Oracle® Endeca Information Discovery Studio

Provisioning Service Administration Guide

Version 3.2.0 • January 2016

ORACLE

Copyright and disclaimer

Table of Contents

Copyright and disclaimer **2**

Preface **4**
About this guide 4
Who should use this guide 4
Conventions used in this document 4
Contacting Oracle Customer Support 5

Chapter 1: Configuring the Provisioning Service **6**
About the Provisioning Service 6
Modifying the Provisioning Service configuration 7
Provisioning Service configuration properties 7
Configuring the Endeca Server to provision 9
Configuring storage of uploaded files 10
Managing memory usage 10
Configuring workflow database cleanup 11
Adding JDBC drivers 11
Changing the Provisioning Service context root 12
Configuring date formats for JSON upload 12
Configuring timeouts 13

Chapter 2: Configuring logging for the Provisioning Service **14**
About logging in the Provisioning Service 14
Changing the logging level for the Provisioning Service 14
Sample Provisioning Service logging.properties file 14

Preface

Endeca Information Discovery Studio is an industry-leading application composition environment and discovery experience that allows business users to easily upload and mash up multiple diverse data sources, and then quickly configure discovery applications - all within the context of an enterprise framework that maintains existing governance and enterprise definitions.

Studio includes world-class search, guided navigation, and filtering, as well as offering an array of powerful interactive visualizations, for rapid intuitive analysis that requires zero training.

About this guide

This guide describes how to configure and manage the Provisioning Service.

The Provisioning Service profiles and transforms data into appropriate formats before sending it to Endeca Server for ingest. Data can be uploaded by Studio users from Excel spreadsheets or JSON files, or extracted from a database or Oracle Business Intelligence Server.

Who should use this guide

This guide is intended for system administrators managing the Provisioning Service.

Conventions used in this document

The following conventions are used in this document.

Typographic conventions

The following table describes the typographic conventions used in this document.

Typeface	Meaning
User Interface Elements	This formatting is used for graphical user interface elements such as pages, dialog boxes, buttons, and fields.
`Code Sample`	This formatting is used for sample code phrases within a paragraph.
Variable	This formatting is used for variable values. For variables within a code sample, the formatting is *`Variable`*.
`File Path`	This formatting is used for file names and paths.

Symbol conventions

The following table describes symbol conventions used in this document.

Symbol	Description	Example	Meaning
>	The right angle bracket, or greater-than sign, indicates menu item selections in a graphic user interface.	File > New > Project	From the File menu, choose New, then from the New submenu, choose Project.

Contacting Oracle Customer Support

Oracle Customer Support provides registered users with important information regarding Oracle software, implementation questions, product and solution help, as well as overall news and updates from Oracle.

You can contact Oracle Customer Support through Oracle's Support portal, My Oracle Support at *https://support.oracle.com*.

Chapter 1

Configuring the Provisioning Service

The Provisioning Service runs as a Web application in a WebLogic Server container. A standard WebLogic Server configuration file (`eidProvisioningConfig/plan.xml`) is provided to configure the Provisioning Service.

About the Provisioning Service

Modifying the Provisioning Service configuration

Configuring the Endeca Server to provision

Configuring storage of uploaded files

Managing memory usage

Configuring workflow database cleanup

Adding JDBC drivers

Changing the Provisioning Service context root

Configuring date formats for JSON upload

Configuring timeouts

About the Provisioning Service

The Provisioning Service enables dynamic application creation from data uploaded from the desktop.

The Provisioning Service profiles and transforms data into appropriate formats before sending it to Endeca Server for ingest.

Users can upload data from the following sources:

- Microsoft Excel spreadsheets
- JSON files, such as Twitter data files
- Relational databases supported by a JDBC driver
- Oracle Business Intelligence Server

Modifying the Provisioning Service configuration

To modify the Provisioning Service configuration, edit the deployment plan (`$DOMAIN_HOME/eidProvisioningConfig/plan.xml`), and save your changes. Changes are applied when you restart the domain.

Provisioning Service configuration properties

This topic describes the available configuration parameters for the Provisioning Service.

