FUJITSU

FUJITSU Software Enterprise Service Catalog Manager V17.9

Shell Integration

December 2018

Contents

About this Manual

This manual describes the integration of the Shell service with) FUJITSU Software Enterprise Service Catalog Manager, hereafter referred to as ESCM.

This manual is structured as follows:

Chapter	Description
Introduction on page 6	Provides an overview of the ESCM Shell integration, the components involved, and the supported usage scenarios.
Creating and Publishing Services on page 8	Describes how to create and publish services for the Shell integration in ESCM.
Administrating the AWS Integration on page 10	Describes administration tasks related to the ESCM Shell integration.
Controller Configuration Settings on page 13	Describes the configuration settings for the Shell integration software.
Service Parameters on page 14	Describes the technical service parameters which are supported by the Shell service controller.
Example Scripts on page 16	Provides some sample Shell scripts.

Readers of this Manual

This manual is intended for operators who want to offer Shell services on a marketplace provided by ESCM. It assumes that you have access to an existing ESCM installation that supports Shell execution. In addition, you should have basic knowledge of Shell scripts and you should be familiar with the concepts and administration of ESCM.

Notational Conventions

This manual uses the following notational conventions:

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

Abbreviations

This manual uses the following abbreviations:

APP	Asynchronous Provisioning Platform
AWS	Amazon Web Services
ESCM	Enterprise Service Catalog Manager
DBMS	Database Management System
IaaS	Infrastructure as a Service
IdP	SAML Identity Provider
SAML	Security Assertion Markup Language
STS	Security Token Service
WSDL	Web Services Description Language
WSIT	Web Services Interoperability Technologies

Available Documentation

The following documentation on ESCM is available:

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

1 Introduction

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

The Shell integration allows users to install any software or execute any task in a Windows environment by running Shell scripts.

When integrating applications with ESCM, the instance provisioning can be done in two provisioning modes: synchronous or asynchronous mode.

Asynchronous provisioning is required if provisioning operations take a long time because long-running processes or manual steps are involved. This is the case, for example, when provisioning virtual machines on a virtual machine server. ESCM supports the integration of such applications with its asynchronous provisioning platform (APP). This is a framework which provides a provisioning service as well as functions, data persistence, and notification features which are always required for integrating applications in asynchronous mode.

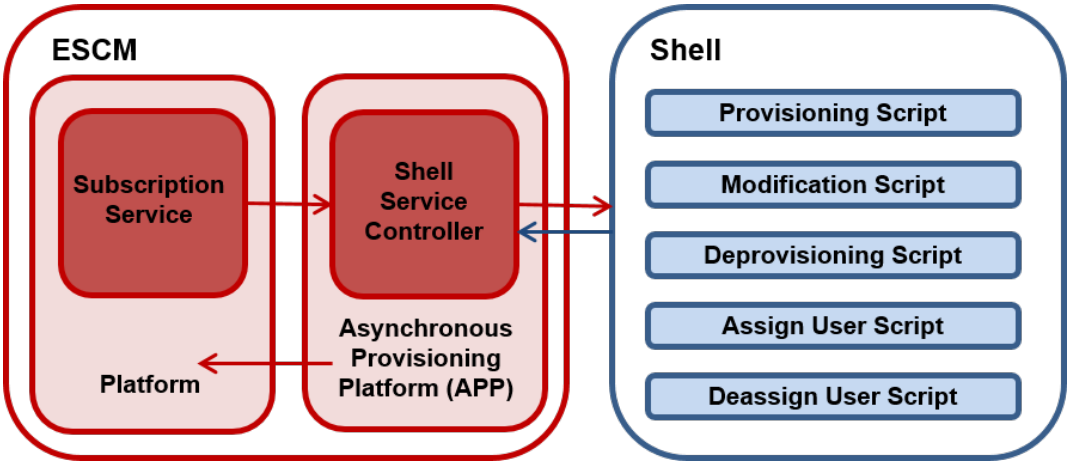
The integration of Shell scripts with ESCM provides for an Infrastructure as a Service (IaaS) solution that leverages the features of both products: Through services, which are published on a marketplace in ESCM, users can request and use Shell scripts. The usage costs can be calculated and charged by means of the ESCM billing and payment services.

When deploying the `oscm-app` container provided with ESCM, a Shell service controller is preconfigured and registered with the asynchronous provisioning platform (APP). This service controller can be used for executing Shell scripts with ESCM.

