**Open Service**

**Catalog Manager V18**

**PowerShell Integration (GlassFish)**

|  |  |
| --- | --- |
| 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, GlassFish, Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle Corporation and/or its affiliates.  Apache Ant, Ant, 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 | |
| High Risk Activity | |
| The Customer acknowledges and agrees that the Product is designed, developed and manufactured as contemplated for general use, including without limitation, general office use, personal use, household use, and ordinary industrial use, but is not designed, developed and manufactured as contemplated for use accompanying fatal risks or dangers that, unless extremely high safety is secured, could lead directly to death, personal injury, severe physical damage or other loss (hereinafter "High Safety Required Use"), including without limitation, nuclear reaction control in nuclear facility, aircraft flight control, air traffic control, mass transport control, medical life support system, missile launch control in weapon system. The Customer shall not use the Product without securing the sufficient safety required for the High Safety Required Use. In addition, FUJITSU (or other affiliate's name) shall not be liable against the Customer and/or any third party for any claims or damages arising in connection with the High Safety Required Use of the Product. | |
| Export Restrictions | |
| Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws. | |

# 

Contents

[About this Manual 4](#_Toc490131796)

[Readers of this Manual 4](#_Toc490131797)

[Notational Conventions 4](#_Toc490131798)

[Abbreviations 4](#_Toc490131799)

[Available Documentation 5](#_Toc490131800)

[Introduction 6](#_Toc490131801)

[Components Involved in the PowerShell Integration 6](#_Toc490131802)

[Usage Scenarios 7](#_Toc490131803)

[Installing the PowerShell Integration Software 8](#_Toc490131804)

[Creating and Publishing Services 8](#_Toc490131805)

[Prerequisites and Preparation 8](#_Toc490131806)

[Creating Technical Services 9](#_Toc490131807)

[Creating and Publishing Marketable Services 10](#_Toc490131808)

[Configure Status Information Panel 10](#_Toc490131809)

[Administrating the PowerShell Integration 11](#_Toc490131810)

[Handling Communication Problems Between APP and OSCM 12](#_Toc490131811)

[Adapting the Log Configuration 12](#_Toc490131812)

[Configuration Settings 13](#_Toc490131813)

[Controller Configuration Settings 13](#_Toc490131814)

[Service Parameters and Operations 14](#_Toc490131815)

[Glossary 17](#_Toc490131816)

# About this Manual

This manual describes the integration of the Windows PowerShell with Open Service Catalog Manager (OSCM). This manual is structured as follows:

| Chapter | Description |
| --- | --- |
| [Introduction](#N90006) | Provides an overview of the OSCM PowerShell integration, the components involved, and the supported usage scenarios. |
| [Installing the PowerShell Integration Software](#NA0006) | Describes how to prepare and carry out the installation of the PowerShell integration software. |
| [Creating and Publishing Services](#NB0006) | Describes how to create and publish services for the PowerShell Integration in OSCM. |
| [Administrating the PowerShell Integration](#ND0006) | Describes administration tasks related to the OSCM PowerShell integration software. |

## Readers of this Manual

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

## 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 |
| **DBMS** | Database Management System |
| **IaaS** | Infrastructure as a Service |
| **IdP** | SAML Identity Provider |
| **OSCM** | Open Service Catalog Manager |
| **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 OSCM is available:

1. *Overview:* A PDF manual introducing OSCM. It is written for everybody interested in OSCM and does not require any special knowledge.
2. *Online Help:* Online help pages describing how to work with the administration portal of OSCM. The online help is intended for and available to everybody working with the administration portal.
3. *Installation Guide (GlassFish):* A PDF manual describing how to install and uninstall OSCM. It is intended for operators who set up and maintain OSCM in their environment.
4. *Operator's Guide:* A PDF manual for operators describing how to administrate and maintain OSCM.
5. *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 OSCM.
6. *Supplier's Guide:* A PDF manual for suppliers describing how to define and manage service offerings for applications that have been integrated with OSCM.
7. *Reseller's Guide:* A PDF manual for resellers describing how to prepare, offer, and sell services defined by suppliers.
8. *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.
9. *Marketplace Owner's Guide:* A PDF manual for marketplace owners describing how to administrate and customize marketplaces in OSCM.
10. *Developer's Guide:* A PDF manual for application developers describing the public Web services and application programming interfaces of OSCM and how to integrate applications and external systems with OSCM.
11. *Amazon Web Services Integration (GlassFish):* 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 OSCM.
12. *OpenStack Integration (GlassFish):* A PDF manual for operators describing how to offer and use virtual systems controlled by OpenStack through services in OSCM.
13. *VMware vSphere Integration (GlassFish):* A PDF manual for operators describing how to offer and use virtual machines provisioned on a VMware vSphere server through services in OSCM.
14. Javadoc and YAML documentation for the public Web services and application programming interfaces of OSCM and additional resources and utilities for application developers.

