description of the XML file elements in the *Developer's Guide*.

### 7.5.5 Export Revenue Share Data

You can export the revenue share data for all organizations with a specific role for a specific time frame. Based on the customer billing data calculated for the given time frame, the costs are analyzed to determine the revenue shares for the operator, marketplace owners, brokers, and resellers and their effects on the suppliers' revenues. Discounts granted by a supplier to his customers are deducted from the revenue shares.

You can use the data to get an overview of who is to receive which revenue shares. The exported data can be forwarded, for example, to an accounting system which continues to process it. For example, you can invoice your revenue share to marketplace owners or suppliers. For details on defining revenue shares, refer to *Defining Revenue Shares* on page 32.

**To export billing data:**

In the ESCM administration portal, choose **Export billing data** in the **Account** menu. For detailed step-by-step instructions, refer to the online help.

The billing data can be saved to an XML file (<date>BillingData.xml) and opened in an editor of your choice. You can edit the data and work with it as required.

For a detailed description of the elements and attributes that may occur in the XML file, refer to the *Developer's Guide*.

## 8 Integrating Certificates for Trusted Communication

Certificates are required for ESCM to allow for trusted communication between ESCM and an application underlying a technical service, a payment service provider (PSP), an Identity Provider (IdP) or a Security Token Service (STS).

The following organizations are involved when using certificates:

- Platform operator
- Technology providers integrating their applications with ESCM.
- PSPs whose services are to be integrated with ESCM for invoicing and payment collection.
- Organizations integrating an external process control system.
- Any other organization using Web service calls to or from ESCM

### 8.1 Introduction

Web service calls coming from ESCM (e.g. for provisioning application instances for services, or for integrating a process control system) or sent to it (e.g. by an application such as a PSP system) can be secured with SSL. SSL is used for authentication and for encryption at the transport level.

Every HTTPS connection involves a client and a server. Depending on the calling direction, ESCM can act as a server (Web service calls to ESCM) or as a client (Web service calls from ESCM).

Every application integrated with ESCM or accessing the platform services acts as a Web service client. ESCM acts as the server, and the client must provide its authenticating data to the server:

The caller sends the key or ID and password of an ESCM user. SOAP-based calls address the ESCM Web services with the `BASIC` suffix. This mechanism does not apply certificates for user authentication. However, certificates are involved because the communication between the client and ESCM must be secured using TLS/SSL and HTTPS. You should also configure your network's firewall to block JNDI lookups from the outside when using basic authentication.

**Note:** Both, the server and the client certificates must be created and signed using the same JRE/JDK. Otherwise, the communication may fail.

#### Terminology

ESCM uses an **X.509** certificate to prove the identity of an entity. This certificate is always used to prove the server's identity and optionally to prove the client's identity.

A certificate has a **subject** which usually identifies the owner of the certificate, and an **issuer** who signed the certificate. A certificate also includes a validity period. Cryptographic algorithms ensure that the information contained in the certificate cannot be changed without breaking the signature of the certificate.

The subject as well as the issuer is given as a **distinguished name (DN)** consisting of a list of key-value pairs. One of the standardized keys is called **common name (CN)**. The CN is of particular importance to HTTPS servers: The CN must contain the server's domain; otherwise the client will refuse the connection.

The process of issuing a certificate for another entity is called **signing**. Certificates always form a chain up to a certain **root certificate**. In a root certificate, the subject and the issuer are one and the same entity. Such certificates are called "**self-signed**".

Signing certificates or proofing that someone is the owner of a certificate requires the possession of the corresponding **private key**. While certificates can be distributed to other parties, special care must be taken to keep the private key secret.

Each client and server may have a keystore and a truststore. A **keystore** is used to keep certificates along with the corresponding private key. This means that a keystore is used to prove your own identity or to sign certificates. A **truststore** contains public certificates of other entities.

## 8.2 Requirements for Web Service Calls from ESCM

For the provisioning of application instances and for integrating external process control or PSP systems, ESCM calls other Web services which can be addressed by HTTPS. In this scenario, ESCM is the Web service client while the other entity is the HTTPS server. The following requirements must be fulfilled to establish a connection to the server:

- The server must present a valid certificate: The CN (common name) must correspond to the server's domain name and it must be valid at the time of calling.
- The client (ESCM) must trust the server's certificate. To this end, the server's certificate must have been imported into the client's truststore, or the client's truststore must contain a root certificate with a valid signing chain to the certificate presented by the server.

## 8.3 Requirements for Web Service Calls to ESCM

ESCM provides Web services (platform services) that can be called by other systems, such as an external process control or a PSP system. In this scenario, ESCM is the HTTPS server while the other system is a Web service client. The following requirements must be met to establish a connection to ESCM:

- The ESCM server must present a valid certificate: The CN (common name) must correspond to the server's domain name and it must be valid at the time of calling.
- The clients must trust the ESCM server certificate for SSL connections: The clients' truststore must contain the ESCM server certificate.
- In case of certificate-based authentication, ESCM must trust the client's certificate: The ESCM truststore must contain a certificate with a valid signing chain to the certificate presented by the client.

## 8.4 Certificate Integration Procedures

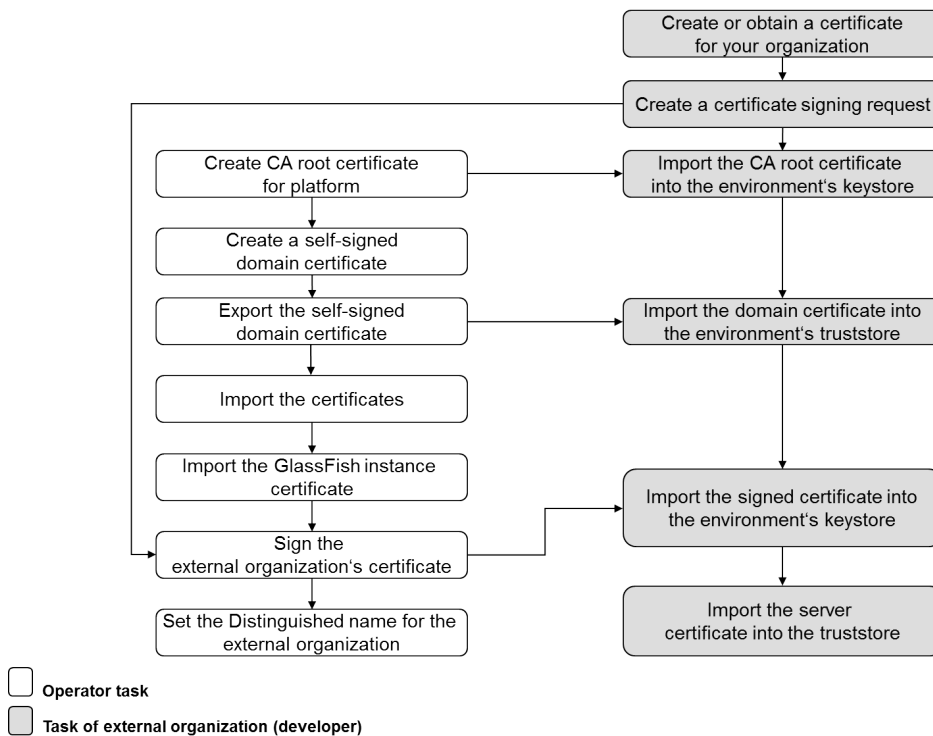
For implementing a secure ESCM environment where the nodes trust each other, the usage of custom root certificates is recommended. The basic procedure is as follows:

The ESCM operator creates the root certificate. This certificate is imported to the ESCM truststore as well as handed out to all technology providers or other organizations calling ESCM Web services.

With this type of implementation, new applications can be added as technical services without touching the ESCM truststore. You must, however, ensure the identity of each organization and manually sign the certificates.



The following figure illustrates the process and tasks involved in using custom root certificates for secure communication:



As an ESCM operator who sets up a secure ESCM environment, you need to perform the tasks depicted in white. These tasks are described in detail in the subsequent sections. The tasks of an external organization that wants to make use of certificates are described in the *Developer's Guide*.

## Prerequisites

- You have a Java environment installed of the Java version supported by ESCM. This is required for creating, exporting and importing certificates.
- A tool for signing certificates is installed in your environment, for example OpenSSL (<http://www.openssl.org/>). The samples mentioned in the next sections assume that OpenSSL is installed.
- You have moved the keystore and truststore files in the application server domain where ESCM has been deployed to a temporary directory. Check for the files  
`<GLASSFISH_HOME>/glassfish/domains/bes-domain/config/cacerts.jks` and  
`<GLASSFISH_HOME>/glassfish/domains/bes-domain/config/keystore.jks`.  
 If they are present, move them to a temporary directory.

### 8.4.1 Creating a CA Root Certificate for ESCM

The creation of the CA (Certification Authority) root certificate for ESCM involves the following steps:

1. Create a private key with the following OpenSSL command:

```
<OPENSSL-Install-Dir>/openssl genrsa -des3 -out ca.key 4096
```

A `ca.key` file is created in the directory where the command was executed. It contains the private key and is required for all subsequent certificate signing operations. 4096 is the number of bits the key will have.

**Note:** Observe applicable national regulations as to the allowed number of bits the key may have.

2. To create the CA root certificate, execute the following command:

```
<OPENSSL-Install-Dir>/openssl req -new -x509 -days 3650  
-key ca.key -out ca.crt
```

Using the private key created before, a certificate file (`ca.crt`) for the CA organization is created. The option `-days 3650` states that the created certificate will be valid for 10 years. The default (if you leave the value empty) is 1 month. The certificate file must be provided to all organizations that request their certificate to be signed.

### 8.4.2 Creating a Self-Signed Domain Certificate

To create a self-signed certificate for the application server domain where ESCM has been deployed, execute the following command of the JRE you are using:

```
<YourJRE>/bin/keytool -genkey -alias slas -keysize 1024  
-keystore keystore.jks
```

### 8.4.3 Exporting the Self-Signed Domain Certificate

The self-signed domain certificate needs to be exported and sent to the external organization that wants to make use of certificates.

Execute the following command to export the certificate:

```
<YourJRE>/bin/keytool -export -file bssDomain.crt  
-alias slas -keystore  
<GLASSFISH_HOME>/glassfish/domains/bes-domain/config/keystore.jks
```

The certificate is stored in the `bssDomain.crt` file.

### 8.4.4 Importing the Certificates

You need to import the following certificate files created in the previous steps into the truststore of your application server:

1. The ESCM domain certificate: `bssDomain.crt`
2. The CA root certificate: `ca.crt`

Execute the following commands:

```
<YourJRE>/bin/keytool -import -trustcacerts -file bssDomain.crt  
-alias slas -keystore  
<GLASSFISH_HOME>/glassfish/domains/bes-domain/config/cacerts.jks
```

```
<YourJRE>/bin/keytool -import -trustcacerts -file CA.crt  
-alias CA -keystore  
<GLASSFISH_HOME>/glassfish/domains/bes-domain/config/cacerts.jks
```

### 8.4.5 Importing the GlassFish Instance Certificate

You need to import the Glassfish instance certificate with the `glassfish-instance` alias into the keystore and truststore of your application server. This certificate is available from the directory to which you moved the keystore and truststore files before starting with the certificate integration procedure.

Execute the following commands:

```
<YourJRE>/bin/keytool -importkeystore -srckeystore  
<TEMP_DIR>/keystore.jks -srcalias glassfish-instance  
-destkeystore keystore.jks -destalias glassfish-instance
```

```
<YourJRE>/bin/keytool -importkeystore -srckeystore  
<TEMP_DIR>/cacerts.jks -srcalias glassfish-instance  
-destkeystore cacerts.jks -destalias glassfish-instance
```

After having imported the certificates, restart the application server and delete the keystore and truststore files from your temporary directory.

### 8.4.6 Signing an External Organization's Certificate

A technology provider (or any other external organization that wants to make use of certificate-based communication) has to provide the certificate for his organization, and a certificate signing request. The ESCM operator needs to sign the certificate.

To sign a certificate, use `OpenSSL`. Make sure that you have access to the private key of the ESCM CA root certificate and the certificate of the external organization (`tpcert.csr`).

Execute the following command:

```
<OPENSSL-Install-Dir>/openssl x509 -req -days 3650  
-in tpcert.csr -CA ca.crt -CAkey ca.key -set_serial 01  
-out tpcert.crt
```

The serial number (01 in the sample above) must be increased with every invocation of the command.

Proceed with sending the output file `tpcert.crt`, the CA root certificate, and the `bssDomain.crt` (domain certificate) to the organization requesting certificate-based communication.

### 8.4.7 Setting the Distinguished Name for an Organization

The logic of the ESCM call handling checks the distinguished name in a client's certificate. If it matches the one specified by the ESCM operator, the caller is assumed to be authorized to connect to ESCM.

