

Oracle® Fusion Middleware
Oracle Data Integrator Tools Reference
12c (12.2.1.2.0)
E77138-01

September 2016

E77138-01

Copyright © 2010, 2016 Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface	vii
Audience	vii
Documentation Accessibility	vii
Related Documents	vii
Conventions	viii
1 Using Oracle Data Integrator Open Tools	
Using Oracle Data Integrator Tools	1-1
Using a Tool in a Package	1-1
Using a Tool in a Knowledge Module or Procedure Command	1-2
Using a Tool From a Command Line	1-2
Using Open Tools	1-2
Installing and Declaring an Open Tool	1-3
Installing an Open Tool	1-3
Declaring a New Open Tool	1-3
Using Open Tools in a Package or Procedure	1-4
Developing Open Tools	1-4
Classes	1-4
Developing a New Open Tool	1-4
Implementing the Class	1-5
Declaration	1-5
Importing Packages	1-5
Defining the Parameters	1-5
Implementing Informational Functions	1-6
Execution	1-7
Open Tools at Runtime	1-7
2 Oracle Data Integrator Tools	
Oracle Data Integrator Tools by Category	2-1
Metadata	2-1
Oracle Data Integrator Objects	2-1
Utilities	2-1
Internet Related Tasks	2-2
Files	2-2
SAP	2-3

XML.....	2-3
Event Detection	2-3
Changed Data Capture.....	2-3
Alphabetical List of Oracle Data Integrator Tools.....	2-3
OdiAnt.....	2-6
OdiBeep.....	2-8
OdiDeleteScen.....	2-9
OdiEnterpriseDataQuality.....	2-10
OdiExportAllScen.....	2-11
OdiExportEnvironmentInformation.....	2-13
OdiExportLog.....	2-15
OdiExportMaster.....	2-17
OdiExportObject.....	2-19
OdiExportScen.....	2-22
OdiExportWork.....	2-24
OdiFileAppend.....	2-25
OdiFileCopy.....	2-26
OdiFileDelete.....	2-28
OdiFileMove.....	2-30
OdiFileWait.....	2-32
OdiFtp.....	2-35
OdiFtpGet.....	2-37
OdiFtpPut.....	2-39
OdiGenerateAllScen.....	2-41
OdiImportObject.....	2-43
OdiImportScen.....	2-45
OdiInvokeWebService.....	2-46
OdiKillAgent.....	2-49
OdiManageOggProcess.....	2-50
OdiMkdir.....	2-51
OdiOSCommand.....	2-52
OdiOutFile.....	2-54
OdiPingAgent.....	2-55
OdiPurgeLog.....	2-56
OdiReadMail.....	2-59
OdiRefreshJournalCount.....	2-62
OdiReinitializeSeq.....	2-64
OdiRemoveTemporaryObjects.....	2-65
OdiRetrieveJournalData.....	2-66
OdiReverseGetMetaData.....	2-68
OdiReverseManageShortcut.....	2-69
OdiReverseResetTable.....	2-70
OdiReverseSetMetaData.....	2-71
OdiSAPALEClient and OdiSAPALEClient3.....	2-72
OdiSAPALEServer and OdiSAPALEServer3.....	2-74
OdiScpGet.....	2-76
OdiScpPut.....	2-79

OdiSendMail	2-82
OdiSftp	2-84
OdiSftpGet	2-86
OdiSftpPut.....	2-89
OdiSleep.....	2-92
OdiSqlUnload	2-93
OdiRetrieveHadoopLog.....	2-96
OdiStartLoadPlan.....	2-97
OdiStartOwbJob.....	2-99
OdiStartScen	2-101
OdiUnZip.....	2-103
OdiUnlockOdiRepository	2-104
OdiUpdateAgentSchedule	2-105
OdiWaitForChildSession.....	2-106
OdiWaitForData	2-108
OdiWaitForLoadPlans	2-113
OdiWaitForLogData.....	2-114
OdiWaitForTable.....	2-117
OdiXMLConcat	2-118
OdiXMLSplit.....	2-120
OdiZip	2-123

Preface

This guide describes how to use and develop Open Tools using Oracle Data Integrator to design integration scenarios.

This preface contains the following topics:

- [Audience](#)
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Conventions](#)

Audience

This document is intended for Oracle Data Integrator application developers who will use Open Tools to design integration scenarios.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Documents

For more information, see the following documents in *Oracle Data Integrator Library*:

- *Release Notes for Oracle Data Integrator Release Notes*
- *Understanding Oracle Data Integrator*
- *Developing Integration Projects with Oracle Data Integrator*
- *Installing and Configuring Oracle Data Integrator*
- *Upgrading Oracle Data Integrator*
- *Integrating Big Data with Oracle Data Integrator*

- *Application Adapters Guide for Oracle Data Integrator*
- *Developing Knowledge Modules with Oracle Data Integrator*
- *Connectivity and Knowledge Modules Guide for Oracle Data Integrator*
- *Migrating From Oracle Warehouse Builder to Oracle Data Integrator*
- *Oracle Data Integrator Tools Reference*
- *Data Services Java API Reference for Oracle Data Integrator*
- *Open Tools Java API Reference for Oracle Data Integrator*
- *Getting Started with SAP ABAP BW Adapter for Oracle Data Integrator*
- *Java API Reference for Oracle Data Integrator*
- *Getting Started with SAP ABAP ERP Adapter for Oracle Data Integrator*
- *Oracle Data Integrator 12c Online Help*, which is available in ODI Studio through the JDeveloper Help Center when you press **F1** or from the main menu by selecting **Help**, and then **Search** or **Table of Contents**.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Using Oracle Data Integrator Open Tools

This appendix provides a reference of Oracle Data Integrator (ODI) tools. It describes how to use Open Tools to develop new scenarios in Oracle Data Integrator.

This appendix includes the following sections:

- [Using Oracle Data Integrator Tools](#)
- [Using Open Tools](#)
- [Developing Open Tools](#)

Using Oracle Data Integrator Tools

Oracle Data Integrator tools (also called Oracle Data Integrator commands) are commands provided for performing specific tasks at runtime. These tasks can be as simple as waiting for a certain time or producing a sound, or as sophisticated as executing Ant scripts or reading e-mail from a server.

Oracle Data Integrator tools are used in Packages, Procedure Commands, Knowledge Modules Commands, or directly from a command line.

Note: Previous versions of Oracle Data Integrator supported calling built-in tools from Jython or Java scripts using their internal Java classes (such as `SnpsSendMail` and `SendMail`). This approach is no longer supported.

Note: Carriage returns in commands are not permitted.

Using a Tool in a Package

Adding and using an Oracle Data Integrator tool in a Package is described in "Adding Oracle Data Integrator Tool Steps" in *Developing Integration Projects with Oracle Data Integrator*.

You can sequence the tool steps within the package and organize them according to their success and failure. For more information about sequencing, see "Arranging the Steps Layout" and "Defining the Sequence of Steps" in *Developing Integration Projects with Oracle Data Integrator*.

You can use variable values, sequences, or Oracle Data Integrator substitution method calls directly in tool parameters.

Using a Tool in a Knowledge Module or Procedure Command

Using an Oracle Data Integrator tool in a Knowledge Module or Procedure is described in "Working with Procedures" in *Developing Integration Projects with Oracle Data Integrator*.

You can use variable values, sequences, Oracle Data Integrator substitution method calls, or the results from a `SELECT` statement directly in tool parameters.

Using a Tool From a Command Line

Command line scripts for Oracle Data Integrator tools are run from the `DOMAIN_HOME/bin` directory. To run a tool from a command line, you must first create an ODI Physical Agent instance in the ODI Topology and configure an ODI Standalone Agent instance in a Domain. For more information about performing these tasks, see *Installing and Configuring Oracle Data Integrator*.

When you run a tool from a command line, you must specify the `-INSTANCE` parameter, where `<agent_name>` is the name of the physical agent you configured (for example, `OracleDIAgent1`).

To use an Oracle Data Integrator tool from a command line:

1. Launch the command shell for your environment (Windows or UNIX).
2. Navigate to the `DOMAIN_HOME/bin` directory.
3. Launch the `startcmd.cmd` (Windows) or `startcmd.sh` (UNIX) command and run an Oracle Data Integrator tool with the following syntax:

```
startcmd.<cmd|sh> -INSTANCE=<agent_name> <command_name> [<command_parameters>]*
```

Command names and command parameters are case-sensitive.

Important Notes

Note the following:

- On Windows platforms, command arguments that contain equal (=) signs or spaces must be surrounded with double quotation marks. This differs from the UNIX command call. For example:

```
startcmd.cmd OdiSleep "-INSTANCE=OracleDIAgent1" "-DELAY=5000"  
./startcmd.sh OdiSleep -INSTANCE=OracleDIAgent1 -DELAY=5000
```

- The following tools do not support direct invocation through a command line:
 - `OdiRetrieveJournalData`
 - `OdiRefreshJournalCount`

Using Open Tools

The Open Tools feature provides an extensible platform for developing custom third-party tools that you can use in Packages and Procedures. As with the standard tools delivered with Oracle Data Integrator, Open Tools can interact with the operating system and manipulate data.

Open Tools are written in Java. Writing your own Open Tools is described in ["Developing Open Tools"](#) on page 1-4.

Open Tools are delivered as a Java package (`.zip` or `.jar`) that contains several files:

- A compiled Java .class file
- Other resources, such as icon files

Installing and Declaring an Open Tool

Before you can use an Open Tool, you must install and add it.

Installing an Open Tool

To install an Open Tool, you must add the Open Tool JAR into the classpath or the component using the tool.

Open Tool JARs must be added to the `DOMAIN_HOME/lib` directory. Drivers are added to the same location.

To deploy an Open Tool JAR with a Java EE agent, generate a server template for this agent. The Open Tool is displayed in the **Libraries and Drivers** list in the Template Generation Wizard. See "Creating a Server Template for the Java EE Agent" in *Administering Oracle Data Integrator* for more information.

Note: This operation must be performed for each Oracle Data Integrator Studio from which the tool is being used, and for each agent that will run sessions using this tool.

Declaring a New Open Tool

This operation declares an Open Tool in a master repository and enables the tool to be displayed in Oracle Data Integrator Studio.

To declare an Open Tool, a JAR must be added in `<ide.user.dir>/oracledi/userlib`.

To declare a new tool:

1. In Oracle Data Integrator Studio, select the **ODI** menu and then select **Add Remove/Open Tools**. The **Add Open Tools** dialog is displayed.
2. Enter the name of the class in the **Open Tool Class Name** field.

or:

1. Click **Find in the ClassPath**, then browse to the name of the Open Tool's Java class. To search for the class by name, enter part of the name in the field at the top.
2. Click **OK**.

Note that all classes currently available to Oracle Data Integrator are displayed, including those that are not Open Tools. You must know the name of your class in order to add it.

3. Click **Add Open Tool**.
4. Select the line containing your Open Tool.
 - If the tool was correctly found on the classpath, the supplied icons and the tool's syntax, description, provider, and version number are displayed.
 - If the tool was not found, an error message is displayed. Change the classpath, or move the Open Tool to the correct directory.

Note: This operation to declare a new Open Tool must be performed only once for a given master repository.

Note: An Open Tool, a name cannot start with `Snp` or `Odi`. An Open Tool with a name that starts with these strings is ignored.

Using Open Tools in a Package or Procedure

You can use Open Tools in a Package or Procedure, similar to the tools provided with Oracle Data Integrator.

Developing Open Tools

An Open Tool is a Java package that contains a compiled Java class that implements the interface `oracle.odi.sdk.opentools.IOpenTool`. For a complete description of classes and methods, see the *Oracle Data Integrator Open Tools Java API Reference* (JavaDoc).

An Open Tool package typically should also contain two icons, which are used to represent the Open Tool in the Oracle Data Integrator graphical interface.

Classes

The following table lists and describes Open Tool classes and interfaces.

Class or Interface	Description
<code>IOpenTool</code>	Interface that every Open Tool must implement.
<code>OpenToolAbstract</code>	Abstraction of the interface with some helper methods. Preferably extend this class rather than implementing the interface directly.
<code>IOpenToolParameter</code>	Interface that parameters used by Open Tools must implement. In most cases, <code>OpenToolParameter</code> should be used rather than implementing this interface.
<code>OpenToolParameter</code>	Complete implementation of <code>IOpenToolParameter</code> . Each <code>OpenToolParameter</code> holds one parameter.
<code>OpenToolsExecutionException</code>	Exception class that should be thrown if necessary by Open Tool methods.
<code>SimpleOpenToolExample</code>	A simple example of an Open Tool, which can be used as a starting point.

Developing a New Open Tool

The following steps describe the development of a basic Open Tool, `SimpleMessageBox`. The source code for this class is available in the `demo/plugins/src` directory.

1. Define the syntax. In this example, the Open Tool is called as follows:


```
SimpleMessageBox "--TEXT=<text message>" "--TITLE=<window title>"
```
2. Create 16x16 and 32x32 icons (usually in `.gif` format).
3. Create and implement the class. See "[Implementing the Class](#)" on page 1-5.
4. Compile the class and create a package with the two icon files.
5. Install and declare the Open Tool as described in "[Installing and Declaring an Open Tool](#)" on page 1-3.

Implementing the Class

Implementing the class consists of the following steps:

1. [Declaration](#)
2. [Importing Packages](#)
3. [Defining the Parameters](#)
4. [Implementing Informational Functions](#)
5. [Execution](#)

Declaration Before you declare the class, you must name the package.

Naming the Package

Put the class in a package named appropriately. The package name is used to identify the Open Tool when installing it.

```
package com.myCompany.OpenTools;
```

Declaring the Class

There are two basic approaches to developing an Open Tool:

- Extend an existing class that you want to convert into an Open Tool. In this case, simply implement the interface `IOpenTool` directly on the existing class.
- Develop a new class. In this case, it is easiest to extend the abstract class `OpenToolAbstract`. This abstract class also contains additional helper methods for working with parameters.

```
public class SimpleMessageBox extends OpenToolAbstract {
```

Importing Packages Almost every Open Tool must import the following Open Tool SDK packages:

```
import oracle.odi.sdk.opentools.IOpenTool; /* All Open Tool classes need these
three classes */
```

```
import oracle.odi.sdk.opentools.IOpenToolParameter;
```

```
import oracle.odi.sdk.opentools.OpenToolExecutionException;
```

```
import oracle.odi.sdk.opentools.OpenToolAbstract; /* The abstract extended for the
Open Tool */
```

```
import oracle.odi.sdk.opentools.OpenToolParameter; /* The class used for
parameters */
```

In this particular example, a package to create the message box is also needed:

```
import javax.swing.JOptionPane; /* Needed for the message box used in this example
*/
```

Defining the Parameters Add a property to store the `OpenToolParameter` objects. This is used to both define them for the syntax, and to retrieve the values of the parameters from the eventual user. It is easiest to define the parameters of the Open Tool with a static array as follows. This array should be private, as it will be accessed through an accessor function.

```
private static final IOpenToolParameter[] mParameters = new IOpenToolParameter[]
```

```
{
    new OpenToolParameter("-TEXT", "Message text", "Text to show in the messagebox
(Mandatory).", true),
    new OpenToolParameter("-TITLE", "Messagebox title", "Title of the
messagebox.", false)
};
```

The four parameters passed to the `OpenToolParameter()` constructor are as follows:

1. The code of the parameter, including the initial hyphen. This code must correspond to the syntax returned by `getSyntax()`.
2. The user-friendly name, which is used if the user is using the graphical interface to set parameters.
3. A descriptive help text.
4. Whether the parameter is mandatory. This is an indication to the user.

Note: Oracle Data Integrator does not enforce the mandatory flag on parameters. Your class must be able to handle any combination of parameters being provided.

You must implement the accessor function `getParameters()` to retrieve the parameters:

```
public IOpenToolParameter[] getParameters()
{
    return mParameters;
}
```

Implementing Informational Functions Implement functions to return information about your Open Tool: `getDescription()`, `getVersion()`, `getProvider()`.

```
public String getDescription() { return "This Open Tool displays a message box
when executed."; }
public String getVersion() { return "v1.0"; }
public String getProvider() { return "My Company, Inc."; }
```

The `getSyntax()` function determines the name of the Open Tool as it is displayed in the Oracle Data Integrator graphical interface, and also the initial values of the parameter. Make sure the names of the parameters here match the names of the parameters returned by `getParameters()`.

```
public String getSyntax()
{
    return "SimpleMessageBox \"-TEXT=<text message>\" \"-TITLE=<window
title>\";
}
```

The `getIcon()` method should then return paths to two appropriately sized images. It should look something like this:

```
public String getIcon(int pIconType)
{
    switch (pIconType)
    {
        case IOpenTool.SMALL_ICON:
            return "/com/myCompany/OpenTools/images/SimpleMessageBox_16.gif";
        case IOpenTool.BIG_ICON:

```

```

        return "/com/myCompany/OpenTools/images/SimpleMessageBox_32.gif";
    default:
        return "";
    }
}

```

Execution Finally, the `execute()` method, which carries out the functionality provided by the Open Tool. In this case, a message box is shown. If you are extending the `OpenToolAbstract` class, use the `getParameterValue()` method to easily retrieve the values of parameters, as they are set at runtime.

Note: You must catch all exceptions and only raise an `OpenToolExecutionException`.

```

public void execute() throws OpenToolExecutionException
{
    try
    {
        if (getParameterValue("-TITLE") == null ||
getParameterValue("-TITLE").equals("")) /* title was not filled in by user */
        {
            JOptionPane.showMessageDialog(null, (String)
getParameterValue("-TEXT"), (String) "Message", JOptionPane.INFORMATION_MESSAGE);
        } else
        {
            JOptionPane.showMessageDialog(null, (String)
getParameterValue("-TEXT"),
                (String) getParameterValue("-TITLE"),
                JOptionPane.INFORMATION_MESSAGE);
        }
    }
    /* Traps any exception and throw them as OpenToolExecutionException */
    catch (IllegalArgumentException e)
    {
        throw new OpenToolExecutionException(e);
    }
}

```

Open Tools at Runtime

In general, your Open Tool class is instantiated only very briefly, and is used in the following ways.