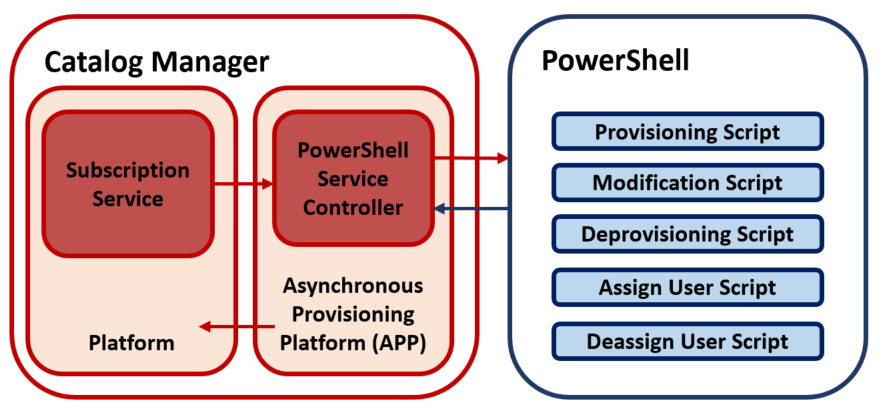
# Introduction

Open Service Catalog Manager (OSCM) 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 PowerShell Integration allows users to install any software or execute any task in a Windows environment by running PowerShell scripts. This manual describes how to deploy this package and how to create and use services for PowerShell Scripts on a OSCM marketplace.

## Components Involved in the PowerShell Integration

The following picture provides an overview of the main components involved in the integration of OSCM and PowerShell:



In OSCM, 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 a PowerShell through the **Asynchronous Provisioning Platform (APP)** and the **PowerShell service controller**: Scripts are executed in a PowerShell.

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 OSCM in asynchronous mode. The actual communication with the applications is carried out by service controllers. APP and the PowerShell service controller are the main components that make up the PowerShell integration software.

Each APP installation supports one PowerShell service controller. This limitation can be overcome by installing APP several times to different application server domains. The need for more than one service controller may arise because multiple technology provider organizations have to be used.

## Usage Scenarios

The OSCM PowerShell integration supports the following usage scenarios:

1. **Executing a script for Provisioning**: When a customer subscribes to a corresponding service on a OSCM marketplace, the service controller triggers the execution of a PowerShell script.
2. **Executing a script for modifying Subscription**: When a customer reconfigures an existing subscription, the service controller triggers the execution of a PowerShell script.
3. **Executing a script for Deprovisioning**: When a customer terminates a subscription, the service controller triggers the execution of a PowerShell script. The subscription terminates in OSCM independent of whether the script execution is successful.
4. **Executing a script for assigning users**: When a customer assigns users to a subscription, the service controller triggers the execution of a PowerShell script.
5. **Executing a script for removing users**: When a customer removes users from a subscription, the service controller triggers the execution of a PowerShell script.

# Installing the PowerShell Integration Software

The following sections describe how to install and configure the PowerShell integration software as well as the preparations you need to take beforehand.

1. Deploy the PowerShell service controller powershell-controller-0.0.1.war to the app-domain domain. To do this, you use the GlassFish administration console, for example http://127.0.0.1:8848/
2. Register the PowerShell service controller as follows in APP:
   1. In a Web browser, access the base URL of APP, for example: http://127.0.0.1:8880/oscm-app
   2. Log in with the ID and password of the user and organization defined as the APP owner.
   3. Specify the controller ID (ess.powershell) and the technology provider organization responsible for the PowerShell service controller.
   4. Click **Save Configuration** to save the settings.

# Creating and Publishing Services

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

## Prerequisites and Preparation

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

1. To create technical services for the PowerShell integration in OSCM, you must have access to OSCM as a technology manager. You must be a member of the technology provider organization responsible for the PowerShell service controller as specified in the configuration settings for the installation.
2. PowerShell scripts must exist. They form the basis for the technical services in OSCM. PowerShell scripts can be provided in one of the following ways:
   1. The technology provider organization responsible for the PowerShell service controller provides them on an external host in a location whose URL can be reached from OSCM via HTTP or HTTPS.
   2. The OSCM operator provides them on the OSCM host in a location whose URL can be reached from OSCM via HTTP or HTTPS.
   3. The OSCM operator provides them on the OSCM host on the filesystem where they will be referenced by the absolute filesystem path.
3. PowerShell scripts used for provisioning are subject to certain restrictions:
   1. 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.
   2. Terminate scripts without calling exit. When calling exit the PowerShell 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 PowerShell pool.
   3. Interactive scripts will cause a timeout. Remove all interactive commands from the script like Out-GridView, Read-Host, ShowDialog, -Confirm, PromptForChoice and Prompt.
4. To create marketable services for the PowerShell integration in OSCM, you must have access to OSCM 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.
5. To publish your marketable services, you must have access to an appropriate marketplace in OSCM in your service manager role.