The following example illustrates the available Provisioning Service configuration properties:

```
<variable-definition>
  <variable>
    <name>endeca-server-host-name</name>
    <value>localhost</value>
  </variable>
  <variable>
    <name>endeca-server-ws-port</name>
    <value>7001</value>
  </variable>
  <variable>
    <name>endeca-server-context-root</name>
    <value>/endeca-server</value>
  </variable>
  <variable>
    <name>endeca-server-data-domain-profile</name>
    <value>default</value>
  </variable>
  <variable>
    <name>endeca-server-security-enabled</name>
    <value>true</value>
  </variable>
  <variable>
    <name>upload-file-directory</name>
    <value>/eidProvisioningUpload</value>
  </variable>
  <variable>
    <name>transport-guarantee</name>
    <value>NONE</value>
  </variable>
  <variable>
    <name>protected-url-pattern</name>
    <value>/DISABLED</value>
  </variable>
  <variable>
    <name>message-driven-bean-limit</name>
    <value>6</value>
  </variable>
  <variable>
    <name>days-to-keep-workflows</name>
    <value>7</value>
  </variable>
  <variable>
    <name>jdbc-driver-classnames</name>
    <value>com.mysql.jdbc.Driver:oracle.bi.jdbc.AnaJdbcDriver:oracle.jdbc.OracleDriver</value>
  </variable>
  <variable>
    <name>context-root</name>
    <value>eid-ps</value>
  </variable>
</variable-definition>
```

The following table describes the Provisioning Service configuration properties.

Table 1.1: Provisioning Service configuration properties

Property name	Description	Default value	Valid values
`endeca-server-host-name`	Specifies the name of the host for the Endeca Server where you want to provision data. The default value assumes that the Endeca Server is installed on the same machine as the Provisioning Service.	localhost	Valid host names or IP addresses
`endeca-server-ws-port`	Specifies the Web services port for the Endeca Server where you want to provision data. The default value is the default Endeca Server Web services port.	7002	Valid port numbers
`endeca-server-context-root`	Specifies the WebLogic application root context of the Endeca Server.	/endeca-server	Valid root context names in WebLogic
`endeca-server-data-domain-profile`	Specifies the name of the Endeca data domain profile that will be used to create data domains when data is uploaded through the Provisioning Service.	default	Valid data domain profile names.
`endeca-server-security-enabled`	Specifies whether the Provisioning Service should use SSL to communicate with Endeca Server.	true	• true The Provisioning Service uses SSL to communicate with Endeca Server. • false The Provisioning Service does not use SSL to communicate with Endeca Server.

Property name	Description	Default value	Valid values
`upload-file-directory`	Specifies the directory on the Provisioning Service host machine where uploaded files will be stored. The directory is relative to the domain home directory.	`eidProvisioningUpload`	Valid path to an existing directory. The directory must be writeable by the Provisioning Service user.
`message-driven-bean-limit`	Specifies the maximum number of message-driven beans allowed in the pool on the WebLogic Server host of the . The number of message-driven beans defines the number of upload processes that the Provisioning Service can run simultaneously .	6	Positive integers
`days-to-keep-workflows`	Specifies the number of days that workflow data will be stored in the backend database before being cleaned up. After workflow data has been cleaned up, any related new application wizards that are not complete cannot be resumed.	7	Positive integers
`jdbc-driver-classnames`	Colon-separated list of JDBC drivers available to be used to connect to databases.	com.mysql.jdbc.Driver:oracle.bi.jdbc.AnaJdbcDriver:oracle.jdbc.OracleDriver	valid JDBC driver
`context-root`	Specifies the context root of the Provisioning Service web application.	`eid-ps`	Valid WebLogic application context root

Configuring the Endeca Server to provision

The default configuration of the Provisioning Service defines a connection to an Endeca Server running on the same machine and WebLogic server instance as the Provisioning Service, and listening on the default Endeca

Server port. Standard practice in production environments, however, is to run Endeca Servers on different machines from the Provisioning Service.

Each Provisioning Service provisions one Endeca Server with data uploaded by users.

To configure the Provisioning Service to connect to an Endeca Server other than the default, modify the following configuration properties:

- `endeca-server-host-name`

  The `endeca-server-host-name` property specifies the name or IP address of the host machine where you run the Endeca Server that you want to provision.

- `endeca-server-ws-port`

  The `endeca-server-ws-port` property specifies the Web services port on which the Endeca Server listens. The default Endeca Server Web services port is `7001`. The default secure port is `7002`. If you use a different port for your Endeca Server, change the value of this property to match the port you use.

The `endeca-server-context-root` property specifies the application root context of the Endeca Server on the WebLogic container. The default root context is `/endeca-server` and it cannot be changed.

Configuring storage of uploaded files

Each file uploaded is stored on the file system of the Provisioning Service.

The default location of uploaded files is `$DOMAIN_HOME/eidProvisioningUpload`.

If you want to configure a different directory to store uploaded files, use the `upload-file-directory` configuration property to specify the path to the new directory. The specified directory must be writeable for the user that runs the WebLogic Server.

For example, if you specify a value of `ps-file` for this property, you must create the `ps-file` directory and configure its permissions to allow write permissions to the WebLogic Server user.

