FUJITSU Software Enterprise Service Catalog Manager V17.6.0

APP Integration Guide

December 2017 - Draft for review

High Risk Activity

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

Export Restrictions

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

Contents

About this Manual.. 4

1 Integrating Applications with ESCM Using APP..........................6

2 Configuring APP... 10

2.1 Hardware and Operating Systems...10

2.2 Java and Ant.. 10

2.3 Application Server.. 11

2.4 Relational Database...11

2.5 Mail Server..12

2.6 Preparing the Software and Setup Utilities.. 12

2.7 Configuring APP.. 13

2.8 Setting up the Database... 14

2.9 Setting up the Application Server Resources.. 15

2.10 Exchanging Certificates..16

2.11 Update Installation..17

3 Implementing a Service Controller.. 20

3.1 Registering a Service Controller..21

3.2 Deploying a Service Controller.. 21

4 Administrating APP... 22

4.1 Handling Problems in the Provisioning Process... 22

4.2 Handling Communication Problems Between APP and ESCM......................23

4.3 Backup and Recovery... 23

4.4 Updating Configuration Settings.. 24

4.5 Adapting the Log Configuration... 25

Appendix A Configuration Settings... 27

A.1 APP Configuration Settings...27

A.2 Database Configuration Settings...28

About this Manual

This manual describes the asynchronous provisioning platform (APP) of) FUJITSU Software Enterprise Service Catalog Manager, hereafter referred to as ESCM.

Readers of this Manual

This manual is directed to developers who carry out tasks for operator organizations involved in integrating payment service providers with ESCM.

This manual assumes that you are familiar with the following:

- Basic Web service concepts
- XML and the XSD language
- Web services standards SOAP and WSDL
- A programming language that can be used to create and invoke Web services, for example, Java
- Installation and basic administration of Web servers

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 user guide uses the following abbreviations:

APP	Asynchronous Provisioning Platform
ESCM	Enterprise Service Catalog Manager
SaaS	Software as a Service
SOAP	Simple Object Access Protocol
WSDL	Web Services Description Language

Available Documentation

The following documentation on ESCM is available:

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

1 Integrating Applications with ESCM Using APP

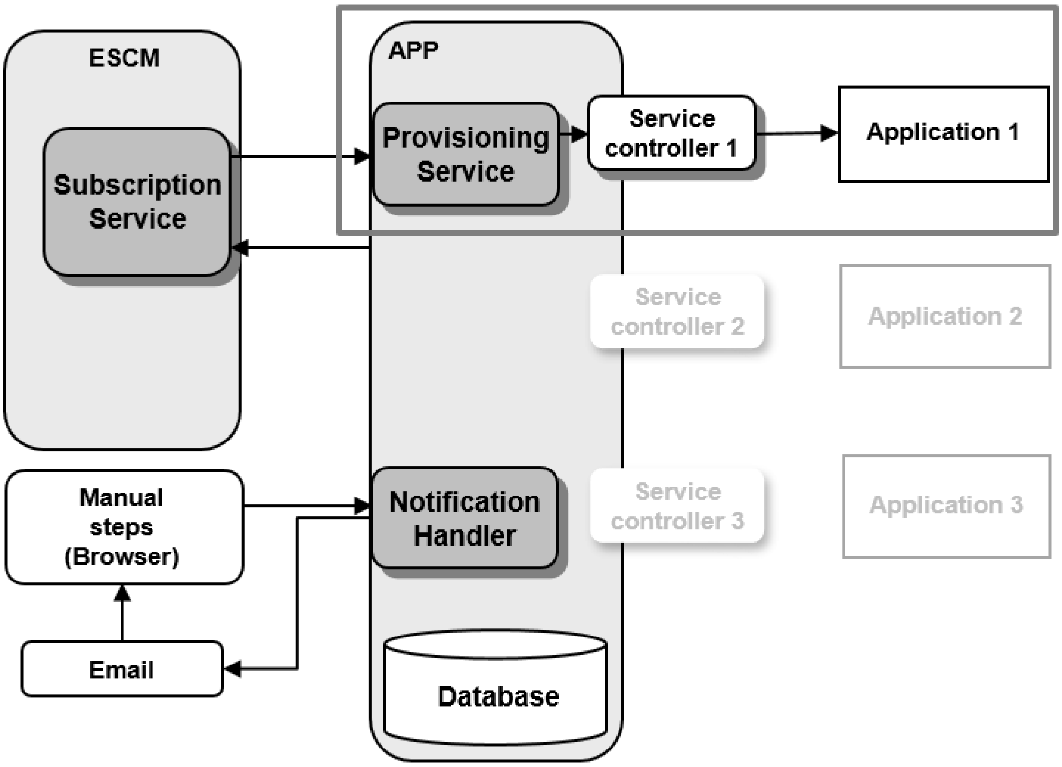
When integrating applications with ESCM, the instance provisioning can be done in two provisioning modes: synchronous or asynchronous mode.

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.

APP also includes the operation service interface for executing technical service operations on the integrated applications from the ESCM user interface.