Installation

When the user chooses to install an Open Tool, Oracle Data Integrator instantiates the class and calls the methods `getDescription()`, `getProvider()`, `getIcon()`, and `getVersion()` to retrieve information about the class.

Use in a Package

When the Open Tool is used in a package, the class is instantiated briefly to call the methods `getDescription()`, `getProvider()`, `getIcon()`, and `getVersion()`. Additionally, `getSyntax()` is called to retrieve the code name of the Open Tool and its default arguments. The method `getParameters()` is called to display the list of arguments to the user.

Execution

Each time the Open Tool is executed in a package or procedure, the class is instantiated again; it has no persistence after its execution. The `execute()` method is called just once.

Tip: See also [Using Open Tools](#) and Open Tools SDK documentation (JavaDoc).

Oracle Data Integrator Tools

This chapter lists all the Oracle Data Integrator Tools by category and describes its commands and parameters.

This section lists Oracle Data Integrator tools by category.

Oracle Data Integrator Tools by Category

This section lists Oracle Data Integrator tools by category.

Metadata

- [OdiReverseGetMetaData](#)
- [OdiReverseManageShortcut](#)
- [OdiReverseResetTable](#)
- [OdiReverseSetMetaData](#)

Oracle Data Integrator Objects

- [OdiDeleteScen](#)
- [OdiExportAllScen](#)
- [OdiExportEnvironmentInformation](#)
- [OdiExportLog](#)
- [OdiExportMaster](#)
- [OdiExportObject](#)
- [OdiExportScen](#)
- [OdiExportWork](#)
- [OdiGenerateAllScen](#)
- [OdiImportObject](#)
- [OdiImportScen](#)
- [OdiUnlockOdiRepository](#)

Utilities

- [OdiAnt](#)

- [OdiBeep](#)
- [OdiEnterpriseDataQuality](#)
- [OdiKillAgent](#)
- [OdiOSCommand](#)
- [OdiPingAgent](#)
- [OdiPurgeLog](#)
- [OdiReinitializeSeq](#)
- [OdiRemoveTemporaryObjects](#)
- [OdiRetrieveHadoopLog](#)
- [OdiStartLoadPlan](#)
- [OdiStartOwbJob](#)
- [OdiStartScen](#)
- [OdiUpdateAgentSchedule](#)

Internet Related Tasks

- [OdiFtp](#)
- [OdiFtpGet](#)
- [OdiFtpPut](#)
- [OdiInvokeWebService](#)
- [OdiReadMail](#)
- [OdiScpGet](#)
- [OdiScpPut](#)
- [OdiSftp](#)
- [OdiSftpGet](#)
- [OdiSftpPut](#)
- [OdiSendMail](#)

Files

- [OdiFileAppend](#)
- [OdiFileCopy](#)
- [OdiFileDelete](#)
- [OdiFileMove](#)
- [OdiFileWait](#)
- [OdiMkDir](#)
- [OdiOutFile](#)
- [OdiSqlUnload](#)
- [OdiUnZip](#)
- [OdiZip](#)

SAP

- [OdiSAPALEClient](#) and [OdiSAPALEClient3](#)
- [OdiSAPALEServer](#) and [OdiSAPALEServer3](#)

XML

- [OdiXMLConcat](#)
- [OdiXMLSplit](#)

Event Detection

- [OdiFileWait](#)
- [OdiReadMail](#)
- [OdiSleep](#)
- [OdiWaitForChildSession](#)
- [OdiWaitForData](#)
- [OdiWaitForLoadPlans](#)
- [OdiWaitForLogData](#)
- [OdiWaitForTable](#)

Changed Data Capture

- [OdiManageOggProcess](#)
- [OdiRefreshJournalCount](#)
- [OdiRetrieveJournalData](#)
- [OdiWaitForData](#)
- [OdiWaitForLogData](#)
- [OdiWaitForTable](#)

Alphabetical List of Oracle Data Integrator Tools

This section lists Oracle Data Integrator tools in alphabetical order.

- [OdiAnt](#)
- [OdiBeep](#)
- [OdiDeleteScen](#)
- [OdiEnterpriseDataQuality](#)
- [OdiExportAllScen](#)
- [OdiExportEnvironmentInformation](#)
- [OdiExportLog](#)
- [OdiExportMaster](#)
- [OdiExportObject](#)
- [OdiExportScen](#)

- [OdiExportWork](#)
- [OdiFileAppend](#)
- [OdiFileCopy](#)
- [OdiFileDelete](#)
- [OdiFileMove](#)
- [OdiFileWait](#)
- [OdiFtp](#)
- [OdiFtpGet](#)
- [OdiFtpPut](#)
- [OdiGenerateAllScen](#)
- [OdiImportObject](#)
- [OdiImportScen](#)
- [OdiInvokeWebService](#)
- [OdiKillAgent](#)
- [OdiManageOggProcess](#)
- [OdiMkDir](#)
- [OdiOSCommand](#)
- [OdiOutFile](#)
- [OdiPingAgent](#)
- [OdiPurgeLog](#)
- [OdiReadMail](#)
- [OdiRefreshJournalCount](#)
- [OdiReinitializeSeq](#)
- [OdiRemoveTemporaryObjects](#)
- [OdiRetrieveHadoopLog](#)
- [OdiRetrieveJournalData](#)
- [OdiReverseGetMetaData](#)
- [OdiReverseManageShortcut](#)
- [OdiReverseResetTable](#)
- [OdiReverseSetMetaData](#)
- [OdiSAPALEClient and OdiSAPALEClient3](#)
- [OdiSAPALEServer and OdiSAPALEServer3](#)
- [OdiScpGet](#)
- [OdiScpPut](#)
- [OdiSendMail](#)
- [OdiSftp](#)
- [OdiSftpGet](#)

- OdiSftpPut
- OdiSleep
- OdiSqlUnload
- OdiStartLoadPlan
- OdiStartOwbJob
- OdiStartScen
- OdiUnZip
- OdiUnlockOdiRepository
- OdiUpdateAgentSchedule
- OdiWaitForChildSession
- OdiWaitForData
- OdiWaitForLoadPlans
- OdiWaitForLogData
- OdiWaitForTable
- OdiXMLConcat
- OdiXMLSplit
- OdiZip

OdiAnt

Use this command to execute an Ant buildfile.

For more details and examples of Ant buildfiles, refer to the online documentation:
<http://jakarta.apache.org/ant/manual/index.html>

Usage

```
OdiAnt -BUILDFILE=<file> -LOGFILE=<file> [-TARGET=<target>]
[-D<property name>=<property value>]* [-PROJECTHELP] [-HELP]
[-VERSION] [-QUIET] [-VERBOSE] [-DEBUG] [-EMACS]
[-LOGGER=<classname>] [-LISTENER=<classname>] [-FIND=<file>]
```

Parameters

Parameters	Mandatory	Description
-BUILDFILE=<file>	Yes	Ant buildfile. XML file containing the Ant commands.
-LOGFILE=<file>	Yes	Use given file for logging.
-TARGET=<target>	No	Target of the build process.
-D<property name>=<property value>	No	List of properties with their values.
-PROJECTHELP	No	Displays the help on the project.
-HELP	No	Displays Ant help.
-VERSION	No	Displays Ant version.
-QUIET	No	Run in nonverbose mode.
-VERBOSE	No	Run in verbose mode.
-DEBUG	No	Prints debug information.
-EMACS	No	Displays the logging information without adornments.
-LOGGER=<classname>	No	Java class performing the logging.
-LISTENER=<classname>	No	Adds a class instance as a listener.
-FIND=<file>	No	Looks for the Ant buildfile from the root of the file system and uses it.

Examples

Download the *.html files from the directory /download/public using FTP from ftp.mycompany.com to the directory C:\temp.

Step 1: Generate the Ant buildfile.

```
OdiOutFile -FILE=c:\temp\ant_cmd.xml
<?xml version="1.0"?>
<project name="myproject" default="ftp" basedir="/">
  <target name="ftp">
    <ftp action="get" remotedir="/download/public"
server="ftp.mycompany.com" userid="anonymous"
password="me@mycompany.com">
      <fileset dir="c:\temp">
```

```
        <include name="**/*.html"/>
        </fileset>
    </ftp>
</target>
</project>
```

Step 2: Run the Ant buildfile.

```
OdiAnt -BUILDFILE=c:\temp\ant_cmd.xml -LOGFILE=c:\temp\ant_cmd.log
```

OdiBeep

Use this command to play a default beep or sound file on the machine hosting the agent.

The following file formats are supported by default:

- WAV
- AIF
- AU

Note: To play other file formats, you must add the appropriate JavaSound Service Provider Interface (JavaSound SPI) to the application classpath.

Usage

```
OdiBeep [-FILE=<sound_file>]
```

Parameters

Parameters	Mandatory	Description
-FILE	No	Path and file name of sound file to be played. If not specified, the default beep sound for the machine is used.

Examples

```
OdiBeep -FILE=c:\wav>alert.wav
```

OdiDeleteScen

Use this command to delete a given scenario version.

Usage

```
OdiDeleteScen -SCEN_NAME=<name> -SCEN_VERSION=<version>
```

Parameters

Parameters	Mandatory	Description
-SCEN_NAME=<name>	Yes	Name of the scenario to delete.
-SCEN_VERSION=<version>	Yes	Version of the scenario to delete.

Examples

Delete the DWH scenario in version 001.

```
OdiDeleteScen -SCEN_NAME=DWH -SCEN_VERSION=001
```

OdiEnterpriseDataQuality

Use this command to invoke an Oracle Enterprise Data Quality (Datanomic) job.

Note: The OdiEnterpriseDataQuality tool supports Oracle Enterprise Data Quality version 8.1.6 and later.

Usage

```
OdiEnterpriseDataQuality "-JOB_NAME=<EDQ job name>"
"-PROJECT_NAME=<EDQ project name>" "-CONTEXT=<context>"
"-LSHEMA=<logical_schema>" "-SYNCHRONOUS=<yes|no>"
```

Parameters

Parameters	Mandatory	Description
-JOB_NAME=<EDQ job name>	Yes	Name of the Enterprise Data Quality job.
-PROJECT_NAME=<EDQ project name>	Yes	Name of the Enterprise Data Quality project.
-SYNCHRONOUS=<yes no>	No	If set to Yes (default), the tool waits for the quality process to complete before returning, with possible error code. If set to No, the tool ends immediately with success and does not wait for the quality process to complete.

Examples

Execute the Enterprise Data Quality job CLEANSE_CUSTOMERS located in the project CUSTOMERS.

```
OdiEnterpriseDataQuality "-JOB_NAME=CLEANSE_CUSTOMERS" "-PROJECT_NAME=CUSTOMERS"
"-CONTEXT=Development" "-LSHEMA=EDQ Logical Schema" "-SYNCHRONOUS=yes"
```

OdiExportAllScen

Use this command to export a group of scenarios from the connected repository.

The export files are named SCEN_<scenario name><scenario version>.xml. This command reproduces the behavior of the export feature available in Designer Navigator and Operator Navigator.

Usage

```
OdiExportAllScen -TODIR=<directory> [-FORCE_OVERWRITE=<yes|no>]
[-FROM_PROJECT=<project_id>] [-FROM_FOLDER=<folder_id>]
[-FROM_PACKAGE=<package_id>] [-RECURSIVE_EXPORT=<yes|no>]
[-XML_VERSION=<1.0>] [-XML_CHARSET=<charset>]
[-JAVA_CHARSET=<charset>] [-EXPORT_KEY=<key>] [-EXPORT_MAPPING=<yes|no>]
[-EXPORT_PACK=<yes|no>] [-EXPORT_POP=<yes|no>]
[-EXPORT_TRT=<yes|no>] [-EXPORT_VAR=<yes|no>]
[EXPORT_WITHOUT_CIPHER_DATA=<yes|no>]
```

Parameters

Parameters	Mandatory	Description
-TODIR=<directory>	Yes	Directory into which the export files are created.
-FORCE_OVERWRITE=<yes no>	No	If set to Yes, existing export files are overwritten without warning. The default value is No.
-FROM_PROJECT=<project_id>	No	ID of the project containing the scenarios to export. This value is the Global ID that displays in the Version tab of the project window in Studio. If this parameter is not set, scenarios from all projects are taken into account for the export.
-FROM_FOLDER=<folder_id>	No	ID of the folder containing the scenarios to export. This value is the Global ID that displays in the Version tab of the folder window in Studio. If this parameter is not set, scenarios from all folders are taken into account for the export.
-FROM_PACKAGE=<package_id>	No	ID of the source package of the scenarios to export. This value is the Global ID that displays in the Version tab of the package window in Studio. If this parameter is not set, scenarios from all components are taken into account for the export.
-RECURSIVE_EXPORT=<yes no>	No	If set to Yes (default), all child objects (schedules) are exported with the scenarios.
-XML_VERSION=<1.0>	No	Sets the XML version shown in the XML header. The default value is 1.0.
-XML_CHARSET=<charset>	No	Encoding specified in the XML export file in the tag <?xml version="1.0" encoding="ISO-8859-1"?>. The default value is ISO-8859-1. For the list of supported encodings, see: http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html

Parameters	Mandatory	Description
-JAVA_CHARSET=<charset>	No	Target file encoding. The default value is ISO8859_1. For the list of supported encodings, see: http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html
-EXPORT_KEY=<key>	No ¹	Specifies a cryptographic private key used to encrypt sensitive cipher data. You must specify this key again when importing the exported object in order to import the cipher data.
-EXPORT_MAPPING=<yes no>	No	Indicates if the mapping scenarios should be exported. The default value is No.
-EXPORT_PACK=<yes no>	No	Indicates if the scenarios attached to packages should be exported. The default value is Yes.
-EXPORT_POP=<yes no>	No	Indicates if the scenarios attached to mappings should be exported. The default value is No.
-EXPORT_TRT=<yes no>	No	Indicates if the scenarios attached to procedures should be exported. The default value is No.
-EXPORT_VAR=<yes no>	No	Indicates if the scenarios attached to variables should be exported. The default value is No.
-EXPORT_WITHOUT_CIPHER_DATA=<yes no>	No ²	When set to Yes, specifies that sensitive (cipher) values should be set to null in the object when it is exported. When set to No or when this parameter is omitted, you must include the -EXPORT_KEY parameter and specify a valid key. The default value is No.

¹ If the -EXPORT_KEY parameter is not specified, the -EXPORT_WITHOUT_CIPHER_DATA parameter must be specified, and must be set to Yes.

² If -EXPORT_WITHOUT_CIPHER_DATA is not specified, or if it is specified and set to No, you must specify the -EXPORT_KEY parameter with a valid key value.

Examples

Export all scenarios from the DW01 project of **Global ID** 2edb524d-eb17-42ea-8aff-399ea9b13bf3 into the /temp/ directory, with all dependent objects, using the key examplekey1 to encrypt sensitive data.

```
OdiExportAllScen -FROM_PROJECT=2edb524d-eb17-42ea-8aff-399ea9b13bf3 -TODIR=/temp/
-RECURSIVE_EXPORT=yes -EXPORT_KEY=examplekey1
```

OdiExportEnvironmentInformation

Use this command to export the details of the technical environment into a comma separated (.csv) file into the directory of your choice. This information is required for maintenance or support purposes.

Usage

```
OdiExportEnvironmentInformation -TODIR=<toDir> -FILE_NAME=<FileName>
[-CHARSET=<charset>] [-SNP_INFO_REC_CODE=<row_code>]
[-MASTER_REC_CODE=<row_code>] [-WORK_REC_CODE=<row_code>]
[-AGENT_REC_CODE=<row_code>] [-TECHNO_REC_CODE=<row_code>]
[-RECORD_SEPARATOR_HEXA=<rec_sep>] [-FIELD_SEPARATOR_HEXA=<field_sep>]
[-TEXT_SEPARATOR=<text_sep>]
```

Parameter

Parameters	Mandatory	Description
-TODIR=<toDir>	Yes	Target directory for the export.
-FILE_NAME=<FileName>	Yes	Name of the CSV export file. The default value is snps_tech_inf.csv.
-CHARSET=<charset>	No	Character set of the export file.
-SNP_INFO_REC_CODE=<row_code>	No	Code used to identify rows that describe the current version of Oracle Data Integrator and the current user. This code is used in the first field of the record. The default value is SUNOPSIS.
-MASTER_REC_CODE=<row_code>	No	Code for rows containing information about the master repository. The default value is MASTER.
-WORK_REC_CODE=<row_code>	No	Code for rows containing information about the work repository. The default value is WORK.
-AGENT_REC_CODE=<row_code>	No	Code for rows containing information about the various agents that are running. The default value is AGENT.
-TECHNO_REC_CODE=<row_code>	No	Code for rows containing information about the data servers, their versions, and so on. The default value is TECHNO.
-RECORD_SEPARATOR_HEXA=<rec_sep>	No	One or several characters in hexadecimal code separating lines (or records) in the file. The default value is 00D0A.
-FIELD_SEPARATOR_HEXA=<field_sep>	No	One or several characters in hexadecimal code separating the fields in a record. The default value is 2C.
-TEXT_SEPARATOR=<text_sep>	No	Character in hexadecimal code delimiting a STRING field. The default value is 22.