You can retrieve the distinguished name of an organization from its certificate with the following call in your Java runtime environment:

```
<YourJRE>/bin/keytool -printcert -file <filename.crt>
```

The distinguished name is composed of the following elements:

- CN (common name)
- OU (organizational unit)
- O (organization)
- L (location)
- ST (state)
- C (country)

Sample output:

```
Owner: EMAILADDRESS=test@test.com, CN=test, OU=test, O=Test, L=Test,
ST=Test, C=GE
Issuer: EMAILADDRESS=test@test.com, CN=test, OU=test, O=Test, L=Test,
ST=Test, C=GE
Serial number: 932577b192765bf6
Valid from: Mon Jan 23 16:26:58 CET 2012 until: Thu Jan 20 16:26:58 CET
2022
Certificate fingerprints:
    MD5: 46:2E:CE:05:1A:57:B6:16:15:8A:4B:8A:CE:1D:F8:21
    SHA1:
8A:A6:B9:44:FE:E5:18:0B:92:FD:04:D4:4E:69:0F:09:FB:7B:02:34
```

**Note:** When integrating the services of commercial payment service providers (PSPs), the distinguished name must always be set. You set the distinguished name when registering the PSP with ESCM. For details, refer to *Registering a Payment Service Provider With ESCM* on page 36.

**To set the distinguished name for an organization:**

In the ESCM administration portal, choose **Manage organization** in the **Operation** menu. For detailed step-by-step instructions, refer to the online help.

## 9 Managing Marketplaces

This chapter describes

- How to create marketplaces.
- How to change the owner of a marketplace.
- How to delete a marketplace.
- How to configure marketplaces.
- How to update the search index.

For details on administrating and customizing marketplaces, refer to the *Marketplace Owner's Guide*.

Use the ESCM reporting facilities, for example, to retrieve information on the services published to your marketplace. For details, refer to *Reporting* on page 30.

### 9.1 Creating a Marketplace

You are responsible for creating marketplaces for the organizations that want to authorize suppliers, brokers, and resellers to publish their services using the facilities of ESCM. Creating a marketplace includes defining the marketplace properties and assigning an organization as the owner of the marketplace. All administrators of the assigned owner organization automatically receive the marketplace manager role. The marketplace manager role enables them to administrate and customize the marketplace.

As a prerequisite for creating a marketplace, the organization to be assigned as the marketplace owner must already exist.

**To create a marketplace:**

In the ESCM administration portal, choose **Create marketplace** in the **Marketplace** menu. For detailed step-by-step instructions, refer to the online help.

### 9.2 Changing the Owner of a Marketplace

The owner of a marketplace is responsible for administrating and customizing the marketplace to which suppliers, brokers, and resellers can publish their services. Assigning a new owner to a marketplace may remove the marketplace owner role from the previous owner organization and the marketplace manager role from its users. This is the case if the marketplace for which you change the owner is the last one owned by the organization.

**To assign another owner to a marketplace:**

In the ESCM administration portal, choose **Update marketplace** in the **Marketplace** menu. For detailed step-by-step instructions, refer to the online help.

### 9.3 Deleting a Marketplace

When deleting a marketplace with activated services, these services are automatically deactivated. Customers can no longer subscribe to them. Existing subscriptions, however, are not affected.

**To delete a marketplace:**

In the ESCM administration portal, choose **Delete marketplace** in the **Marketplace** menu. For detailed step-by-step instructions, refer to the online help.

## 9.4 Configuring Marketplaces

ESCM provides several configuration settings that influence the behavior of marketplaces. For example:

- `CUSTOMER_SELF_REGISTRATION_ENABLED`: Specifies whether customer organizations can register on a marketplace.
- `MP_ERROR_REDIRECT_HTTP` or `MP_ERROR_REDIRECT_HTTPS` : URL specifying a Web page that is to be displayed in case a visitor tries to access a marketplace without a valid marketplace ID. Note that this URL is used platform-wide. It is not tenant-specific.
- `TAGGING_MAX_TAGS`: The maximum number of tags composing the tag cloud.
- `TAGGING_MIN_SCORE`: The minimum number of times a tag must be used in services to be shown in the tag cloud.

For details on the settings, refer to *ESCM Configuration Settings* on page 48.

For details on how to change configuration settings, refer to *Updating Configuration Settings* on page 17.

## 9.5 Updating the Search Index

The search index is automatically and periodically created by the system. In some cases, for example, after a recovery of the database, the search index may become inconsistent. In order to make sure that the index becomes consistent again, you can either manually delete the index files and restart the respective application server domain, or use the **Update search index** menu option to explicitly re-create the search index at any time. The search index is re-built based on the current data in the database. The existing index structure is overwritten. The search index is re-created online, and you do not need to restart any application server domain.

The search indexer application on the `master-indexer-domain` is triggered, the old index structure is deleted, a new one is created, and the collection of the current indexing data from the `bss` database is started. The time it takes to collect the data depends on the size of the database. When the collection is finished, the `hibernate.search.default.refresh` configuration setting defined in the `glassfish.properties` file located in `<install_pack_dir>/domains/indexer_domain` is evaluated. The interval in seconds defined in this setting (default: 300 seconds) determines the additional time it will take until the new search index is available for users on marketplaces to search for subscriptions or services.

**To re-create the search index, you have the following option:**

- ESCM administration portal - Choose **Update search index** in the **Operation** menu.  
For detailed step-by-step instructions, refer to the online help.

## Appendix A: Configuration Settings

The configuration settings for ESCM are provided in the following files in subdirectories of the directory to which you extracted the `oscm-install-pack.zip` file (`<install_pack_dir>`):

- `domains/indexer_domain/glassfish.properties`
- `domains/bes_domain/glassfish.properties`
- `domains/indexer_domain/glassfishJMSBroker.properties`
- `domains/bes_domain/glassfishJMSBroker.properties`
- `databases/bes_db/db.properties`
- `databases/bes_db/configsettings.properties`
- `databases/bes_db/sso.properties`

This appendix describes the settings in detail.

### A.1 Database Configuration Settings

The `db.properties` file located in `<install_pack_dir>/databases/bes_db` contains the configuration settings for database access. This configuration is used for the initial setup and schema updates.

#### **db.driver.class**

The Java class of the JDBC driver.

Default: `org.postgresql.Driver`

#### **db.host**

The database host.

Default: `localhost`

#### **db.port**

The database port.

Default: `5432`

#### **db.name**

The name of the database.

Default: `bss`

#### **db.user**

The name of the user to connect to the database.

Default: `bssuser`

#### **db.pwd**

The password of the user to connect to the database.

Default: `bssuser`

**db.type**

The type of the database.

Default: `postgresql`

## A.2 ESCM Configuration Settings

The ESCM configuration settings can be viewed and (partially) changed in the ESCM administration portal. The default settings are specified in the `###` Docker files `###` as used during the deployment.

**AUDIT\_LOG\_ENABLED**

Optional. Specifies whether user operations related to subscriptions, marketable services, and price models are logged and stored in the database. If set to `true`, the operator can export audit log data to retrieve information on the user operations.

Allowed values: `true`, `false`

Default: `false`

**AUDIT\_LOG\_MAX\_ENTRIES\_RETRIEVED**

Optional. Specifies how many log entries are retrieved in one export of audit log data. If this number is exceeded, a warning is displayed asking the operator to change his filter criteria and start the export again. This setting is required to keep the number of SQL requests to the database low when audit log data is exported. Too many requests may lead to a decrease in system performance.

Allowed values: Any value between 1 and 1000

Default: 100

**AUTH\_MODE**

Mandatory. Specifies that ESCM is used for user authentication. This configuration setting is evaluated at the first startup of ESCM and cannot be changed.

Default: `INTERNAL`

**BASE\_URL**

Optional. The base URL is used to access the ESCM home page if ESCM does not require HTTPS for communication. If left empty, the `BASE_URL_HTTPS` setting is used.

Syntax: `http://<host>:<port>/<context-root>`

The default for `<context-root>` is `oscm-portal`.

**Note:** If the SSL/HTTPS port was changed, then this setting must also be updated.

**BASE\_URL\_HTTPS**

Mandatory. The base URL is used to access the ESCM home page and to create the URL for accessing services via HTTPS.

Syntax: `https://<host>:<port>/<context-root>`

The default for `<context-root>` is `oscm-portal`.

**Note:** If the SSL/HTTPS port was changed, then this setting must also be updated.



## CUSTOMER\_SELF\_REGISTRATION\_ENABLED

Optional. Specifies whether customer organizations can register on a marketplace. If set to `false`, the operator needs to create an organization for the customer who wants to register, or a seller (supplier, broker, reseller) needs to register the customer.

Allowed values: `true`, `false`

Default: `true`

## DECIMAL\_PLACES

Optional. Specifies the number of decimal places in which usage charges are calculated. This setting is needed only when migrating from ESCM V14.1 to a higher version.

Allowed values: 2, 3, 4, 5, 6

Default: 2

## HIDDEN\_UI\_ELEMENTS

Optional. Specifies user interface elements to be hidden from the ESCM administration portal and the marketplaces operated on your platform. You can use this setting to hide user interface elements both from the marketplaces and the administration portal.

### Marketplaces

If you want to hide a menu option from the **Account** menu of the marketplaces operated on your platform, enter one of the following values:

- `marketplace.navigation.Profile`: **Profile** menu
- `marketplace.navigation.Payment`: **Payment** menu
- `marketplace.navigation.Subscriptions`: **Subscriptions** menu
- `marketplace.navigation.Users`: **Users & Units** menu
- `marketplace.navigation.Reports`: **Reports** menu
- `marketplace.navigation.Processes`: **Processes** menu
- `marketplace.navigation.Operations`: **Operations** menu

To hide several options from the **Account** menu, separate the options by a comma.

### Administration Portal

If you want to hide a specific page from the ESCM administration portal, you can find out which value needs to be specified here as follows:

1. Open the respective page at the administration portal.
2. Display the online help for this page.
3. Have a look at the name of the online help HTML page.
4. Omit the file extension `.htm` and replace the underscore by a dot.

Example:

You want to hide the **Manage VAT rates** page. The online help HTML page name is `organization_manageVats.htm`. Thus, the respective administration portal page is `organization.manageVats`. You need to set the configuration setting as follows:

```
HIDDEN_UI_ELEMENTS=organization.manageVats
```

To hide several pages from the administration portal, separate the entries by a comma.

Below, you find some more examples of values that can be used to hide a specific page. The list is not complete.

- `organization.edit`: **Edit profile** page
- `shop.editSkin`: **Customize layout** page
- `techService.edit`: **Update service definition** page

To hide a complete menu from the administration portal, enter one of the following values:

- `navigation.myAccount`: **Account** menu
- `navigation.customer`: **Customer** menu
- `navigation.operator`: **Operation** menu
- `navigation.techService`: **Technical service** menu
- `navigation.service`: **Marketable service** menu
- `navigation.priceModel`: **Price model** menu
- `navigation.marketplace`: **Marketplace** menu

**Note:** The **Update configuration settings** page in the **Operation** menu is the default page the operator is directed to when logging in. If you hide the page from the menu or hide the complete menu, you are still directed to the **Update configuration settings** page where you can make changes, if required.

## HIDE\_PAYMENT\_INFORMATION

Optional. Determines whether customers need to specify payment information for subscribing to services that are not free of charge. If set to `true`, the dialogs and options for specifying payment types and billing addresses are hidden on the marketplaces, and customer payment information does not appear in billing and payment preview reports. In the administration portal, suppliers and resellers cannot manage payment types for their customers. Billing runs and payment processing are not affected by the setting.

The setting is evaluated only once when ESCM is installed or updated. The value can no longer be changed after ESCM has been started, not even with a further update installation.

Allowed values: `true`, `false`

Default: `false`

## HTTP\_PROXY

Optional. The proxy to be used for PSP-related HTTP connections, if any.

Example: `proxy.domain` or the proxy server IP address.

## HTTP\_PROXY\_PORT

Optional. The proxy port to be used for PSP-related HTTP connections, if any.

Allowed values: Any value between 1 and 65535

Default: 1080

## KAFKA\_BOOTSTRAP\_SERVERS

Optional. Comma-separated list of host and port pairs that are the addresses of Kafka brokers. Bootstrap servers provide the initial hosts that act as the starting point for a Kafka client. In ESCM, they are used for event-based provisioning.

Default: no value, i.e. empty

Example: `localhost:9092`

**Note:** Be aware that this setting is provided for future use and is currently only available for testing purposes.

## KEY\_FILE\_PATH

Optional. The path to the file containing the key required for encryption and decryption of service parameters with data type `PWD` and custom attributes marked for encryption.

Upon the start of ESCM, it is checked whether a key file exists in the location specified in this setting. If this is not the case, the key file is automatically generated and stored in the location specified here. If nothing is specified, the file will be named `key` and generated into the following location:

`<GLASSFISH_HOME>/glassfish/domains/bes-domain/config`

Default:

`./key`

**Note:** Be aware that the key file must not be deleted. Otherwise, encryption and decryption is no longer possible. It is recommended to create a backup of this file once generated.