Components and Communication Paths

The following figure illustrates the components and communication paths involved when using APP to integrate an application with ESCM.



The following components are involved:

- ESCM with the subscription service handling the provisioning tasks.
- APP including the provisioning service, a database, a notification handler, and an API to enable the communication between the application instances and ESCM.

- A service controller for communication with the application.

  The following ready-to-use service controllers are available, registered, and initialized ### true for both controllers? ### when deploying the `oscm-app` container:

  - AWS service controller: Can be used for integrating the Amazon Elastic Compute Cloud Web service with ESCM.
  - OpenStack service controller: Can be used for integrating OpenStack services with ESCM.

- The application with a remote interface and the application instances, if needed.

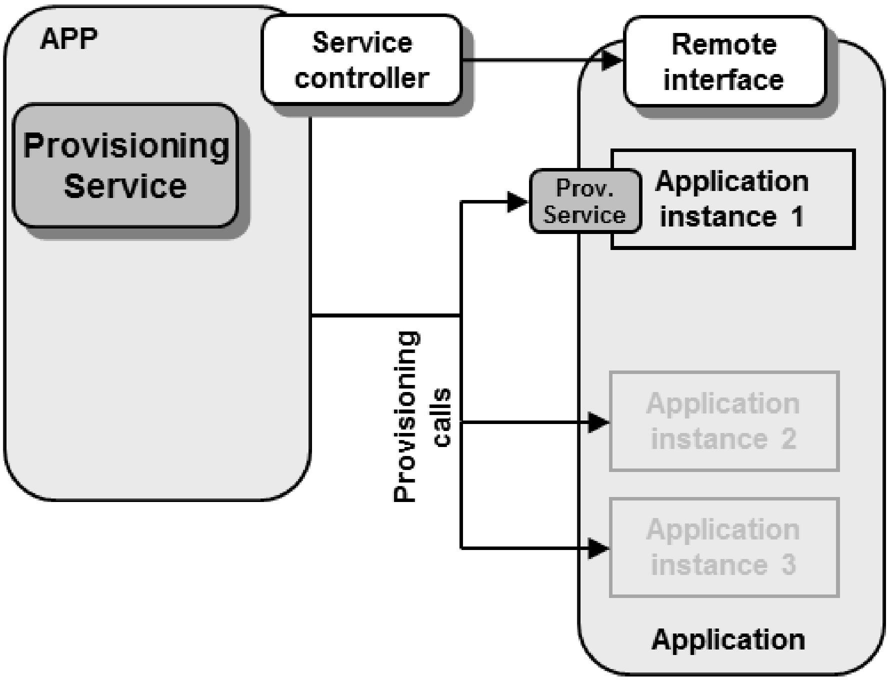
The communication paths are as follows:

- When a subscription is created, deleted, or modified in ESCM, the provisioning service of APP triggers the service controller to execute all the tasks required on the application side.
- It is recommended to divide long-running provisioning operations into several steps using the polling feature of APP. For example, when creating an application instance, the controller could immediately return the instance ID, report the instance status as "not ready", and request APP to poll the status at regular intervals. Only after the instance has actually been set up at the application side, the controller would report its status as "ready" upon the next polling by APP.
- All intermediate artefacts of the tasks are stored in the database of APP.
- When the provisioning operation for a subscription has been successfully completed, ESCM receives an appropriate message, and the status of the subscription is set to `COMPLETE`. When deleting a subscription, ESCM does not expect a response. The task is completed immediately.
- Specific tasks may require manual steps. In this case, the service controller can request the automatic execution to be paused and then send an email notification to the responsible person providing a link to continue the automatic processing. As soon as the link contained in the email is clicked, the task execution is continued.

If the provisioning or modification of an application instance fails or if there are problems in the communication between the participating systems, the corresponding subscription in ESCM remains pending. APP supports you in recovering from such problems with its instance status interface. For details, refer to *Handling Problems in the Provisioning Process* on page 22.

Provisioning by Instance

The following figure shows details of the communication paths between APP and an application, for example, an OpenStack platform.



Besides communicating with the application's general remote interface via the service controller, APP can forward the provisioning calls directly to the application instance in question.

Each provisioning operation first invokes the service controller, which in turn triggers the relevant actions in the application's general remote interface. When the controller returns with an appropriate status setting, APP calls the provisioning service of the application instance in question. This provisioning service must implement the standard `ProvisioningService` interface and work in synchronous mode.

This feature is useful if not all provisioning operations can be executed via the application's general remote interface. For example, the general remote interface of a virtual machine server allows you to provision virtual machines, but you usually cannot create or manage users within these virtual machines. To do this, you have to address the virtual machine itself.

For proper operation, the application's remote interface needs to support the following network connectivity:

- HTTPS/SOAP requests from APP to the application instances within the application.
- HTTPS/SOAP calls from an application instance to APP (in specific scenarios).

Service Operations

APP also includes the ESCM operation service interface for executing technical service operations on the integrated applications from the ESCM user interface.