Examples

Export the details of the technical environment into the /temp/snps_tech_inf.csv export file.

```
OdiExportEnvironmentInformation "-TODIR=/temp/"
"-FILE_NAME=snps_tech_inf.csv" "-CHARSET=ISO8859_1"
"-SNP_INFO_REC_CODE=SUNOPSIS" "-MASTER_REC_CODE=MASTER"
"-WORK_REC_CODE=WORK" "-AGENT_REC_CODE=AGENT"
"-TECHNO_REC_CODE=TECHNO" "-RECORD_SEPARATOR_HEX=0D0A"
"-FIELD_SEPARATOR_HEX=2C" "-TEXT_SEPARATOR_HEX=22"
```

OdiExportLog

Use this command to export the execution log into a ZIP export file.

Usage

```
OdiExportLog -TODIR=<toDir> [-EXPORT_TYPE=<logsToExport>] [-EXPORT_KEY=<key>]
[-ZIPFILE_NAME=<zipFileName>] [-XML_CHARSET=<charset>]
[-JAVA_CHARSET=<charset>] [-FROMDATE=<from_date>] [-TODATE=<to_date>]
[-AGENT=<agent>] [-CONTEXT=<context>] [-STATUS=<status>] [-USER_FILTER=<user>]
[-NAME=<sessionOrLoadPlanName>] [EXPORT_WITHOUT_CIPHER_DATA=<yes|no>]
```

Parameters

Parameters	Mandatory	Description
-EXPORT_TYPE=<logsToExport>	No	Export the log of: <ul style="list-style-type: none"> LOAD_PLAN_RUN: All Load Plan runs that match the export criteria are exported, including all sessions launched by the Load Plan runs along the child session's hierarchy. SESSION: All session logs that match the export filter criteria are exported. All Load Plan sessions will be excluded when exporting the session logs. ALL: All Load Plan runs and session logs that match the filter criteria are exported.
-EXPORT_KEY=<key>	No ¹	Specifies a cryptographic private key used to encrypt sensitive cipher data. You must specify this key again when importing the exported object in order to import the cipher data.
-TODIR=<toDir>	Yes	Target directory for the export.
-ZIPFILE_NAME=<zipFileName>	No	Name of the compressed file.
-XML_CHARSET=<charset>	No	XML version specified in the export file. Parameter <code>xml version</code> in the XML file header. <code><?xml version="1.0" encoding="ISO-8859-1"?></code> . The default value is ISO-8859-1. For the list of supported encodings, see: http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html
-JAVA_CHARSET=<charset>	No	Result file Java character encoding. The default value is ISO8859_1. For the list of supported encodings, see: http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html
-FROMDATE=<from_date>	No	Beginning date for the export, using the format <code>yyyy/MM/dd hh:mm:ss</code> . All sessions from this date are exported.
-TODATE=<to_date>	No	End date for the export, using the format <code>yyyy/MM/dd hh:mm:ss</code> . All sessions to this date are exported.

Parameters	Mandatory	Description
-AGENT=<agent>	No	Exports only sessions executed by the agent <agent>.
-CONTEXT=<context>	No	Exports only sessions executed in the context code <context>.
-STATUS=<status>	No	Exports only sessions in the specified state. Possible states are Done, Error, Queued, Running, Waiting, and Warning.
-USER_FILTER=<user>	No	Exports only sessions launched by <user>.
-NAME=<sessionOrLoadPlanName>	No	Name of the session or Load Plan to be exported.
-EXPORT_WITHOUT_CIPHER_DATA=<yes no>	No ²	When set to Yes, specifies that sensitive (cipher) values should be set to null in the object when it is exported. When set to No or when this parameter is omitted, you must include the -EXPORT_KEY parameter and specify a valid key. The default value is No.

¹ If the -EXPORT_KEY parameter is not specified, the -EXPORT_WITHOUT_CIPHER_DATA parameter must be specified, and must be set to Yes.

² If -EXPORT_WITHOUT_CIPHER_DATA is not specified, or if it is specified and set to No, you must specify the -EXPORT_KEY parameter with a valid key value.

Examples

Export and compress the log into the /temp/log2.zip export file.

```
OdiExportLog "-EXPORT_TYPE=ALL" "-EXPORT_KEY=examplekey1"
"-TODIR=/temp/" "-ZIPFILE_NAME=log2.zip" "-XML_CHARSET=ISO-8859-1"
"-JAVA_CHARSET=ISO8859_1"
```

OdiExportMaster

Use this command to export the master repository to a directory or ZIP file. The versions and/or solutions stored in the master repository are optionally exported.

Usage

```
OdiExportMaster -TODIR=<toDir> [-ZIPFILE_NAME=<zipFileName>]
[-EXPORT_KEY=<key>] [-EXPORT_SOLUTIONS=<yes|no>] [-EXPORT_VERSIONS=<yes|no>]
[-XML_CHARSET=<charset>] [-JAVA_CHARSET=<charset>]
[EXPORT_WITHOUT_CIPHER_DATA=<yes|no>]
```

Parameters

Parameters	Mandatory	Description
-TODIR=<toDir>	Yes	Target directory for the export.
-ZIPFILE_NAME=<zipFileName>	No	Name of the compressed file.
-EXPORT_KEY=<key>	No ¹	Specifies a cryptographic private key used to encrypt sensitive cipher data. You must specify this key again when importing the exported object in order to import the cipher data.
-EXPORT_SOLUTIONS=<yes no>	No	Exports all solutions that are stored in the repository. The default value is No.
-EXPORT_VERSIONS=<yes no>	No	Exports all versions of objects that are stored in the repository. The default value is No.
-XML_CHARSET=<charset>	No	XML version specified in the export file. Parameter <code>xml version="1.0" encoding="ISO-8859-1" ?></code> . The default value is ISO-8859-1. For the list of supported encodings, see: http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html
-JAVA_CHARSET=<charset>	No	Result file Java character encoding. The default value is ISO8859_1. For the list of supported encodings, see: http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html
-EXPORT_WITHOUT_CIPHER_DATA=<yes no>	No ²	When set to Yes, specifies that sensitive (cipher) values should be set to null in the object when it is exported. When set to No or when this parameter is omitted, you must include the <code>-EXPORT_KEY</code> parameter and specify a valid key. The default value is No.

¹ If the `-EXPORT_KEY` parameter is not specified, the `-EXPORT_WITHOUT_CIPHER_DATA` parameter must be specified, and must be set to Yes.

² If `-EXPORT_WITHOUT_CIPHER_DATA` is not specified, or if it is specified and set to No, you must specify the `-EXPORT_KEY` parameter with a valid key value.

Examples

Export and compress the master repository into the `export.zip` file located in the `/temp/` directory.

```
OdiExportMaster "-TODIR=/temp/" "-ZIPFILE_NAME=export.zip"  
"-EXPORT_KEY=examplekey1" "-XML_CHARSET=ISO-8859-1"  
"-JAVA_CHARSET=ISO8859_1" "-EXPORT_VERSIONS=YES"
```

OdiExportObject

Use this command to export an object from the current repository. This command reproduces the behavior of the export feature available in the user interface.

Usage

```
OdiExportObject -CLASS_NAME=<class_name> -I_OBJECT=<object_id>
[-EXPORT_KEY=<key>] [-EXPORT_DIR=<directory>]
[-EXPORT_NAME=<export_name>|-FILE_NAME=<file_name>] [-FORCE_OVERWRITE=<yes|no>]
[-RECURSIVE_EXPORT=<yes|no>] [-XML_VERSION=<1.0>] [-XML_CHARSET=<charset>]
[-JAVA_CHARSET=<charset>] [EXPORT_WITHOUT_CIPHER_DATA=<yes|no>]
```

Parameters

Parameters	Mandatory	Description
-CLASS_NAME=<class_name>	Yes	Class of the object to export (see the following list of classes).
-I_OBJECT=<object_id>	Yes	Object identifier. This value is the Global ID that displays in the Version tab of the object edit window.
-EXPORT_KEY=<key>	No ¹	Specifies a cryptographic private key used to encrypt sensitive cipher data. You must specify this key again when importing the exported object in order to import the cipher data.
-FILE_NAME=<file_name>	No	Export file name. Absolute path or relative path from EXPORT_DIR. This file name may or may not comply with the Oracle Data Integrator standard export file prefix and suffix. To comply with these standards, use the -EXPORT_NAME parameter instead. This parameter cannot be used if -EXPORT_NAME is set.
-EXPORT_DIR=<directory>	No	Directory where the object will be exported. The export file created in this directory is named based on the -FILE_NAME and -EXPORT_NAME parameters. If -FILE_NAME or -EXPORT_NAME are not specified, the export file is automatically named <object_prefix>_<object_name>.xml. For example, a project named Datawarehouse would be exported to PRJ_Datawarehouse.xml.
-EXPORT_NAME=<export_name>	No	Export name. Use this parameter to generate an export file named <object_prefix>_<export_name>.xml. This parameter cannot be used with -FILE_NAME.
-FORCE_OVERWRITE=<yes no>	No	If set to Yes, an existing export file with the same name is forcibly overwritten. The default value is No.
-RECURSIVE_EXPORT=<yes no>	No	If set to Yes (default), all child objects are exported with the current object. For example, if exporting a project, all folders, KMs, and so on in this project are exported into the project export file.
-XML_VERSION=<1.0>	No	Sets the XML version that appears in the XML header. The default value is 1.0.

Parameters	Mandatory	Description
-XML_CHARSET=<charset>	No	Encoding specified in the XML file, in the tag <?xml version="1.0" encoding="ISO-8859-1"?>. The default value is ISO-8859-1. For the list of supported encodings, see: http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html
-JAVA_CHARSET=<charset>	No	Target file encoding. The default value is ISO8859_1. For the list of supported encodings, see: http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html
-EXPORT_WITHOUT_CIPHER_DATA=<yes no>	No ²	When set to Yes, specifies that sensitive (cipher) values should be set to null in the object when it is exported. When set to No or when this parameter is omitted, you must include the -EXPORT_KEY parameter and specify a valid key. The default value is No.

¹ If the -EXPORT_KEY parameter is not specified, the -EXPORT_WITHOUT_CIPHER_DATA parameter must be specified, and must be set to Yes.

² If -EXPORT_WITHOUT_CIPHER_DATA is not specified, or if it is specified and set to No, you must specify the -EXPORT_KEY parameter with a valid key value.

List of Classes

Object	Class Name
Column	SnpCol
Condition/Filter	SnpCond
Context	SnpContext
Data Server	SnpConnect
Datastore	SnpTable
Folder	SnpFolder
Interface	SnpPop
Language	SnpLang
Loadplan	SnpLoadPlan
Mapping	SnpMapping
Model	SnpModel
Package	SnpPackage
Physical Schema	SnpPschema
Procedure or KM	SnpTrt
Procedure or KM Option	SnpUserExit
Project	SnpProject
Reference	SnpJoin
Reusable Mapping	SnpMapping
Scenario	SnpScen
Sequence	SnpSequence
Step	SnpStep

Object	Class Name
Sub-Model	SnpsubModel
Technology	SnpsTechno
User Functions	SnpsUfunc
Variable	SnpsVar
Version of an Object	SnpsVer

Examples

Export the DW01 project of **Global ID** 2edb524d-eb17-42ea-8aff-399ea9b13bf3 into the /temp/dw1.xml export file, with all dependent objects.

```
OdiExportObject -CLASS_NAME=SnpsProject  
-I_OBJECT=2edb524d-eb17-42ea-8aff-399ea9b13bf3 -EXPORT_KEY=examplekey1  
-FILE_NAME=/temp/dw1.xml -FORCE_OVERWRITE=yes  
-RECURSIVE_EXPORT=yes
```

OdiExportScen

Use this command to export a scenario from the current work repository.

Usage

```
OdiExportScen -SCEN_NAME=<scenario_name> -SCEN_VERSION=<scenario_version>
[-EXPORT_KEY=<key>] [-EXPORT_DIR=<directory>]
[-FILE_NAME=<file_name>|EXPORT_NAME=<export_name>] [-FORCE_OVERWRITE=<yes|no>]
[-RECURSIVE_EXPORT=<yes|no>] [-XML_VERSION=<1.0>] [-XML_CHARSET=<encoding>]
[-JAVA_CHARSET=<encoding>] [EXPORT_WITHOUT_CIPHER_DATA=<yes|no>]
```

Parameters

Parameters	Mandatory	Description
-SCEN_NAME=<scenario_name>	Yes	Name of the scenario to be exported.
-SCEN_VERSION=<scenario_version>	Yes	Version of the scenario to be exported.
-EXPORT_KEY=<key>	No ¹	Specifies a cryptographic private key used to encrypt sensitive cipher data. You must specify this key again when importing the exported object in order to import the cipher data.
-FILE_NAME=<file_name>	Yes	Export file name. Absolute path or relative path from -EXPORT_DIR. This file name may or not comply with the Oracle Data Integrator standard export file prefix and suffix for scenarios. To comply with these standards, use the -EXPORT_NAME parameter instead. This parameter cannot be used if -EXPORT_NAME is set.
-EXPORT_DIR=<directory>	No	Directory where the scenario will be exported. The export file created in this directory is named based on the -FILE_NAME and -EXPORT_NAME parameters. If -FILE_NAME or -EXPORT_NAME are not specified, the export file is automatically named SCEN_<scenario_name><scenario_version>.xml.
-EXPORT_NAME=<export_name>	No	Export name. Use this parameter to generate an export file named SCEN_<export_name>.xml. This parameter cannot be used with -FILE_NAME.
-FORCE_OVERWRITE=<yes no>	No	If set to Yes, overwrites the export file if it already exists. The default value is No.
-RECURSIVE_EXPORT=<yes no>	No	Forces the export of the objects under the scenario. The default value is Yes.
-XML_VERSION=<1.0>	No	Version specified in the generated XML file, in the tag <?xml version="1.0" encoding="ISO-8859-1"?>. The default value is 1.0.

Parameters	Mandatory	Description
-XML_CHARSET=<encoding>	No	Encoding specified in the XML file, in the tag <?xml version="1.0" encoding="ISO-8859-1"?>. The default value is ISO-8859-1. For the list of supported encodings, see: http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html
-JAVA_CHARSET=<encoding>	No	Target file encoding. The default value is ISO8859_1. For the list of supported encodings, see: http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html
-EXPORT_WITHOUT_CIPHER_DATA=<yes no>	No ²	When set to Yes, specifies that sensitive (cipher) values should be set to null in the object when it is exported. When set to No or when this parameter is omitted, you must include the -EXPORT_KEY parameter and specify a valid key. The default value is No.

¹ If the -EXPORT_KEY parameter is not specified, the -EXPORT_WITHOUT_CIPHER_DATA parameter must be specified, and must be set to Yes.

² If -EXPORT_WITHOUT_CIPHER_DATA is not specified, or if it is specified and set to No, you must specify the -EXPORT_KEY parameter with a valid key value.

Examples

Export the LOAD_DWH scenario in version 1 into the /temp/load_dwh.xml export file, with all dependent objects.

```
OdiExportScen -SCEN_NAME=LOAD_DWH -SCEN_VERSION=1 -EXPORT_KEY=examplekey1
-FILE_NAME=/temp/load_dwh.xml -RECURSIVE_EXPORT=yes
```

OdiExportWork

Use this command to export the work repository to a directory or ZIP export file.

Usage

```
OdiExportWork -TODIR=<directory> [-ZIPFILE_NAME=<zipFileName>]
[-EXPORT_KEY=<key>] [-XML_CHARSET=<charset>] [-JAVA_CHARSET=<charset>]
[EXPORT_WITHOUT_CIPHER_DATA=<yes|no>]
```

Parameters

Parameters	Mandatory	Description
-TODIR=<directory>	Yes	Target directory for the export.
-ZIPFILE_NAME=<zipFileName>	No	Name of the compressed file.
-EXPORT_KEY=<key>	No ¹	Specifies a cryptographic private key used to encrypt sensitive cipher data. You must specify this key again when importing the exported object in order to import the cipher data.
-XML_CHARSET=<charset>	No	XML version specified in the export file. Parameter <code>xml version</code> in the XML file header. <code><?xml version="1.0" encoding="ISO-8859-1"?></code> . The default value is ISO-8859-1. For the list of supported encodings, see: http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html
-JAVA_CHARSET=<charset>	No	Result file Java character encoding. The default value is ISO8859_1. For the list of supported encodings, see: http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html
-EXPORT_WITHOUT_CIPHER_DATA=<yes no>	No ²	When set to Yes, specifies that sensitive (cipher) values should be set to null in the object when it is exported. When set to No or when this parameter is omitted, you must include the <code>-EXPORT_KEY</code> parameter and specify a valid key. The default value is No.

¹ If the `-EXPORT_KEY` parameter is not specified, the `-EXPORT_WITHOUT_CIPHER_DATA` parameter must be specified, and must be set to Yes.

² If `-EXPORT_WITHOUT_CIPHER_DATA` is not specified, or if it is specified and set to No, you must specify the `-EXPORT_KEY` parameter with a valid key value.

Examples

Export and compress the work repository into the `/temp/workexport.zip` export file.

```
OdiExportWork "-TODIR=/temp/" "-ZIPFILE_NAME=workexport.zip"
"-EXPORT_KEY=examplekey1"
```

OdiFileAppend

Use this command to concatenate a set of files into a single file.

