

# Fusion Middleware Provisioning with Cloud Control

Andreas Chatziantoniou, Foxglove-IT BV  & Ingo Reisky, OPITZ CONSULTING

Oracle Enterprise Manager Cloud Control 12c (EMCC12c) is an excellent tool for monitoring Oracle environments, especially those containing Fusion Middleware (FMW) components. Setting up of such environments in the real world often uses bespoke scripting solutions or just Oracle Configuration Tools (config.sh). Cloud Control offers possibilities in the provisioning arena to assist in creating and administrating Oracle software and FMW domains. The added value is obviously the fact that such an environment can be provided consistently and repeatedly. This is completely embedded in the framework of Cloud Control employing concepts of privileges and roles, but also offering reporting capabilities and configuration auditing.

In this article we would like to demonstrate how such a solution can be designed and implemented. Our main focus lies on the quality of the provisioned environments because variations and discrepancies in the various project phases (DEV, TST, INT, PRD) are avoided.

## Introduction

An Oracle Fusion Middleware (FMW) environment can relatively easy evolve into a very complex entity. Often more than 100 (sub-) activities need to be carried out just to set up one environment only once!

Very soon such provisioning needs to be executed more than once and, in projects with several lifecycles (DEV, TST, INT, PRD), the problem arises to ensure and guarantee that a provisioning is consistent with its predecessor and successor. Furthermore, each provisioning can be very time consuming.

Additionally, each provisioning operation (possibly) needs adaptations according to the project phase, e.g. a production environment has RAC databases while previous stages used single instances, and so on.

Many organisations employ a script-based approach (often its historic roots are buried in time) or execute the tasks manually. Here a number of well-known issues arise: traceability and governance (Who built the INT environment? Which changes were carried out in DEV?).

## What is provisioning anyway?

Provisioning knows a number of aspects that typically cross the boundaries of classical system management teams and realms. Especially in Oracle FMW environments various components are to be addressed:

- Hardware
- Network
- Storage
- OS
- Oracle Software (DB, FMW)

After providing these components they need additional attention - such as a configuration effort, setup of database instances and, last but not least, the application that brings another level of complexity to the party.

The main focus of this article is on the Oracle FMW components. The reason for this is the fact that in each organisation different underlying infrastructures and platforms are present. This does not however rule out an extension of the described concepts of this article towards these components.

What does provisioning of FMW components mean in particular? We need to face the reality and accept that the day to day practice depends on an organisation's choice of components. A classic J2EE application serving the needs of an individual department needs less effort and attention than a highly-available enterprise application that crosses various

network zones, deploys the SOA Suite and Service Bus and needs to be secured with certificates. Furthermore, some Oracle FMW products are not yet integrated into the provisioning mechanisms, which leads to an extended usage of script-based solutions.

However, this does not imply to abandon the idea of using EMCC12c for provisioning. The more components of a system architecture that are provisioned in an integrated way using EMCC12c the better the overall quality of the complete solution will become. Additionally, the number of outages decreases. The necessary steps for quality assurance benefit from this approach because the huge effort to carry out such tasks can be reduced or eliminated. A system which was checked OK once in a previous lifecycle requires less checks when everybody knows that it is an (almost) exact replica of its predecessor.

### Steps for Oracle components

Starting as always one needs to provide and install the Oracle software. Possibly a technical database (Metadata Repository) is required to harbour configuration data and even actual run-time data. Finally, the FMW software is configured (creation of a domain and possibly an extension of domain templates) and then the application is deployed.

### Provisioning with EM Cloud Control 12c

The provisioning process, as stated before, should be based as much as possible on available Oracle tools. Using EM Cloud Control 12c (EMCC12c) is the logical choice. In addition to provisioning, there is the integration of governance aspects (who is allowed to execute this provisioning operation?) and the subsequent monitoring of the components, which is mandatory for a seamless system administration. The new FMW targets are discovered in a post-provisioning step in the EMCC12c. This offers such goodies as an automated QA of EMCC12c standard metrics (e.g. application deployment in status failed, data source in status suspended) as an integral part of the provisioning workflow. If required, such an automated QA can be expanded individually by using "Configuration Extensions". Likewise the "Configuration Management" in EMCC12c offers afterwards a quick insight into configuration changes after the provisioning; which changes happened after completion of the provisioning workflow? (Many configuration parameter of an environment are collected by EMCC12c every 24 hours by default).

Depending on your version of OEM FMW plugin, the following components can be provisioned with the EMCC12c: WebLogic Server, Java EE applications, Coherence nodes and clusters, SOA artifacts, Oracle Service Bus resources, Oracle BPEL processes and Oracle Application Server.

This shows that the complete width of the Oracle FMW components is not yet covered (see also the official "Cloud Control Lifecycle Guide" for updates).