Technical service operations can be used to access the resources of an application and perform administrative tasks without actually opening the application. The operations and the access information of the operation service must be specified in the technical service definition for the application.

APP passes service operation calls from the ESCM user interface directly to the service controller for the underlying application. The service controller is responsible for the execution of the operation in the application.

2 Configuring APP

APP comes with a default configuration that can be adjusted BEFORE deployment; it can still be changed once the oscm-app container has been deployed.

Prerequisites and Preparation

Consider the following before you install APP:

- *Hardware and Operating Systems* on page 10
- *Java and Ant* on page 10
- *Application Server* on page 10
- *Relational Database* on page 11
- *Mail Server* on page 12

Installation

The installation of APP consists of the following main steps:

- *Preparing the Software and Setup Utilities* on page 12
- *Configuring APP* on page 13
- *Setting up the Database* on page 14
- *Setting up the Application Server Resources* on page 15
- *Exchanging Certificates* on page 16

The descriptions assume that you are using the APP setup utilities for setting up the database and the application server resources.

Update Installation

For updating your APP installation, read the following:

- *Update Installation* on page 17

2.1 Hardware and Operating Systems

APP as a Java application does not rely on specific hardware or operating systems. It can be deployed on any platform supported by the application server and the database management system.

2.2 Java and Ant

APP requires a Java Development Kit (JDK), version 8, 64 bit. Deployment with JDK 8, Update 121 has been tested and is recommended.

In order to be able to execute the installation scripts, you need to install the Apache Ant 1.10 (or higher) open source software. In the subsequent sections, `<ANT_HOME>` is the installation directory of Apache Ant.

2.3 Application Server

APP must be deployed on an application server compatible with Java EE version 7. The following application server is supported:

Oracle GlassFish Server, version 4.1.2.

You can deploy APP on the application server you use for ESCM. Alternatively, you can use a separate application server installation.

Be aware that when operating ESCM in SAML_SP mode, every Web service client must run in a separate domain of the application server.

Note: Before installing GlassFish, make sure that the `JAVA_HOME` environment variable points to a Java Development Kit (JDK), version 8, 64 bit.

Proceed as follows:

1. Install the application server as described in its documentation, and configure it as required by your environment.

Note: Make sure that the path of the GlassFish installation directory does not contain blanks.

2. After you have configured GlassFish, make a backup copy of the GlassFish installation.
3. Make sure that GlassFish is running in a JDK 8 environment. Also, make sure that no other applications (e.g. Tomcat) are running on your GlassFish ports.

The installation of APP creates the `app-domain` domain in your application server. If required, you can change the domain name in the `glassfish.properties` file before starting the installation.

In the subsequent sections, `<GLASSFISH_HOME>` is the installation directory of GlassFish.

2.4 Relational Database

APP stores its data in a relational database. The following database management system (DBMS) is supported:

PostgreSQL, version 9.1.12.

Install the DBMS as described in its documentation.

If required, you can use a separate machine for the APP database.

Setup and Configuration

Edit the file

`<postgres_dir>/data/postgresql.conf`

as follows (`<postgres_dir>` is the PostgreSQL installation directory):

1. Set the `max_prepared_transactions` property value to `50`.
2. Set the `max_connections` property value to `210`.

   This property determines the maximum number of concurrent connections to the database server.

   Note the following: This setting is used in combination with the JDBC pool size settings for the domains on your application server. If you change the JDBC pool size, you might need to adapt the `max_connections` setting. Refer to the ESCM *Operator's Guide*, section *Tuning Performance*, for details.

3. Set the `listen_addresses` property value:

   Specify the IP addresses of all application servers on which the database server is to listen for connections from client applications. If you use the entry `'*'`, which corresponds to all available IP addresses, you must be aware of possible security holes.

4. Save the file.

If you use a server name in all configuration files instead of `localhost` during installation, edit the file

`<postgres_dir>/data/pg_hba.conf`

as follows (`<postgres_dir>` is the PostgreSQL installation directory):

1. Add the IP address of the application server that is to host APP.

   For example:

   ```
   host all all 123.123.12.1/32 md5
   ```

   Also add the application server's IPv6 address.

   For example:

   ```
   host all all fe80::cdfb:b6ed:9b38:cf17/128 md5
   ```

   There are authentication methods other than `md5`. For details, refer to the PostgreSQL documentation.

2. Save the file.

Restart your PostgreSQL server for the changes to take effect.

2.5 Mail Server

To inform users about relevant issues, APP requires a mail server in its environment. You can use any mail server that supports SMTP.

The settings for addressing the mail server are defined in the `glassfish.properties` file of APP.

2.6 Preparing the Software and Setup Utilities

The APP software and setup utilities are provided in the `oscm-app-install-pack.zip` installation package file contained in the APP integration package, `oscm-integration-app-pack.zip`. The contents of the installation package need to be made available in your environment as follows:

Extract the contents of the `oscm-app-install-pack.zip` package to a separate temporary directory on the system from where you want to install and deploy APP.

In the following sections, this directory is referred to as `<install_pack_dir>`.

After extraction, the following directories are available:

- **`databases/app_db`**

  Configuration files for setting up the database used by APP.