## Creating Technical Services

The first step in providing OSCM services for PowerShell is to create one or more technical services.

Proceed as follows:

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

The PowerShell integration package includes a technical service as a sample:

TechnicalService.xml

A technical service specifies PowerShell scripts as parameter options. A sample script is also included:

example\_script.ps1

In the technical service definition, be sure to specify:

* 1. The asynchronous provisioning type
  2. The USER access type
  3. Service parameters which correspond to the parameters specified in the PowerShell scripts. For details on the supported service parameters, refer to [Service Parameters and Operations](#N2B0006).

1. Log in to the OSCM administration portal with your technology manager account.
2. 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 OSCM.

## Creating and Publishing Marketable Services

As soon as the technical services for the PowerShell integration exist in OSCM, 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.

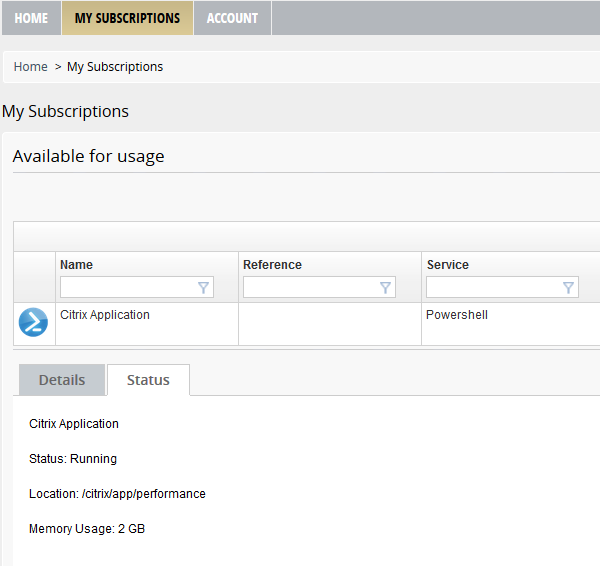
Proceed as follows:

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

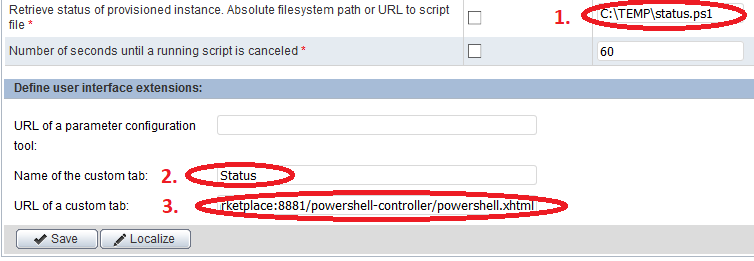
For details on these steps, refer to the *Supplier's Guide* and to the online help of OSCM.

## Configure Status Information Panel

The following screenshot shows where you can see additional information about a provisioned instance. When you login to the marketplace and select a subscription under “My Subscriptions” you see the two tabs “Details” and “Status”.



In order to see the Status tab the marketable service is configured the following way.



1. Enter the script for retrieving the status information
2. Give the custom tab a name. Do not name it “Details” because the first tab is named that way.
3. Enter the URL that points to the file powershell.xhtml wherever you have deployed the powershell controller.

# Administrating the PowerShell Integration

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

## Handling Communication Problems Between APP and OSCM

When the communication between APP and OSCM is no longer possible, for example, because OSCM is stopped, APP suspends the processing of requests. An internal flag is set in the APP database: APP\_SUSPEND=true, and an email is sent to the address specified in the APP\_ADMIN\_MAIL\_ADDRESS configuration setting.

Contact the OSCM operator to make sure that OSCM is up and running again correctly.

You then have the following possibilities to resume the processing of requests by APP:

1. Click the link provided in the email.
2. Log in to APP. APP is restarted instantly. In the APP database, the APP\_SUSPEND key is set to false.

As an alternative, you can proceed as follows:

1. In a Web browser, access the base URL of APP, for example:

http://127.0.0.1:8880/oscm-app

1. Log in with the ID and password of the user and organization that owns the APP (BSS\_USER\_ID and BSS\_USER\_PWD). A message is shown that APP has been suspended due to a communication problem with OSCM.
2. Click **Restart**. APP is restarted instantly. In the APP database, the APP\_SUSPEND key is set to false.