## LDAP\_SEARCH\_LIMIT

Optional. The maximum number of entries that will be returned by an LDAP query in case an organization uses an external LDAP system for user management.

Allowed values: Any value between 1 and 9223372036854775807

Default: 100

## LOG\_CONFIG\_FILE

Optional. The path to the `log4j` configuration file of ESCM.

Default:

`./log4j.properties` in the `<GLASSFISH_HOME>/glassfish/domains/bes-domain/config` directory

## LOG\_FILE\_PATH

Mandatory. The path to the ESCM log files.

Default:

`./logs`, which is the `<GLASSFISH_HOME>/glassfish/domains/bes-domain/logs` directory

If you change this setting, you need to restart ESCM.

## LOG\_LEVEL

Optional. The log level for ESCM. This setting applies to all logging classes if it is not overridden by the content of the `log4j.properties` file.

Allowed values: `ERROR`, `WARN`, `INFO`, `DEBUG`

Default: `INFO`

If you change this setting, you need to restart ESCM.

**MAIL\_JA\_CHARSET**

Optional. Special character encoding for emails sent in Japanese.

Default: UTF-8

**MAX\_NUMBER\_ALLOWED\_USERS**

Mandatory. The maximum number of users that can be registered within the ESCM installation.

Allowed values: Any value between 1 and 9223372036854775807

Default: 10

**MAX\_NUMBER\_LOGIN\_ATTEMPTS**

Optional. The maximum number of allowed login attempts to ESCM. If a user does not log in successfully with this number of attempts, his account is locked.

Allowed values: Any value between 1 and 9223372036854775807

Default: 3

**MP\_ERROR\_REDIRECT\_HTTP**

Optional. The URL of a Web page that is to be displayed in case a visitor tries to access a marketplace without a valid marketplace ID by HTTP. This Web page will be shown instead of the default error message. Note that this URL is used platform-wide.

Syntax: `http://<your Web page>`

Make sure to specify a valid URL that does not exceed a maximum of 255 characters.

**MP\_ERROR\_REDIRECT\_HTTPS**

Optional. The URL of a Web page that is to be displayed in case a visitor tries to access a marketplace without a valid marketplace ID by HTTPS. This Web page will be shown instead of the default error message. Note that this URL is used platform-wide.

Syntax: `https://<your Web page>`

Make sure to specify a valid URL that does not exceed a maximum of 255 characters.

**PERMITTED\_PERIOD\_INACTIVE\_ON\_BEHALF\_USERS**

Optional. The time in milliseconds after which a user who logged in on behalf of a customer and was inactive will be removed from the system.

Allowed values: Any value between 1 and 9223372036854775807

Default: 604800000, i.e. 7 days

**PERMITTED\_PERIOD\_UNCONFIRMED\_ORGANIZATIONS**

Optional. The maximum time in milliseconds until an organization's initial administrative account must be confirmed. When this time has passed, the account is removed.

Allowed values: Any value between 1 and 9223372036854775807

Default: 604800000, i.e. 7 days

**PSP\_USAGE\_ENABLED**

Mandatory. Specifies whether PSP integration is used for the current environment.

Allowed values: `true`, `false`

Default: `false`

If you change this setting, you need to restart ESCM.

## REPORT\_ENGINEURL

Mandatory if you want to use the ESCM reports. The URL template of the report engine. If you do not specify a correct URL template, ESCM will not be able to generate any reports, since the Report Web service cannot be called correctly.

If your installation is configured to use HTTP access, the required value is:

```
http://<host IP address>:<port>/birt/frameset?
    _report=${reportname}.rptdesign&SessionId=${sessionid}
    &__locale=${locale}&WSDLURL=${wsdlurl}&SOAPEndPoint=${soapendpoint}
    &wsname=Report&wsport=ReportPort
```

If your installation is configured to use HTTPS access, the required value is:

```
https://<host IP address>:<port>/birt/frameset?
    _report=${reportname}.rptdesign&SessionId=${sessionid}
    &__locale=${locale}&WSDLURL=${wsdlurl}&SOAPEndPoint=${soapendpoint}
    &wsname=ReportSecure&wsport=ReportSecurePort
```

The `<host IP address>` and `<port>` are the IP address and port of the application server on which the BIRT report engine has been deployed.

**Note:** The above value must be used as indicated. Do not change this value.

## REPORT\_SOAP\_ENDPOINT

Mandatory if you want to use the ESCM reports. The SOAP end point of the Report Web service. All report data is retrieved via a call to the Report Web service. If you do not specify a correct value, ESCM will not be able to generate any reports, since the Report Web service cannot be called correctly.

If your installation is configured to use HTTP access, the required value is:

```
http://<host IP address>:<port>/Report/ReportingServiceBean
```

If your installation is configured to use HTTPS access, the required value is:

```
https://<host IP address>:<port>/ReportSecure/ReportingServiceSecureBean
```

**Note:** The above value must be used as indicated. Do not change this value.

The `<host IP address>` and `<port>` are the IP address and port of the application server on which ESCM has been deployed.

## REPORT\_WSDLURL

Mandatory if you want to use the ESCM reports. The URL of the WSDL file of the Report Web service. All report data is retrieved via a call to the Report Web service. If you do not specify a correct value, ESCM will not be able to generate any reports, since the Report Web service cannot be called correctly.

If your installation is configured to use HTTP access, the required value is:

```
http://<host IP address>:<port>/Report/ReportingServiceBean?wsdl
```

If your installation is configured to use HTTPS access, the required value is:

`https://<host IP address>:<port>/ReportSecure/ReportingServiceSecureBean?wsdl`

**Note:** The above value must be used as indicated. Do not change this value.

The `<host IP address>` and `<port>` are the IP address and port of the application server on which ESCM has been deployed.

## **SSO\_SIGNING\_KEY\_ALIAS**

Mandatory if requests are to be signed by ESCM. The alias of the private key of ESCM to be used for signing requests.

Requests are signed for the communication with Web pages or applications from custom tabs in subscriptions. The application server used for the Web page or Web application needs the corresponding certificate of ESCM in its truststore for verifying the signature.

Default: `slas`

If you change this setting, you need to restart ESCM.

## **SSO\_SIGNING\_KEYSTORE**

Mandatory if requests are to be signed by ESCM. The path and name of the application server's keystore where the private key of ESCM specified in the `SSO_SIGNING_KEY_ALIAS` setting is stored.

Default: `./keystore.jks`

If you change this setting, you need to restart ESCM.

## **SSO\_SIGNING\_KEYSTORE\_PASS**

Mandatory if requests are to be signed by ESCM. The password for accessing the keystore specified in the `SSO_SIGNING_KEYSTORE` setting.

Default: `changeit`

If you change this setting, you need to restart ESCM.

## **SUPPLIER\_SETS\_INVOICE\_AS\_DEFAULT**

Optional. Specifies whether invoice is to be used as the default payment type for all customers.

Allowed values: `true`, `false`

Default: `false`

## **TAGGING\_MAX\_TAGS**

Mandatory. The maximum number of tags composing the tag cloud.

The tag cloud is the area of a marketplace containing defined search terms (tags). The more often a tag is used in services, the bigger the characters of the tag are displayed. Customers can use the tags to search for services, provided that the tag cloud is enabled for the marketplace by the marketplace owner.

Allowed values: Any value between 0 and 2147483647

Default: 20

## **TAGGING\_MIN\_SCORE**

Mandatory. The minimum number of times a tag must be used in services to be shown in the tag cloud.

The tag cloud is the area of a marketplace containing defined search terms (tags). The more often a tag is used in services, the bigger the characters of the tag are displayed. Customers can use the tags to search for services, provided that the tag cloud is enabled for the marketplace by the marketplace owner.

Allowed values: Any value between 1 and 2147483647

Default: 1, i.e. a tag must have been used at least once so that it is shown in the tag cloud.

## **TIME\_ZONE\_ID**

Optional. The time zone to be used for display.

Allowed values: All time zones supported by Java. This can be an abbreviation such as `PST`, a full name such as `America/Los_Angeles`, or a custom ID such as `GMT-8:00`. For a list of IDs, refer to `java.util.TimeZone`.

Default: `GMT`

## **TIMER\_INTERVAL\_BILLING\_OFFSET**

Optional. The offset in milliseconds for the timer for billing runs calculating subscription usage costs (customer billing data) or revenue share data. The interval for this timer is one day and cannot be changed. If no offset is defined, the default offset of 4 days is applied.

Customer billing data is calculated for a period of one month (billing period). Suppliers and resellers can define individual start days for their billing periods. Revenue share data is always calculated for the past month on the first day of a month.

The offset for the billing run timer defines the following:

- Number of days after which the billing run calculating the customer billing data or the revenue share data is executed.
- Time the timer for the daily billing runs expires on the current day.

Example:

A supplier defines the 10th of a month as the billing period start date. The offset is set to 4 days and 4 hours. The billing run that calculates the customer billing data for the past billing period of this supplier is started on the 14th of the following month at 04:00:00.000. The revenue share data is calculated on the 5th of the following month at 04:00:00.000. The daily check whether a billing period of any supplier has ended is started at 04:00:00.000 every day.

Allowed values: Any value between 0 and 2419200000 (28 days)

Default: 345600000, i.e. 4 days.

## **TIMER\_INTERVAL\_DISCOUNT\_END\_NOTIFICATION\_OFFSET**

Optional. The offset in milliseconds for the timer for terminating the discounts for all organizations. The timer interval is one day and cannot be changed.

Allowed values: Any value between 0 and 9223372036854775807

Default: 0

## **TIMER\_INTERVAL\_INACTIVE\_ON\_BEHALF\_USERS**

Optional. The time interval in milliseconds at which a check for non-existing users acting on behalf of another organization is executed. A value of 0 indicates that this timer is disabled.

A technical service definition may contain a flag (`allowingOnBehalfActing`) to indicate that an organization can act in the name of another organization. The organization must be a customer

of the other organization, which must have both the technology provider and supplier role. Additionally, the customer organization must have allowed the other organization to log in on its behalf. This is achieved via a subscription whose underlying technical service has the `allowingOnBehalfActing` flag set to `true`.

When an organization acts in the name of another organization, an artificial user ID is generated. Cleaning up the ESCM database from time to time to remove such users who no longer exist might be required since it cannot be ensured that a technical service always removes such users itself.

Allowed values: 0 and any value between 10000 (10 seconds) and 9223372036854775807

Default: 0

### **TIMER\_INTERVAL\_INACTIVE\_ON\_BEHALF\_USERS\_OFFSET**

Optional. The offset in milliseconds for the timer for removing inactive "on behalf" users.

Allowed values: Any value between 0 and 9223372036854775807

Default: 0

### **TIMER\_INTERVAL\_ORGANIZATION**

Optional. The time interval in milliseconds at which tasks related to organizations are executed. A value of 0 indicates that this timer is disabled.

Allowed values: 0 and any value between 10000 (10 seconds) and 9223372036854775807

Default: 0

### **TIMER\_INTERVAL\_ORGANIZATION\_OFFSET**

Optional. The offset in milliseconds for the timer for organization-related tasks.

Allowed values: Any value between 0 and 9223372036854775807

Default: 0

### **TIMER\_INTERVAL\_SUBSCRIPTION\_EXPIRATION**

Optional. The time interval in milliseconds at which a check for expired subscriptions is executed. This timer cannot be disabled, i.e. it cannot be set to 0.

Allowed values: Any value between 10000 (10 seconds) and 9223372036854775807

Default: 86400000, i.e. 1 day

### **TIMER\_INTERVAL\_SUBSCRIPTION\_EXPIRATION\_OFFSET**

Optional. The offset in milliseconds for the timer for subscription expiration checks.

Allowed values: Any value between 0 and 9223372036854775807

Default: 0

### **TIMER\_INTERVAL\_TENANT\_PROVISIONING\_TIMEOUT**

Optional. The time interval in milliseconds at which a check for timed-out subscriptions is executed. A value of 0 indicates that this timer is disabled.

Allowed values: 0 and any value between 10000 (10 seconds) and 9223372036854775807

Default: 0



**TIMER\_INTERVAL\_TENANT\_PROVISIONING\_TIMEOUT\_OFFSET**

Optional. The offset in milliseconds for the timer for pending subscription checks.

Allowed values: Any value between 0 and 9223372036854775807

Default: 0

**TIMER\_INTERVAL\_USER\_COUNT**

Mandatory. The time interval in milliseconds at which the amount of users registered with the platform is checked. This timer cannot be disabled, i.e. it cannot be set to 0.

Allowed values: Any value between 1 and 9223372036854775807

Default: 43200000, i.e. 12 hours

**WS\_TIMEOUT**

Mandatory. The timeout for outgoing Web service calls in milliseconds. After this time has passed, a timeout exception is thrown by the JAX-WS framework.

An outgoing Web service call is a call initiated by ESCM. A typical example is the invocation of the `createUsers` method of the `ProvisioningService` interface, which is implemented by an application. If the timeout is reached before the Web service call returns, the operation is aborted and an exception is thrown.