- **`domains/app_domain`**

  - Configuration file (`glassfish.properties`) for setting up the application server resources for the domain to which APP is to be deployed.

  - `applications`: Contains the APP application (`oscm-app.ear`)

- **`install`**

  XML files that support you in setting up the databases and application server resources for APP.

2.7 Configuring APP

The APP software requires a number of settings. These settings are provided in environment variables in Docker files when deploying the APP containers.

- **`databases/app_db`**
  - `db.properties`: Settings for the database setup and access. For details, refer to *Database Configuration Settings* on page 28.
  - `configsettings.properties`: Configuration settings for the APP services. For details, refer to *APP Configuration Settings* on page 27.

    The initial installation stores these settings in the `bssapp` 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.
  - `configsettings_controller.properties`: In this file, you can specify settings for a service controller which you want to register when installing APP.

    The initial installation stores these settings in the `bssapp` database. You can change them later using a graphical user interface.

    For details, refer to *Registering a Service Controller* on page 20.
- **`domains/app_domain`**

  The configuration settings for setting up the application server domain to which APP will be deployed are provided in the following file:

  `glassfish.properties`

Additional configuration files contained in other subdirectories are used internally and must not be changed.

You need to adapt the settings in the files above to your environment. In particular, server names, ports, paths, user IDs, and passwords require adaptation.

Proceed as follows to view and adjust the configuration settings:

1. Open each of the configuration files listed above with an editor.
2. Check the settings in each file and adapt them to your environment.
3. Save the files to their original location in `<install_pack_dir>/<subdirectory>`. For future reference, it is a good idea to create a backup of the files.

Observe the following configuration issues:

- The specified ports are suggestions and work with the default settings used in the files.
- If you install everything on the local system, use either the server name or `localhost` in all configuration files for all URLs that need to be resolved by APP.

  Do not mix the specification of server names and `localhost`.

The `APP_BASE_URL` setting in the `configsettings.properties` file must be resolved by clients. They always require that the server name be specified.

Specify the `APP_BASE_URL` setting as follows:

```
APP_BASE_URL=http://<host>:<port>/oscm-app
```

If you have changed the `glassfish.domain.portbase` setting in the `glassfish.properties` file, you must change the port here accordingly.

2.8 Setting up the Database

APP requires and stores its data in the `bssapp` PostgreSQL database.

The database is created by executing installation scripts. It is initialized with the appropriate schema and settings.

> **Note:** When initializing the database, you can also register a service controller with its settings. For details, refer to *Registering a Service Controller* on page 20.

Proceed as follows:

1. Make sure that the database server is running.
2. Open the command prompt (Windows) or a terminal session (UNIX/Linux).
3. Set the following environment variable for your current session:

   `DB_INTERPRETER`: The absolute path and name of the `psql` executable of PostgreSQL. The executable is usually located in the `bin` subdirectory of the PostgreSQL installation directory.

   Example (Unix/Linux):

   ```
   export DB_INTERPRETER="/opt/PostgreSQL/9.1/bin/psql"
   ```

   Example (Windows):

   ```
   set DB_INTERPRETER="C:\Program Files\PostgreSQL\9.1\bin\psql"
   ```

4. Create the database by executing the `build-db.xml` file in `<install_pack_dir>/install` as follows:

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

   If you set an ID or password other than `postgres` for the PostgreSQL user account (`postgres`) when installing the database management system, you have to specify the ID or password with the call to the `build-db.xml` file as follows:

   ```
   <ANT_HOME>/bin/ant -f build-db.xml initDB
      -Ddb.admin.user=<user ID> -Ddb.admin.pwd=<password>
   ```

> **Note:** It may be required to enclose the `-Ddb.admin.user=<user ID>` and `-Ddb.admin.pwd=<password>` in double or single quotes depending on the operating system.

If the setup of the database fails with errors, proceed as follows:

1. Check and correct the configuration files.

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

   ```
   <ANT_HOME>/bin/ant -f build-db.xml DROP.dbsAndUsers
   ```

3. Repeat the setup.

2.9 Setting up the Application Server Resources

APP requires specific settings and resources in the application server, such as mail settings or a data source.

Note: When creating the application server resources, you can also deploy a service controller. For details, refer to *Deploying a Service Controller* on page 21.

Proceed as follows to create the resources and make the required settings in the application server:

1. Open the command prompt (Windows) or a terminal session (UNIX/Linux).
2. Execute the `build-glassfish.xml` file in `<install_pack_dir>/install` as follows:

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

   This has the following results:

   1. **Domains and resources**:
      - The `app-domain` domain is created and started.
      - The settings and resources for APP are created in the application server.
      - APP (`oscm-app.ear`) is deployed to the `app-domain` domain.
      - A key file for encryption and decryption is generated in the location you specified in the `APP_KEY_PATH` setting in the `configsettings.properties` file. By default, this file is named `key` and located in

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

   2. **JVM options set for the `app-domain` domain** during the installation:

      ```
      -Dorg.apache.catalina.loader.WebappClassLoader.
         ENABLE_CLEAR_REFERENCES=false
      -Dfile.encoding=UTF8
      -Dsun.java2d.print.polling=false
      -Dsun.net.inetaddr.ttl=3600
      -XX:MaxMetaspaceSize=512m
      -Dproduct.name=
      ```

3. Depending on your environment, it may be required to define a proxy server for the `app-domain` domain in the **JVM Options** of the application server.