The complete process is embedded in the framework of the Oracle Enterprise Manager, so that the creation of a software library in the EMCC12c, granting of privileges and roles to users,

the creation of provisioning profiles and the creation/extension of WebLogic domains need to be executed in that particular order. The transition of lifecycles (DEV, TST, INT, PRD) calls for a decreased list of entitled users (larger user group or self-service in DEV <--> Super Administrator in PRD).

Application of provisioning profiles to target systems requires an OEM agent on those systems. Therefore the target systems need to be "discovered" by the OEM as "Host Targets" before provisioning takes place.

How is a provisioning of the FMW components executed? Globally this can be divided into two areas: *Profiles* and *Deployment Procedures*. Profiles describe the Oracle software (installation media), the Oracle Home and the WebLogic domain. A profile is created in the Profile Wizard and will then be stored in the Software Library. The steps depend primarily on the desired task, but typically it is a combination of the following activities:

- Creation of an Installation Media Profile
- Creation of a Middleware Home Profile
- Creation of a WebLogic Domain Profile
- Execute the Provisioning Profile

The *Deployment Procedures* describe the provisioning of the FMW domains and Oracle Homes, scaling out of SOA, Oracle Service Bus, and WebLogic Server domains, the deployment of J2EE applications up to the provisioning of BPEL processes and OSB resources. Adapted procedures (Perl, WLST or shell scripts) can be stored in Cloud Control (inside the EMCC12c Software Library) and executed from the EMCC12c Console (GUI) as well as by EMCLI (command line).

### Execution of a Provisioning Profile

Comparably to the set up without EMCC12c, a number of prerequisites need to be met before the successful provisioning can happen. For example, the target machines need to be known as targets in Cloud Control. Numerous read- and write permissions on those machines are to be set for the user executing the commands. Furthermore, the ports for the Administration Server, Managed Server and Node Manager are not to be occupied by other processes.

In an interactive session, a defined sequence of dialogs is to be executed to provide the necessary data.

The first activity is to call the FMW Provisioning Page. From here a *Profile* or a *Deployment Procedure* is initiated. This is followed by the *Middleware Target Home Page*. The targets are selected here (e.g. a WebLogic domain). This is then cloned from an existing WebLogic domain or from a Middleware home. Further dialogs ask for *Source Environment Details*, *Destination Environment Details*, *Domain Configuration Details*, passwords and a start time.

Such an interactive approach obviously suits only the preparation of a provisioning. Repeated execution of such provisionings will therefore use the "EMCLI" (Enterprise Manager

Command-Line Interface).

The following example shows the creation of a profile:

```
emcli create_fmws_domain_profile
-name="Profile Name"
-ref_target="Reference Target Name"
[-description="Profile Description"]
[-oh_cred="Oracle Home Owner
Credentials"]
[-includeOh]
[-schedule=start_time:yyyy/MM/dd HH:mm];
```

Selecting the *Profiles* or the *Deployments Procedure* as starting point requires the GUID of the selection:

```
emcli get_procedures
```

This command provides a unique global ID (GUID) of the existing *Profiles/Deployment Procedures*. These have been created manually in the EMCC12c Console. With the GUID a *Properties File Template* is created.

```
emcli describe_procedure_input -procedure=<proc_guid> -
name=<proc_name>
```

The output can then be redirected into a file and will be edited. Such a *Properties File Template* contains a large number of parameters - which can be adapted for the more specific needs and parameters that were not set by the interactive dialog.

```
CREATE_DOMAIN=true
DEST_ADMIN_HOST.0.ADMIN_SQL_HOME=
DEST_ADMIN_HOST.0.DATASOURCE_PROPERTY_FILE_DIR=
DEST_ADMIN_HOST.0.DATASOURCE_PROPERTY_FILE_NAME=
DEST_ADMIN_HOST.0.DOMAIN_HOME_DEST_ADMIN_HOST=/opt/oracle/
domains/MY_DOMAIN
DEST_ADMIN_HOST.0.DOMAIN_NAME_DEST_ADMIN_HOST=MY_DOMAIN
...
DEST_ADMIN_HOST.0.PORT_DETAILS_DEST_ADMIN_HOST=7001:Listen
Port,7002:SSL Listen Port
...
DEST_ADMIN_HOST.0.WLS_PASSWORD_DEST_ADMIN_HOST=welcome1
DEST_ADMIN_HOST.0.WLS_USERNAME_DEST_ADMIN_HOST=weblogic
...
DEST_ADMIN_HOST.0.name=as01.mydomain.com
...
DEST_FMWS_HOST.0.name=srv01.mydomain.com
...
DEST_MANAGED_SERVERS.0.NM_LISTEN_PORT=5556
```

Handing over the *Properties File Template* to the provisioning tool is achieved by the following command:

```
emcli submit_procedure -input_file=data:<input_properties_file>
-procedure=<proc_guid> -instance_name=<optnl_DP_Instance_Name>
```

This will set up the environment like any interactive session of the EMCC12c Console would do.