Allowed values: Any value between 1 and 9223372036854775807

Default: 30000, i.e. 30 seconds

**Note:** Make sure that timeouts set in the GlassFish application server, e.g. request timeouts, do not conflict with or overrule the timeouts defined in the ESCM configuration settings.

## Appendix B: 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: <code>LDAP_URL=ldap://myldapserver.lan.est.company.de:389</code>
LDAP_BASE_DN	Mandatory. Position in the LDAP directory tree at which to start looking for users. Example: <code>LDAP_BASE_DN=ou=people,dc=est,dc=mycompany,dc=de</code>
LDAP_PRINCIPAL	Optional. Name of the user who is allowed to query the LDAP server. Example: <code>LDAP_PRINCIPAL=uid=admin,ou=system</code>
LDAP_CREDENTIALS	Optional. Password of the user who is allowed to query the LDAP server. Example: <code>LDAP_CREDENTIALS=secret</code>
LDAP_ATTR_UID	Mandatory. LDAP attribute from which a user ID is read. The default used when an organization is created is <code>uid</code> unless the operator has defined a different value in the platform LDAP settings. Example: <code>LDAP_ATTR_UID=uid</code>
LDAP_ATTR_EMAIL	Optional. LDAP attribute from which the email address of a user is read. Example: <code>LDAP_ATTR_EMAIL=scalixEmailAddress</code>
LDAP_ATTR_FIRST_NAME	Optional. LDAP attribute from which the first name of a user is read. Example: <code>LDAP_ATTR_FIRST_NAME=givenName</code>
LDAP_ATTR_LAST_NAME	Optional. LDAP attribute from which the last name of a user is read. Example: <code>LDAP_ATTR_LAST_NAME=sn</code>
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: <code>LDAP_ATTR_LOCALE=locale</code>

---

LDAP_CONTEXT_FACTORY	<p>Mandatory. Context factory which provides the API to query the LDAP server. The default used when an organization is created is <code>com.sun.jndi.ldap.LdapCtxFactory</code> unless the operator has defined a different value in the platform LDAP settings.</p> <p>Example:</p> <pre>LDAP_CONTEXT_FACTORY=com.sun.jndi.ldap.LdapCtxFactory</pre>
LDAP_ATTR_REFERRAL	<p>Optional. Property defining how LDAP referrals are to be processed.</p> <p>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.</p> <p>This property can take on the following values:</p> <p><code>follow</code>: Referrals are followed, i.e. users are imported from all referenced directory partitions or sub-domains.</p> <p><code>ignore</code>: Referrals are ignored (default), i.e. users are imported from the current domain directory only.</p> <p>Example: <code>LDAP_ATTR_REFERRAL=ignore</code></p>

## Appendix C: Audit Log

The operator can view and export audit log data on all kinds of user operations related to subscriptions and marketable services including their price models. This may be useful, for example, to check when and by whom a price model was changed, when subscriptions were created, when and by whom a license description was changed.

Single entries of the audit log consist of a **header** and a **message**, separated by a comma.

The header consists of the following elements, separated by a blank:

- `MM/DD/YYYY_hh:mm:ss.SSS`: The local server date and time.
- `FSP_SW/CT-MG_CTMG-BSS`: The predefined log label for log entries resulting from user operations on your platform. This label cannot be changed.
- `INFO`: The default log level. This level cannot be changed.
- `3<nnnn>`: The ID of the operation that was logged. This is a number between 30000 and 39999. The ID is unique for each operation. For example, the ID of the `Define service` operation is 30090, the ID of the `Subscribe to service` operation is 30000.
- `<operation>`: The operation that was logged, for example, `Define service`.

Following the header, the detailed log message is appended. It consists of name-value pairs with additional information, separated by vertical bars (`|`).

Every message starts with the following name-value pairs:

- `userId=<user ID>`: The ID of the user who executed the operation.
- `orgId=<organization ID>`: The ID of the organization the user belongs to.
- `orgName=<organization name>`: The name of the organization the user belongs to.
- `<additional name-value pairs>`: Additional name-value pairs specific to the logged operation, separated by vertical bars (`|`).

The log entries are categorized by the role of the user who executed an operation:

- *User Operations* on page 60
- *Administrator Operations* on page 61
- *Service Manager Operations* on page 67

The log entries show the information as stored in the database. If there are updated values, the initial values are not contained in the audit log file.

### C.1 User Operations

This section explains the messages that are output in the audit log for operations executed by standard users on a marketplace.

The following `<additional name-value pairs>` are appended to the header and the name-value pairs common to all log entries:

- `serviceId=<service ID>`: The ID of a marketable service as entered during its creation.
- `serviceName=<service name>`: The service name for customers of a marketable service as entered during its creation.
- `subscriptionName=<subscription name>`: The name of the subscription as entered when subscribing to a service.

Additional name-value pairs may be appended depending on the logged operation.

---

**Operation: Execute service operation**

Description: A user selected an operation to be executed for the service he subscribed to. The operations and their parameters are defined in the technical service underlying the marketable service.

Additional name-value pairs:

- `serviceOperation=<operation>`: The ID of the service operation executed by the user.
- `<parameter ID>=<parameter value>`: List of operation parameters, separated by vertical bars (|). For every parameter, its ID and its corresponding value are indicated.

## C.2 Administrator Operations

This section explains the messages that are output in the audit log for operations executed by an administrator of an organization with any role on a marketplace. Most of these operations can also be executed by OU administrators and subscription managers.

*<additional name-value pairs>* are appended depending on the logged operation.

---

**Operation: Assign service role**

Description: An administrator, OU administrator, or subscription manager set or changed a service role for a user assigned to a subscription.

Additional name-value pairs:

- `serviceId=<service ID>`: The ID of a marketable service as entered during its creation.
- `serviceName=<service name>`: The service name for customers of a marketable service as entered during its creation.
- `subscriptionName=<subscription name>`: The name of a subscription as entered when subscribing to a service.
- `user=<user ID>`: The ID of the user who was assigned the service role.
- `userRole=<service role>`: The name of the service role as defined in the underlying technical service.

---

**Operation: Assign user to organizational unit**

Description: An administrator or OU administrator added one or more users to an organizational unit.

Additional name-value pairs:

- `user=<user ID>`: The IDs of the users who were assigned to the organizational unit, separated by commas.
- `organizationalUnit=<organizational unit name>`: The name of the organizational unit to which the users were added.

---

**Operation: Assign user to subscription**

Description: An administrator, OU administrator, or subscription manager assigned a user to a subscription.

Additional name-value pairs:

- `serviceId=<service ID>`: The ID of a marketable service as entered during its creation.
- `serviceName=<service name>`: The service name for customers of a marketable service as entered during its creation.
- `subscriptionName=<subscription name>`: The name of a subscription as entered when subscribing to a service.
- `user=<user ID>`: The ID of the user who was assigned to the subscription.

---

**Operation: Deassign service role**

Description: An administrator, OU administrator, or subscription manager removed a service role from a user assigned to a subscription.

Additional name-value pairs:

- `serviceId=<service ID>`: The ID of a marketable service as entered during its creation.
- `serviceName=<service name>`: The service name for customers of a marketable service as entered during its creation.
- `subscriptionName=<subscription name>`: The name of a subscription as entered when subscribing to a service.
- `user=<user ID>`: The ID of the user who was deassigned the service role.
- `userRole=<service role>`: The name of the service role as defined in the underlying technical service.

---

**Operation: Deassign user from subscription**

Description: An administrator, OU administrator, or subscription manager removed the assignment of a user to a subscription.

Additional name-value pairs:

- `serviceId=<service ID>`: The ID of a marketable service as entered during its creation.
- `serviceName=<service name>`: The service name for customers of a marketable service as entered during its creation.
- `subscriptionName=<subscription name>`: The name of a subscription as entered when subscribing to a service.
- `user=<user ID>`: The ID of the user who was deassigned from the subscription.

---

**Operation: Disable access to services**

Description: An administrator or OU administrator specified the services that can no longer be accessed by the members of an organizational unit. For every service, a separate log entry is written.

Additional name-value pairs:

- `organizationalUnit=<organizational unit name>`: The name of the organizational unit whose members cannot access the service.
- `marketplaceId=<marketplace ID>`: The ID of the marketplace on which the members of the organizational unit were able to access the service.
- `marketplaceName=<marketplace name>`: The name of the marketplace on which the members of the organizational unit were able to access the service.
- `serviceID=<service ID>`: The ID of a marketable service as entered during its creation.
- `serviceName=<service name>`: The service name for customers of a marketable service as entered during its creation.
- `sellerID=<seller ID>`: The organization ID of the supplier, broker, or reseller who offers the service.

---

**Operation: Edit customer attribute by customer**

Description: An administrator or subscription manager defined or changed an attribute value for a customer attribute. For every attribute value, a separate log entry is written.

Additional name-value pairs:

- `customerKey=<organization ID>`: The ID of the customer organization.
- `customerName=<organization name>`: The name of the customer organization.
- `attributeName=<attribute name>`: The name of the customer attribute.
- `attributeValue=<attribute value>`: The value of the customer attribute.

---

**Operation: Edit subscription attribute by customer**

Description: An administrator, OU administrator, or subscription manager defined or changed an attribute value for a custom subscription attribute. For every attribute value, a separate log entry is written.

Additional name-value pairs:

- `serviceId=<service ID>`: The ID of a marketable service as entered during its creation.
  - `serviceName=<service name>`: The service name for customers of a marketable service as entered during its creation.
  - `subscriptionName=<subscription name>`: The name of a subscription as entered when subscribing to a service.
  - `attributeName=<attribute name>`: The name of the custom attribute.
  - `attributeValue=<attribute value>`: The value of the custom attribute.
-

---

**Operation: Edit subscription owner**

Description: An administrator or OU administrator set or changed the owner of a subscription.

Additional name-value pairs:

- `serviceId=<service ID>`: The ID of a marketable service as entered during its creation.
- `serviceName=<service name>`: The service name for customers of a marketable service as entered during its creation.
- `subscriptionName=<subscription name>`: The name of a subscription as entered when subscribing to a service.
- `subscriptionOwner=<user ID>`: The ID of the administrator, OU administrator, or subscription manager who was set as the new owner of the subscription.

---

**Operation: Edit subscription parameter configuration**

Description: An administrator, OU administrator, or subscription manager changed the parameter options for a subscription. For every parameter option, a separate log entry is written.

Additional name-value pairs:

- `serviceId=<service ID>`: The ID of a marketable service as entered during its creation.
- `serviceName=<service name>`: The service name for customers of a marketable service as entered during its creation.
- `subscriptionName=<subscription name>`: The name of a subscription as entered when subscribing to a service.
- `parameterName=<option name>`: The name of the parameter option.
- `parameterValue=<option value>`: The new option value. For boolean parameter options, `ON` or `OFF` (`ON`: the option has been selected, `OFF`: the option has not been selected). For enumerations, the value set for the parameter option.

---

**Operation: Enable access to services**

Description: An administrator or OU administrator specified the services that can be accessed by the members of an organizational unit. For every service, a separate log entry is written.

Additional name-value pairs:

- `organizationalUnit=<organizational unit name>`: The name of the organizational unit whose members can access the service.
  - `marketplaceId=<marketplace ID>`: The ID of the marketplace on which the members of the organizational unit can access the service.
  - `marketplaceName=<marketplace name>`: The name of the marketplace on which the members of the organizational unit can access the service.
  - `serviceID=<service ID>`: The ID of a marketable service as entered during its creation.
  - `serviceName=<service name>`: The service name for customers of a marketable service as entered during its creation.
  - `sellerID=<seller ID>`: The organization ID of the supplier, broker, or reseller who offers the service.
-



---

**Operation: Remove user from organizational unit**

Description: An administrator or OU administrator removed one or more users from an organizational unit.

Additional name-value pairs:

- `user=<user IDs>`: The IDs of the users who were removed from the organizational unit, separated by commas.
- `organizationalUnit=<organizational unit name>`: The name of the organizational unit from which the users were removed.

---

**Operation: Report issue**

Description: An administrator, OU administrator, or subscription manager reported an issue on a subscription to the responsible supplier or reseller.

Additional name-value pairs:

- `serviceId=<service ID>`: The ID of a marketable service as entered during its creation.
- `serviceName=<service name>`: The service name for customers of a marketable service as entered during its creation.
- `subscriptionName=<subscription name>`: The name of a subscription as entered when subscribing to a service.
- `issue=<text>`: The subject of the issue as entered by the administrator, OU administrator, or subscription manager.

---

**Operation: Set subscription billing address**

Description: An administrator, OU administrator, or subscription manager set or changed the billing address for a subscription.

Additional name-value pairs:

- `serviceId=<service ID>`: The ID of a marketable service as entered during its creation.
- `serviceName=<service name>`: The service name for customers of a marketable service as entered during its creation.
- `subscriptionName=<subscription name>`: The name of a subscription as entered when subscribing to a service.
- `address=<display name>`: The name of the billing address as shown to the user.
- `addressDetails=<address details>`: The name of the organization, email, and postal address to which invoices are to be sent.