   To define a proxy server, specify the following JVM options:

   - `-Dhttps.proxyHost`
   - `-Dhttps.proxyPort`

   If authentication is required, specify the following additional settings:

   - `-Dhttps.proxyUser`
   - `-Dhttps.proxyPassword`

For all direct communication you need to bypass the proxy server. Specify the hosts which are to be addressed directly and not through the proxy server in the following setting:

- `-Dhttp.nonProxyHosts`

For example, APP must not use the configured proxy for Web service calls to ESCM:

```
-Dhttp.nonProxyHosts=localhost|127.0.0.1|myServer*
```

where `myServer` is the host on which ESCM is running.

In case several service controllers are to run in the same APP domain, and only one of them is to communicate via a proxy server, you need to exclude the target systems of the other service controllers, for example, as follows:

```
-Dhttps.proxyHost=proxy.intern.myserver.com
-Dhttps.proxyPort=8081
-Dhttp.nonProxyHosts=myServer.com|localhost|127.0.0.1|
http://10.140.18.112*|http://myServer.com:8880/templates/*
```

After having configured the proxy server, restart the `app-domain`.

4. Depending on your environment, it may be required to provide separate certificates for external and internal communication between ESCM and APP. In this case, you can use the secure HTTP listener port (default: `8082`).

If the setup of the application server domain fails with errors, proceed as follows:

1. Stop the `app-domain` domain.
2. Delete the `app-domain` domain.
3. Repeat the setup.

2.10 Exchanging Certificates

For secure communication of APP with ESCM and integrated applications, you need to exchange the corresponding certificates.

ESCM is the HTTPS server while APP is a Web service client. The Web service calls are secured with SSL. The following requirements must be met to establish a connection to ESCM:

- The ESCM server must present a valid certificate.
- The ESCM certificate must be trusted by APP.

You need to:

- Import the certificates of ESCM and the applications into the truststore of the `app-domain` application server domain of APP.
- Export the certificate of the `app-domain` domain and ask the ESCM operator to import it into the `bes-domain` application server domain of ESCM.

Proceed as follows:

1. Obtain a `.crt` file with the certificate of ESCM .
2. Obtain a `.cer` file with the certificate of each application to be integrated.
3. Import the certificates of ESCM and the applications into the truststore of the `app-domain` application server domain.

To do this, you can use, for example, the following command at the command prompt (Windows) or in a terminal session (UNIX/Linux) on the application server:

```
<AppServerJRE>/bin/keytool -import -trustcacerts -alias <alias>
   -file <filename>.crt -storepass <password> -keystore
   <GLASSFISH_HOME>/glassfish/domains/app-domain/config/cacerts.jks
```

4. Create a `.crt` file with the certificate of the `app-domain` domain in which you have deployed APP.

   The `.crt` file can be created, for example, by executing the following command at the command prompt (Windows) or in a terminal session (UNIX/Linux) on the application server:

```
<AppServerJRE>/bin/keytool -export -rfc -alias s1as
   -file ctmgapp.crt -storepass <password> -keystore
   <GLASSFISH_HOME>/glassfish/domains/app-domain/config/keystore.jks
```

5. Ask the ESCM operator to import the certificate of the `app-domain` domain into the `bes-domain` application server domain of ESCM.
6. If APP and ESCM are configured for SAML_SP authentication mode, obtain the relevant certificates from the IdP system and import them into the truststore of the `app-domain` domain.

   For example, when using Microsoft Active Directory as the IdP, you need to obtain and import the service communications and token-signing certificates.
7. Stop and restart the `app-domain` domain for the certificates to become effective.

2.11 Update Installation

Before updating your installation of APP, read the *Release Notes* of the new release. They contain information on compatibility issues, changes and enhancements, and known restrictions.

The platform operator and technology managers must make sure that the following rules are observed: The ESCM version must be higher or equal to the APP version. The APP version must be equal to the controller version.

Example: If you want to use the VMware service controller included in the V17.3 release, you must upgrade ESCM and APP to V17.3 first.

Preparing the Update

Before you start with the update installation, carry out the following steps:

1. Make sure that all provisioning operations are complete. Follow the steps as described in *Handling Problems in the Provisioning Process* on page 22.
2. If you are upgrading from V17.0:

   In the `app-domain` application server domain, disable or undeploy the following application:

   `oscm-app`
3. If you are upgrading from V17.0:

   Check for `.glassfishStaleFiles` files in the `app-domain` domain. If there are any, delete them. The files are located in

   `app-domain/applications/<application name>/.glassfishStaleFiles`

   For example:

   `app-domain/applications/oscm-app/.glassfishStaleFiles`

4. Set the following environment variable for your current session:

   `DB_INTERPRETER`: The absolute path and name of the `psql` executable of PostgreSQL. The executable is usually located in the `bin` subdirectory of the PostgreSQL installation directory.