This manual describes how to create and use services for Shell scripts on an ESCM marketplace.

1.1 Components Involved in the Shell Integration

The following picture provides an overview of the main components involved in the integration of ESCM and Shell scripts:



In ESCM, customer subscriptions are managed by means of the **Subscription service**. When a customer creates or terminates a subscription, the Subscription service asynchronously triggers

the corresponding actions in Shell through the **Asynchronous Provisioning Platform (APP)** and the **Shell service controller**: Scripts are executed in a Shell.

APP is a framework which provides a provisioning service, an operation service, as well as functions, data persistence, and notification features which are required for integrating applications with ESCM in asynchronous mode. The actual communication with the applications is carried out by service controllers. For each application, a specific and dedicated service controller is required. Refer to the *Technology Provider's Guide* for details on APP.

Amazon EC2 allows customers to provision and use virtual servers on which to run their applications. Each virtual server is based on an Amazon Machine Image (AMI). An AMI serves as the basic unit of deployment for services delivered with Amazon EC2. AWS customers can either request pre-configured AMIs or they can create their own images. They can provision their images with a variety of operating systems and load them with custom application environments.

When APP is deployed, an AWS service controller is also deployed, registered, and initialized.

1.2 Usage Scenarios

The ESCM AWS integration supports the following usage scenarios:

- **Executing a Script for provisioning**: When a customer subscribes to a corresponding service on an ESCM marketplace, the Shell service controller triggers the execution of a Shell script.
- **Executing a Script for modifying a subscription**: When a customer reconfigures an existing subscription, the Shell service controller triggers the execution of a Shell script.
- **Executing a Script for deprovisioning**: When a customer terminates a subscription, the Shell service controller triggers the execution of a Shell script. The subscription is terminated in ESCM independent of whether the Script execution is successful.
- **Executing a Script for assigning users**: When a customer assigns users to a subscription, the Shell service controller triggers the execution of a Shell script.
- **Executing a Script for deassigning users**: When a customer removes users from a subscription, the Shell service controller triggers the execution of a Shell script.

2 Creating and Publishing Services

The following sections describe how to create and publish services in ESCM by means of which customers can execute Shell scripts.

2.1 Prerequisites and Preparation

The following prerequisites must be fulfilled before you can create and publish services in ESCM:

- To create technical services for the Shell integration integration in ESCM, you must have access to ESCM as a technology manager. You must be a member of the technology provider organization responsible for the Shell service controller as specified in the configuration settings for the installation.
- Shell scripts must exist. They form the basis for the technical services in ESCM. Shell scripts can be provided in the following ways:
  - The technology provider organization responsible for the Shell service controller provides them on an external host in a location whose URL can be reached from ESCM via HTTP or HTTPS.
  - The ESCM operator provides them on an ESCM host in a location whose URL can be reached from ESCM via HTTP or HTTPS.
  - The ESCM operator provides them on the ESCM host on the file system where they will be referenced by the absolute file path.
- Shell Scripts used for provisioning are subject to the following restrictions:
  - The Script must return the following command at the end of execution: `Write-Output "END_OF_SCRIPT"`. Otherwise the Script execution will run into a timeout.
  - Terminate Scripts without calling `exit`. When calling `exit`, the Shell is no longer available for other Scripts and will be removed from the pool. This is not a problem, just an ineffective way of using the Shell pool.
  - Interactive Scripts will cause a timeout. Remove all interactive command from the Script, such as `Out-GridView`, `Read-Host`, `ShowDialog`, `-Confirm`, `PromptForChoice`, and `Prompt`.
- To create marketable services for the Shell integration in ESCM, you must have access to ESCM as a service manager of an organization with the supplier role. This may be the same organization as the technology provider organization or a different one.
- To publish your marketable services, you must have access to an appropriate marketplace in ESCM in your service manager role.

2.2 Creating Technical Services

The first step in providing ESCM services for Shell is to create one or more technical services.

Proceed as follows:

1. Define one or more technical services in an XML file.

   As a basis, you can use the technical service sample provided on *https://github.com/servicecatalog/oscm/tree/master/oscm-app-shell/resources*.

   A technical service specifies Shell scripts as parameter options. A sample script is also available.

In the technical service definition, be sure to specify:

- The asynchronous provisioning type
- The USER access type
- Service parameters which correspond to the parameters specified in the Shell Scripts. For details on the supported service parameters, refer to *Service Parameters* on page 14.