Usage

```
OdiFileAppend -FILE=<file> -TOFILE=<target_file> [-OVERWRITE=<yes|no>]
[-CASESENS=<yes|no>] [-HEADER=<n>] [-KEEP_FIRST_HEADER=<yes|no>]
```

Parameters

Parameters	Mandatory	Description
-FILE=<file>	Yes	Full path of the files to concatenate. Use * to specify generic characters. Examples: /var/tmp/*.log (all files with the log extension in the folder /var/tmp) arch_*.lst (all files starting with arch_ and with the extension lst) The file location is always relative to the data schema directory of its logical schema.
-TOFILE=<target_file>	Yes	Target file.
-OVERWRITE=<yes no>	No	Indicates if the target file must be overwritten if it already exists. The default value is No.
-CASESENS=<yes no>	No	Indicates if file search is case-sensitive. By default, Oracle Data Integrator searches files in uppercase (set to No).
-HEADER=<n>	No	Number of header lines to be removed from the source files before concatenation. By default, no lines are removed. When the -HEADER parameter is omitted, the concatenation does not require file edition, and therefore runs faster.
-KEEP_FIRST_HEADER=<yes no>	No	Keep the header lines of the first file during the concatenation. The default value is Yes.

Examples

Concatenate the files *.log of the folder /var/tmp into the file /home/all_files.log.

```
OdiFileAppend -FILE=/var/tmp/*.log -TOFILE=/home/all_files.log
```

OdiFileCopy

Use this command to copy files or folders.

Usage

```
OdiFileCopy -DIR=<directory> -TODIR=<target_directory> [-OVERWRITE=<yes|no>]
[-RECURSE=<yes|no>] [-CASESENS=<yes|no>]
```

```
OdiFileCopy -FILE=<file> -TOFILE=<target_file>|-TODIR=<target_directory>
[-OVERWRITE=<yes|no>] [-RECURSE=<yes|no>] [-CASESENS=<yes|no>]
```

Parameters

Parameters	Mandatory	Description
-DIR=<directory>	Yes if -FILE is omitted	Directory (or folder) to copy. The directory location is always relative to the data schema directory of its logical schema.
-FILE=<file>	Yes if -DIR is omitted	The full path of the files to copy. Use * to specify the generic character. Examples: /var/tmp/*.log (all files with the log extension in folder /var/tmp) arch_*.lst (all files starting with arch_ and with the extension lst) The file location is always relative to the data schema directory of its logical schema.
-TODIR=<target_directory>	Yes if -DIR is specified	Target directory for the copy. If a directory is copied (-DIR), this parameter indicates the name of the copied directory. If one or several files are copied (-FILE), this parameter indicates the destination directory.
-TOFILE=<target_file>	Yes if -TODIR is omitted	Destination file(s). This parameter cannot be used with parameter -DIR. This parameter contains: <ul style="list-style-type: none"> ■ The name of the destination file if only one file is copied (no generic character). ■ The mask of the new name of the destination files if several files are copied. Note that -TODIR and -TOFILE are exclusive parameters. If both are specified, only -TODIR is taken into account, and -TOFILE is ignored.

Parameters	Mandatory	Description
-TGT_LSCHEMA=<target_file>	No	The file located on a data server, based on the Logical Schema value. For example, the LSCHEMA may point to a Hadoop Data Server and the tool will access the file from that data server if the file needs to be accessed from HDFS.
-OVERWRITE=<yes no>	No	Indicates if the files of the folder are overwritten if they already exist. The default value is No.
-RECURSE=<yes no>	No	Indicates if files are copied recursively when the directory contains other directories. The value No indicates that only the files within the directory are copied, not the subdirectories. The default value is Yes.
-CASESENS=<yes no>	No	Indicates if file search is case-sensitive. By default, Oracle Data Integrator searches for files in uppercase (set to No).

Examples

Copy the file `hosts` from the directory `/etc` to the directory `/home`.

```
OdiFileCopy -FILE=/etc/hosts -TOFILE=/home/hosts
```

Copy all `*.csv` files from the directory `/etc` to the directory `/home` and overwrite.

```
OdiFileCopy -FILE=/etc/*.csv -TODIR=/home -OVERWRITE=yes
```

OdiFileDelete

Use this command to delete files or directories.

The most common uses of this tool are described in the following table where:

- x means is supplied
- o means is omitted

-DIR	-FILE	-RECURSE	Behavior
x	x	x	Every file with the name or with a name matching the mask specified in -FILE is deleted from -DIR and from all of its subdirectories.
x	o	x	The subdirectories from -FILE are deleted.
x	x	o	Every file with the name or with a name matching the mask specified in -FILE is deleted from -DIR.
x	o	o	The -DIR is deleted.

Usage

```
OdiFileDelete -DIR=<directory> -FILE=<file> [-RECURSE=<yes|no>]
[-CASESENS=<yes|no>] [-NOFILE_ERROR=<yes|no>] [-FROMDATE=<from_date>]
[-TODATE=<to_date>]
```

Parameters

Parameters	Mandatory	Description
-DIR=<directory>	Yes if -FILE is omitted	<p>If -FILE is omitted, specifies the name of the directory (folder) to delete.</p> <p>If -FILE is supplied, specifies the path where files should be deleted from.</p> <p>The directory location is always relative to the data schema directory of its logical schema.</p>
-FILE=<file>	Yes if -DIR is omitted	<p>Name or mask of file(s) to delete. If -DIR is not specified, provide the full path. Use * to specify wildcard characters.</p> <p>Examples:</p> <p><code>/var/tmp/*.log</code> (all files with the log extension of the directory <code>/var/tmp</code>)</p> <p><code>/arch_*.lst</code> (all files starting with <code>arch_</code> and with the extension <code>lst</code>)</p> <p>The file location is always relative to the data schema directory of its logical schema.</p>
-RECURSE=<yes no>	No	<p>If -FILE is omitted, the -RECURSE parameter has no effect: all subdirectories are implicitly deleted.</p> <p>If -FILE is supplied, the -RECURSE parameter specifies if the files should be deleted from this directory and from all of its subdirectories.</p> <p>The default value is Yes.</p>

Parameters	Mandatory	Description
-CASESENS=<yes no>	No	Specifies that Oracle Data Integrator should distinguish between uppercase and lowercase when matching file names. The default value is No.
-NOFILE_ERROR=<yes no>	Yes	Indicates that an error should be generated if the specified directory or files are not found. The default value is Yes.
-FROMDATE=<from_date>	No	All files with a modification date later than this date are deleted. Use the format yyyy/MM/dd hh:mm:ss. The -FROM_DATE is not inclusive. If -FROMDATE is omitted, all files with a modification date earlier than the -TODATE date are deleted. If both -FROMDATE and -TODATE are omitted, all files matching the -FILE parameter value are deleted.
-TODATE=<to_date>	No	All files with a modification date earlier than this date are deleted. Use the format yyyy/MM/dd hh:mm:ss. The TO_DATE is not inclusive. If -TODATE is omitted, all files with a modification date later than the -FROMDATE date are deleted. If both -FROMDATE and -TODATE parameters are omitted, all files matching the -FILE parameter value are deleted.

Note: You cannot delete a file and a directory at the same time by combining the -DIR and -FILE parameters. To achieve that, you must make two calls to OdiFileDelete.

Examples

Delete the file `my_data.dat` from the directory `c:\data\input`, generating an error if the file or directory is missing.

```
OdiFileDelete -FILE=c:\data\input\my_data.dat -NOFILE_ERROR=yes
```

Delete all `.txt` files from the `bin` directory, but not `.TXT` files.

```
OdiFileDelete "-FILE=c:\Program Files\odi\bin\*.txt" -CASESENS=yes
```

This statement has the same effect:

```
OdiFileDelete "-DIR=c:\Program Files\odi\bin" "-FILE=*.txt" -CASESENS=yes
```

Delete the directory `/bin/usr/nothingToDoHere`.

```
OdiFileDelete "-DIR=/bin/usr/nothingToDoHere"
```

Delete all files under the `C:\temp` directory whose modification time is between `10/01/2008 00:00:00` and `10/31/2008 22:59:00`, where `10/01/2008` and `10/31/2008` are not inclusive.

```
OdiFileDelete -DIR=C:\temp -FILE=* -NOFILE_ERROR=NO -FROMDATE=FROMDATE=10/01/2008 00:00:00 -TODATE=10/31/2008 22:59:00
```

Delete all files under the `C:\temp` directory whose modification time is earlier than `10/31/2008 17:00:00`.

```
OdiFileDelete -DIR=C:\temp -FILE=* -NOFILE_ERROR=YES -TODATE=10/31/2008 17:00:00
```

Delete all files under the `C:\temp` directory whose modification time is later than `10/01/2008 08:00:00`.

```
OdiFileDelete -DIR=C:\temp -FILE=* -NOFILE_ERROR=NO -FROMDATE=10/01/2008 08:00:00
```

OdiFileMove

Use this command to move or rename files or a directory into files or a directory.

Usage

```
OdiFileMove -FILE=<file> -TODIR=<target_directory> -TOFILE=<target_file>
[-OVERWRITE=<yes|no>] [-RECURSE=<yes|no>] [-CASESENS=<yes|no>]
```

```
OdiFileMove -DIR=<directory> -TODIR=<target_directory> [-OVERWRITE=<yes|no>]
[-RECURSE=<yes|no>] [-CASESENS=<yes|no>]
```

Parameters

Parameters	Mandatory	Description
-DIR=<directory>	Yes if -FILE is omitted	Directory (or folder) to move or rename. The directory location is always relative to the data schema directory of its logical schema.
-FILE=<file>	Yes if -DIR is omitted	Full path of the file(s) to move or rename. Use * for generic characters. Examples: <i>/var/tmp/*.log</i> (all files with the log extension in the directory <i>/var/tmp</i>) <i>arch_*.lst</i> (all files starting with arch_ and with the extension lst) The file location is always relative to the data schema directory of its logical schema.
-TODIR=<target_directory>	Yes if -DIR is specified	Target directory of the move. If a directory is moved (-DIR), this parameter indicates the new name of the directory. If a file or several files are moved (-FILE), this parameter indicates the target directory.
-TOFILE=<target_file>	Yes if -TODIR is omitted	Target file(s). This parameter cannot be used with parameter -DIR. This parameter is: <ul style="list-style-type: none"> ■ The new name of the target file if one single file is moved (no generic character). ■ The mask of the new file names if several files are moved.
-OVERWRITE=<yes no>	No	Indicates if the files or directory are overwritten if they exist. The default value is No.
-RECURSE=<yes no>	No	Indicates if files are moved recursively when the directory contains other directories. The value No indicates that only files contained in the directory to move (not the subdirectories) are moved. The default value is Yes.

Parameters	Mandatory	Description
-CASESENS=<yes no>	No	Indicates if file search is case-sensitive. By default, Oracle Data Integrator searches for files in uppercase (set to No).

Examples

Rename the hosts file to hosts.old.

```
OdiFileMove -FILE=/etc/hosts -TOFILE=/etc/hosts.old
```

Move the file hosts from the directory /etc to the directory /home/odi.

```
OdiFileMove -FILE=/etc/hosts -TOFILE=/home/odi/hosts
```

Move all files *.csv from directory /etc to directory /home/odi with overwrite.

```
OdiFileMove -FILE=/etc/*.csv -TODIR=/home/odi -OVERWRITE=yes
```

Move all *.csv files from directory /etc to directory /home/odi and change their extension to .txt.

```
OdiFileMove -FILE=/etc/*.csv -TOFILE=/home/odi/*.txt -OVERWRITE=yes
```

Rename the directory C:\odi to C:\odi_is_wonderful.

```
OdiFileMove -DIR=C:\odi -TODIR=C:\odi_is_wonderful
```

Move the directory C:\odi and its subfolders into the directory C:\Program Files\odi.

```
OdiFileMove -DIR=C:\odi "-TODIR=C:\Program Files\odi" -RECURSE=yes
```

OdiFileWait

Use this command to manage file events. This command regularly scans a directory and waits for a number of files matching a mask to appear, until a given timeout is reached. When the specified files are found, an action on these files is triggered.

Usage

```
OdiFileWait -DIR=<directory> -PATTERN=<pattern>
[-ACTION=<DELETE|COPY|MOVE|APPEND|ZIP|NONE>] [-TODIR=<target_directory>]
[-TOFILE=<target_file>] [-OVERWRITE=<yes|no>] [-CASESENS=<yes|no>]
[-FILECOUNT=<n>] [-TIMEOUT=<n>] [-POLLINT=<n>] [-HEADER=<n>]
[-KEEP_FIRST_HEADER=<yes|no>] [-NOFILE_ERROR=<yes|no>]
```

Parameters

Parameters	Mandatory	Description
-ACTION= <DELETE COPY MOVE APPEND ZIP NONE>	No	Action taken on the files found: DELETE: Delete the files found. COPY: Copy the files found into the directory -TODIR. MOVE: Move or rename the files found into folder -TODIR by naming them as specified by -TOFILE. APPEND: Concatenates all files found and creates a result file -TOFILE. Source files are deleted. ZIP: Compress the files found and store them in ZIP file -TOFILE. NONE (default): No action is performed.
-DIR=<directory>	Yes	Directory (or folder) to scan. The directory location is always relative to the data schema directory of its logical schema.
-PATTERN=<pattern>	Yes	Mask of file names to scan. Use * to specify the generic characters. Examples: *.log (all files with the log extension) arch_*.lst (all files starting with arch_ and with the extension lst)
-TODIR=<target_directory>	No	Target directory of the action. When the action is: COPY: Directory where the files are copied. MOVE: Directory where the files are moved.

Parameters	Mandatory	Description
-TOFILE=<target_file>	No	<p>Destination file(s). When the action is:</p> <p>MOVE: Renaming mask of the moved files.</p> <p>APPEND: Name of the file resulting from the concatenation.</p> <p>ZIP: Name of the resulting ZIP file.</p> <p>COPY: Renaming mask of the copied files.</p> <p>Renaming rules:</p> <ul style="list-style-type: none"> ■ Any alphanumeric character is replaced in the original file name with the alphanumeric characters specified for <target_file>. ■ ? at -TOFILE leaves origin symbol on this position. ■ * at -TOFILE means all remaining symbols from origin file name.
-OVERWRITE=<yes no>	No	<p>Indicates if the destination file(s) will be overwritten if they exist. The default value is No.</p> <p>Note that if this option is used with APPEND, the target file will only contain the contents of the latest file processed.</p>
-CASESENS=<yes no>	No	<p>Indicates if file search is case-sensitive. By default, Oracle Data Integrator searches files in uppercase (set to No).</p>
-FILECOUNT=<n>	No	<p>Maximum number of files to wait for (the default value is 0). If this number is reached, the command ends.</p> <p>The value 0 indicates that Oracle Data Integrator waits for all files until the timeout is reached.</p> <p>If this parameter is 0 and the timeout is also 0, this parameter is then forced implicitly to 1.</p>
-TIMEOUT=<n>	No	<p>Maximum waiting time in milliseconds (the default value is 0).</p> <p>If this delay is reached, the command yields control to the following command and uses its value -FILECOUNT.</p> <p>The value 0 is used to specify an infinite waiting time (wait until the maximum number of messages to read as specified in the parameter -FILECOUNT).</p>
-POLLINT=<n>	No	<p>Interval in milliseconds to search for new files. The default value is 1000 (1 second), which means that Oracle Data Integrator looks for new messages every second. Files written during the OdiFileWait are taken into account only after being closed (file size unchanged) during this interval.</p>

Parameters	Mandatory	Description
-HEADER=<n>	No	This parameter is valid only for the APPEND action. Number of header lines to suppress from the files before concatenation. The default value is 0 (no processing).
-KEEP_FIRST_HEADER=<yes no>	No	This parameter is valid only for the APPEND action. Keeps the header lines of the first file during the concatenation. The default value is Yes.
-NOFILE_ERROR=<yes no>	No	Indicates the behavior if no file is found. The default value is No, which means that no error is generated if no file is found.

Examples

Wait indefinitely for file `flag.txt` in directory `c:\events` and proceed when this file is detected.

```
OdiFileWait -ACTION=NONE -DIR=c:\events -PATTERN=flag.txt -FILECOUNT=1
-TIMEOUT=0 -POLLINT=1000
```

Wait indefinitely for file `flag.txt` in directory `c:\events` and suppress this file when it is detected.

```
OdiFileWait -ACTION=DELETE -DIR=c:\events -PATTERN=flag.txt -FILECOUNT=1
-TIMEOUT=0 -POLLINT=1000
```

Wait for the sales files `*.dat` for 5 minutes and scan every second in directory `c:\sales_in`, then concatenate into file `sales.dat` in directory `C:\sales_ok`. Keep the header of the first file.

```
OdiFileWait -ACTION=APPEND -DIR=c:\sales_in -PATTERN=*.dat
TOFILE=c:\sales_ok\sales.dat -FILECOUNT=0 -TIMEOUT=350000 -POLLINT=1000
-HEADER=1 -KEEP_FIRST_HEADER=yes -OVERWRITE=yes
```

Wait for the sales files `*.dat` for 5 minutes every second in directory `c:\sales_in`, then copy these files into directory `C:\sales_ok`. Do not overwrite.

```
OdiFileWait -ACTION=COPY -DIR=c:\sales_in -PATTERN=*.dat -TODIR=c:\sales_ok
-FILECOUNT=0 -TIMEOUT=350000 -POLLINT=1000 -OVERWRITE=no
```

Wait for the sales files `*.dat` for 5 minutes every second in directory `c:\sales_in` and then archive these files into a ZIP file.

```
OdiFileWait -ACTION=ZIP -DIR=c:\sales_in -PATTERN=*.dat
-TOFILE=c:\sales_ok\sales.zip -FILECOUNT=0 -TIMEOUT=350000
-POLLINT=1000 -OVERWRITE=yes
```

Wait for the sales files `*.dat` for 5 minutes every second into directory `c:\sales_in`, then move these files into directory `C:\sales_ok`. Do not overwrite. Append `.bak` to the file names.

```
OdiFileWait -ACTION=MOVE -DIR=c:\sales_in -PATTERN=*.dat
-TODIR=c:\sales_ok -TOFILE=*.bak -FILECOUNT=0 -TIMEOUT=350000
-POLLINT=1000 -OVERWRITE=no
```

OdiFtp

Use this command to use the FTP protocol to connect to a remote system and to perform standard FTP commands on the remote system. Trace from the script is recorded against the Execution Details of the task representing the OdiFtp step in Operator Navigator.