The above command "**emcli submit\_procedure ...**" delivers (when successful) a so-called Instance ID (IID), which can be

used to check the status of the set-up activities (comparable to the overview present in the "Procedure Activity" of the EMCC12c Console (GUI)):

```
emcli get_instance_status -instance=<IID> -details
-showJobOutput
```

The above mentioned properties-file can also serve as input for different environments. This makes transition between the different lifecycles easier. If, for example, an environment contains a larger number of hosts, the file can be simply edited and adapted by multiplying the number of blocks describing a target host.

The properties file can also be saved or updated in the EMCC12c Procedure Library:

```
emcli save_procedure_input -name=<...> -procedure=<...> -input_
file=<MYFILE>
emcli update_procedure_input -name=<...> -procedure=<...> -input_
file=<MYFILE>
```

## Deployment

Provisioning the environment (platform) is a big step forward. Now the application should also be provided automatically. Such a deployment can take different forms depending on the used FMW components. Take the example of a WebLogic environment with a J2EE application. This can be compared to the **emcli get\_procedures** approach of the provisioning. The main difference is the usage of Procedures that employ a *J2EE Application Provisioning Template*. Even in this case a parameter file is generated with **emcli describe\_procedure\_input**. The structure of this file is comparable to the one seen before.

```
deployMode=true
domains.0.continueOnDeployError=true
domains.0.domainName=MY_DOMAIN
domains.0.javaeeApps.0.appName=MY_APPLICATION
...
domains.0.javaeeApps.0.archivePath=/tmp/MY_APPLICATION.ear
domains.0.javaeeApps.0.copyComponents=true
domains.0.javaeeApps.0.copiedComponentsList.0.componentPath=sw
lib/MY_APPLICATION
...
domains.0.javaeeApps.0.copiedComponentsList.0.name=srv01.
mydomain.com
...
domains.0.javaeeApps.0.deplMode=Deploy
...
domains.0.javaeeApps.0.stageMode=DEFAULT
domains.0.javaeeApps.0.startMode=full
domains.0.javaeeApps.0.targets="MS_001_APPLICATIONS"
domains.0.javaeeApps.0.type=host
domains.0.javaeeApps.0.wlsAdminURL=t3s://as01.mydomain.com:7002
...
undeployMode=false
```

Modifying manually the properties-file happens as described above when the application should be deployed in the various lifecycle environments (more Managed Server, different ports, etc.).

## Conclusion

Employing EMCC12c for the provisioning it most certainly useful when addressing the ever-growing complexity of environments. One should keep in mind that the preparation is not trivial as it “executes” the manual provisioning steps in order to automate them. Furthermore, a number of FMW products is not yet supported. However when regarding FMW products or J2EE applications that need for example a database, the advantages become obvious. Although this article has not shown the database and schema provisioning, it is very similar to the concepts depicted above. Combining the FMW and database activities in one Provisioning Workflow will take a large burden off the shoulders of system administrators as a large number of repetitive tasks can be automated with EMCC12c. ■



## ABOUT THE AUTHORS

### Andreas Chatziantoniou

Fusion Middleware Specialist, Foxglove-IT

Andreas (Diplom-Informatiker TU Berlin, OCA, RHCT, Oracle ACE, Oracle SOA Suite specialised, Oracle Enterprise Manager specialised) is a freelance Oracle Fusion Middleware specialist with his own company Foxglove-IT. His focus is the technology area of all Oracle Fusion Middleware products (except BI), WebLogic Server, Enterprise Manager and the Oracle Database. Years of hands-on experience in literally 100s of environments ensure a backpack full of tricks to get systems running. Andreas is a frequent speaker at various Oracle events.

**Blog:** [achatzia.blogspot.com](http://achatzia.blogspot.com)



[www.linkedin.com/in/andreaschatziantoniou](http://www.linkedin.com/in/andreaschatziantoniou)



@MagicChatzi



### Ingo Reisky

Senior Consultant, OPITZ CONSULTING Deutschland GmbH

Ingo (Diplom-Wirtschaftsingenieur TH Karlsruhe, OCP, RHCT, LPIC-2) is a Senior Consultant for the Infrastructure Consulting division of OPITZ CONSULTING Deutschland GmbH. He is specialised for the Oracle Fusion Middleware, with a focus on WebLogic Server, SOA Suite and Enterprise Manager. Broad practical experience and years of administration of very complex system environments (databases, application servers, system administration of Linux, as well as large VM farms) are the foundation of his knowledge. Alongside this he works as trainer for Oracle University delivering WebLogic Server Administration courses.

**Blog:** [thecattlecrew.wordpress.com](http://thecattlecrew.wordpress.com)



[www.linkedin.com/in/ireisky](http://www.linkedin.com/in/ireisky)



@IngoReisky