2. Log in to the ESCM administration portal with your technology manager account.
3. Import the technical services you created and appoint one or more supplier organizations for them.

For details on these steps, refer to the *Technology Provider's Guide* and to the online help of ESCM.

2.3 Creating and Publishing Marketable Services

As soon as the technical services for the Shell integration exist in ESCM, you can define and publish marketable services based on them. Your cost calculation for the services should include any external costs for operating the virtual systems.

Proceed as follows:

1. Log in to the ESCM administration portal with your service manager account.
2. Define one or more marketable services based on the technical services you created for the Shell.
3. Define price models for your marketable services.
4. Publish the services to a marketplace.

For details on these steps, refer to the *Supplier's Guide* and to the online help of ESCM.

3 Administrating the AWS Integration

The following sections describe administration tasks you may need to perform in your role as an operator of the Shell integration software.

3.1 Controlling the Provisioning Process

The Shell integration provides you with the following feature for controlling the execution of Shell scripts:

In the definition of the technical services for Shell, you can specify the `MAIL_FOR_COMPLETION` parameter. This is an address to which emails are to be sent in case manual steps are required to complete an operation.

If you specify this parameter, the Shell service controller interrupts the processing of each operation before its completion and waits for a notification about the execution of a manual action. This notification consists in opening the link given in the email.

Omit the `MAIL_FOR_COMPLETION` parameter if you do not want to interrupt the processing.

3.2 Handling Problems in the Provisioning Process

If the provisioning of fails on the side or if there are problems in the communication between the participating systems, the corresponding subscription in ESCM remains pending. The service controller informs the technology managers of its responsible technology provider organization by email of any incomplete operation in .

You can then take the appropriate actions to solve the problem in or in the communication. For example, you could remove an incomplete , or you could restore a missing connection.

After solving the problem, the integration components and ESCM need to be synchronized accordingly. You do this by triggering a corresponding action in the APP component. Proceed as follows:

1. Work as a technology manager of the technology provider organization responsible for the service controller.
2. Invoke the instance status interface of APP for the service controller of the application by opening it's URL in a Web browser.

   The access URL has the following format:

   `https://<hostname.fqdn>:<port>/oscm-app/controller/?controllerid=ess.`

   `<hostname.fqdn>` is the name and the fully qualified domain name of the machine where the `oscm-app` container has been deployed, `<port>` is the port to address the machine (default: `8881`), `oscm-app/controller/?controllerid=ess.` is the default context root of the service controller and cannot be changed.

   The Web page shows all subscriptions for the application, including detailed information such as the customer organization, the ID of the related application instance, and the provisioning status.
3. Find the subscription for which you solved the problem in the most recent operation.
4. In the **Action** column, select the action for the integration components to execute next. Possible actions are the following:
   - `RESUME` - to resume the processing of a provisioning operation in APP which was suspended.

- `SUSPEND` - to suspend the processing of a provisioning operation in APP, for example when does not respond.
- `UNLOCK` - to remove the lock for instance in APP.
- `DELETE` - to terminate the subscription in ESCM and remove the instance in APP, but keep the for later use. The service manager role is required for this action.
- `DEPROVISION` - to terminate the subscription in ESCM, remove the instance in APP, and delete the . The service manager role is required for this action.
- `ABORT_PENDING` - to abort a pending in ESCM. ESCM is notified to roll back the changes made for the subscription and return it to its previous state. In , no actions are carried out.
- `COMPLETE_PENDING` - to complete a pending in ESCM. ESCM is notified to complete the changes for the subscription and set the subscription status to **ready** (or **suspended** if it was suspended before). This is possible only if the operations of the service controller are already completed.

5. Click **Execute** to invoke the selected action.

The instance status interface provides the following additional functionality that is useful for problem-solving purposes:

- You can display service instance details for each subscription by clicking the corresponding entry in the table. This displays all subscription-related information that is stored in the `bssapp` database.

3.3 Updating Service Controller Settings in the Database

During deployment, several configuration settings are written to the `bssapp` database. This configuration is used for the initial setup of the Shell service controller and its registration with APP. It is up to the platform operator for taking care that the initial settings are correct. Refer to the *Operator's Guide* for details on the initial configuration using a `var.env` file.