Usage

```
OdiFtp -HOST=<ftp server host name> -USER=<ftp user>
[-PASSWORD=<ftp user password>] -REMOTE_DIR=<remote dir on ftp host>
-LOCAL_DIR=<local dir> [-PASSIVE_MODE=<yes|no>] [-TIMEOUT=<time in seconds>]
[-STOP_ON_FTP_ERROR=<yes|no>] -COMMAND=<command>
```

Parameters

Parameters	Mandatory	Description
-HOST=<ftp server host name>	Yes	Host name of the FTP server.
-USER=<ftp user>	Yes	User on the FTP server.
-PASSWORD=<ftp user password>	No	Password of the FTP user.
-REMOTE_DIR=<remote dir on ftp host>	Yes	Directory path on the remote FTP host.
-LOCAL_DIR=<local dir>	Yes	Directory path on the local machine.
-PASSIVE_MODE=<yes no>	No	If set to No, the FTP session uses Active Mode. The default value is Yes, which means the session runs in passive mode.
-TIMEOUT=<time in seconds>	No	Time in seconds after which the socket connection times out.
-STOP_ON_FTP_ERROR=<yes no>	No	If set to Yes (default), the step stops when an FTP error occurs instead of running to completion.
-COMMAND=<command>	Yes	Raw FTP command to execute. For a multiline command, pass the whole command as raw text after the OdiFtp line without the -COMMAND parameter. Supported commands: APPE, CDUP, CWD, DELE, LIST, MKD, NLST, PWD, QUIT, RETR, RMD, RNFR, RNTO, SIZE, STOR

Examples

Execute a script on a remote host that makes a directory, changes directory into the directory, puts a file into the directory, and checks its size. The script appends another file, checks the new size, and then renames the file to `dailyData.csv`. The `-STOP_ON_FTP_ERROR` parameter is set to No so that the script continues even if the directory exists.

```
OdiFtp -HOST=machine.oracle.com -USER=odiftpuser -PASSWORD=<password>
-LOCAL_DIR=/tmp -REMOTE_DIR=c:\temp -PASSIVE_MODE=YES -STOP_ON_FTP_ERROR=No
MKD dataDir
CWD dataDir
STOR customers.csv
```

```
SIZE customers.csv  
APPE new_customers.csv customers.csv  
SIZE customers.csv  
RNFR customers.csv  
RNTO dailyData.csv
```

OdiFtpGet

Use this command to download a file from an FTP server.

Usage

```
OdiFtpGet -HOST=<ftp server host name> -USER=<ftp user>
[PASSWORD=<ftp user password>] -REMOTE_DIR=<remote dir on ftp host>
[-REMOTE_FILE=<file name under the -REMOTE_DIR>] -LOCAL_DIR=<local dir>
[-LOCAL_FILE=<file name under the -LOCAL_DIR>] [-PASSIVE_MODE=<yes|no>]
[-TIMEOUT=<time in seconds>]
```

Note: If a Local or Remote file name needs to have % as part of its name, %25 needs to be passed instead of just %.

%25 will resolve automatically to %.

For example, if file name needs to be temp%result, it should be passed as REMOTE_FILE=temp%25result or -LOCAL_FILE=temp%25result.

Parameters

Parameters	Mandatory	Description
-HOST=<host name of the ftp server>	Yes	Host name of the FTP server.
-USER=<host name of the ftp user>	Yes	User on the FTP server.
-PASSWORD=<password of the ftp user>	No	Password of the FTP user.
-REMOTE_DIR=<dir on the ftp host>	Yes	Directory path on the remote FTP host.
-REMOTE_FILE=<file name under -REMOTE DIR>	No	File name under the directory specified in the -REMOTE_DIR argument. If this argument is missing, the file is copied with the -LOCAL_FILE file name. If the -LOCAL_FILE argument is also missing, the -LOCAL_DIR is copied recursively to the -REMOTE_DIR.
-LOCAL_DIR=<local dir path>	Yes	Directory path on the local machine.
-LOCAL_FILE=<local file>	No	File name under the directory specified in the -LOCAL_DIR argument. If this argument is missing, all files and directories under the -LOCAL_DIR are copied recursively to the -REMOTE_DIR. To filter the files to be copied, use * to specify the generic characters. Examples: <ul style="list-style-type: none"> ▪ *.log (all files with the log extension) ▪ arch_*.lst (all files starting with arch_ and with the extension lst)
-PASSIVE_MODE=<yes no>]	No	If set to No, the FTP session uses Active Mode. The default value is Yes, which means the session runs in passive mode.

Parameters	Mandatory	Description
-TIMEOUT=<time in seconds>	No	The time in seconds after which the socket connection times out.
-TGT_LSCHEMA=<target_file>	No	The file located on a data server resolved based on the Logical Schema value. For example, the LSCHEMA may point to a Hadoop Data Server and the tool will access the file from that data server.

Examples

Copy the remote directory /test_copy555 on the FTP server recursively to the local directory C:\temp\test_copy.

```
OdiFtpGet -HOST=machine.oracle.com -USER=test_ftp -PASSWORD=<password> -LOCAL_
DIR=C:\temp\test_copy -REMOTE_DIR=/test_copy555
```

Copy all files matching the Sales*.txt pattern under the remote directory / on the FTP server to the local directory C:\temp\ using Active Mode for the FTP connection.

```
OdiFtpGet -HOST=machine.oracle.com -USER=test_ftp -PASSWORD=<password> -LOCAL_
DIR=C:\temp -LOCAL_FILE=Sales*.txt -REMOTE_DIR=/ -PASSIVE_MODE=NO
```

OdiFtpPut

Use this command to upload a local file to an FTP server.

Usage

```
OdiFtpPut -HOST=<ftp server host name> -USER=<ftp user>
[PASSWORD=<ftp user password>] -REMOTE_DIR=<remote dir on ftp host>
[-REMOTE_FILE=<file name under the -REMOTE_DIR>] -LOCAL_DIR=<local dir>
[-LOCAL_FILE=<file name under the -LOCAL_DIR>] [-PASSIVE_MODE=<yes|no>]
[-TIMEOUT=<time in seconds>]
```

Note: If a Local or Remote file name needs to have % as part of its name, %25 needs to be passed instead of just %.

%25 will resolve automatically to %.

For example, if file name needs to be temp%result, it should be passed as REMOTE_FILE=temp%25result or -LOCAL_FILE=temp%25result.

Parameters

Parameters	Mandatory	Description
-HOST=<host name of the ftp server>	Yes	Host name of the FTP server.
-USER=<host name of the ftp user>	Yes	User on the FTP server.
-PASSWORD=<password of the ftp user>	No	Password of the FTP user.
-REMOTE_DIR=<dir on the ftp host>	Yes	Directory path on the remote FTP host.
-REMOTE_FILE=<file name under -REMOTE DIR>	No	File name under the directory specified in the -REMOTE_DIR argument. If this argument is missing, the file is copied with the -LOCAL_FILE file name. If the -LOCAL_FILE argument is also missing, the -LOCAL_DIR is copied recursively to the -REMOTE_DIR.
-LOCAL_DIR=<local dir path>	Yes	Directory path on the local machine.
-LOCAL_FILE=<local file>	No	File name under the directory specified in the -LOCAL_DIR argument. If this argument is missing, all files and directories under the -LOCAL_DIR are copied recursively to the -REMOTE_DIR. To filter the files to be copied, use * to specify the generic characters. Examples: <ul style="list-style-type: none"> ■ *.log (all files with the log extension) ■ arch_*.lst (all files starting with arch_ and with the extension lst)
-PASSIVE_MODE=<yes no>	No	If set to No, the FTP session uses Active Mode. The default value is Yes, which means the session runs in passive mode.

Parameters	Mandatory	Description
-TIMEOUT=<time in seconds>	No	The time in seconds after which the socket connection times out.

Note: For OdiFtp execution to be successful, you must have LIST privilege in the user's home directory.

Examples

Copy the local directory C:\temp\test_copy recursively to the remote directory /test_copy555 on the FTP server.

```
OdiFtpPut -HOST=machine.oracle.com -USER=test_ftp -PASSWORD=<password>
-LLOCAL_DIR=C:\temp\test_copy -REMOTE_DIR=/test_copy555"
```

Copy all files matching the Sales*.txt pattern under the local directory C:\temp\ to the remote directory / on the FTP server.

```
OdiFtpPut -HOST=machine.oracle.com -USER=test_ftp -PASSWORD=<password> -LOCAL_
DIR=C:\temp -LOCAL_FILE=Sales*.txt -REMOTE_DIR=/"
```

Copy the Sales1.txt file under the local directory C:\temp\ to the remote directory / on the FTP server as a Sample1.txt file.

```
OdiFtpPut -HOST=machine.oracle.com -USER=test_ftp -PASSWORD=<password> -LOCAL_
DIR=C:\temp -LOCAL_FILE=Sales1.txt -REMOTE_DIR=/Sample1.txt
```

OdiGenerateAllScen

Use this command to generate a set of scenarios from design-time components (Packages, Mappings, Procedures, or Variables) contained in a folder or project, filtered by markers.

Usage

```
OdiGenerateAllScen -PROJECT=<project_id> [-FOLDER=<folder_id>]
[-MODE=<REPLACE|CREATE>] [-GRPMARKER=<marker_group_code>]
[-MARKER=<marker_code>] [-MATERIALIZED=<yes|no>]
[-GENERATE_MAP=<yes|no>] [-GENERATE_PACK=<yes|no>]
[-GENERATE_POP=<yes|no>] [-GENERATE_TRT=<yes|no>]
[-GENERATE_VAR=<yes|no>]
```

Parameters

Parameters	Mandatory	Description
-PROJECT=<project_id>	Yes	ID of the Project containing the components to generate scenarios for.
-FOLDER=<folder_id>	No	ID of the Folder containing the components to generate scenarios for.
-MODE=<REPLACE CREATE>	No	Scenario generation mode: <ul style="list-style-type: none"> REPLACE (default): Causes the last scenario generated for the component to be replaced by the new one generated, with no change of name or version. Any schedules linked to this scenario are deleted. If no scenario exists, a new one is generated. CREATE: Creates a new scenario with the same name as the latest scenario generated for the component, with the version number automatically incremented (if the latest version is an integer) or set to the current date (if the latest version is not an integer). If no scenario has been created for the component, a first version of the scenario is automatically created. New scenarios are named after the component according to the <i>Scenario Naming Convention</i> user parameter.
-GRPMARKER=<marker_group_code>	No	Group containing the marker used to filter the components for which scenarios must be generated. When -GRPMARKER and -MARKER are specified, scenarios will be (re-)generated only for components flagged with the marker identified by the marker code and the marker group code.

Parameters	Mandatory	Description
-MARKER=<marker_code>	No	Marker used to filter the components for which scenarios must be generated. When -GRPMARKER and -MARKER are specified, scenarios will be (re-)generated only for components flagged with the marker identified by the marker code and the marker group code.
-MATERIALIZED=<yes no>	No	Specifies whether scenarios should be generated as if all underlying objects are materialized. The default value is No.
-GENERATE_MAP=<yes no>	No	Specifies whether scenarios should be generated from the mapping. The default value is No.
-GENERATE_PACK=<yes no>	No	Specifies whether scenarios attached to packages should be (re-)generated. The default value is Yes.
-GENERATE_POP=<yes no>	No	Specifies whether scenarios attached to mappings should be (re-)generated. The default value is No.
-GENERATE_TRT=<yes no>	No	Specifies whether scenarios attached to procedures should be (re-)generated. The default value is No.
-GENERATE_VAR=<yes no>	No	Specifies whether scenarios attached to variables should be (re-)generated. The default value is No.

Examples

Generate all scenarios in the project whose ID is 1003 for the current repository.

```
OdiGenerateAllScen -PROJECT=1003
```

OdiImportObject

Use this command to import the contents of an export file into a repository. This command reproduces the behavior of the import feature available from the user interface.

Use caution when using this tool. It may work incorrectly when importing objects that depend on objects that do not exist in the repository. It is recommended that you use this API for importing high-level objects (projects, models, and so on).

WARNING: The import type and the order in which objects are imported into a repository should be carefully specified. Refer to the chapter *Exporting and Importing in Developing Integration Projects with Oracle Data Integrator* for more information on import.

Usage

```
OdiImportObject -FILE_NAME=<FileName> [-WORK_REP_NAME=<workRepositoryName>]
-IMPORT_MODE=<DUPLICATION|SYNONYM_INSERT|SYNONYM_UPDATE|SYNONYM_INSERT_UPDATE>]
[-IMPORT_SCHEDULE=<yes|no>] [-EXPORT_KEY=<key>] [-UPGRADE_KEY=<upgradeKey>]
[IMPORT_WITHOUT_CIPHER_DATA=<yes|no>]
```

Parameters

Parameters	Mandatory	Description
-FILE_NAME=<FileName>	Yes	Name of the XML export file to import.
-WORK_REP_NAME=<workRepositoryName>	No	Name of the work repository into which the object must be imported. This work repository must be defined in the connected master repository. If this parameter is not specified, the object is imported into the current master or work repository.
-IMPORT_MODE=<DUPLICATION SYNONYM_INSERT SYNONYM_UPDATE SYNONYM_INSERT_UPDATE>	Yes	Import mode for the object. The default value is DUPLICATION. For more information about import types, see <i>Import Modes in Developing Integration Projects with Oracle Data Integrator</i> .
-IMPORT_SCHEDULE=<yes no>	No	If the selected file is a scenario export, imports the schedules contained in the scenario export file. The default value is No.
-EXPORT_KEY=<key>	No ¹	Specifies a cryptographic private key used to encrypt sensitive cipher data. You must specify this key when importing the exported object in order to import the cipher data.
-UPGRADE_KEY=<upgradeKey>	No	Upgrade key to import repository objects from earlier versions of Oracle Data Integrator (pre-12c).

Parameters	Mandatory	Description
-IMPORT_WITHOUT_CIPHER_DATA=<yes no>	No ²	When set to Yes, specifies that sensitive (cipher) values should be set to null in the object when it is imported. When set to No or when this parameter is omitted, you must include the -EXPORT_KEY parameter and specify a valid key. The default value is No.

¹ If the -EXPORT_KEY parameter is not specified, the -IMPORT_WITHOUT_CIPHER_DATA parameter must be specified, and must be set to Yes.

² If -EXPORT_WITHOUT_CIPHER_DATA is not specified, or if it is specified and set to No, you must specify the -EXPORT_KEY parameter with a valid key value.

Examples

Import the /temp/DW01.xml export file (a project) into the WORKREP work repository using DUPLICATION mode.

```
OdiImportObject -FILE_NAME=/temp/DW01.xml -WORK_REP_NAME=WORKREP
-IMPORT_MODE=DUPLICATION -EXPORT_KEY=examplekey1
```

OdiImportScen

Use this command to import a scenario into the current work repository from an export file.

Usage

```
OdiImportScen -FILE_NAME=<FileName>
[-IMPORT_MODE=<DUPLICATION|SYNONYM_INSERT|SYNONYM_UPDATE|SYNONYM_INSERT_UPDATE>]
[-EXPORT_KEY=<key>] [-IMPORT_SCHEDULE=<yes|no>] [-FOLDER=<parentFolderGlobalId>]
[-UPGRADE_KEY=<upgradeKey>] [IMPORT_WITHOUT_CIPHER_DATA=<yes|no>]
```

Parameters

Parameters	Mandatory	Description
-FILE_NAME=<FileName>	Yes	Name of the export file.
-IMPORT_MODE=<DUPLICATION SYNONYM_INSERT SYNONYM_UPDATE SYNONYM_INSERT_UPDATE>	No	Import mode of the scenario. The default value is DUPLICATION. For more information about import types, see <i>Import Modes in Developing Integration Projects with Oracle Data Integrator</i> .
-EXPORT_KEY=<key>	No ¹	Specifies a cryptographic private key used to encrypt sensitive cipher data. You must specify this key when importing the exported object in order to import the cipher data.
-IMPORT_SCHEDULE=<yes no>	No	Imports the schedules contained in the scenario export file. The default value is No.
-FOLDER=<parentFolderGlobalId>	No	Global ID of the parent scenario folder.
-UPGRADE_KEY=<upgradeKey>	No	Upgrade key to import repository objects from earlier versions of Oracle Data Integrator (pre-12c).
-IMPORT_WITHOUT_CIPHER_DATA=<yes no>	No ²	When set to Yes, specifies that sensitive (cipher) values should be set to null in the object when it is imported. When set to No or when this parameter is omitted, you must include the -EXPORT_KEY parameter and specify a valid key. The default value is No.

¹ If the -EXPORT_KEY parameter is not specified, the -IMPORT_WITHOUT_CIPHER_DATA parameter must be specified, and must be set to Yes.

² If -IMPORT_WITHOUT_CIPHER_DATA is not specified, or if it is specified and set to No, you must specify the -EXPORT_KEY parameter with a valid key value.

Examples

Import the /temp/load_dwh.xml export file (a scenario) into the current work repository using DUPLICATION mode.

```
OdiImportScen -FILE_NAME=/temp/load_dwh.xml -IMPORT_MODE=DUPLICATION
-EXPORT_KEY=examplekey1
```

OdiInvokeWebService

Note: This tool replaces the OdiExecuteWebService tool.

Use this command to invoke a web service over HTTP/HTTPS and write the response to an XML file.

This tool invokes a specific *operation* on a *port* of a web service whose description file (WSDL) *URL* is provided.