   Example:

   ```
   export DB_INTERPRETER="/opt/PostgreSQL/9.1/bin/psql"
   ```

5. Create a backup of the key file required for the encryption and decryption of service parameters with data type `PWD` and custom attributes marked for encryption. By default, the file is named `key` and located in the following directory:

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

Updating the Database

Proceed with updating the database as follows:

1. Check whether the file

   `postgresql-9.4-1206-jdbc42.jar`

   is contained in the following directories of the application server:

   - `lib` directory of the `app-domain` domain
   - `<GLASSFISH_HOME>/mq/lib/ext`

   If it is not, copy the file from the `<install_pack_dir>/install/lib` directory to the location where it is missing.

2. Create a backup of the `bssapp` database using the standard PostgreSQL commands. The database backup must be compatible with PostgreSQL 9.1.12.
3. Update the following configuration files so that the settings match your current installation:

   - `db.properties`
   - `configsettings.properties`
   - `configsettings_controller.properties`

   Particularly take care of settings which have been introduced or changed with the new release to which you are upgrading.

4. Update the schema and configuration settings of the `bssapp` database by executing the `build-db.xml` file in `<install_pack_dir>/install` as follows:

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

Note: Make sure that Ant runs in a Java 8 runtime environment when calling the `build-db.xml` file.

Updating the Application Server

After you have executed the preparation steps:

1. If you are upgrading from a release prior to V17.0:

   Copy the key file of which you have created a backup copy into the following directory of the extracted installation package:

   `oscm-app-install-pack\domains\app_domain\installer`

Make sure that the key file is named `key`. The installation script will then copy the key file to its default location in the updated version of the application server:

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

Note: For the update installation, make sure that the default path and name are specified in the `configsettings.properties` file (`APP_KEY_PATH="./key"`). The installation script expects this default and does not look up the database.

2. Redeploy or deploy `oscm-app` in the `app-domain` domain.
3. In the **JVM Options** of the application server:
   a. Set the following setting to `false`:

      `-Dorg.apache.catalina.loader.WebappClassLoader.ENABLE_CLEAR_REFERENCES=false`

   b. Delete the following setting:

      `-XX:MaxPermSize`

   c. Add the following setting:

      `XX:MaxMetaspaceSize=512m`

4. If you are upgrading from a release prior to V17.0:

   Export the certificate of the `app-domain` and import it into the `bes-domain` application server domain of ESCM.

5. Restart the `app-domain` domain.

Updating the Configuration for SAML_SP Mode

If you are running ESCM and the APP software in SAML_SP mode, and if the IdP metadata of your SSO environment have changed, you need to update your WSIT files accordingly:

1. Extract the `OSCM-wsit.jar` file into a separate directory.

   The `.jar` file is located in

   `<GLASSFISH_HOME>/domains/app_domain/lib`

   The `OSCM-wsit.jar` file contains the following files:

   `wsit-client.xml`

   `STSService.xml`

2. Adapt the `.xml` files as required by your environment. For details, refer to *Configuring APP* on page 13.
3. Recreate the `OSCM-wsit.jar` file with the modified contents.
4. Stop the `app-domain` domain.
5. Copy the adapted `OSCM-wsit.jar` to the following directory:

   `<GLASSFISH_HOME>/domains/app_domain/lib`

6. Restart the `app-domain` domain.

Note: If, for some reason, you recreate the `app-domain` domain, you also need to recreate the `OSCM-wsit.jar` in the `<GLASSFISH_HOME>/domains/app_domain/lib` directory.

3 Implementing a Service Controller

For each application to be integrated with ESCM by means of APP, you need to implement a service controller using the APP API.

The API includes two interfaces for implementing a service controller, the `APPlatformController` and the `APPlatformService` interface. The `APPlatformController` interface provides the methods required to access an application which is to be integrated with the subscription management of ESCM. The `APPlatformService` interface provides the methods by which service controllers implemented in APP can access common APP utilities. A detailed documentation of the API is provided as Javadoc with the APP integration package (`oscm-integration-app-pack.zip`).

The service controller sample included in the integration package is a fully functional implementation of a service controller. The following packages also include ready-to-use service controllers:

- `oscm-aws-install-pack.zip`:

  Can be used for integrating the Amazon Elastic Compute Cloud Web service with ESCM.

- `oscm-openstack-install-pack.zip`:

  Can be used for integrating OpenStack services with ESCM.

To implement a service controller, you need to perform the following steps:

- Identify the API of the application which is to be integrated.
- Define the data which are to be persisted by APP.

  This includes all settings required during the provisioning process (e.g. configuration, target environment, current status etc.). APP can store arbitrary key/value pairs and provide them to the service controller in the subsequent steps.

- Identify the required steps for each of the provisioning operations, and split them into a synchronous and an asynchronous phase.