A technology provider can define service parameters in the technical service definition. If such a parameter has the same ID as a controller configuration setting stored in the APP database, it overrules the configuration setting in the database when the marketable service based on such a technical service is subscribed to. By default the values in the controller configuration settings are used. Refer to the *Technology Provider's Guide* for details on defining technical services.

In addition, a supplier can define custom attributes for subscriptions and for customers. If such an attribute has the same ID as a controller configuration setting stored in the APP database as well as a corresponding technical service parameter, it overrules the technical service parameter as well as the configuration setting in the database when the marketable service based on such a technical service is subscribed to.

The controller configuration settings are evaluated as follows:

1. Configuration setting as stored in the APP database.
2. Technical service parameter. If defined, it overrules 1.
3. Custom attribute for customer. If defined, it overrules 1. and 2.
4. Custom attribute for subscription. If defined, it overrules 1. and 2. and 3.

To update the controller settings in the APP database:

1. Invoke the instance status interface of APP for the Shell service controller of the application by opening it's URL in a Web browser.

   The access URL has the following format:

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

`<hostname.fqdn>` is the name and the fully qualified domain name of the machine where the `oscm-app` container has been deployed, `<port>` is the port to address the machine (default: `8881`), `oscm-app-shell` is the default context root of the service controller and cannot be changed.

2. Log in with the ID and password of the user specified in the configuration settings for the Shell service controller by the platform operator in the `BSS_USER_ID` and `BSS_USER_PWD` configuration settings, or as another technology manager registered for the same organization.
3. The following settings can be changed:
   - **User ID**: The identifier of the user responsible for the Shell service controller.
   - **User Key**: The user key for accessing ESCM. You receive this key with the confirmation email for your user account. The user must have the technology manager role in ESCM and belong to the technology provider organization responsible for the service controller.

     It is recommended that the user account is used only for carrying out actions on behalf of the service controller in ESCM.
   - **Password**: The password of the user for accessing ESCM.
4. Save the settings.

3.4 Changing the Responsible Organization

You can change the technology provider organization responsible for the OpenStack service controller using the Web interface of APP:

1. In a Web browser, access the Web interface (base URL) of APP.

   The access URL has the following format:

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

   `<hostname.fqdn>` is the name and the fully qualified domain name of the machine where the `oscm-app` container has been deployed, `<port>` is the port to address the machine (default: `8881`), `oscm-app` is the default context root of APP and cannot be changed.
2. Log in with the ID and password of the user specified for `BSS_USER_KEY` in the configuration settings for APP or as another administrator of the same organization.
3. Specify the technology provider organization for the Shell service controller, `ess.shell`.
4. Save the settings.
5. Make sure that the configuration settings for the Shell service controller are updated.

   Any technology manager registered for the technology provider organization you specified can log in to the graphical user interface for updating the controller configuration settings (see above). At least the ID and password of the user to be used for accessing ESCM must be changed in the controller configuration settings.

Appendix A: Controller Configuration Settings

This appendix describes the controller configuration settings as stored in the `bssapp` database. For details on how to update them, refer to *Updating Service Controller Settings in the Database* on page 11 and *Changing the Responsible Organization* on page 12.

CONTROLLER_ID

`CONTROLLER_ID=ess.shell`

The identifier of the service controller. This setting cannot be changed.

BSS_ORGANIZATION_ID

`BSS_ORGANIZATION_ID=<organizationID>`

The ID of the organization in ESCM which is responsible for the service controller. The organization must have the technology provider role. It is created and initially assigned to the Shell service controller by the platform operator.

BSS_USER_ID

`BSS_USER_ID=<userId>`

The identifier of the user specified in `BSS_USER_KEY` for accessing ESCM.

BSS_USER_KEY

`BSS_USER_KEY=<userKey>`

The user key for accessing ESCM.

The user key for accessing ESCM. You receive this key with the confirmation email for your user account. The user must have the technology manager role in ESCM and belong to the technology provider organization responsible for the service controller.

It is recommended that the user account is used only for carrying out actions on behalf of the service controller in ESCM.

BSS_USER_PWD

`BSS_USER_PWD=_crypt:<password>`

The password of the user for accessing ESCM.

Appendix B: Service Parameters

The following section describes the technical service parameters which are supported by the Shell service controller.

You find a sample service on GitHub:
https://github.com/servicecatalog/oscm/tree/master/oscm-app-shell/resources.