If this operation requires a web service request, it is provided either in a request file, or directly written out in the tool call (<XML Request>). This request file can have two different formats (XML, which corresponds to the XML body only, or SOAP, which corresponds to the full-formed SOAP envelope including a SOAP header and body) specified in the `-RESPONSE_FILE_FORMAT` parameter. The response of the web service request is written to an XML file that can be processed afterwards in Oracle Data Integrator. If the web service operation is one-way and does not return any response, no response file is generated.

Note: This tool cannot be executed in a command line with `startcmd`.

Usage

```
OdiInvokeWebService -URL=<url> -PORT=<port> -OPERATION=<operation>
[<XML Request>] [-REQUEST_FILE=<xml_request_file>]
[-RESPONSE_MODE=<NO_FILE|NEW_FILE|FILE_APPEND>]
[-RESPONSE_FILE=<xml_response_file>] [-RESPONSE_XML_ENCODING=<charset>]
[-RESPONSE_FILE_CHARSET=<charset>] [-RESPONSE_FILE_FORMAT=<XML|SOAP>]
[-HTTP_USER=<user>]
[-HTTP_PASS=<password>] [-TIMEOUT=<timeout>]
```

Parameters

Parameters	Mandatory	Description
<code>-LSHEMA=<logical_schema></code>	No	Logical schema containing the journalized tables (optional parameter). If LSHEMA is specified, then OdiInvokeWebService will use URL, PORT, HTTP_USER, and HTTP_PASS configured at mapped SOAP WS Physical Schema and/or SOAP WS Data Server.
<code>-CONTEXT=<odi context></code>	No	Context in which the logical schema will be resolved. If no context is specified, the execution context is used (optional parameter).
<code>-URL=<url></code>	No	URL of the Web Service Description File (WSDL) describing the web service.
<code>-PORT_TYPE=<port_type></code>	No	Name of the WSDL port type to invoke.
<code>-OPERATION=<operation></code>	Yes	Name of the web service operation to invoke.

Parameters	Mandatory	Description
<XML Request>	No	Request message in SOAP (Simple Object Access Protocol) format. This message should be provided on the line immediately following the OdiInvokeWebService call. The request can alternately be passed through a file whose location is provided with the <code>-REQUEST_FILE</code> parameter.
<code>-REQUEST_FILE=<xml_request_file></code>	No	Location of the XML file containing the request message in SOAP format. The request can alternately be directly written out in the tool call (<code><xmlRequest></code>).
<code>-RESPONSE_MODE=<NO_FILE NEW_FILE FILE_APPEND></code>	No	Generation mode for the response file. This parameter takes the following values: <ul style="list-style-type: none"> ■ <code>NO_FILE</code> (default): No response file is generated. ■ <code>NEW_FILE</code>: A new response file is generated. If the file already exists, it is overwritten. ■ <code>FILE_APPEND</code>: The response is appended to the file. If the file does not exist, it is created.
<code>-RESPONSE_FILE=<file></code>	Depends	The name of the result file to write. Mandatory if <code>-RESPONSE_MODE</code> is <code>NEW_FILE</code> or <code>APPEND</code> .
<code>-RESPONSE_FILE_CHARSET=<charset></code>	Depends	Response file character encoding. See the following table. Mandatory if <code>-RESPONSE_MODE</code> is <code>NEW_FILE</code> or <code>APPEND</code> .
<code>-RESPONSE_XML_ENCODING=<charset></code>	Depends	Character encoding that will be indicated in the XML declaration header of the response file. See the following table. Mandatory if <code>-RESPONSE_MODE</code> is not <code>NO_FILE</code> .
<code>-RESPONSE_FILE_FORMAT=<XML SOAP></code>	No	Format of the request and response file. <ul style="list-style-type: none"> ■ If <code>XML</code> is selected (default), the request is processed as a SOAP body. The tool adds a default SOAP header and envelope content to this body before sending the request. The response is stripped from its SOAP envelope and headers and only the response's body is written to the response file. ■ If <code>SOAP</code> is selected, the request is processed as a full-formed SOAP envelope and is sent as is. The response is also written to the response file with no processing.
<code>-HTTP_USER=<user></code>	No	User account authenticating on the HTTP server.

Parameters	Mandatory	Description
-HTTP_PASS=<password>	No	Password of the HTTP user. Note: When using an ODI variable as the password, the variable content must be encrypted using the encode script.
-TIMEOUT=<timeout>	No	The web service request waits for a reply for this amount of time before considering that the server will not provide a response and an error is produced. The default value is 15 seconds.

The following table lists the most common XML/Java character encoding schemes. For a more complete list, see:

<http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html>

XML Charset	Java Charset
US-ASCII	ASCII
UTF-8	UTF8
UTF-16	UTF-16
ISO-8859-1	ISO8859_1

Examples

The following web service call returns the capital city for a given country (the ISO country code is sent in the request). Note that the request and response format, as well as the port and operations available, are defined in the WSDL passed in the URL parameter.

```
OdiInvokeWebService -
-URL=http://www.oorsprong.org/websamples.countryinfo/CountryInfoService.wso
?WSDL -PORT_TYPE=CountryInfoServiceSoapType -OPERATION=CapitalCity
-RESPONSE_MODE=NEW_FILE -RESPONSE_XML_ENCODING=ISO-8859-1
"-RESPONSE_FILE=/temp/result.xml" -RESPONSE_FILE_CHARSET=ISO8859_1 -RESPONSE_FILE_
FORMAT=XML
<CapitalCityRequest>
<sCountryISOCODE>US</sCountryISOCODE>
</CapitalCityRequest>
```

The generated /temp/result.xml file contains the following:

```
<CapitalCityResponse>
<m:CapitalCityResponse>
<m:CapitalCityResult>Washington</m:CapitalCityResult>
</m:CapitalCityResponse>
</CapitalCityResponse>
```

Packages

Oracle Data Integrator provides a special graphical interface for calling OdiInvokeWebService in packages. See the chapter Using Web Services in *Developing Integration Projects with Oracle Data Integrator* for more information.

OdiKillAgent

Use this command to stop a standalone agent.

Java EE Agents deployed in an application server cannot be stopped using this tool and must be stopped using the application server utilities.

Usage

```
OdiKillAgent (-PORT=<TCP/IP Port>|-NAME=<physical_agent_name>)
[-IMMEDIATE=<yes|no>] [-MAX_WAIT=<timeout>]
```

Parameter

Parameters	Mandatory	Description
-PORT=<TCP/IP Port>	No	If this parameter is specified, the agent running on the local machine with the specified port is stopped.
-NAME=<physical_agent_name>	Yes	If this parameter is specified, the physical agent whose name is provided is stopped. This agent may be a local or remote agent. It must be declared in the master repository.
-IMMEDIATE=<yes no>	No	If this parameter is set to Yes, the agent is stopped without waiting for its running sessions to complete. If this parameter is set to No, the agent is stopped after its running sessions reach completion or after the -MAX_WAIT timeout is reached. The default value is No.
-MAX_WAIT=<timeout>	No	This parameter can be used when -IMMEDIATE is set to No. The parameter defines a timeout in milliseconds after which the agent is stopped regardless of the running sessions. The default value is 0, which means no timeout and the agent is stopped after its running sessions complete.

Examples

Stop the ODI_AGT_001 physical agent immediately.

```
OdiKillAgent -NAME=ODI_AGT_001 -IMMEDIATE=yes
```

OdiManageOggProcess

Use this command to start and stop Oracle GoldenGate processes.

The `-NB_PROCESS` parameter specifies the number of processes on which to perform the operation and applies only to Oracle GoldenGate Delivery processes.

If `-NB_PROCESS` is not specified, the name of the physical process is derived from the logical process. For example, if logical schema `R1_LS` maps to physical process `R1`, an Oracle GoldenGate process named `R1` is started or stopped.

If `-NB_PROCESS` is specified with a positive value, sequence numbers are appended to the process and all processes are started or stopped with the new name. For example, if the value is set to 3, and logical schema `R2_LS` maps to physical process `R2`, processes `R21`, `R22` and `R23` are started or stopped.

If `Start Journal` is used to start the CDC (Changed Data Capture) process with Oracle GoldenGate JKMs (Journalizing Knowledge Modules), Oracle Data Integrator generates the Oracle GoldenGate Delivery process with the additional sequence number in the process name. For example, if Delivery process `RP` is used for the `Start Journal` action, `Start Journal` generates an Oracle GoldenGate Delivery process named `RP1`. To stop and start the process using the `OdiManageOggProcess` tool, set `-NB_PROCESS` to 1. The maximum value of `-NB_PROCESS` is the value of the `-NB_APPLY_PROCESS` parameter of the JKM within the model.

Usage

```
OdiManageOggProcess -OPERATION=<start|stop>
-PROCESS_LSCHEMA=<OGG logical schema> [-NB_PROCESS=<number of processes>]
```

Parameters

Parameters	Mandatory	Description
<code>-OPERATION=<start stop></code>	Yes	Operation to perform on the process.
<code>-PROCESS_LSCHEMA=<OGG logical schema></code>	Yes	Logical schema of the process.
<code>-NB_PROCESS=<number of processes></code>	No	Number of processes on which to perform the operation.

Examples

Start Oracle GoldenGate process `R1`, which maps to logical schema `R1_LS`.

```
OdiManageOggProcess "-OPERATION=START" "-PROCESS_LSCHEMA=R1_LS"
```

OdiMkDir

Use this command to create a directory structure.

If the parent directory does not exist, this command recursively creates the parent directories.

Usage

```
OdiMkDir -DIR=<directory>
```

Parameters

Parameters	Mandatory	Description
-DIR=<directory>	Yes	Directory (or folder) to create.
-TO_HDFS=<yes no>	No	Indicates if the target is HDFS

Examples

Create the directory `odi` in `C:\temp`. If `C:\temp` does not exist, it is created.

```
OdiMkDir "-DIR=C:\temp\odi"
```

OdiOSCommand

Use this command to invoke an operating system command shell to carry out a command, and redirect the output result to files.

The following operating systems are supported:

- Windows operating systems, using `cmd`
- POSIX-compliant operating systems, using `sh`

The following operating system is not supported:

- Mac OS

Usage

```
OdiOSCommand [-OUT_FILE=<stdout_file>] [-ERR_FILE=<stderr_file>]
[-FILE_APPEND=<yes|no>] [-WORKING_DIR=<workingdir>] [-SYNCHRONOUS=<yes|no>]
[CR/LF <command> | -COMMAND=<command>]
```

Parameters

Parameters	Mandatory	Description
-COMMAND=<command>	Yes	Command to execute. For a multiline command, pass the whole command as raw text after the OdiOSCommand line without the -COMMAND parameter.
-OUT_FILE=<stdout_file>	No	Absolute name of the file to redirect standard output to.
-ERR_FILE=<stderr_file>	No	Absolute name of the file to redirect standard error to.
-FILE_APPEND=<yes no>	No	Whether to append to the output files, rather than overwriting them. The default value is Yes.
-WORKING_DIR=<workingdir>	No	Directory in which the command is executed.
-SYNCHRONOUS=<yes no>	No	If set to Yes (default), the session waits for the command to terminate. If set to No, the session continues immediately with error code 0. The default is synchronous mode.
-CAPTURE_OUT_STREAM=[ON_ERROR[,]][ALL NONE [NSTART[,NEND]]]	No	Use to capture some of the content that is written to the output stream and display in the Task Execution details in Operator. If set to ON_ERROR , the content will be captured only if the task fails. If set to ALL or NONE , either all or none of the output stream will be captured. Use NSTART and NEND to specify the number of lines to be captured (from the start and end).

Parameters	Mandatory	Description
<code>-CAPTURE_ERR_STREAM=[ON_ERROR[,]][ALL NONE][NSTART[, NEND]]</code>	No	Use to capture some of the content that is written to the error stream and display in the Task Error Message in Operator. If set to ON_ERROR , the content will be captured only if the task fails. If set to ALL or NONE , either all or none of the output stream will be captured. Use NSTART and NEND to specify the number of lines to be captured (from the start and end).

Examples

Execute the file `c:\work\load.bat` (on a Windows machine) and append the output streams to files.

```
OdiOSCommand "-OUT_FILE=c:\work\load-out.txt"
"-ERR_FILE=c:\work\load-err.txt" "-FILE_APPEND=YES"
"-WORKING_DIR=c:\work" c:\work\load.bat
```

OdiOutFile

Use this command to write or append content to a text file.

Usage

```
OdiOutFile -FILE=<file_name> [-APPEND] [-CHARSET_ENCODING=<encoding>]
[-XROW_SEP=<hexadecimal_line_break>] [CR/LF <text> | -TEXT=<text>]
```

Parameters

Parameters	Mandatory	Description
-FILE=<file_name>	Yes	Target file. The file location is always relative to the data schema directory of its logical schema.
-APPEND	No	Indicates whether <text> must be appended at the end of the file. If this parameter is not specified, the file is overwritten if it exists.
-CHARSET_ENCODING=<encoding>	No	Target file encoding. The default value is ISO-8859-1. For the list of supported encodings, see: http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html
-XROW_SEP=<hexadecimal_line_break>	No	Hexadecimal code of the character used as a line separator (line break). The default value is 0A (UNIX line break). For a Windows line break, the value is 0D0A.
CR/LF <text> or -TEXT=<text>	No	Text to write in the file. This text can be typed on the line following the OdiOutFile command (a carriage return - CR/LF - indicates the beginning of the text), or can be defined with the -TEXT parameter. The -TEXT parameter should be used when calling this Oracle Data Integrator command from an OS command line. The text can contain variables or substitution methods.
-TO_HDFS=<yes no>	No	Indicates if the output file is created in HDFS
-TGT_LSCHEMA	No	Indicates if the file is located on a data server resolved based on the Logical Schema value.

Examples

Generate the file `/var/tmp/my_file.txt` on the UNIX system of the agent that executed it.

```
OdiOutFile -FILE=/var/tmp/my_file.txt
Welcome to Oracle Data Integrator
```

```
This file has been overwritten by <%=odiRef.getSession("SESS_NAME")%>
```

Add the entry PLUTON into the file hosts of the Windows system of the agent that executed it.

```
OdiOutFile -FILE=C:\winnt\system32\drivers\etc\hosts -APPEND
195.10.10.6 PLUTON pluton
```

OdiPingAgent

Use this command to perform a test on a given agent. If the agent is not started, this command raises an error.

Usage

```
OdiPingAgent -AGENT_NAME=<physical_agent_name>
```

Parameters

Parameters	Mandatory	Description
-AGENT_NAME=<physical_agent_name>	Yes	Name of the physical agent to test.

Examples

Test the physical agent AGENT_SOLARIS_DEV.

```
OdiPingAgent -AGENT_NAME=AGENT_SOLARIS_DEV
```

OdiPurgeLog

Use this command to purge the execution logs.

The OdiPurgeLog tool purges all session logs and/or Load Plan runs that match the filter criteria.

The `-PURGE_TYPE` parameter defines the objects to purge:

- Select `SESSION` to purge all session logs matching the criteria. Child sessions and grandchild sessions are purged if the parent session matches the criteria. Note that sessions launched by a Load Plan execution, including the child sessions, are *not* purged.
- Select `LOAD_PLAN_RUN` to purge all load plan logs matching the criteria. Note that all sessions launched from the Load Plan run are purged even if the sessions attached to the Load Plan runs themselves do not match the criteria.
- Select `ALL` to purge both session logs and Load Plan runs matching the criteria.

The `-COUNT` parameter defines the number of sessions and/or Load Plan runs (after filter) to preserve in the log. The `-ARCHIVE` parameter enables automatic archiving of the purged sessions and/or Load Plan runs.

Note: Load Plans and sessions in running, waiting, or queued status are not purged.

Usage

```
OdiPurgeLog
[-PURGE_TYPE=<SESSION|LOAD_PLAN_RUN|ALL>]
[-COUNT=<session_number>] [-FROMDATE=<from_date>] [TODATE=<to_date>]
[-CONTEXT_CODE=<context_code>] [-USER_NAME=<user_name>]
[-AGENT_NAME=<agent_name>] [-PURGE_REPORTS=<Yes|No>] [-STATUS=<D|E|M>]
[-NAME=<session_or_load_plan_name>] [-ARCHIVE=<Yes|No>] [-EXPORT_KEY=<key>]
[-TODIR=<directory>]
[-ZIPFILE_NAME=<zipfile_name>] [-XML_CHARSET=<charset>] [-JAVA_CHARSET=<charset>]
[-REMOVE_TEMPORARY_OBJECTS=<yes|no>] [ARCHIVE_WITHOUT_CIPHER_DATA=<yes|no>]
```

Parameters

Parameters	Mandatory	Description
<code>-PURGE_TYPE=<SESSION LOAD_PLAN_RUN ALL></code>	No	Purges only session logs, Load Plan logs, or both. The default is session.
<code>-COUNT=<session_number></code>	No	Retains the most recent count number of sessions and/or Load Plan runs that match the specified filter criteria and purges the rest. If this parameter is not specified or equals 0, purges all sessions and/or Load Plan runs that match the filter criteria.
<code>-FROMDATE=<from_date></code>	No	Starting date for the purge, using the format <code>yyyy/MM/dd hh:mm:ss</code> . If <code>-FROMDATE</code> is omitted, the purge is done starting with the oldest session and/or Load Plan run.