Managing memory usage

You may notice reduced performance when loading large files. In that case, you may want to modify the memory usage of the Provisioning Service.

Two configuration settings control memory usage in the Provisioning Service:

- The maximum number of Message-Driven Beans (MDBs) allowed in the pool.

  The maximum number of Message-Driven Beans for the Provisioning Service is controlled by the `message-driven-bean-limit` parameter in `plan.xml`. The default value of this variable is `2`.

- The Java heap space. Java heap space is controlled by the `USER_MEM_ARGS=` parameter in the file `$DOMAIN_HOME/bin/setDomainEnv.sh` in Linux, or `$DOMAIN_HOME/bin/setDomainEnv.cmd` in Windows.

  In the line

  ```
  USER_MEM_ARGS="-Xms256m -Xmx6144m -XX:PermSize=64M -XX:MaxPermSize=512M"
  ```

  change `-Xmx6144m` to the amount of heap space you want to use. The default heap space is 6GB.

These two configuration settings are related. If you specify a larger pool of MDBs, you should also specify more heap space for the Provisioning Service. Conversely, if you reduce the amount of heap space, you should reduce the number of MDBs in the pool.

The starting estimate is 1 GB of heap space per unit in the pool. Thus, the default pool size of 6 calls for 6GB of heap space. As you use the Provisioning Service, you can fine tune these estimates to match your own needs and experience.

Configuring workflow database cleanup

You can configure the duration that workflow data will be stored before being cleaned from the database.

Data for upload workflow processes is stored in the backend database of the Provisioning Service. The Provisioning Service automatically cleans up old data from this database.

You can configure the duration the Provisioning Service will store this data before removing it from the database during cleanup. Use the `duration-to-keep-workflows` configuration property to configure the length of time that the Provisioning Service will store workflow data in the database before deleting it. The value of this property specifies the number of days that workflow data will be stored before being deleted. The default value is 7 (one week). Any workflow data older than the specified number of days will be deleted.

Specify a value of `-1` to disable database cleanup.

Adding JDBC drivers

You can configure support for JDBC drivers beyond those supported by default.

A default installation of the Provisioning Service includes drivers to support the following relational database management systems:

- Oracle 11g

  Oracle Business Intelligence Server is also supported.
- MySQL

To support additional third party databases:

1. Copy the JDBC driver file to `$DOMAIN_HOME/config/jdbc`.
2. Add the driver class name to the value of the `jdbc-driver-classnames` configuration property.

   The value of this property is a colon-separated list of third-part JDBC driver classpaths.

For example, the default value of the `jdbc-driver-classnames` property is:

```
com.mysql.jdbc.Driver:oracle.bi.jdbc.AnaJdbcDriver:oracle.jdbc.OracleDriver
```

.

If you want to add support for Hadoop Hive clusters, copy the Hadoop Hive JDBC driver to `$DOMAIN_HOME/config/jdbc` and update the value of the `jdbc-driver-classnames` configuration parameter to:

```
com.mysql.jdbc.Driver:oracle.bi.jdbc.AnaJdbcDriver:oracle.jdbc.OracleDriver:org.apache.hadoop.hive.jdbc.HiveDriver
```

.

Changing the Provisioning Service context root

You can change the context root for the Provisioning Service.

Use the `context-root` configuration property to specify the context root of the Provisioning Service web application. The default value of this property is `eid-ps`.

For example, the address for your Provisioning Service might be:

```
http://localhost:8201/eid-ps/CoreService?wsdl
```

If you want to change the *eid-ps* to another root, you would change the value of this property. For example, if you wanted the root to be *endeca*, you would change the value of this property to `endeca`, and the address of your Provisioning Service would be:

```
http://localhost:8201/endeca/CoreService?wsdl
```

If you use a context root other than the default, you must modify the Studio connection to the Provisioning Service. For details, see "Configuring the Connection to the Provisioning Service" in the *Studio Administration and Customization Guide*.

Configuring date formats for JSON upload

You can configure the supported date formats for JSON uploads in the `eidProvisioningConfig/dateFormats.txt` file.

The content of this file is a list of the date formats that are checked when processing JSON input. If an input date does not match any of the formats defined in this file, it will not be processed as a date.

Date formats must conform to Java SimpleDateFormat: *http://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html*.

Processing of date format patterns is not lenient. For details, see *http://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#setLenient%28boolean%29*.

Dates that do not include a time zone are processed as Greenwich Mean Time (GMT).

