**Import and Export MDS from Enterprise Manager and other tools as well**

1. Import and Export Metadata From EnterpriseManager :- Exporting a SOA composite via Enterprise Manager.

2. Validating the MDS information using Enterprise Manager.

3. Exporting the whole MDS via Enterprise Manager.

4. Reviewing the export contents of a composite on the file system.

5. Reviewing the export contents of an MDS on the file system.

6. Browsing the MDS via Oracle JDeveloper.

7. Importing a composite into JDeveloper for review.

8. Exporting from the MDS via command line (WLST).

In the following examples, it is assumed you have deployed a SOA composite to the SOA midtier either from JDeveloper or you are reviewing an Oracle Fusion Applications installation for any pillar family (e.g. HVM, CRM, SCM). Also, the administrative server and SOA domain must be fully started.

Steps 6+7 require that JDeveloper is installed with the Oracle SOA Suite extension (JDEV: help / check for updates / search for SOA). The latest JDeveloper can be downloaded fromJDeveloper Download. Ensure the JDeveloper version matches the SOA Suite version (see Step 2 below for version information).

Obtain the username/password logins for Enterprise Manager and the MDS schemas used by the SOA-INFRA layer.

MDS Types

An MDS can be file based on a file system or within an RDBMS. For production SOA environments the database option should be used. Similarly, all Fusion Applications installations use a database based MDS. This note focuses on the database MDS, but the concepts also apply to file based MDS stores.

Within the MDS there are component configurations, deployed applications and SOA composites. These MDS schemas are loaded at Weblogic managed server start-up time and then database connections are established from the weblogic jdbc connection pools.

Schema's

On a standard SOA Suite install, the schemas used by the SOA Suite are loaded at creation time by the RCU utility prior to Weblogic domain creation. Typically their names are <prefix>\_soainfra, where <prefix> is user defined eg: DEV\_SOAINFRA, TEST\_SOAINFRA, PROD\_SOAINFRA. The Fusion Applications Framework seeds the schemas during installation time within Fusion Applications. They are named <FAMILY>\_FUSION\_SOAINFRA; eg: CRM\_FUSION\_SOAINFRA, FSCM\_FUSION\_ORABAM, HCM\_FUSION\_SOAINFRA, SCM\_FUSION\_MDS\_SOA, SCM\_FUSION\_\* etc.

The \*\_SOAINFRA schema is used to manage the SOA Suite instance information such as Rules and Workflows. The MDS schemas are where the SOA composites are deployed and stored within the RDBMS. Like a call to a remote webservice, SOA composites can make calls to suitable referenced composites and applications within the MDS layer for that Oracle Fusion Middleware deployment. As a Fusion Applications installation will prefix the SOA MDS and be named like CRM\_FUSION\_MDS\_SOA - a normal SOA Suite install will end in \*MDS\_SOA eg: TEST\_MDS\_SOA, PROD\_MDS\_SOA etc.

SOA composites are checked and loaded at Weblogic SOA domain start-up. Modifications to some component parameters are also stored as xml files.

1. Exporting a SOA composite via Enterprise Manager

First , connect and login to the Enterprise Manager on the Admin server for that SOA domain at http://<host><domain>:<port>/em.

• Once connected , open the FARM\_domain link on the left and drill down to SOA. Open the SOA node by clicking on the plus icons until you can see the soa-infra node. Everything underneath this node is read from the MDS and contains all of the deployed SOA composites.

• Expand all of the nodes under soa-infra (you may have multiple soa\_servers) and see the composite names followed by their version number; eg HelloWorld(1.0).

• 'Right click' on the desired SOA composite for export and then select EXPORT from the menu.

• On the right is the Export Composite screen, which contains 4 options and allows the modification of the exported filename. Typically you would accept the default, Option 1: Export with all post-deploy changes. This option gives all design time and also runtime changes and saves as one file. You can rename the file, or accept the default which will generate a file named like sca\_<compositeName>\_rev<versionNumber>.jar - eg: sca\_HelloWorld\_rev1.0.jar.

• Click the EXPORT button. This choice will retrieve the SOA composite from the MDS with all the files that match the export option and then prompt to save the file locally.

• Option 4 below reviews the contents of your export.

2. Validating the MDS information using Enterprise Manager

When debugging an issue, users need to know what database and / or schema and version of the SOA Suite is used. This information can be validated directly from Enterprise Manager.

Connect and login again to Enterprise Manager at http://<host><domain>:<port>/em.

• Once connected, expand the +Metadata Repositories node to view a list of MDS stores in use. There are MDS stores like mds-owsm and mds-soa .

• Click on the mds-soa name and the page to the right will populate with information about that MetaData Store. Validate if the <XXprefix>\_soa-infra is using that MDS. Click on mds-soa to see a list of Repository Partitions like obpm, owsm and soa-infra. The SOA-INFRA application (used in standalone SOA Suite and underlying SOA Suites to Fusion Applications) will use the mds-soa MDS.