## Adapting the Log Configuration

The PowerShell integration software records information and problems in the following log files on the application server:

1. <GLASSFISH\_HOME>/domains/<DOMAIN\_NAME>/logs/app-powershell.log: Log of the PowerShell service controller
2. <GLASSFISH\_HOME>/domains/<DOMAIN\_NAME>/logs/app-core.log: Log of the APP component

The logging is based on log4j. The default log level is INFO, which may not be sufficient depending on the circumstances. In such a case, you will need to adapt the log level in the configuration files. The following configuration files are of relevance:

1. <GLASSFISH\_HOME>/domains/<DOMAIN\_NAME>/config/log4j.ess.powershell.properties: Log configuration of the PowerShell service controller
2. <GLASSFISH\_HOME>/domains/<DOMAIN\_NAME>/config/log4j.app.core.properties: Log configuration of the APP component

Proceed as follows to adapt the log level:

1. Open the relevant configuration file.
2. Find the string log4j.logger.org.oscm.app in the configuration file.
3. Change the log level as desired to one of the following:
   1. ERROR - designates error events that might still allow the application to continue running.
   2. WARN - designates potentially harmful situations.
   3. INFO - designates informational messages that highlight the progress of the application at coarse-grained level.
   4. DEBUG - designates fine-grained informational events that are most useful to debug an application.

Example:

log4j.logger.org.oscm.app=INFO

Every 60 seconds, the PowerShell integration software checks for changes in the log configuration. There is no need to restart the application.

# Configuration Settings

## Controller Configuration Settings

A technology provider can define service parameters in the technical service definition. Refer to the *Technology Provider's Guide* for details on defining technical services.

**CONTROLLER\_ID**

CONTROLLER\_ID=ess.PowerShell

The identifier of the service controller.

**BSS\_ORGANIZATION\_ID**

BSS\_ORGANIZATION\_ID=<organizationID>

The ID of the organization in OSCM responsible for the service controller. The organization must have the technology provider role.

**BSS\_USER\_ID**

BSS\_USER\_ID=<userId>

The identifier of the user specified in BSS\_USER\_KEY for accessing OSCM.

Replace <userId> with the user ID.

**BSS\_USER\_KEY**

BSS\_USER\_KEY=<userKey>

The user key for accessing OSCM.

Replace <userKey> with the user key which you receive with the confirmation email for your user account.

The user specified here must have the technology manager role in OSCM and belong to the organization specified in the BSS\_ORGANIZATION\_ID setting.

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

**BSS\_USER\_PWD**

BSS\_USER\_PWD=\_crypt:<password>

The password of the user specified in BSS\_USER\_KEY for accessing OSCM.

Replace <password> with the plain text password. The password is encrypted when the app domain is restarted.

# Service Parameters and Operations

The following sections describe the technical service parameters which are supported by the PowerShell service controller.

**Service Parameters**

The PowerShell service controller supports the parameters below.

|  |
| --- |
| **APP\_CONTROLLER\_ID**  Mandatory. The ID of the service controller as defined in its implementation.  Default (must not be changed): ess.powershell |

|  |
| --- |
| **CONSOLE\_FILE**  Optional. The absolute file system path to a PowerShell configuration file.  The PowerShell will be executed with the argument –PSConsoleFile if the filename is specified.  Example: C:/TEMP/Console.psc1 |

|  |
| --- |
| **PROVISIONING\_SCRIPT**  Mandatory. The absolute file system path or URL to a PowerShell script file. This script will be executed when a user subscribes to a service offering on the marketplace.  Example: C:/TEMP/example\_script.ps1 |

|  |
| --- |
| **DEPROVISIONING\_SCRIPT**  Mandatory. The absolute file system path or URL to a PowerShell script file. This script will be executed when the subscription is terminated.  Example: C:/TEMP/example\_script.ps1 |

|  |
| --- |
| **UPDATE\_SCRIPT**  Mandatory. The absolute file system path or URL to a PowerShell script file. This script will be executed when 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 PowerShell 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 PowerShell 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 PowerShell script file. This script will be executed when a user opens the “Details” view under “My Subscription”. It shows the status of the whatever the provisioning script has been installed.    Example: C:/TEMP/example\_script.ps1 |

|  |
| --- |
| **SCRIPT\_TIMEOUT\_SECONDS**  Mandatory. The number of seconds until a running script is canceled.    Example: 600 |

|  |
| --- |
| **<freely definable service parameter>**  Optional. Any number of parameters that are mapped from the parameters defined in the PowerShell script files. For each parameter in the script file, there must be a corresponding parameter in the technical service definition.  All service parameters a patched into the script file at the top of the file. For example a service parameter “MY\_PARAM” can be used as “$MY\_PARAM” in the script file. |

# 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. OSCM comes with a native billing system, but can also be integrated with external ones.

**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 OSCM 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 OSCM, 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 OSCM to provide their services to customers.

**Marketplace Owner**

An organization which holds a marketplace in OSCM, 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 OSCM.

**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 OSCM; 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. OSCM 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 OSCM. 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 OSCM for offering applications provisioned by technology providers to customers.

**Technical Service**

The representation of an application in OSCM. 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 OSCM.