Parameters	Mandatory	Description
-TODATE=<to_date>	No	Ending date for the purge, using the format yyyy/MM/dd hh:mm:ss. If -TODATE is omitted, the purge is done up to the most recent session and/or Load Plan run.
-CONTEXT_CODE=<context_code>	No	Purges only sessions and/or Load Plan runs executed in <context_code>. If -CONTEXT_CODE is omitted, the purge is done on all contexts.
-USER_NAME=<user_name>	No	Purges only sessions and/or Load Plan runs launched by <user_name>.
-AGENT_NAME=<agent_name>	No	Purges only sessions and/or Load Plan runs executed by <agent_name>.
-PURGE_REPORTS=<0 1>	No	If set to 1, scenario reports (appearing under the execution node of each scenario) are also purged.
-STATUS=<D E M>	No	Purges only the sessions and/or Load Plan runs with the specified state: <ul style="list-style-type: none"> ■ D: Done ■ E: Error ■ M: Warning If this parameter is not specified, sessions and/or Load Plan runs in all of these states are purged.
-NAME=<session_or_load_plan_name>	No	Session name or Load Plan name.
-ARCHIVE=<Yes No>	No	If set to Yes, exports the sessions and/or Load Plan runs before they are purged.
-EXPORT_KEY=<key>	No ¹	Specifies a cryptographic private key used to encrypt sensitive cipher data. You must specify this key again when importing the exported object in order to import the cipher data.
-ARCHIVE_WITHOUT_CIPHER_DATA=<yes no>	No ²	When set to Yes, specifies that sensitive (cipher) values should be set to null in the object when it is archived. When set to No or when this parameter is omitted, you must include the -EXPORT_KEY parameter and specify a valid key. The default value is No.
-TODIR=<directory>	No	Target directory for the export. This parameter is required if -ARCHIVE is set to Yes.
-ZIPFILE_NAME=<zipfile_name>	No	Name of the compressed file. Target directory for the export. This parameter is required if -ARCHIVE is set to Yes.
-XML_CHARSET=<charset>	No	XML encoding of the export files. The default value is ISO-8859-1. For the list of supported encodings, see: http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html
-JAVA_CHARSET=<charset>	No	Export file encoding. The default value is ISO8859_1. For the list of supported encodings, see: http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html

Parameters	Mandatory	Description
-REMOVE_TEMPORARY_OBJECTS=<yes no>	No	If set to Yes (default), cleanup tasks are performed before sessions are purged so that any temporary objects are removed.

¹ If the -EXPORT_KEY parameter is not specified, the -ARCHIVE_WITHOUT_CIPHER_DATA parameter must be specified, and must be set to Yes.

² If -ARCHIVE_WITHOUT_CIPHER_DATA is not specified, or if it is specified and set to No, you must specify the -EXPORT_KEY parameter with a valid key value.

Examples

Purge all sessions executed between 2001/03/25 00:00:00 and 2001/08/31 21:59:00.

```
OdiPurgeLog "-FROMDATE=2001/03/25 00:00:00" "-TODATE=2001/08/31 21:59:00"
```

Purge all Load Plan runs that were executed in the GLOBAL context by the Internal agent and that are in Error status.

```
OdiPurgeLog "-PURGE_TYPE=LOAD_PLAN_RUN" "-CONTEXT_CODE=GLOBAL" "-AGENT_NAME=Internal" "-STATUS=E"
```

OdiReadMail

Use this command to read emails and attachments from a POP or IMAP account.

This command connects the mail server `-MAILHOST` using the connection parameters specified by `-USER` and `-PASS`. The execution agent reads messages from the mailbox until `-MAX_MSG` messages are received or the maximum waiting time specified by `-TIMEOUT` is reached. The extracted messages must match the filters such as those specified by the parameters `-SUBJECT` and `-SENDER`. When a message satisfies these criteria, its content and its attachments are extracted in a directory specified by the parameter `-FOLDER`. If the parameter `-KEEP` is set to No, the retrieved message is suppressed from the mailbox.

Usage

```
OdiReadMail -MAILHOST=<mail_host> -USER=<mail_user>
-PASS=<mail_user_password> -FOLDER=<folder_path>
[-PROTOCOL=<pop3|imap>] [-FOLDER_OPT=<none|sender|subject>]
[-KEEP=<no|yes>] [-EXTRACT_MSG=<yes|no>] [-EXTRACT_ATT=<yes|no>]
[-MSG_PRF=<my_prefix>] [-ATT_PRF=<my_prefix>] [-USE_UCASE=<no|yes>]
[-NOMAIL_ERROR=<no|yes>] [-TIMEOUT=<timeout>] [-POLLINT=<pollint>]
[-MAX_MSG=<max_msg>] [-SUBJECT=<subject_filter>] [-SENDER=<sender_filter>]
[-TO=<to_filter>] [-CC=<cc_filter>]
```

Parameters

Parameters	Mandatory	Description
<code>-MAILHOST=<mail_host></code>	Yes	IP address of the POP or IMAP mail server.
<code>-USER=<mail_user></code>	Yes	Valid mail server account.
<code>-PASS=<mail_user_password></code>	Yes	Password of the mail server account.
<code>-FOLDER=<folder_path></code>	Yes	Full path of the storage folder for attachments and messages.
<code>-PROTOCOL=<pop3 imap></code>	No	Type of mail accessed (POP3 or IMAP). The default is POP3.
<code>-FOLDER_OPT=<none sender subject></code>	No	Allows the creation of a subdirectory in the directory <code>-FOLDER</code> according to the following parameters: <ul style="list-style-type: none"> ■ <code>none</code> (default): No action. ■ <code>sender</code>: A subdirectory is created with the external name of the sender. ■ <code>subject</code>: A subdirectory is created with the subject of the message. For the <code>sender</code> and <code>subject</code> folder options, the spaces and nonalphanumeric characters (such as @) are replaced by underscores in the generated folder's name.

Parameters	Mandatory	Description
-KEEP=<no yes>	No	If set to Yes, keeps the messages that match the filters in the mailbox after reading them. If set to No (default), deletes the messages that match the filters of the mailbox after reading them.
-EXTRACT_MSG=<yes no>	No	If set to Yes (default), extracts the body of the message into a file. If set to No, does not extract the body of the message into a file.
-EXTRACT_ATT=<yes no>	No	If set to Yes (default), extracts the attachments into files. If set to No, does not extract attachments.
-MSG_PRF=<my_prefix>	No	Prefix of the file that contains the body of the message. The default is MSG.
-ATT_PRF=<my_prefix>	No	Prefix of the files that contain the attachments. The original file names are kept.
-USE_UCASE=<no yes>	No	If set to Yes, forces the file names to uppercase. If set to No (default), keeps the original letter case.
-NOMAIL_ERROR=<no yes>	No	If set to Yes, generates an error when no mail matches the specified criteria. If set to No (default), does not generate an error when no mail corresponds to the specified criteria.
-TIMEOUT=<timeout>	No	Maximum waiting time in milliseconds. If this waiting time is reached, the command ends. The default value is 0, which means an infinite waiting time (as long as needed for the maximum number of messages specified with -MAX_MSG to be received).
-POLLINT=<pollint>	No	Searching interval in milliseconds to scan for new messages. The default value is 1000 (1 second).
-MAX_MSG=<max_msg>	No	Maximum number of messages to extract. If this number is reached, the command ends. The default value is 1.
-SUBJECT=<subject_filter>	No	Parameter used to filter the messages according to their subjects.
-SENDER=<sender_filter>	No	Parameter used to filter messages according to their sender.
-TO=<to_filter>	No	Parameter used to filter messages according to their addresses. This option can be repeated to create multiple filters.
-CC=<cc_filter>	No	Parameter used to filter messages according to their addresses in copy. This option can be repeated to create multiple filters.

Examples

Automatic reception of the mails of support with attachments detached in the folder C:\support on the system of the agent. Wait for all messages with a maximum waiting time of 10 seconds.

```
OdiReadMail -MAILHOST=mail.mymail.com -USER=myaccount -PASS=mypass  
-KEEP=no -FOLDER=c:\support -TIMEOUT=0 -MAX_MSG=0  
-SENDER=support@mycompany.com -EXTRACT_MSG=yes -MSG_PRF=TXT  
-EXTRACT_ATT=yes
```

Wait indefinitely for 10 messages and check for new messages every minute.

```
OdiReadMail -MAILHOST=mail.mymail.com -USER=myaccount -PASS=mypass  
-KEEP=no -FOLDER=c:\support -TIMEOUT=0 -MAX_MSG=10 -POLLINT=60000  
-SENDER=support@mycompany.com -EXTRACT_MSG=yes -MSG_PRF=TXT  
-EXTRACT_ATT=yes
```

OdiRefreshJournalCount

Use this command to refresh for a given journalizing subscriber the number of rows to consume for the given table list or CDC set. This refresh is performed on a logical schema and a given context, and may be limited.

Note: This command is suitable for journalized tables in simple or consistent mode and cannot be executed in a command line with `startcmd`.

Usage

```
OdiRefreshJournalCount -LSHEMA=<logical_schema>
-SUBSCRIBER_NAME=<subscriber_name>
(-TABLE_NAME=<table_name> | -CDC_SET_NAME=<cdc set name>)
[-CONTEXT=<context>] [-MAX_JRN_DATE=<to_date>]
```

Parameters

Parameters	Mandatory	Description
-LSHEMA=<logical_schema>	Yes	Logical schema containing the journalized tables.
-TABLE_NAME=<table_name>	Yes for working with simple CDC	<p>Journalized table name, mask, or list to check. This parameter accepts three formats:</p> <ul style="list-style-type: none"> ■ Table Name ■ Table Name Mask: This mask selects the tables to poll. The mask is specified using the SQL LIKE syntax: the % symbol replaces an unspecified number of characters and the _ symbol acts as a wildcard. ■ Table Names List: List of table names separated by commas. Masks as defined above are not allowed. <p>Note that this option works only for tables in a model journalized in simple mode.</p> <p>This parameter cannot be used with -CDC_SET_NAME. It is mandatory if -CDC_SET_NAME is not set.</p>
-CDC_SET_NAME=<cdcSetName>	Yes for working with consistent set CDC	<p>Name of the CDC set to check.</p> <p>Note that this option works only for tables in a model journalized in consistent mode.</p> <p>This parameter cannot be used with -TABLE_NAME. It is mandatory if -TABLE_NAME is not set.</p>
-SUBSCRIBER_NAME=<subscriber_name>	Yes	Name of the subscriber for which the count is refreshed.
-CONTEXT=<context>	No	Context in which the logical schema will be resolved. If no context is specified, the execution context is used.
-MAX_JRN_DATE=<to_date>	No	Date (and time) until which the journalizing events are taken into account.

Examples

Refresh for the CUSTOMERS table in the SALES_APPLICATION schema the count of modifications recorded for the SALES_SYNC subscriber. This datastore is journalized in simple mode.

```
OdiRefreshJournalCount -LSHEMA=SALES_APPLICATION  
-TABLE_NAME=CUSTOMERS -SUBSCRIBER_NAME=SALES_SYNC
```

Refresh for all tables from the SALES CDC set in the SALES_APPLICATION schema the count of modifications recorded for the SALES_SYNC subscriber. These datastores are journalized with consistent set CDC.

```
OdiRefreshJournalCount -LSHEMA=SALES_APPLICATION  
-SUBSCRIBER_NAME=SALES_SYNC -CDC_SET_NAME=SALES
```

OdiReinitializeSeq

Use this command to reinitialize an Oracle Data Integrator sequence.

Usage

```
OdiReinitializeSeq -SEQ_NAME=<sequence_name> -CONTEXT=<context>
-STD_POS=<position>
```

Parameters

Parameters	Mandatory	Description
-SEQ_NAME=<sequence_name>	Yes	Name of the sequence to reinitialize. It must be prefixed with GLOBAL. for a global sequence, or by <project code> . for a project sequence.
-CONTEXT=<context>	Yes	Context in which the sequence must be reinitialized.
-STD_POS=<position>	Yes	Position to which the sequence must be reinitialized.

Examples

Reset the global sequence SEQ_I to 0 for the GLOBAL context.

```
OdiReinitializeSeq -SEQ_NAME=GLOBAL.SEQ_I -CONTEXT=GLOBAL
-STD_POS=0
```

OdiRemoveTemporaryObjects

Use this command to remove temporary objects that could remain between executions. This is performed by executing the cleanup tasks for the sessions identified by the parameters specified in the tool parameters.

Usage

```
OdiRemoveTemporaryObjects [-COUNT=<session_number>] [-FROMDATE=<from_date>]
[-TODATE=<to_date>] [-CONTEXT_CODE=<context_code>]
[-AGENT_NAME=<agent_name>] [-USER_NAME=<user_name>]
[-NAME=<session_name>] [-ERRORS_ALLOWED=<number_of_errors_allowed>]
```

Parameter

Parameters	Mandatory	Description
-COUNT=<session_number>	No	Number of sessions to skip cleanup for. The most recent number of sessions (<session_number>) is kept and the rest are cleaned up.
-FROMDATE=<from_date>	No	Start date for the cleanup, using the format yyyy/MM/dd hh:mm:ss. All sessions started after this date are cleaned up. If -FROMDATE is omitted, the cleanup starts with the oldest session.
-TODATE=<to_date>	No	End date for the cleanup, using the format yyyy/MM/dd hh:mm:ss. All sessions started before this date are cleaned up. If -TODATE is omitted, the cleanup starts with the most recent session.
-CONTEXT_CODE=<context_code>	No	Cleans up only those sessions executed in this context (<context_code>). If -CONTEXT_CODE is omitted, cleanup is performed on all contexts.
-AGENT_NAME=<agent_name>	No	Cleans up only those sessions executed by this agent (<agent_name>).
-USER_NAME=<user_name>	No	Cleans up only those sessions launched by this user (<user_name>).
-NAME=<session_name>	No	Session name.
-ERRORS_ALLOWED=<number_of_errors_allowed>	No	Number of errors allowed before the step ends with OK. If set to 0, the step ends with OK regardless of the number of errors encountered during the cleanup phase.

Examples

Remove the temporary objects by performing the cleanup tasks of all sessions executed between 2013/03/25 00:00:00 and 2013/08/31 21:59:00.

```
OdiRemoveTemporaryObjects "-FROMDATE=2013/03/25 00:00:00" "-TODATE=2013/08/31 21:59:00"
```

Remove the temporary objects by performing the cleanup tasks of all sessions executed in the GLOBAL context by the Internal agent.

```
OdiRemoveTemporaryObjects "-CONTEXT_CODE=GLOBAL" "-AGENT_NAME=Internal"
```

OdiRetrieveJournalData

Use this command to retrieve the journalized events for a given journalizing subscriber, a given table list or CDC set. The retrieval is performed specifically for the technology containing the tables. This retrieval is performed on a logical schema and a given context.

Note: This tool works for tables journalized using simple or consistent set modes and cannot be executed in a command line with startcmd.

Usage

```
OdiRetrieveJournalData -LSHEMA=<logical_schema>
-SUBSCRIBER_NAME=<subscriber_name>
(-TABLE_NAME=<table_name> | -CDC_SET_NAME=<cdc_set_name>)
[-CONTEXT=<context>] [-MAX_JRN_DATE=<to_date>]
```

Parameters

Parameters	Mandatory	Description
-LSHEMA=<logical_schema>	Yes	Logical schema containing the journalized tables.
-TABLE_NAME=<table_name>	No	Journalized table name, mask, or list to check. This parameter accepts three formats: <ul style="list-style-type: none"> Table Name Table Name Mask: This mask selects the tables to poll. The mask is specified using the SQL LIKE syntax: the % symbol replaces an unspecified number of characters and the _ symbol acts as a wildcard. Table Names List: List of table names separated by commas. Masks as defined above are not allowed. <p>Note that this option works only for tables in a model journalized in simple mode.</p> <p>This parameter cannot be used with -CDC_SET_NAME. It is mandatory if -CDC_SET_NAME is not set.</p>
-CDC_SET_NAME=<cdc_set_name>	No	Name of the CDC set to update. <p>Note that this option works only for tables in a model journalized in consistent mode.</p> <p>This parameter cannot be used with -TABLE_NAME. It is mandatory if -TABLE_NAME is not set.</p>
-SUBSCRIBER_NAME=<subscriber_name>	Yes	Name of the subscriber for which the data is retrieved.
-CONTEXT=<context>	No	Context in which the logical schema will be resolved. If no context is specified, the execution context is used.
-MAX_JRN_DATE=<to_date>	No	Date (and time) until which the journalizing events are taken into account.

Examples

Retrieve for the CUSTOMERS table in the SALES_APPLICATION schema the journalizing events for the SALES_SYNC subscriber.

```
OdiRetrieveJournalData -LSHEMA=SALES_APPLICATION  
-TABLE_NAME=CUSTOMERS -SUBSCRIBER_NAME=SALES_SYNC
```

OdiReverseGetMetaData

Use this command to reverse-engineer metadata for the given model in the reverse tables using the JDBC driver capabilities. This command is typically preceded by OdiReverseResetTable and followed by OdiReverseSetMetaData.

Notes:

- This command uses the same technique as the standard reverse-engineering, and depends on the capabilities of the JDBC driver used.
 - The use of this command is restricted to DEVELOPMENT type Repositories because the metadata is not available on EXECUTION type Repositories.
-
-

Usage

```
OdiReverseGetMetaData -MODEL=<model_id>
```

Parameters

Parameters	Mandatory	Description
-MODEL=<model_id>	Yes	Model to reverse-engineer.

Examples

Reverse the RKM's current model.

```
OdiReverseGetMetaData -MODEL=<%=odiRef.getModel("ID")%>
```

OdiReverseManageShortcut

Use this command to define how to handle shortcuts when they are reverse-engineered in a model.