---

**Operation: Set subscription payment type**

Description: An administrator, OU administrator, or subscription manager set or changed the payment type for a subscription.

Additional name-value pairs:

- `serviceId=<service ID>`: The ID of a marketable service as entered during its creation.
- `serviceName=<service name>`: The service name for customers of a marketable service as entered during its creation.
- `subscriptionName=<subscription name>`: The name of a subscription as entered when subscribing to a service.
- `paymentName=<display name>`: The name of the payment type as shown to the user.
- `paymentType=<payment type>`: The payment type name as offered by the supplier or PSP.

---

**Operation: Subscribe to service**

Description: An administrator, OU administrator, or subscription manager completed subscribing to a marketable service offered on a marketplace.

Additional name-value pairs:

- `serviceId=<service ID>`: The ID of a marketable service as entered during its creation.
- `serviceName=<service name>`: The service name for customers of a marketable service as entered during its creation.
- `subscriptionName=<subscription name>`: The name of a subscription as entered when subscribing to a service.

---

**Operation: Unsubscribe from service**

Description: An administrator, OU administrator, or subscription manager terminated a subscription.

Additional name-value pairs:

- `serviceId=<service ID>`: The ID of a marketable service as entered during its creation.
- `serviceName=<service name>`: The service name for customers of a marketable service as entered during its creation.
- `subscriptionName=<subscription name>`: The name of a subscription as entered when subscribing to a service.

---

**Operation: Up/downgrade subscription**

Description: An administrator, OU administrator, or subscription manager upgraded or downgraded a subscription.

Additional name-value pairs:

- `serviceId=<service ID>`: The ID of a marketable service as entered during its creation.
- `serviceName=<service name>`: The service name for customers of a marketable service as entered during its creation.
- `subscriptionName=<subscription name>`: The name of a subscription as entered when subscribing to a service.
- `newServiceId=<service ID>`: The ID of the marketable service to which the subscription was upgraded or downgraded.
- `newServiceName=<service name>`: The service name for customers of the service to which the subscription was upgraded or downgraded.

## C.3 Service Manager Operations

This section explains the messages that are output in the audit log for operations that can be executed in the administration portal by a service manager of an organization with the supplier role. Some of these operations can also be executed by brokers or resellers.

The following *<additional name-value pairs>* are appended to the header and the name-value pairs common to all log entries:

- `serviceId=<service ID>`: The ID of a marketable service as entered during its creation.
- `serviceName=<service name>`: The service name for customers of a marketable service as entered during its creation.

Additional name-value pairs may be appended depending on the logged operation.

---

**Operation: Activate/deactivate service**

Description: A service manager, broker, or reseller activated or deactivated a marketable service.

Additional name-value pairs:

- `marketplaceId=<ID>`: The ID of the marketplace to which the service is published.
- `marketplaceName=<name>`: The name of the marketplace to which the service is published.
- `activation=<on or off>`: Specifies whether the service was activated or deactivated. Can be ON or OFF.
- `inCatalog=<on or off>`: Specifies whether the service is to be displayed in the service catalog. Can be ON or OFF.

---

**Operation: Assign brokers**

Description: In the publishing options, a service manager authorized a broker to sell the supplier's services.

Additional name-value pair:

`BrokerId=<ID>`: The organization ID of the assigned broker.

---

**Operation: Assign categories**

Description: In the publishing options, a service manager, broker, or reseller assigned or deassigned one or several categories to/from a marketable service.

Additional name-value pair:

`listOfCategories=<names>`: The categories that are assigned, separated by commas.

---

**Operation: Assign resellers**

Description: In the publishing options, a service manager authorized a reseller to sell the supplier's services.

Additional name-value pair:

`ResellerId=<ID>`: The organization ID of the assigned reseller.

---

**Operation: Assign service to marketplace**

Description: In the publishing options, a service manager, broker, or reseller specified or changed the marketplace to which a marketable service is to be published.

Additional name-value pairs:

- `marketplaceId=<ID>`: The ID of the marketplace.
  - `marketplaceName=<name>`: The name of the marketplace.
- 

**Operation: Copy service**

Description: A service manager created a copy of a marketable service.

Additional name-value pairs:

- `copyId=<service ID>`: The ID of the marketable service copy as stored in the database.
  - `copyName=<service name>`: The name of the marketable service copy as entered when the copy was created.
- 

**Operation: Deassign brokers**

Description: In the publishing options, a service manager removed a broker's right to sell the supplier's services.

Additional name-value pair:

`BrokerId=<ID>`: The organization ID of the deassigned broker.

---

**Operation: Deassign resellers**

Description: In the publishing options, a service manager removed a reseller's right to sell the supplier's services.

Additional name-value pair:

`ResellerId=<ID>`: The organization ID of the deassigned reseller.

---

---

**Operation: Define service**

Description: A service manager defined a marketable service. One additional **Update service parameter** log entry is created for each parameter and parameter option defined in the service.

Additional name-value pairs:

- `technServiceName=<name>`: The name of the technical service the marketable service is based on.
- `shortDescription=<yes or no>`: Specifies whether the short description of the marketable service was entered. Can be YES or NO.
- `description=<yes or no>`: Specifies whether the description of the marketable service was entered. Can be YES or NO.
- `locale=<language code>`: The code of the language in which the descriptions were saved.
- `autoAssignUser=<yes or no>`: Specifies whether the user subscribing to the service is automatically assigned to the subscription. Can be YES or NO.

---

**Operation: Define up/downgrade options**

Description: A service manager defined options to which a subscription can be upgraded or downgraded, or he removed a service from the list of up/downgrade options.

Additional name-value pairs:

- `targetId=<service ID>`: The ID of the marketable service to which subscriptions can be up/downgraded as stored in the database.
- `targetName=<service name>`: The name of the marketable service to which subscriptions can be up/downgraded.
- `upDownGrade=<on or off>`: Specifies whether the service was added or removed as an up/downgrade option. Can be ON or OFF.

---

**Operation: Delete customer price model**

Description: A service manager deleted a customer-specific price model.

Additional name-value pairs:

- `customerKey=<organization ID>`: The ID of the customer organization.
- `customerName=<organization name>`: The name of the customer organization.

---

**Operation: Delete service**

Description: A service manager deleted a marketable service.

No additional name-value pairs.

---

**Operation: Edit event price in customer price model**

Description: A service manager defined or changed the price for an event in a customer-specific price model.

Additional name-value pairs:

- `customerKey=<organization ID>`: The ID of the customer organization.
- `customerName=<organization name>`: The name of the customer organization.
- `currency=<currency code>`: The ISO code of the currency in which a customer is charged for using the service.
- `eventName=<name>`: The name of the event as defined in the underlying technical service.
- `range=<value>`: If stepped prices are defined, this is the step limit up to which the price applies. If no stepped prices are defined, the value is 1-ANY.
- `action=<action>`: If stepped prices are defined, action that was executed. Can be `INSERT` (when a new stepped price was defined), `UPDATE` (when a stepped price was updated), or `DELETE` (when a stepped price was deleted). If no stepped prices are defined, the `action` parameter is not written to the audit log.
- `price=<price>`: The price as defined in the price model.

---

**Operation: Edit event price in service price model**

Description: A service manager defined or changed the price for an event in a service price model.

Additional name-value pairs:

- `currency=<currency code>`: The ISO code of the currency in which a customer is charged for using the service.
- `eventName=<name>`: The name of the event as defined in the underlying technical service.
- `range=<value>`: If stepped prices are defined, this is the step limit up to which the price applies. If no stepped prices are defined, the value is 1-ANY.
- `action=<action>`: If stepped prices are defined, action that was executed. Can be `INSERT` (when a new stepped price was defined), `UPDATE` (when a stepped price was updated), or `DELETE` (when a stepped price was deleted). If no stepped prices are defined, the `action` parameter is not written to the audit log.
- `price=<price>`: The price as defined in the price model.

---

**Operation: Edit event price in subscription price model**

Description: A service manager defined or changed the price for an event in a subscription-specific price model.

Additional name-value pairs:

- `customerKey=<organization ID>`: The ID of the customer organization.
- `customerName=<organization name>`: The name of the customer organization.
- `subscriptionName=<subscription name>`: The name of the subscription as entered when subscribing to a service.
- `currency=<currency code>`: The ISO code of the currency in which a customer is charged for using the service.
- `eventName=<name>`: The name of the event as defined in the underlying technical service.
- `range=<value>`: If stepped prices are defined, this is the step limit up to which the price applies. If no stepped prices are defined, the value is 1-ANY.
- `action=<action>`: If stepped prices are defined, action that was executed. Can be `INSERT` (when a new stepped price was defined), `UPDATE` (when a stepped price was updated), or `DELETE` (when a stepped price was deleted. If no stepped prices are defined, the `action` parameter is not written to the audit log.
- `price=<price>`: The price as defined in the price model.

---

**Operation: Edit one time fee in customer price model**

Description: A service manager defined or changed a one-time fee in a customer-specific price model.

Additional name-value pairs:

- `customerKey=<organization ID>`: The ID of the customer organization.
- `customerName=<organization name>`: The name of the customer organization.
- `currency=<currency code>`: The ISO code of the currency in which a customer is charged for using the service.
- `timeUnit=<time unit>`: The time unit defined for recurring charges. Can be `MONTH`, `WEEK`, `DAY`, or `HOURL`.
- `oneTimeFee=<fee>`: The one-time fee for a subscription.

---

**Operation: Edit one time fee in service price model**

Description: A service manager defined or changed a one-time fee in a service price model.

Additional name-value pairs:

- `currency=<currency code>`: The ISO code of the currency in which a customer is charged for using the service.
  - `timeUnit=<time unit>`: The time unit defined for recurring charges. Can be `MONTH`, `WEEK`, `DAY`, or `HOURL`.
  - `oneTimeFee=<fee>`: The one-time fee for a subscription.
-

---

**Operation: Edit customer price model**

Description: A service manager defined or changed the currency, time unit, calculation mode, and/or a free trial period of a customer-specific price model.

Additional name-value pairs:

- `customerKey=<organization ID>`: The ID of the customer organization.
  - `customerName=<organization name>`: The name of the customer organization.
  - `currency=<currency code>`: The ISO code of the currency in which a customer is charged for using the service.
  - `timeUnit=<time unit>`: The time unit for recurring charges and per time unit calculation. Can be MONTH, WEEK, DAY, or HOUR.
  - `calculationMode=<calculation option>`: The way charges for using a service are calculated. Can be PRO\_RATA or PER\_UNIT.
  - `trialPeriod=<on or off>`: Specifies whether a free trial period has been defined in the price model. Can be ON or OFF.
  - `daysOfTrial=<Number of days>`: The number of days defined for the free trial period.
- 

**Operation: Edit service price model**

Description: A service manager defined or changed the currency, time unit, calculation mode and/or a free trial period of a service price model.

Additional name-value pairs:

- `currency=<currency code>`: The ISO code of the currency in which a customer is charged for using the service.
  - `timeUnit=<time unit>`: The time unit for recurring charges and per time unit calculation. Can be MONTH, WEEK, DAY, or HOUR.
  - `calculationMode=<calculation option>`: The way charges for using a service are calculated. Can be PRO\_RATA or PER\_UNIT.
  - `trialPeriod=<on or off>`: Specifies whether a free trial period has been defined in the price model. Can be ON or OFF.
  - `daysOfTrial=<Number of days>`: The number of days defined for the free trial period.
- 

**Operation: Edit price model type to free of charge for customer**

Description: A service manager defined or changed a free-of-charge, customer-specific price model so that no costs are charged for using the service.

Additional name-value pairs:

- `customerKey=<organization ID>`: The ID of the customer organization.
  - `customerName=<organization name>`: The name of the customer organization.
- 

**Operation: Edit price model type to free of charge for service**

Description: A service manager defined or changed a free-of-charge service price model so that no costs are charged for using the service.

No additional name-value pairs.



---

**Operation: Edit price per subscription parameter in customer price model**

Description: A service manager defined or changed the price per subscription for a parameter or a parameter option in a customer-specific price model.

Additional name-value pairs:

- `customerKey=<organization ID>`: The ID of the customer organization.
- `customerName=<organization name>`: The name of the customer organization.
- `currency=<currency code>`: The ISO code of the currency in which a customer is charged for using the service.
- `parameterName=<name>`: The name of the parameter as defined in the underlying technical service.
- `range=<value>`: For numeric parameters, stepped prices can be defined. If this is the case, this is the step limit up to which the price applies. If no stepped prices are defined, the value is 1-ANY.
- `optionName=<name>`: The name of the parameter option as defined in the underlying technical service.
- `action=<action>`: If stepped prices are defined, action that was executed. Can be `INSERT` (when a new stepped price was defined), `UPDATE` (when a stepped price was updated), or `DELETE` (when a stepped price was deleted. If no stepped prices are defined, the `action` parameter is not written to the audit log.
- `price=<price>`: The price as defined in the price model.