• Validate the MDS by clicking on the dropdown list under mds-soa in the top left corner of the right hand page (in the blue bar). The path is HOME, ADMINISTRATION AND GENERAL INFORMATION options.

• Click GENERAL INFORMATION to see a small popup which confirms if this MDS is file or database based (Fusion Applications and SOA production installs will need to be database based). Notice the database version , the database type (XE in development or 11.x in production etc), the <prefix>\_MDS schema used by the RDBMS version ,and the SOA Suite schema version (e.g. 11.1.1.x and above).

• The SOA Suite is a J2EE application within an Oracle Weblogic domain and JVM container. Validate the JDBC connection used by this MDS via the Weblogic console itself at http://<host><domain>:<port>

/console --> <Domain> // Services // JDBC // Data Sources. From that list click into mds-soa and check the connection pool to validate the connection from the Weblogic layer to the remote RDBMS for connection issues. The connection pool schema should directly tie back to the MDS information from within /em.

3. Exporting the whole MDS via Enterprise Manager

Connect and login again as per step 1 to /em and expand down to the SOA / SOA-INFRA node of the midtier.

• Instead of clicking into a particular composite as in step 1 , export the whole MDS by right clicking on the SOA-INFRA node and drilling down to ADMINISTRATION and then MDS CONFIGURATION.

• Properties within the page will update on the right. See the Repository name used by the SOA-INFRA (as described in step 2 above) and the options to Export, Import and Purge.

• Do not click Import or Purge at this time. Click the Export button to be prompted on where and what to name the export filename.jar/.zip.

NOTE\_1: Exporting a whole MDS means not just exporting a single SOA composite or config file, but exporting everything within that MDS. Depending on the installation type or Fusion Applications pillar this could generate a very large archive filename including all the Applications, SOA Composites and component configuration files. A domain with a few SOA Composites might save as a 10MB sized file, while a default install of an SCM Fusion Applications domain may save as a 25-45MB sized file. Ensure there is ample disk available to export the MDS.

NOTE\_2: The MDS archive expects to be saved to a file system location on the Weblogic managed server itself. The browser will not prompt to browse for a file system on a local PC. A suitable name to give the export might be /tmp/scm\_dev\_soamds.zip.

• Once the MDS is exported to the server file system , ftp or move it to another location for further review.

4. Reviewing the export contents of a composite on the file system

Review the exported SOA composite with an archive program which can extract and expand the exported composite.jar/.zip on the file system such as 7zip, or Winzip. Also, within a regular java installation run <javaHome/bin/jar -xvf <filename.jar> to extract the contents of a .jar file.

• Open and extract the soa composite to the local filesystem.

• A SOA developer or administrator can provide files for an SR or review code to validate a recent change, or see the config and code files that make up a composite. For example;

<javahome\_/jdk1.6.0\_20>/bin/jar xvf sca\_SimpleApproval\_rev1.0.jar

inflated: SCA-INF/classes/META-INF/adf-config.xml

inflated: SCA-INF/classes/META-INF/weblogic-application.xml

inflated: SimpleApprovalTaskSimpleApprovalTask.componentType

inflated: SimpleApprovalTaskSimpleApprovalTask.task

inflated: composite.xml

inflated: monitor.config

inflated: testsuites/fileList.xml

inflated: xsd/SimpleApprovalTaskPayloadSimpleApprovalTaskPayload.xsd

inflated: xsd/SimpleApprovalTaskPayloadParametersSimpleApprovalTaskPayloadParameters.xsd

inflated: xsd/SimpleApprovalTaskWorkflowTaskSimpleApprovalTaskWorkflowTask.xsd

inflated: xsd/TaskEvidenceServiceTaskEvidenceService.xsd

inflated: xsd/TaskSequenceChangesTaskSequenceChanges.xsd

inflated: xsd/TaskStateMachineTaskStateMachine.xsd

inflated: xsd/WorkflowCommon.xsd

- the XSD files used for validate, the composite.xml and in this case a .task file indicating a Human Worklist Task component.

• Open and review any of the files using a text editor to validate code or parameters.

• In Step 8 below take the exported .jar file and import it for review into an Oracle JDeveloper environment.

• This exported .jar file could be archived and / or deployed to a different SOA suite environment.

5. Reviewing the export contents of an MDS on the file system

Using the directions in step 4 above, extract the contents of your MDS export using a suitable archive tool to view the contents on the file system.

• In addition to every SOA Composite deployed to that MDS (and all the various files making up those composites) , see additional directory structures that have been brought out of the export.