  The synchronous phase should not execute long-running commands or communicate with remote systems. Instead, it should perform plausibility checks and parameter validation, for example, determine the necessary actions. All actual provisioning tasks are to be handled in the asynchronous phase by the polling feature. For details on the polling feature, refer to the Javadoc of the APP API.

- Implement all the identified provisioning steps using the appropriate API commands of the application.
- Optionally, implement some kind of dispatcher that checks the current status and determines the next task to be executed depending on the current provisioning status and the target configuration.

  The dispatcher can also schedule manual execution steps by switching off the automatic polling by APP. Finally, it signals the completion of the steps to APP.

- Carefully consider the error handling: Decide whether abortion of the process is necessary. In most cases, you can suspend the process and trigger manual steps in error situations.
- Optionally, identify and implement service operations that users can execute from the ESCM user interface.

3.1 Registering a Service Controller

Each service controller needs to be registered with APP. Registration means entering the controller with its ID and its responsible organization in the `configurationsetting` table of the `bssapp` database.

The organization responsible for a service controller is usually your own organization. However, it may also be another organization with the technology provider role registered with ESCM. In the controller registration, you specify the organization ID.

You can register a service controller in one of the following ways:

- **Using the Web interface of APP**:
  1. In a Web browser, access the base URL of APP, for example, `https://127.0.0.1:8880/oscm-app`.
  2. Log in with the ID and password of the user you specified for `BSS_USER_KEY` in the APP configuration settings (see *configuration settings* or as another administrator of the same organization.
  3. Specify the IDs of the service controller and of the responsible organization.
  4. Click **Save Configuration** to save the settings.
- **Using a standard table editor or SQL statements**:

  Connect to the database with the user ID and password specified for the database access in the `db.properties` file when configuring APP (see *configuring APP*. In the `configurationsetting` table, create the corresponding entries. The SQL statement to do this would be:

  ```
  INSERT INTO configurationsetting (controllerid, settingkey, settingvalue)
  VALUES ('<controllerID>', 'BSS_ORGANIZATION_ID', '<organizationID>';
  ```

  Example:

  ```
  INSERT INTO configurationsetting (controllerid, settingkey, settingvalue)
  VALUES ('mycontroller', 'BSS_ORGANIZATION_ID', 'baaf0c7d';
  ```

3.2 Deploying a Service Controller

After having implemented a service controller, you need to deploy it. You usually do this in the `app-domain` domain on the application server where you have installed APP.

You can deploy a service controller in the following way using the standard utilities of the application server:

1. You need an `.ear` or `.war` file with the code and resources of the service controller.
2. Deploy the `.ear` or `.war` file of the service controller to the `app-domain` domain.

4 Administrating APP

The following sections describe administration tasks you may need to perform in your role as an operator of APP.

4.1 Handling Problems in the Provisioning Process

If the provisioning or modification of an application instance fails or if there are problems in the communication between the participating systems, the corresponding subscription in ESCM remains pending.