---

**Operation: Edit price per subscription parameter in service price model**

Description: A service manager defined or changed the price per subscription for a parameter or a parameter option in a service price model.

Additional name-value pairs:

- `currency=<currency code>`: The ISO code of the currency in which a customer is charged for using the service.
  - `parameterName=<name>`: The name of the parameter as defined in the underlying technical service.
  - `range=<value>`: For numeric parameters, stepped prices can be defined. If this is the case, this is the step limit up to which the price applies. If no stepped prices are defined, the value is 1-ANY.
  - `optionName=<name>`: The name of the parameter option as defined in the underlying technical service.
  - `action=<action>`: If stepped prices are defined, action that was executed. Can be `INSERT` (when a new stepped price was defined), `UPDATE` (when a stepped price was updated), or `DELETE` (when a stepped price was deleted. If no stepped prices are defined, the `action` parameter is not written to the audit log.
  - `price=<price>`: The price as defined in the price model.
-

---

**Operation: Edit price per subscription parameter in subscription price model**

Description: A service manager defined or changed the price per subscription for a parameter or a parameter option in a subscription-specific price model.

Additional name-value pairs:

- `customerKey=<organization ID>`: The ID of the customer organization.
- `customerName=<organization name>`: The name of the customer organization.
- `subscriptionName=<subscription name>`: The name of the subscription as entered when subscribing to a service.
- `currency=<currency code>`: The ISO code of the currency in which a customer is charged for using the service.
- `parameterName=<name>`: The name of the parameter as defined in the underlying technical service.
- `range=<value>`: For numeric parameters, stepped prices can be defined. If this is the case, this is the step limit up to which the price applies. If no stepped prices are defined, the value is 1-ANY.
- `optionName=<name>`: The name of the parameter option as defined in the underlying technical service.
- `action=<action>`: If stepped prices are defined, action that was executed. Can be `INSERT` (when a new stepped price was defined), `UPDATE` (when a stepped price was updated), or `DELETE` (when a stepped price was deleted). If no stepped prices are defined, the `action` parameter is not written to the audit log.
- `price=<price>`: The price as defined in the price model.

---

**Operation: Edit price per user parameter in customer price model**

Description: A service manager defined or changed the price per user for a parameter or a parameter option in a customer-specific price model.

Additional name-value pairs:

- `customerKey=<organization ID>`: The ID of the customer organization.
- `customerName=<organization name>`: The name of the customer organization.
- `currency=<currency code>`: The ISO code of the currency in which a customer is charged for using the service.
- `parameterName=<name>`: The name of the parameter as defined in the underlying technical service.
- `optionName=<name>`: The name of the parameter option as defined in the underlying technical service.
- `price=<price>`: The price as defined in the price model.

---

**Operation: Edit price per user parameter in service price model**

Description: A service manager defined or changed the price per user for a parameter or a parameter option in a service price model.

Additional name-value pairs:

- `currency=<currency code>`: The ISO code of the currency in which a customer is charged for using the service.
- `parameterName=<name>`: The name of the parameter as defined in the underlying technical service.
- `optionName=<name>`: The name of the parameter option as defined in the underlying technical service.
- `price=<price>`: The price as defined in the price model.

---

**Operation: Edit price per user parameter in subscription price model**

Description: A service manager defined or changed the price per user for a parameter or a parameter option in a subscription-specific price model.

Additional name-value pairs:

- `customerKey=<organization ID>`: The ID of the customer organization.
- `customerName=<organization name>`: The name of the customer organization.
- `subscriptionName=<subscription name>`: The name of the subscription as entered when subscribing to a service.
- `currency=<currency code>`: The ISO code of the currency in which a customer is charged for using the service.
- `parameterName=<name>`: The name of the parameter as defined in the underlying technical service.
- `optionName=<name>`: The name of the parameter option as defined in the underlying technical service.
- `price=<price>`: The price as defined in the price model.

---

**Operation: Edit price per user role parameter in customer price model**

Description: A service manager defined or changed the price per user with a specific service role for a parameter or a parameter option in a customer-specific price model.

Additional name-value pairs:

- `customerKey=<organization ID>`: The ID of the customer organization.
- `customerName=<organization name>`: The name of the customer organization.
- `currency=<currency code>`: The ISO code of the currency in which a customer is charged for using the service.
- `timeUnit=<time unit>`: The time unit defined for recurring charges. Can be MONTH, WEEK, DAY, or HOUR.
- `userRole=<role name>`: The name of the service role as defined in the underlying technical service.
- `parameterName=<name>`: The name of the parameter as defined in the underlying technical service.
- `optionName=<name>`: The name of the parameter option as defined in the underlying technical service.
- `price=<price>`: The price as defined in the price model.

---

**Operation: Edit price per user role parameter in service price model**

Description: A service manager defined or changed the price per user with a specific service role for a parameter or a parameter option in a service price model.

Additional name-value pairs:

- `currency=<currency code>`: The ISO code of the currency in which a customer is charged for using the service.
- `timeUnit=<time unit>`: The time unit defined for recurring charges. Can be MONTH, WEEK, DAY, or HOUR.
- `userRole=<role name>`: The name of the service role as defined in the underlying technical service.
- `parameterName=<name>`: The name of the parameter as defined in the underlying technical service.
- `optionName=<name>`: The name of the parameter option as defined in the underlying technical service.
- `price=<price>`: The price as defined in the price model.

---

**Operation: Edit price per user role parameter in subscription price model**

Description: A service manager defined or changed the price per user with a specific service role for a parameter or a parameter option in a subscription-specific price model.

Additional name-value pairs:

- `customerKey=<organization ID>`: The ID of the customer organization.
- `customerName=<organization name>`: The name of the customer organization.
- `subscriptionName=<subscription name>`: The name of the subscription as entered when subscribing to a service.
- `currency=<currency code>`: The ISO code of the currency in which a customer is charged for using the service.
- `timeUnit=<time unit>`: The time unit defined for recurring charges. Can be MONTH, WEEK, DAY, or HOUR.
- `userRole=<role name>`: The name of the service role as defined in the underlying technical service.
- `parameterName=<name>`: The name of the parameter as defined in the underlying technical service.
- `optionName=<name>`: The name of the parameter option as defined in the underlying technical service.
- `price=<price>`: The price as defined in the price model.

---

**Operation: Edit recurring charge for subscription in customer price model**

Description: A service manager defined or changed the recurring charge per subscription in a customer-specific price model.

Additional name-value pairs:

- `customerKey=<organization ID>`: The ID of the customer organization.
- `customerName=<organization name>`: The name of the customer organization.
- `currency=<currency code>`: The ISO code of the currency in which a customer is charged for using the service.
- `timeUnit=<time unit>`: The time unit defined for recurring charges. Can be MONTH, WEEK, DAY, or HOUR.
- `oneTimeFee=<fee>`: The one-time fee for a subscription.
- `recurringCharge=<charge>`: The recurring charge per subscription.

---

**Operation: Edit recurring charge for subscription in service price model**

Description: A service manager defined or changed the recurring charge per subscription in a service price model.

Additional name-value pairs:

- `currency=<currency code>`: The ISO code of the currency in which a customer is charged for using the service.
  - `timeUnit=<time unit>`: The time unit defined for recurring charges. Can be MONTH, WEEK, DAY, or HOUR.
  - `oneTimeFee=<fee>`: The one-time fee for a subscription.
  - `recurringCharge=<charge>`: The recurring charge per subscription.
-

---

**Operation: Edit recurring charge for subscription in subscription price model**

Description: A service manager defined or changed the recurring charge per subscription in a subscription-specific price model.

Additional name-value pairs:

- `customerKey=<organization ID>`: The ID of the customer organization.
- `customerName=<organization name>`: The name of the customer organization.
- `subscriptionName=<subscription name>`: The name of the subscription as entered when subscribing to a service.
- `currency=<currency code>`: The ISO code of the currency in which a customer is charged for using the service.
- `timeUnit=<time unit>`: The time unit defined for recurring charges. Can be MONTH, WEEK, DAY, or HOUR.
- `oneTimeFee=<fee>`: The one-time fee for a subscription.
- `recurringCharge=<charge>`: The recurring charge per subscription.

---

**Operation: Edit recurring charge for users in customer price model**

Description: A service manager defined or changed the recurring charge for users in a customer-specific price model.

Additional name-value pairs:

- `customerKey=<organization ID>`: The ID of the customer organization.
- `customerName=<organization name>`: The name of the customer organization.
- `currency=<currency code>`: The ISO code of the currency in which a customer is charged for using the service.
- `timeUnit=<time unit>`: The time unit defined for recurring charges. Can be MONTH, WEEK, DAY, or HOUR.
- `range=<value>`: If stepped prices are defined, this is the step limit up to which the price applies. If no stepped prices are defined, the value is 1-ANY.
- `action=<action>`: If stepped prices are defined, action that was executed. Can be INSERT (when a new stepped price was defined), UPDATE (when a stepped price was updated), or DELETE (when a stepped price was deleted. If no stepped prices are defined, the `action` parameter is not written to the audit log.
- `recurringCharge=<charge>`: The recurring charge for users.

---

**Operation: Edit recurring charge for users in service price model**

Description: A service manager defined or changed the recurring charge for users in a service price model.

Additional name-value pairs:

- `currency=<currency code>`: The ISO code of the currency in which a customer is charged for using the service.
- `timeUnit=<time unit>`: The time unit defined for recurring charges. Can be MONTH, WEEK, DAY, or HOUR.
- `range=<value>`: If stepped prices are defined, this is the step limit up to which the price applies. If no stepped prices are defined, the value is 1-ANY.
- `action=<action>`: If stepped prices are defined, action that was executed. Can be INSERT (when a new stepped price was defined), UPDATE (when a stepped price was updated), or DELETE (when a stepped price was deleted. If no stepped prices are defined, the `action` parameter is not written to the audit log.
- `recurringCharge=<charge>`: The recurring charge for users.

---

**Operation: Edit recurring charge for users in subscription price model**

Description: A service manager defined or changed the recurring charge for users in a subscription-specific price model.

Additional name-value pairs:

- `customerKey=<organization ID>`: The ID of the customer organization.
- `customerName=<organization name>`: The name of the customer organization.
- `subscriptionName=<subscription name>`: The name of the subscription as entered when subscribing to a service.
- `currency=<currency code>`: The ISO code of the currency in which a customer is charged for using the service.
- `timeUnit=<time unit>`: The time unit defined for recurring charges. Can be MONTH, WEEK, DAY, or HOUR.
- `range=<value>`: If stepped prices are defined, this is the step limit up to which the price applies. If no stepped prices are defined, the value is 1-ANY.
- `action=<action>`: If stepped prices are defined, action that was executed. Can be INSERT (when a new stepped price was defined), UPDATE (when a stepped price was updated), or DELETE (when a stepped price was deleted. If no stepped prices are defined, the `action` parameter is not written to the audit log.
- `recurringCharge=<charge>`: The recurring charge for users.

---

**Operation: Edit service role prices for customer price model**

Description: A service manager defined or changed the recurring charge for users with a given service role in a customer-specific price model.

Additional name-value pairs:

- `customerKey=<organization ID>`: The ID of the customer organization.
- `customerName=<organization name>`: The name of the customer organization.
- `currency=<currency code>`: The ISO code of the currency in which a customer is charged for using the service.
- `timeUnit=<time unit>`: The time unit defined for recurring charges. Can be MONTH, WEEK, DAY, or HOUR.
- `userRole=<role name>`: The name of the service role as defined in the underlying technical service.
- `price=<price>`: The recurring charge for users having the selected role.

---

**Operation: Edit service role prices for service price model**

Description: A service manager defined or changed the recurring charge for users with a given service role in a service price model.

Additional name-value pairs:

- `currency=<currency code>`: The ISO code of the currency in which a customer is charged for using the service.
- `timeUnit=<time unit>`: The time unit defined for recurring charges. Can be MONTH, WEEK, DAY, or HOUR.
- `userRole=<role name>`: The name of the service role as defined in the underlying technical service.
- `price=<price>`: The recurring charge for users having the selected role.

---

**Operation: Edit service role prices for subscription price model**

Description: A service manager defined or changed the recurring charge for users with a given service role in a subscription-specific price model.

Additional name-value pairs:

- `customerKey=<organization ID>`: The ID of the customer organization.
  - `customerName=<organization name>`: The name of the customer organization.
  - `subscriptionName=<subscription name>`: The name of the subscription as entered when subscribing to a service.
  - `currency=<currency code>`: The ISO code of the currency in which a customer is charged for using the service.
  - `timeUnit=<time unit>`: The time unit defined for recurring charges. Can be MONTH, WEEK, DAY, or HOUR.
  - `userRole=<role name>`: The name of the service role as defined in the underlying technical service.
  - `price=<price>`: The recurring charge for users having the selected role.
-



---

**Operation: Edit subscription attribute by service manager**

Description: A service manager defined or changed an attribute value for a custom subscription attribute. For every attribute value, a separate log entry is written.

