An Oracle White Paper
September 2010

Building Custom Plug-ins for Oracle Virtual Directory 11g

Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

[

Overview

This document complements the Oracle Virtual Directory 11g Product Manual for customers and partners who need to write custom plug-ins.

Specifically this document details how to build a plug-in using JDeveloper 11g. The package include a JDeveloper 11g workspace containing a sample plug-in (the same example plug-in documented in 11g documentation) as well as a screencast showing how to deploy to the server.

Plug-In Basics

OVD plug-ins are Java code that are used to solve use cases not provided by the default adapter or plug-ins that ship with the product.

Customers may use them to solve problems such as how to call a database stored procedure or to enable OVD to query a Web Service.

The plug-in API including Javadoc are provided by the Identity Management documentation.

Building Plug-ins in JDeveloper 11g

OVD plug-ins can be written using any Java IDE but this document will describe how to do this with JDeveloper 11g.

The first step is to get a copy of two JAR files – vde.jar, ovdcommon.jar and asn1c.jar from the OVD server. These files are in $ORACLE_HOME/ovd/jlib

Next create a new application and generic project that will be used to build the plug-ins.

Create a new Java class that extends com.octetstring.vde.chain.BasePlugin.

A plug-in only needs to override the methods needed for its logic.

After the Java class is written and it compiles correctly – it must be packaged into a Java Archive (JAR). The JAR file must include specific values in its META-INF/MANIFEST.MF file.

The developer must a new text file called vde-properties.txt that will store these properties. In the JAR deployment option for the project – configure JDeveloper to include this file into the MANIFEST.MF.

This is a commented example from the included project:

```
#commented
#this is class of the plug-in. Important if multiple classes in the plug-in
vde-package-classname: demo.BadPasswordCount
```

```
#next parameter tells OVD that this is infact a plug-in
vde-package-type: 0
#name to display in UI
vde-package-name: BadPasswordCount
#define which operations the plug-in will operate on
vde-package-ops-add: true
vde-package-ops-delete: false
vde-package-ops-bind: true
vde-package-ops-modify: true
vde-package-ops-rename: false
vde-package-ops-get: true
#the version of this specific plug-in. increase for each production update
vde-package-version: 0
#description of the plugin
vde-package-description: This plug-in keeps track of number of times password is incorrectly called

#the followin is optional. It defines parameters for the plug-in.
#the value after param is name of the attribute. Parameters can be retrieved in init method
vde-package-param-countattribute-description: Name of the LDAP attribute to store password count
vde-package-param-addoncreate-description: true to support adds
vde-package-param-objectclass-description: Name of objectclass that contains the countattribute.
vde-package-param-ignoremodify-description: true to not reset count
vde-package-param-maxtry-description: The maximum number of tries before locking account
vde-package-param-usememory-description: Set to true if wish to only keep password count in memory
```

Once the file is created, use JDeveloper’s Deploy option to compile and build the JAR.

After the JAR is built – use Oracle Directory Services Manager to upload the JAR to OVD server to make it available as a plug-in. The included Flash screenscast demonstrates how to do this.

Building Custom Plug-ins for Oracle Virtual
Directory 11g
September 2010
Author: Mark Wilcox
Contributing Authors:

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

Oracle is committed to developing practices and products that help protect the environment