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

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

1. Work as a technology manager of the technology provider organization which is responsible for the relevant 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/?ess.openstack
   ```

   `<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/?ess.openstack` 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 provisioning, modification, or deprovisioning operation.
4. In the **Action** column, select the action for APP 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 the application does not respond.
   - `UNLOCK` - to remove the lock for an application instance in APP.
   - `DELETE` - to terminate the subscription in ESCM and remove the instance in APP, but keep the actual application instance for later use. For this action, the service manager role is required in addition.
   - `DEPROVISION` - to terminate the subscription in ESCM, remove the instance in APP, and deprovision the application instance. For this action, the service manager role is required in addition.
   - `ABORT_PENDING` - to abort a pending provisioning or modification operation in ESCM. ESCM is notified to roll back the changes made for the subscription and return it to its previous state. In the application, no actions are carried out.

  - `COMPLETE_PENDING` - to complete a pending provisioning or modification operation 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.
- The **Run with timer** column indicates whether the timer for the interval at which APP polls the status of instances is running..

4.2 Handling Communication Problems Between APP and ESCM

When the communication between APP and ESCM is no longer possible, for example, because ESCM 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 ESCM operator to make sure that ESCM 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 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 and organization defined by the platform operator in the `BSS_USER_ID` and `BSS_USER_PWD` configuration settings (default: `administrator`, `admin123`).

   A message is shown that APP has been suspended due to a communication problem with ESCM.

3. Click **Restart**.

   APP is restarted instantly. In the APP database, the `APP_SUSPEND` key is set to `false`.

4.3 Backup and Recovery

The APP integration software does not offer integrated backup and recovery mechanisms. Use the standard file system, application server, and database mechanisms instead.

Backup

It is recommended that you create a regular backup of the following data according to the general guidelines of your organization:

- Database (`bssapp`). The frequency of database backups depends on the amount of changes and 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:

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

  The location of this file can be changed using the `APP_KEY_PATH` configuration setting.
- Certificates contained in the truststore of the `app-domain` domain (`cacerts.jks` file).
- Configuration files.

Note: When preparing for an update installation of the current APP installation, always create a backup of the data mentioned above.

Recovery

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

1. Restore the `bssapp` database from the backup using the relevant PostgreSQL commands.
2. 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 `APP_KEY_PATH` configuration setting. By default, this file is to be named `key` and located in:

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

   If the file is missing, copy it from your backup to the correct location.
3. Stop the `app-domain` domain of the application server.
4. Restore the certificate truststore of the `app-domain` domain (`cacerts.jks` file) from the backup.
5. Start the `app-domain` domain.

4.4 Updating Configuration Settings

The APP software requires a number of settings for configuring its container environment. These settings are provided as environment variables in Docker files when deploying the containers. Before the deployment, the platform operator adapted the initial settings to your environment, in particular server names, ports, paths, and user IDs.

The initial configuration settings are described in the *Operator's Guide*.

Some of the initial settings can be changed after the deployment. Proceed as described in the following sections.

To update the configuration settings for APP, use 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 administrator of the technology provider organization specified by the platform operator.
3. Edit the settings as required.
4. Click **Save Configuration** to save the settings.

To update the configuration settings for a service controller integrated with APP:

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

   The access URL has the following format:

   `https://<hostname.fqdn>:<port>/<controller-id>`

   `<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`), `<controller-id>` is the default context root of the service controller (`oscm-app-aws` or `oscm-app-openstack`).
2. Log in with the ID and password of the user specified in the configuration settings for the service controller (`CONTROLLER_USER_NAME` and `CONTROLLER_USER_PASS` settings during the deployment) or as another administrator registered for the same organization.
3. Enter the required settings.
4. Save the settings.

If you want to **change the technology provider** organization responsible for the service controller, you can use the Web interface of APP:

1. In a Web browser, access the base URL of APP, for example:
   `https://127.0.0.1:8881/oscm-app`
2. Log in with the ID and password of the user specified for `CONTROLLER_USER_KEY` in the configuration settings for APP or as another administrator of the same organization.
3. Specify the technology provider organization for the service controller, .
4. Save the settings.
5. Make sure that the configuration settings for the 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.

Note: ### issue https://github.com/servicecatalog/oscm/issues/46

4.5 Adapting the Log Configuration

APP records information and problems such as connection issues in the following log file on the application server:

`<GLASSFISH_HOME>/domains/<DOMAIN_NAME>/logs/app-core.log`

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 file. The following configuration file is of relevance:

`<GLASSFISH_HOME>/domains/<DOMAIN_NAME>/config/log4j.app.core.properties`

Proceed as follows to adapt the log level:

1. Open the 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:
   - `ERROR` - designates error events that might still allow the application to continue running.
   - `WARN` - designates potentially harmful situations.
   - `INFO` - designates informational messages that highlight the progress of the application at coarse-grained level.
   - `DEBUG` - designates fine-grained informational events that are most useful to debug an application.

   Example:

   ```
   log4j.logger.org.oscm.app=INFO
   ```

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

Appendix A: Configuration Settings

The configuration settings for APP are provided in the following files in subfolders of the directory to which you extracted the `oscm-app-install-pack.zip` file contained in the `oscm-integration-app-pack.zip` package (`<install_pack_dir>`):

- `databases/app_db/db.properties`
- `domains/app_domain/glassfish.properties`
- `databases/app_db/configsettings.properties`

The settings are described in detail in

- *Database Configuration Settings* on page 28
- *APP Configuration Settings* on page 27

A.1 APP Configuration Settings

This section describes the configuration settings that are written to the `bssapp` database during deployment.

APP_BASE_URL

`APP_BASE_URL=https://<server>:<port>/oscm-app`

The URL used to access APP. The default port is `8881`.

APP_TIMER_INTERVAL

`APP_TIMER_INTERVAL=15000`

The interval (in milliseconds) at which APP polls the status of instances. This setting cannot be changed.

APP_TIMER_REFRESH_SUBSCRIPTIONS

`APP_TIMER_REFRESH_SUBSCRIPTIONS=86400000` (once a day)

The interval (in milliseconds) at which APP polls the status of instances and updates the number of virtual machines (VMs) provisioned for subscriptions to IaaS services, for example in OpenStack. The number is updated only in case the technical service definition specifies a `VMS_NUMBER` parameter.

This setting cannot be changed.

APP_ADMIN_MAIL_ADDRESS

`APP_ADMIN_MAIL_ADDRESS=admin@example.com`

The email address to which email notifications are sent.

BSS_USER_KEY

`BSS_USER_KEY=<userKey>`

The user key for accessing ESCM.

You receive this key with the confirmation email for your user account.

The user specified here must have the administrator role for your organization in ESCM. The user account is used for carrying out actions on behalf of APP in ESCM. In addition, the user is allowed to register service controllers in APP.

BSS_USER_ID

```
BSS_USER_ID=<userId>
```

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

BSS_USER_PWD

```
BSS_USER_PWD=_crypt:<password>
```

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

The password is encrypted when it is stored in the database.

A.2 Database Configuration Settings

The `db.properties` file located in `<install_pack_dir>/databases/app_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: `bssapp`

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`