Additional name-value pairs:

- `subscriptionName=<subscription name>`: The name of the subscription as entered when subscribing to a service.
- `attributeName=<attribute name>`: The name of the custom attribute.
- `attributeValue=<attribute value>`: The value of the custom attribute.

---

**Operation: Localize price model for customer**

Description: A service manager translated the price model elements for a customer into a given language.

Additional name-value pairs:

- `customerKey=<ID>`: The ID of the customer organization.
- `customerName=<name>`: The name of the customer organization.
- `locale=<language code>`: The code of the language in which the texts were saved.
- `description=<yes or no>`: Specifies whether the description of the price model was changed. Can be YES or NO.
- `license=<yes or no>`: Specifies whether the license information was changed. Can be YES or NO.

---

**Operation: Localize price model for service**

Description: A service manager translated the price model elements for a service into a given language.

Additional name-value pairs:

- `locale=<language code>`: The code of the language in which the texts were saved.
- `description=<yes or no>`: Specifies whether the description of the price model was changed. Can be YES or NO.
- `license=<yes or no>`: Specifies whether the license information was changed. Can be YES or NO.

---

**Operation: Localize price model for subscription**

Description: A service manager translated the price model elements for a subscription into a given language.

Additional name-value pairs:

- `customerKey=<ID>`: The ID of the customer organization.
- `customerName=<name>`: The name of the customer organization.
- `subscriptionName=<subscription name>`: The name of the subscription as entered when subscribing to a service.
- `locale=<language code>`: The code of the language in which the texts were saved.
- `description=<yes or no>`: Specifies whether the description of the price model was changed. Can be YES or NO.
- `license=<yes or no>`: Specifies whether the license information was changed. Can be YES or NO.

---

**Operation: Localize service**

Description: A service manager added a service description or short description in another language.

Additional name-value pairs:

- `shortDescription=<yes or no>`: Specifies whether the short description of the marketable service was changed. Can be YES or NO.
- `description=<yes or no>`: Specifies whether the description of the marketable service was changed. Can be YES or NO.
- `locale=<language code>`: The code of the language in which the descriptions were saved.

---

**Operation: Set service as public**

Description: In the publishing options, a service manager, broker, or reseller specified that a marketable service is to be public or that a public marketable service is no longer to be public.

Additional name-value pair:

`public=<true or false>`: Specifies whether the service has been marked as public. Can be `true` (service is public) or `false` (service is not public).

---

**Operation: Terminate subscription**

Description: A service manager or reseller explicitly terminated a customer's subscription.

Additional name-value pairs:

- `subscriptionName=<subscription name>`: The name of the subscription as entered when subscribing to a service.
- `reason=<termination reason>`: Text explaining the reason for terminating the subscription.

---

**Operation: Update service**

Description: A service manager changed a marketable service.

Additional name-value pairs:

- `shortDescription=<yes or no>`: Specifies whether the short description of the marketable service was changed. Can be YES or NO.
- `description=<yes or no>`: Specifies whether the description of the marketable service was changed. Can be YES or NO.
- `locale=<language code>`: The code of the language in which the descriptions were saved.
- `autoAssignUser=<yes or no>`: Specifies whether the user subscribing to the service is automatically assigned to the subscription. Can be YES or NO.

---

**Operation: Update service parameter**

Description: A service manager defined or changed a parameter or parameter option.

Additional name-value pairs:

- `parameterName=<name>`: Name of the parameter as defined in the underlying technical service.
- `userOption=<on or off>`: Specifies whether the parameter is offered as an option to customers who subscribe to the marketable service. Can be ON or OFF.
- `parameterValue=<value>`: For a parameter, the actual string or number; for a parameter option, this can be YES or NO.

---

**Operation: View subscription**

Description: A service manager, broker, or reseller displayed the details of a subscription.

Additional name-value pairs:

- `subscriptionName=<subscription name>`: The name of the subscription as entered when subscribing to a service.
- `customerKey=<organization ID>`: The ID of the customer organization.
- `customerName=<organization name>`: The name of the customer organization.

## Appendix D: Language Resource Bundles

A language bundle consists of the following resources:

- **User interface resources:** All texts that appear on the user interface or in generated emails.
- **Online help** and **FAQ** HTML files: The online help topics are available in the administration portal, the FAQs are available on the marketplaces.
- **Report labels:** The labels and texts for the content of generated reports.

This appendix provides details on how to translate the language resources and how to provide them to the system so that a new language is available to users to set it in their user profile:

- *User Interface Resources* on page 84
- *Online Help and FAQs* on page 85
- *Report Labels* on page 88

### D.1 User Interface Resources

The user interface resources are translated or updated in a Microsoft Excel file generated with the **Export** function on the **Manage languages** page.

The exported Microsoft Excel file contains three worksheets:

- **User interface:** All texts, labels, and messages that appear on the user interface of the platform's administration portal and the marketplaces.
- **Email:** The subject and body texts of all email messages which are generated automatically by the platform.
- **Platform objects:** The names of service parameters and events, report titles and names of payment types provided by the platform. The parameter and event names are visible to suppliers when defining a marketable service as well as to users subscribing to a service.

Each worksheet contains five columns:

- **Key** used by the platform to identify the label or string. The keys must not be changed.
- **de system, en system, ja system:** The system default names and labels provided with the language bundles after installation (German, English, Japanese). The system default strings are for your reference when providing translations.
- **Add your language code here** or **<Language Code>**: In this column, the strings of the new language to be supported are to be entered.

**Add your language code here** is shown if you did not select a language on the **Manage languages** page before exporting the data. In this case, you usually want to provide a new language. The column is empty.

If you selected a language on the **Manage languages** page, the column is headed by the code of this language. In this case, you usually update existing translations of the selected language. You can also select the system default languages (de, en, or jp) if you want to customize the system default strings.

The strings can be also be changed by marketplace managers using the **Marketplace --> Customize texts** menu function. Their changes apply to the marketplaces owned by the organization of the marketplace managers.

Proceed as follows to translate the file:

1. **On the three worksheets**, enter the ISO language code of the language you want to provide in the header of the fifth column.

2. For each key, enter your translation in the fifth column.

If a translation is missing, the English label or string is used. If the English label or string is not defined either, the language-independent technical key is displayed.

Be aware of the following:

- You can use HTML markups in the texts for all keys which represent a descriptive text (keys ending with `.description`). For example, you can use `<b>` for bold text, `<br>` for line breaks, and so on.
- You can use the complete Unicode character set.
- You can remove the text from the table cells which you do not want to change. This may be useful for managing and tracking your changes to the user interface.
- You must make sure that HTML fields contain valid data and do not break the page layout. For example, text which is too long will be broken into multiple lines on the user interface. Use a separate test environment for testing your translations.

3. Save the Microsoft Excel file in `.xls` format.

4. Import the file on the **Manage languages** page.

By importing the file, existing labels and strings in the language with the code entered in the fifth column are overwritten. The language can be used instantly after its activation. Users who have set the language in their profile will instantly see your customizations.

---

**Note:** Make sure that all cells on all worksheets of the Excel file are formatted as text.

---

## D.2 Online Help and FAQs

In the administration portal, online help is available in the language the user has set in his profile. The same applies to the FAQs that can be opened on a marketplace.

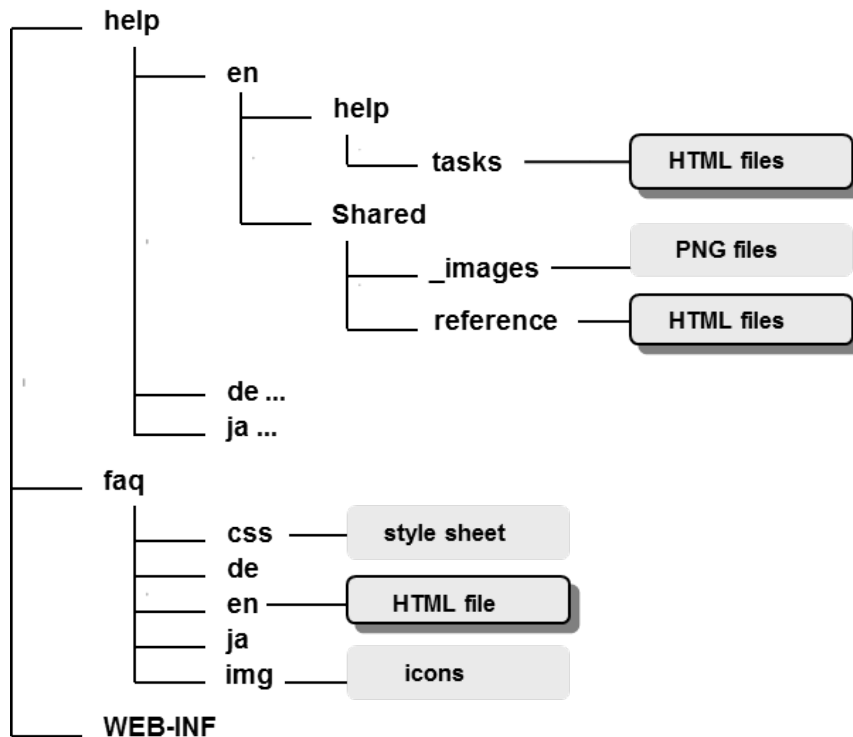
The online help and FAQ files are provided after installation in a separate `.war` file contained in the `oscm-install-pack.zip` file:

```
<install_pack_dir>/domains/bes_domain/oscm-portal-help.war.
```

You can update the existing online help and FAQ files and/or provide the files in another language:

1. Extract the content of the `oscm-portal-help.war` file to a separate directory on your file system.

You see the following directory structure:



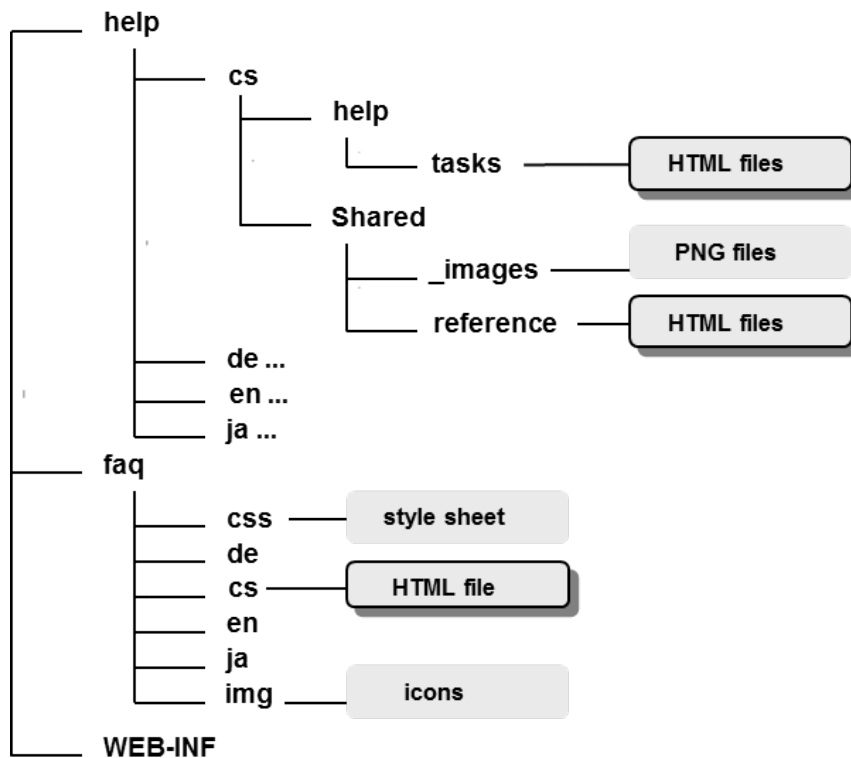
The languages provided after installation are English (en), German (de), and Japanese (ja). You can update the texts in their respective directories. The procedure is the same as when providing the texts in a different language (see below).

2. Copy the `help/en` directory and name it `help/<ISO code>`.

Copy the `faq/en` directory and name it `faq/<ISO code>`.

The ISO language code denotes the language for which you want to provide translations. For example, add a `cs` directory to the `help` as well as to the `faq` directory.

The directory structure now looks as follows:



### 3. Translate the HTML files.

Be aware of the following:

- Make sure not to change any style sheet or icon.
- Do not remove any files that you do not translate.
- In the HTML files, make sure not to enter tags or invalid HTML code that may affect the page layout.
- Use a separate test environment for testing your translations.
- Images used in the online help topics can be translated as follows:
  1. Create the image with an image drawing tool.
  2. Save the image with the file name referenced in the related HTML file to the `Shared/_images` directory.
  3. Check the size of the image in the HTML topic and adapt it, if required.