The Shell service controller supports the parameters below.

Note: All parameters defined in the technical service definition must be one-time parameters, since the modification of parameters is not supported. Be sure to set their `modificationType` to `ONE_TIME`.

APP_CONTROLLER_ID

Mandatory. The ID of the service controller as defined in its implementation. The ID is set during the installation of the Shell integration software.

Default (must not be changed): `ess.shell`

CONSOLE_FILE

Optional. The absolute file system path of the Shell configuration file.

The Shell will be executed with the `-PSConsoleFile` argument in case the file name is specified.

Example: `C:/TEMP/Console.psc1`

PROVISIONING_SCRIPT

Mandatory. The absolute file system path or URL to a Shell script file. This script will be executed when a user subscribes to a service on the marketplace.

Example: `C:/TEMP/example_script.ps1`

DEPROVISIONING_SCRIPT

Mandatory. The absolute file system path or URL to a Shell script file. This script will be executed when a user terminates the subscription.

Example: `C:/TEMP/example_script.ps1`

UPDATE_SCRIPT

Mandatory. The absolute file system path or URL to a Shell script file. This script will be executed when a the configuration of an existing subscription is changed.

Example: `C:/TEMP/example_script.ps1`

ASSIGN_USER_SCRIPT

Mandatory. The absolute file system path or URL to a Shell script file. This script will be executed when a user is assigned to a subscription.

Example: `C:/TEMP/example_script.ps1`

DEASSIGN_USER_SCRIPT

Mandatory. The absolute file system path or URL to a Shell script file. This script will be executed when a user is removed from a subscription.

Example: `C:/TEMP/example_script.ps1`

CHECK_STATUS_SCRIPT

Mandatory. The absolute file system path or URL to a Shell script file. This script will be executed when a user opens the Details view under My Subscriptions. It shows the status of the results of the provisioning script execution.

Example: `C:/TEMP/example_script.ps1`

SCRIPT_TIMEOUT_SECONDS

Mandatory. The number of seconds until a running script will be canceled.

Example: `600`

<freely definable service parameter

Optional. Any number of parameters that are mapped from the parameters defined in the Shell script files. For each parameter in the script file, a corresponding parameter must be specified in the technical service definition.

All service parameters are patched into the script file at the top of the file. For example, a service parameter called `MY_PARAM` can be used as `$MY_PARAM` in the script file.

Example: `600`

Appendix C: Example Scripts

Citrix.XenApp.Sdk.psc1

```
<?xml version="1.0" encoding="utf-8"?>
<PSConsoleFile ConsoleSchemaVersion="1.0">
  <PSVersion>2.0</PSVersion>
  <PSSnapIns>
    <PSSnapIn Name="Citrix.Common.Commands" />
    <PSSnapIn Name="Citrix.XenApp.Commands" />
    <PSSnapIn Name="Citrix.Common.GroupPolicy" />
  </PSSnapIns>
</PSConsoleFile>
```

Console.psc1

```
<?xml version="1.0" encoding="utf-8"?>
<PSConsoleFile ConsoleSchemaVersion="1.0">
  <PSVersion>5.1.15063.483</PSVersion>
  <PSSnapIns />
</PSConsoleFile>
```

example_script.ps1

```
Write-Output "All Serviceparameters are patched into the script."
Write-Output "$MY_SCRIPT_PARAM_1"
sleep 2
Write-Output "END_OF_SCRIPT"
```

Glossary

Administrator

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

Application

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

Billing System

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

Broker

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

Cloud

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

Cloud Computing

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

Customer

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

Infrastructure as a Service (IaaS)

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

Marketable Service

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

Marketplace

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

Marketplace Owner

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

Marketplace Manager

A privileged user role within a marketplace owner organization.

Operator

An organization or person responsible for maintaining and operating ESCM.

Organization

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

Organizational Unit

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

OU Administrator

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

Payment Type

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

Platform as a Service (PaaS)

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

Price Model

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

Reseller

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

Role

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

Seller

Collective term for supplier, broker, and reseller organizations.

Service

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

Service Manager

A privileged user role within a supplier organization.

Standard User

A non-privileged user role within an organization.

Software as a Service (SaaS)

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

Subscription

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

Subscription Manager

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

Supplier

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

Technical Service

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

Technology Manager

A privileged user role within a technology provider organization.

Technology Provider

An organization which provisions applications as technical services in ESCM.