• See the configuration/default directory which stores component related config files; eg:

inflating: soa/configuration/folders.xml

inflating: soa/configuration/logmetadata.xml

inflating: soa/configuration/default/bpel-config.xml

inflating: soa/configuration/default/workflow-identity-config.xml

inflating: soa/configuration/default/soa-infra-config.xml

inflating: soa/configuration/default/mediator-config.xml

inflating: soa/configuration/default/cep-config.xml

inflating: soa/configuration/default/bpmn-config.xml

inflating: soa/configuration/default/workflow-notification-config.xml

inflating: soa/configuration/default/edn-config.xml

inflating: soa/configuration/default/b2b-config.xml

inflating: soa/configuration/default/workflow-config.xml

inflating: soa/configuration/default/adapter-config.xml

inflating: soa/configuration/default/businessrules-config.xml

For example, the soa-infra-config.xml file contains the audit trail parameter levels, transaction retry intervals and other various parameters.

• Open and review any of the files using a text editor to validate code or parameters or upload one or all files to an Oracle SR.

6. Browsing the MDS via Oracle JDeveloper

If developing or extending a SOA composite or reviewing code, and JDeveloper is the preferred editing environment, make a connection to the MDS and browse the contents similarly to extracting and editing the MDS contents via step 5 above.

Within an open JDeveloper instance;

• On the left is the Application Navigator (View / Application Navigator) and under PROJECTS see APPLICATION RESOURCES which contains connections and descriptors. Right click on Connections and create a new SOA-MDS connection.

• The Create SOA-MDS connection screen appears. Complete the details .

• Give the connection a name.

• Select DB Based MDS, as the connection type.

• Select the connection to the RDBMS where this MDS is stored.

NOTE: If not already defined , click on the + symbol and enter in the hostname, SID and port information. Give the Username and Password details as per your installation (Refer to Step 2 to obtain the schema name of this MDS - eg: <prefix>\_MDS or <CRM\_FUSION>\_MDS as examples. The password will have been defined during installation). Test this new RDBMS connection to ensure it works and click OK to return back to the SOA-MDS screen.

• Select SOA-INFRA from the MDS partition. Obtain this information from Step 2 above.

• Test the connection and click OK when successful.

• With the connection now created, on the right side of JDeveloper see the Resource Palette (view / Resource Palette). Open the IDE Connections section see the custom named and newly created MDS connection.

• Browse the contents of the MDS directly without exporting anything via Enterprise Manager.

• The structure will be the same as if the MDS file was exported using the /em url and extracted locally for review. It contains the same component configurations, deployed composites - including the xml, xsd, edl code and such. Double click to open this files directly within JDeveloper for review.

• Open the component configuration files such as soa-infra-config.xml. For example. the structure view (Bottom left of JDeveloper / Application Navigator) will read the xml elements and allow traversal of the structure of that document; showing you the values for the audit trail levels for example.

7. Importing a composite into JDeveloper for review

Users may want to review the code that resides within a SOA composite , extend the composite to add additional scopes for further inbound or outbound integrations, or check that recent changes are implemented in the current composite revision. In addition to reviewing an exported composite with a text editor, you can import the composite within JDeveloper.

See the GUI designer view of the composite flow, showing the wiring of the composite to exposed services and referenced in addition to reviewing and modifying code directly (e.g. changing a paramater of an exposed service or reference).

• Complete Step 1 above to save your sca\_<exportedComposite>\_rev<version>.jar

• In JDeveloper, select FILE / IMPORT, and select SOA Archive Into SOA Project and click OK

• JDeveloper will import the composite into a new project that was defined, or an existing project that was selected via the subsequent import screens.

• With the exported .jar file, and the project defined import the file.

NOTE: Unless you defined all of the connections that are referenced within the SOA composite then JDeveloper will throw connection errors as a popup dialog box at the final stage of importing. This dialog box has an OK button, so press ENTER to navigate past those errors.

• With the composite imported , open the composite.xml file normally available within the SOA content node of the project. See the wiring of the composite components and double click through the BPEL flow. Notice the yellow warning triangles on any references to services which use undefined connections. These warnings can be ignored for the purposes of code review. Clicking on the warnings will explain what the problem was - likely showing a reference to an external http://<host><domain>:<port>

/..../resource/..../something.wsdl link.

• With a connected developer environment , modify and link to external web services exposed on your network or on a partners network across the internet.

8. Exporting from the MDS via commandline (WLST)

In addition to using Enterprise Manager via http://<host><domain>:<port>/em , use the command line weblogic WLST command to export items from within an MDS.

The example below shows how to export the workflow-identity-config.xml configuration file;

wls:/offline> connect()

Please enter your username [weblogic] :weblogic

Please enter your server URL [t3://localhost:7001] :t3://<host><domain>:<port>

exportMetadata(application='soa-infra',server='soa\_server1',toLocation='any Location on server',docs='/soa/configuration/default/\*\*')

Notice the path to the config file is identical to what a full MDS export

would provide; that is because it's traversing the same path internally to the MDS to find the desired xml

config file. Like in Step 5 above, see how the root of the MDS export has a SOA directory, with a CONFIGURATION child directory that in turn has a directory called DEFAULT. Within that is the workflow-config.xml file.