### 4. Create a new `oscm-portal-help.war` file containing the additional subdirectories for the new language.

### 5. Undeploy and deploy or redeploy the customized `oscm-portal-help.war` file to the `bes-domain` domain on your application server.

By deploying the file, existing HTML files are overwritten.

**Note:** You are responsible for keeping the directory structures and files consistent!

## D.3 Report Labels

Reports are generated in the language the user has set in his profile. The texts used in the reports are provided after installation in properties files contained in a separate `.war` file contained in the `oscm-install-pack.zip` file:

`<install_pack_dir>/domains/bes_domain/birt.war`.

The `birt.war` archive comes with the following properties files:

- `reportLabels.properties` (default language English)
- `reportLabels_de.properties` (German)
- `reportLabels_ja.properties` (Japanese)

You can also change the English, German, or Japanese strings if you want to customize the reports in these languages to your needs.

Proceed as follows to have reports generated in another language:

1. Extract the `birt.war` file to a local directory.
2. Go to the `WEB-INF` subdirectory.
3. Copy the English properties file, `reportLabels.properties`, and name it `reportLabels_<ISO code>.properties`.

The ISO code denotes the language for which you want to provide translations. For example, add a `reportLabels_cs.properties` file to the `WEB-INF` subdirectory.

4. Open the new properties file in a text editor.

The lines in the file are in the following format:

- Key in uppercase characters.
- Equals sign (=).
- String in the language as indicated by the file suffix.

Translate the strings after the equals sign into the desired language. Observe the following:

- Do not modify the keys.
- If you want to translate into a language with non-ASCII characters:  
Use Unicode encoding preceded by escaping backslashes for these characters.  
For example in Japanese: `BASEPRICE=\u30B5\u30FC\u30D3\u30B9\u57FA\u672C\u6599\u91D1` or in German for Umlauts: `INVALID=ung\u00FCltig`
- Special characters such as colons must be escaped using a backslash (`\`). For example:  
`ROLESPEPRICEMODEL=Role costs per subscription\`
- Save the file in UTF-8 encoding.

5. Create a new `birt.war` file containing the additional properties file for the new language.
6. Undeploy and deploy or redeploy the customized `birt.war` file to the `bes-domain` domain on your application server.

With this, existing report labels are overwritten.

---

**Note:** You are responsible for keeping the directory structures and files consistent!

---



## D.4 Supported Language Codes

Below is the list of languages and their codes that can be used for user interface and email texts, parameter and event names, report labels, as well as online help and FAQs:

Language Code	Language
az	Azerbaijani
be	Belarusian
bg	Bulgarian
bn	Bengali
br	Breton
bs	Bosnian
ca	Catalan
ch	Chamorro
cs	Czech
cy	Welsh
da	Danish
de	German
el	Greek
en	English
es	Spanish
et	Estonian
fi	Finnish
fr	French
gl	Gallegan
gu	Gujarati
hi	Hindi
hr	Croatian
hu	Hungarian
ia	Interlingua
in	Indonesian
ii	Sichuan Yi
is	Icelandic
it	Italian

---

Language Code	Language
ja	Japanese
ko	Korean
lt	Lithuanian
lv	Latvian
mk	Macedonian
ml	Malayalam
mn	Mongolian
ms	Malay
nb	Norwegian Bokmål
nl	Dutch
nn	Norwegian Nynorsk
no	Norwegian
pl	Polish
pt	Portuguese
ro	Romanian
ru	Russian
sc	Sardinian
se	Northern Sami
si	Sinhalese
sk	Slovak
sl	Slovenian
sq	Albanian
sr	Serbian
sv	Swedish
ta	Tamil
te	Telugu
th	Thai
tr	Turkish
tt	Tatar
tw	Twi
uk	Ukrainian

Language Code	Language
vi	Vietnamese
zh	Chinese

## Appendix E: User Data File for Multiple User Import

An administrator can import multiple users of his organization and register them with the platform. A platform operator can import multiple users of his own organization as well as of any organization managed on his platform.

The user data must be provided in a file in `csv` (comma-separated values) format. This file can then be imported into ESCM.

The following rules apply:

- The data for one user is provided in one line. Empty lines are ignored.
- The user data file is saved in UTF-8 encoding.
- The user data is provided in the following sequence:
  1. User ID (mandatory)
  2. Email address (mandatory)
  3. Language (mandatory)
  4. Title (optional)
  5. First name (optional)
  6. Last name (optional)
  7. One or several user roles to be assigned to the imported user (mandatory if no marketplace is selected at the user interface)
- The user data fields are separated by a comma each. If an optional field does not contain any data, it must be empty and separated by a comma from the next field.

The fields can take on the following values:

Field	Value(s)
User ID	Mandatory. ID with which the user is to log in to the platform. User IDs are restricted to 100 characters and must not contain any of the following characters: ! " # \$ % & ' * + , / : ; < = > ? \ ^ `
Email address	Mandatory. Email address of the user. It is used for notifying the user about the registration. The system checks whether the syntax of the given email address is valid, and whether the domain name corresponds to the standards as defined and maintained by the Internet Assigned Numbers Authority (IANA). <b>Examples:</b> <code>user.name@domain.arpa</code> , <code>user.name@domain.org</code> , <code>user@mycompany.lan.uk</code>
Language	Mandatory. ISO code of the language in which the user will work by default, for example, <code>en</code> (English).
Title	Optional. Salutation. Allowed values: <code>MR</code> or <code>MS</code> Observe that the value is case-sensitive and must be specified as indicated.
First name	Optional.

Field	Value(s)
Last name	Optional.
User role	<p>Mandatory if no marketplace is selected at the user interface. If a marketplace is selected and no role is specified, a standard user without any privileges is registered.</p> <p>The user roles that can be specified depend on the role of the organization for which the users are to be imported. Valid user roles are:</p> <p>ORGANIZATION_ADMIN (administrator) for organizations with any role.</p> <p>SUBSCRIPTION_MANAGER (subscription manager) for organizations with any role.</p> <p>PLATFORM_OPERATOR (operator) for platform operator organizations.</p> <p>MARKETPLACE_OWNER (marketplace manager) for marketplace owner organizations.</p> <p>SERVICE_MANAGER (service manager) for supplier organizations.</p> <p>TECHNOLOGY_MANAGER (technology manager) for technology provider organizations.</p> <p>BROKER_MANAGER (broker manager) for broker organizations.</p> <p>RESELLER_MANAGER (reseller manager) for reseller organizations.</p> <p>Observe the following:</p> <ul style="list-style-type: none"> <li>• The role names are case-sensitive and must be specified as indicated above.</li> <li>• If several user roles are specified, they must be separated by a comma and enclosed in double quotes (Example: "ORGANIZATION_ADMIN, RESELLER_MANAGER"). Blanks between role names and fields are ignored.</li> <li>• If one user role is specified that is not available for the organization for which to import users, it is ignored and the affected users are not registered.</li> <li>• If several roles are specified for a user, and at least one of these roles is valid for the organization, the user is imported and registered successfully, invalid roles are ignored.</li> </ul>

**Sample user data file:**

```

user1,user1@company.com,en,MS,Jane,Smith,ORGANIZATION_ADMIN
user2,user2@company.com,de,MR,John F.,Cool,SUBSCRIPTION_MANAGER
user3,user3@company.com,en,,,ORGANIZATION_ADMIN
user4,user4@company.com,en,,,Admin,"SERVICE_MANAGER,ORGANIZATION_ADMIN"
user5,user5@company.com,en,MR,,Mueller-Siegel,SERVICE_MANAGER
user6,user6@company.com,en,MR,Joe,StandardUser,

```

## Appendix F: Menu Options and Required Roles

The roles of an organization determine which functions are available to its users at the ESCM user interface and which roles the users can be assigned. The user roles control the actions an individual user is allowed to execute.

This appendix provides a list of the available user interface functions and shows which user of which organization is allowed to execute a function.

### Account Menu

This menu is available to organizations of all roles.

Function	Organization Role	User Role
<b>Edit profile</b>	Any	Any
<b>Import users</b> (if the organization uses LDAP-based user management)	Any	Administrator
<b>Change password</b> (if the organization does not use an external system for user management)	Any	Any
<b>Register new users</b> (if the organization does not use LDAP-based user management)	Any	Administrator
<b>Manage users</b>	Any	Administrator
<b>LDAP settings</b> (if the organization uses LDAP-based user management)	Any	Administrator
<b>Create report</b>	Any	Administrator
<b>Manage suppliers</b>	Technology provider	Technology manager
<b>Manage custom attributes</b>	Supplier	Service manager
<b>Process triggers</b>	Any	Administrator
<b>Manage processes</b> (if the organization is connected to an external process control system)	Any	Any
<b>Export billing data</b>	Supplier, reseller, broker, marketplace owner, operator	Service manager, reseller, broker, marketplace manager, operator

Function	Organization Role	User Role
<b>Define billing period</b>	Supplier, reseller	Service manager, reseller

### Customer Menu

This menu is available to supplier, reseller, and broker organizations.

Function	Organization Role	User Role
<b>Register customer</b>	Supplier, reseller, broker	Service manager, reseller, broker
<b>View customer</b>	Reseller, broker	Reseller, broker
<b>Manage customer</b>	Supplier	Service manager
<b>Manage payment types</b>	Supplier, reseller	Service manager, reseller
<b>Manage VAT rates</b>	Supplier	Service manager
<b>View subscription</b>	Supplier, reseller, broker	Service manager, reseller, broker
<b>Manage subscription attributes</b>	Supplier	Service manager
<b>Terminate subscription</b>	Supplier, reseller	Service manager, reseller

### Marketable Service Menu

This menu is available to supplier, reseller, and broker organizations.

Function	Organization Role	User Role
<b>Define service</b>	Supplier	Service manager
<b>Update service</b>	Supplier	Service manager
<b>View service</b>	Broker	Broker
<b>Manage service</b>	Reseller	Reseller
<b>Copy service</b>	Supplier	Service manager
<b>Delete service</b>	Supplier	Service manager
<b>Define up/downgrade options</b>	Supplier	Service manager
<b>Define publishing options</b>	Supplier, reseller, broker	Service manager, reseller, broker
<b>Activate or deactivate services</b>	Supplier, reseller, broker	Service manager, reseller, broker

### Marketplace Menu

This menu is available to marketplace owner and operator organizations.

Function	Organization Role	User Role
<b>Manage categories</b>	Marketplace owner	Marketplace manager
<b>Manage access</b>	Marketplace owner	Marketplace manager
<b>Manage sellers</b>	Marketplace owner	Marketplace manager
<b>Create marketplace</b>	Operator	Operator
<b>Update marketplace</b>	Marketplace owner, operator	Marketplace manager or operator. The operator can assign a different marketplace owner and define revenue shares; the marketplace manager can change the name and settings of the marketplace.
<b>Delete marketplace</b>	Operator	Operator
<b>Manage broker revenue share</b>	Operator	Operator
<b>Manage reseller revenue share</b>	Operator	Operator
<b>Add tracking code</b>	Marketplace owner	Marketplace manager
<b>Define featured services</b>	Marketplace owner	Marketplace manager
<b>Customize stage</b>	Marketplace owner	Marketplace manager
<b>Customize texts</b>	Marketplace owner	Marketplace manager
<b>Customize layout</b>	Marketplace owner	Marketplace manager

## Operation Menu

This menu is available to operator organizations only.

Function	Organization Role	User Role
<b>Manage users</b>	Operator	Operator
<b>Create organization</b>	Operator	Operator
<b>Create payment service provider</b>	Operator	Operator
<b>Manage organization</b>	Operator	Operator
<b>Manage operator revenue share</b>	Operator	Operator
<b>Manage timers</b>	Operator	Operator
<b>Manage payment service provider</b>	Operator	Operator



Function	Organization Role	User Role
Manage currencies	Operator	Operator
Manage LDAP settings	Operator	Operator
Update configuration settings	Operator	Operator
Billing data preview	Operator	Operator
Execute billing tasks	Operator	Operator
Export audit log	Operator	Operator
Manage languages	Operator	Operator
Manage billing systems	Operator	Operator
Manage tenants	Operator	Operator

### Price Model Menu

This menu is available to supplier organizations only.

Function	Organization Role	User Role
Define for service	Supplier	Service manager
Define for customer	Supplier	Service manager
Delete for customer	Supplier	Service manager
Define for subscription	Supplier	Service manager

### Technical Service Menu

This menu is available to technology provider organizations only.

Function	Organization Role	User Role
Register service definition	Technology provider	Technology manager
Import service definition	Technology provider	Technology manager
Update service definition	Technology provider	Technology manager
Export service definition	Technology provider	Technology manager
Delete service definition	Technology provider	Technology manager
View billing systems	Technology provider	Technology manager

# 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 Service Provider (PSP)**

A company that offers suppliers or resellers online services for accepting electronic payments by a variety of payment methods including credit card or bank-based payments such as direct debit or bank transfer. Suppliers and resellers can use the services of a PSP for the creation of invoices and payment collection.

**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.