Date formats supported by default include:

- d/M/yy
- d-M-yy
- d.M.yy
- M/d/yy
- M-d-yy
- M.d.yy
- yy/M/d
- yy-M-d
- yy.M.d
- MMM d, yyyy
- EEE, MMM d, yyyy

- yyyy-MM-dd'T'HH-mm-ssZ
- yyyy-MM-dd'T'HH:mm:ss'Z'
- yyyy-MM-dd'T'HH:mm:ss.SSS'Z'
- EEE d MMM yyyy HH:mm:ss Z

If a date matches multiple patterns, an arbitrary pattern is selected when uploading the data to Endeca Server.

Configuring timeouts

Two timeouts are available to configure interaction with data sources.

Stuck thread max time

The *Stuck thread max time* property specifies the amount of time, in seconds, a thread can run processing before WebLogic Server considers it stuck and terminates it.

When the Provisioning Service is installed, the value of this parameter is set to 14400 (four hours).

If you need to load a large amount of data and need processing to run longer, you can change the value of this property.

For additional details about this property and how to set it, see "Tuning stuck thread detection behavior" at *https://docs.oracle.com/middleware/1213/wls/WLACH/taskhelp/tuning/TuningExecuteThreads.html*.

Oracle Business Intelligence Server synchronization operations

When loading data from Oracle Business Intelligence Server, a server under heavy load may not respond quickly, and browsing operations when defining a data source may time out. In that case, you may need to increase the timeout on the operation synchronization bean, `ObiSyncOperationsEjb`. The default timeout for this bean is 600 seconds (ten minutes). Browsing operations should complete in far less time. If browsing operations with the Oracle Business Intelligence Server take longer, you can change the timeout to allow more time.

For details about configuring the timeout in beans in the WebLogic Server, see *https://docs.oracle.com/middleware/1213/wls/WLJTA/trxejb.htm#WLJTA252*.

This setting only affects browsing operations with the Oracle Business Intelligence server. Data loading operations are still handled by the *Stuck thread max time* property.

Chapter 2

Configuring logging for the Provisioning Service

The Provisioning Service uses standard WebLogic Server logging functionality.

About logging in the Provisioning Service

Changing the logging level for the Provisioning Service

Sample Provisioning Service logging.properties file

About logging in the Provisioning Service

By default, the Provisioning Service logs all errors and warnings to the domain log broadcaster, to the server log, or to standard out, depending on the configuration of the WebLogic Server.

For details about configuring logging in WebLogic Server, see "Configuring Log Files and Filtering Log Messages for Oracle WebLogic Server" at *https://docs.oracle.com/middleware/1213/wls/WLLOG/toc.htm*.

Changing the logging level for the Provisioning Service

When resolving issues, technical support may request that you modify the logging level of the Provisioning Service.

The Provisioning Service installation package you download from Oracle includes a `logging.properties` file you can use when configuring the logging level of your Provisioning Service. This file is stored in the `eidProvisioningConfig` directory under your domain.

To change the logging level of the Provisioning Service, edit `logging.properties` to specify the correct level of logging messages (FINE or FINEST) to filter, and then restart the domain.

Logging messages from the Provisioning Service are listed in the log file under the `ProvisioningLogger` class, as illustrated in the following example:

```
<Dec 21, 2015 12:45:40 PM EST> <Warning> <com.oracle.endeca.pdi.logging.ProvisioningLogger>
<BEA-000000> <OurMessage>
```

You can search and filter the logs to find this class so you can easily find logging messages from the Provisioning Service.

Sample Provisioning Service logging.properties file

The code in this topic illustrates an example `logging.properties` file for the Provisioning Service.

```
# Specify the handlers to create in the root logger
handlers = weblogic.logging.ServerLoggingHandler
```

```
# Register handlers for the com.oracle.endeca.pdi and its child loggers
com.oracle.endeca.pdi.logging.ProvisioningLogger.handlers = weblogic.logging.ServerLoggingHandler
# Do not send the com.oracle.endeca.pdi log messages to the root handler
com.oracle.endeca.pdi.logging.ProvisioningLogger.useParentHandlers = false
# Set specific LogLevel
com.oracle.endeca.pdi.logging.ProvisioningLogger.level = FINE
```

The logging properties file defines the logging level following the JDK specification. Use the following logging levels:

- `FINE` (sets Debug logging level in WebLogic server)
- `FINEST` (sets Trace logging level in WebLogic server)
- `ALL` (sets All logging level in WebLogic server)

Index

C

configuration
- available properties 7
- modifying 7

context root 12

D

database cleanup 11

date configuration 12

E

Endeca Server, connecting to 10

J

JDBC drivers 11

JSON date configuration 12

L

logging
- about 14
- changing logging level 14
- example logging.properties file 14

M

memory usage 10

S

storage, configuring for uploaded files 10

T

third-party databases 11

U

uploaded files, configuring storage 10

W

workflow cleanup 11