Usage

```
OdiReverseManageShortcut "-MODEL=<model_id>" "-MODE=MATERIALIZING_MODE"
```

Parameters

Parameters	Mandatory	Description
-MODEL=<model_id>	Yes	Global identifier of the model to be reversed.
-MODE=ALWAYS_MATERIALIZE ALWAYS_SKIP PROMPT	Yes	<p>This parameter is supported only when a package or scenario is run in ODI Studio.</p> <p>This parameter accepts the following values:</p> <ul style="list-style-type: none"> ■ ALWAYS_MATERIALIZE: Conflicted shortcuts are always materialized and datastores are reversed (default). ■ ALWAYS_SKIP: Conflicted shortcuts are always skipped and not reversed. ■ PROMPT: The Shortcut Conflict Detected dialog is displayed. You can define how to handle conflicted shortcuts. Select Materialize, to materialize and reverse-engineer the conflicted datastore shortcut. Leave Materialize unselected, to skip the conflicted shortcuts. Unselected datastores are not reversed and the shortcut remains.

Examples

Reverse model 44fa5543-a378-4442-ac64-3dabab65ef98 in ALWAYS_MATERIALIZE mode.

```
OdiReverseManageShortcut -MODEL=44fa5543-a378-4442-ac64-3dabab65ef98 -MODE=ALWAYS_MATERIALIZE
```

OdiReverseResetTable

Use this command to reset the content of reverse tables for a given model. This command is typically used at the beginning of a customized reverse-engineering process.

Usage

```
OdiReverseResetTable -MODEL=<model_id>
```

Parameters

Parameters	Mandatory	Description
-MODEL=<model_id>	Yes	Global identifier of the model to be reversed.

Examples

```
OdiReverseResetTable -MODEL=44fa5543-a378-4442-ac64-3dabab65ef98
```


OdiUnlockOdiRepository

Use this command to unlock the ODI repository.

Note: Please note the following:

- VCS Administrator privileges are required to run this command.
- This tool can be run only from the command line.

Usage

```
OdiUnlockOdiRepository
```

Parameters

None

OdiUpdateAgentSchedule

Use this command to force an agent to recalculate its schedule of tasks.

Usage

```
OdiUpdateAgentSchedule -AGENT_NAME=<physical_agent_name>
```

Parameters

Parameters	Mandatory	Description
<code>-AGENT_NAME=<physical_agent_name></code>	Yes	Name of the physical agent to update.

Examples

Cause the physical agent agt_s1 to update its schedule.

```
OdiUpdateAgentSchedule -AGENT_NAME=agt_s1
```


Examples

Wait and poll every 5 seconds for all child sessions of the current session with a name filter of LOAD% and keywords MANDATORY and CRITICAL to finish.

```
OdiWaitForChildSession -PARENT_SESS_NO=<%=odiRef.getSession("SESS_NO")%>  
-POLL_INT=5 -SESSION_NAME_FILTER=LOAD%  
-SESSION_KEYWORDS=MANDATORY,CRITICAL
```


Parameters	Mandatory	Description
-POLLINT=<pollInt>	No	The period of time in milliseconds to wait between data polls. The default value is 1000.
-UNIT_ROWCOUNT=<unitRowCount>	No	Number of rows expected in a polled table to terminate the command. The default value is 1.
-GLOBAL_ROWCOUNT=<globalRowCount>	No	Total number of rows expected cumulatively, over the set of tables, to terminate the command. If not specified, the default value 1 is used.
-INCREMENT_DETECTION=<no yes>	No	<p>Defines the mode in which the command considers row count: either in absolute terms (with respect to the total number of rows in the table) or in differential terms (the difference between a stored reference value and the current row count value).</p> <ul style="list-style-type: none">■ If set to Yes, the row count is performed in differential mode. The number of additional rows in the table is compared to a stored reference value. The reference value depends on the <code>-INCREMENT_MODE</code> parameter.■ If set to No, the count is performed in absolute row count mode. <p>The default value is No.</p>


```
to_date('#MAX_DE1_DATMAJ_ORACLE_CHAR', 'DD/MM/YYYY HH24:MI:SS')
```

Wait for a maximum of 4 hours for new data to appear in either the CITY_SRC or the CITY_TRG table in the logical schema SQLSRV_SALES.

```
OdiWaitForData -LSHEMA=SQLSRV_SALES -TABLE_NAME=CITY%  
-TIMEOUT=14400000 -INCREMENT_DETECTION=yes
```

OdiWaitForLoadPlans

Use this command to wait for load plan runs to complete.

Usage

```
OdiWaitForLoadPlans [-PARENT_SESS_NO=<parent_sess_guid>]
[-LP_NAME_FILTER=<load_plan_name_filter>] [-LP_KEYWORDS=<load_plan_keywords>]
[-MAX_LP_ERROR=ALL|<number_of_lp_errors>] [-POLLINT=<polling_interval_msec>]
```

Parameters

Parameters	Mandatory	Description
-PARENT_SESS_NO=<parent_sess_guid>	No	Global ID of the parent session that started the load plan. If this parameter is not specified, the global ID of the current session is used.
-LP_NAME_FILTER=<load_plan_name_filter>	No	Only load plan runs whose name matches this filter are tested for completion status. This filter can be a SQL LIKE-formatted pattern.
-LP_KEYWORDS=<load_plan_keywords>	No	Only load plan runs whose keywords contain all entries in this comma-separated list are tested for completion status. Each element in the list can be a SQL LIKE-formatted pattern.
-MAX_LP_ERROR=ALL <number_of_lp_errors>	No	OdiWaitForLoadPlans terminates in error if a number of load plan runs are in Error status: <ul style="list-style-type: none"> ALL: Error if all load plan runs complete in Error status. <number_of_lp_errors>: Error if the number of load plan runs in Error status is at or above this value <number_of_lp_errors> when all load plan runs are complete. If this parameter is not specified or its value is less than 1, OdiWaitForLoadPlans never terminates in error, regardless of the number of load plan runs in Error status.
-POLLINT=<polling_interval_msec>	No	The time in milliseconds to wait between polling load plan runs status for completion state. The default value is 1000 (1 second). The value must be greater than 0.

Examples

Wait and poll every 5 seconds for all load plan runs started by the current session with a name filter of POPULATE% and keywords MANDATORY and CRITICAL to finish in a Done or Error status. If 2 or more load plan runs are in Error status when execution is complete for all selected load plan runs, OdiWaitForLoadPlans ends in error.

```
OdiWaitForLoadPlans -PARENT_SESS_NO=<%=odiRef.getSession("SESS_GUID")%>
-LP_NAME_FILTER=POPULATE% -LP_KEYWORDS=MANDATORY,CRITICAL
-POLLINT=5000 -MAX_LP_ERROR=2
```

OdiWaitForLogData

Use this command to wait for a number of modifications to occur on a journalized table or a list of journalized tables.

The OdiWaitForLogData command determines whether rows have been modified on a table or a group of tables. These changes are detected using the Oracle Data Integrator changed data capture (CDC) in simple mode (using the `-TABLE_NAME` parameter) or in consistent mode (using the `-CDC_SET_NAME` parameter). The test is repeated every `-POLLINT` milliseconds until one of the following conditions is met: the desired number of row modifications for one of the tables has been detected (`-UNIT_ROWCOUNT`), the desired cumulative number of row modifications for all of the tables has been detected (`-GLOBAL_ROWCOUNT`), or a timeout (`-TIMEOUT`) has been reached.

Note: This command takes into account all journalized operations (inserts, updates, and deletes).

The command is suitable for journalized tables only in simple or consistent mode.

Usage

```
OdiWaitForLogData -LSHEMA=<logical_schema> -SUBSCRIBER_NAME=<subscriber_name>
(-TABLE_NAME=<table_name> | -CDC_SET_NAME=<cdcSetName>)
[-CONTEXT=<context>] [-TIMEOUT=<timeout>] [-POLLINT=<pollInt>]
[-GLOBAL_ROWCOUNT=<globalRowCount>]
[-UNIT_ROWCOUNT=<unitRowCount> [-OPTIMIZED_WAIT=<yes|no|AUTO>]
[-TIMEOUT_WITH_ROWS_OK=<yes|no>]
```

Parameters

Parameters	Mandatory	Description
<code>-CONTEXT=<context></code>	No	Context in which the logical schema will be resolved. If no context is specified, the execution context is used.
<code>-GLOBAL_ROWCOUNT=<globalRowCount></code>	No	Total number of changes expected in the tables or the CDC set to end the command. The default value is 1.
<code>-LSHEMA=<logical_schema></code>	Yes	Logical schema containing the journalized tables.

Parameters	Mandatory	Description
-OPTIMIZED_WAIT=<yes no AUTO>	No	<p>Method used to access the journals.</p> <ul style="list-style-type: none"> yes: Optimized method. This method works for later versions of journalizing. It runs faster than the nonoptimized mode. no: Nonoptimized method. A count is performed on the journalizing table. This method is of lower performance but is compatible with earlier versions of the journalizing feature. AUTO: If more than one table is checked, the optimized method is used. Otherwise, the nonoptimized method is used. <p>The default value is AUTO.</p>
-POLLINT=<pollInt>	No	The period of time in milliseconds to wait between polls. The default value is 2000.
-SUBSCRIBER_NAME=<subscriber_name>	Yes	Name of the subscriber used to get the journalizing information.
-TABLE_NAME=<table_name>	Yes	<p>Journalized table name, mask, or list to check. This parameter accepts three formats:</p> <ul style="list-style-type: none"> Table Name Table Name Mask: This mask selects the tables to poll. The mask is specified using the SQL LIKE syntax: the % symbol replaces an unspecified number of characters and the _ symbol acts as a wildcard. Table Names List: List of table names separated by commas. Masks as defined above are not allowed. <p>Note that this option works only for tables in a model journalized in simple mode.</p> <p>This parameter cannot be used with -CDC_SET_NAME. It is mandatory if -CDC_SET_NAME is not set.</p>
-CDC_SET_NAME=<cdcSetName>	Yes	<p>Name of the CDC set to check. This CDC set name is the fully qualified model code, typically PHYSICAL_SCHEMA_NAME.MODEL_CODE.</p> <p>It can be obtained in the current context using a substitution method API call, as shown below:</p> <pre><%=odiRef.getObjectName("L", "model_code", "logical_schema", "D")%></pre> <p>Note that this option works only for tables in a model journalized in consistent mode.</p> <p>This parameter cannot be used with -TABLE_NAME. It is mandatory if -TABLE_NAME is not set.</p>
-TIMEOUT=<timeout>	No	Maximum period of time in milliseconds over which changes are polled. If this value is equal to 0, the timeout is infinite. The default value is 0.

Parameters	Mandatory	Description
-TIMEOUT_WITH_ROWS_OK=<yes no>	No	If this parameter is set to Yes, at least one row was detected, and the timeout occurs before the predefined number of rows has been polled, the API exits with a return code of 0. Otherwise, it signals an error. The default value is Yes.
-UNIT_ROWCOUNT=<unitRowCount>	No	Number of changes expected in one of the polled tables to end the command. The default value is 1. Note that -UNIT_ROWCOUNT is not taken into account with -CDC_SET_NAME.

Examples

Wait for the CUSTOMERS table in the SALES_APPLICATION schema to have 200 row modifications recorded for the SALES_SYNC subscriber.

```
OdiWaitForLogData -LSHEMA=SALES_APPLICATION  
-TABLE_NAME=CUSTOMERS -GLOBAL_ROWCOUNT=200  
-SUBSCRIBER_NAME=SALES_SYNC
```

OdiWaitForTable

Use this command to wait for a table to be created and populated with a predefined number of rows.

The OdiWaitForTable command regularly tests whether the specified table has been created and has been populated with a number of records. The test is repeated every `-POLLINT` milliseconds until the table exists and contains the desired number of rows (`-GLOBAL_ROWCOUNT`), or until a timeout (`-TIMEOUT`) is reached.

Usage

```
OdiWaitForTable -CONTEXT=<context> -LSHEMA=<logical_schema>
-TABLE_NAME=<table_name> [-TIMEOUT=<timeout>] [-POLLINT=<pollInt>]
[-GLOBAL_ROWCOUNT=<globalRowCount>] [-TIMEOUT_WITH_ROWS_OK=<yes|no>]
```

Parameters

Parameters	Mandatory	Description
<code>-CONTEXT=<context></code>	No	Context in which the logical schema will be resolved. If no context is specified, the execution context is used.
<code>-GLOBAL_ROWCOUNT=<globalRowCount></code>	No	Total number of rows expected in the table to terminate the command. The default value is 1. If not specified, the command finishes when a new row is inserted into the table.
<code>-LSHEMA=<logical_schema></code>	Yes	Logical schema in which the table is searched for.
<code>-POLLINT=<pollInt></code>	No	Period of time in milliseconds to wait between each test. The default value is 1000.
<code>-TABLE_NAME=<table_name></code>	Yes	Name of table to search for.
<code>-TIMEOUT=<timeout></code>	No	Maximum time in milliseconds the table is searched for. If this value is equal to 0, the timeout is infinite. The default value is 0.
<code>-TIMEOUT_WITH_ROWS_OK=<yes no></code>	No	If this parameter is set to Yes, at least one row is detected, and the timeout occurs before the expected number of records is detected, the API exits with a return code of 0. Otherwise, it signals an error. The default value is Yes.

Examples

Wait for the DE1P1 table in the ORA_WAITFORDATA schema to exist, and to contain at least 1 record.

```
OdiWaitForTable -LSHEMA=ORA_WAITFORDATA -TABLE_NAME=DE1P1
-GLOBAL_ROWCOUNT=1
```

OdiXMLConcat

Use this command to concatenate elements from multiple XML files into a single file.

This tool extracts all instances of a given element from a set of source XML files and concatenates them into one target XML file. The tool parses and generates well formed XML. It does not modify or generate a DTD for the generated files. A reference to an existing DTD can be specified in the `-HEADER` parameter or preserved from the original files using `-KEEP_XML_PROLOGUE`.

Note: XML namespaces are not supported by this tool. Provide the local part of the element name (without the namespace or prefix value) in the `-ELEMENT_NAME` parameter.

Usage

```
OdiXMLConcat -FILE=<file_filter> -TOFILE=<target_file>
-XML_ELEMENT=<element_name> [-CHARSET_ENCODING=<encoding>]
[-IF_FILE_EXISTS=<overwrite|skip|error>]
[-KEEP_XML_PROLOGUE=<all|xml|doctype|none>] [-HEADER=<header>]
[-FOOTER=<footer>]
```

Parameters

Parameters	Mandatory	Description
<code>-FILE=<file_filter></code>	Yes	<p>Filter for the source XML files. This filter uses standard file wildcards (?,*). It includes both file names and directory names. Source files can be taken from the same folder or from different folders.</p> <p>The following file filters are valid:</p> <ul style="list-style-type: none"> ■ /tmp/files_*/customer.xml ■ /tmp/files_*/*.* ■ /tmp/files_??/customer.xml ■ /tmp/files/customer_*.xml ■ /tmp/files/customer_?.xml
<code>-TOFILE=<target_file></code>	Yes	Target file into which the elements are concatenated.
<code>-XML_ELEMENT=<element_name></code>	Yes	<p>Local name of the XML element (without enclosing <> characters, prefix, or namespace information) to be extracted with its content and child elements from the source files.</p> <p>Note that this element detection is not recursive. If a given instance of <element_name> contains other instances of <element_name>, only the element of higher level is taken into account and child elements are only extracted as a part of the top element's content.</p>

Parameters	Mandatory	Description
<code>-CHARSET_ENCODING=<encoding></code>	No	Target files encoding. The default value is ISO-8859-1. For the list of supported encodings, see: http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html
<code>-IF_FILE_EXISTS=<overwrite skip error></code>	No	Define behavior when the target file exists. <ul style="list-style-type: none"> ■ <code>overwrite</code>: Overwrite the target file if it exists. ■ <code>skip</code>: Do nothing for this file. ■ <code>error</code>: Raise an error.
<code>-KEEP_XML_PROLOGUE=<all xml doctype none></code>	No	Copies the source file XML prologue in the target file. Depending on this parameter's value, the following parts of the XML prologue are preserved: <ul style="list-style-type: none"> ■ <code>all</code>: Copies all of the prologue (XML and document type declaration). ■ <code>xml</code>: Copies only the XML declaration <code><?xml...?></code> and not the document type declaration. ■ <code>doctype</code>: Copies only the document type declaration and not the XML declaration. ■ <code>none</code>: Does not copy the prologue from the source file. <p>Note: If all or part of the prologue is not preserved, it should be specified in the <code>-HEADER</code> parameter.</p>
<code>-HEADER=<header></code>	No	String that is appended after the prologue (if any) in each target file. You can use this parameter to create a customized XML prologue or root element.
<code>-FOOTER=<footer></code>	No	String that is appended at the end of each target file. You can use this parameter to close a root element added in the header.

Examples

Concatenate the content of the IDOC elements in the files `ord1.xml`, `ord2.xml`, and so on in the `ord_i` subfolder into the file `MDSLS.TXT.XML`, with the root element `<WMMBID02>` added to the target.

```
OdiXMLConcat "-FILE=./ord_i/ord*.xml" "-TOFILE=./MDSLS.TXT.XML" -XML_ELEMENT=IDOC
"-CHARSET_ENCODING=UTF-8" -IF_FILE_EXISTS=overwrite -KEEP_XML_PROLOGUE=xml
"-HEADER=<WMMBID02>" "-FOOTER=</WMMBID02>"
```

```
OdiXMLConcat "-FILE=./o?d_*/ord*.xml" "-TOFILE=./MDSLS.TXT.XML" -XML_ELEMENT=IDOC
"-CHARSET_ENCODING=UTF-8" -IF_FILE_EXISTS=overwrite -KEEP_XML_PROLOGUE=none
"-HEADER=<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<WMMBID02>"
"-FOOTER=</WMMBID02>"
```

Concatenate the EDI elements of the files `ord1.xml`, `ord2.xml`, and so on in the `ord_i` subfolder into the file `MDSLS2.XML`. This file will have the new root element `EDI_BATCH` above all `<EDI>` elements.

```
OdiXMLConcat "-FILE=./o?d_*/ord*.xml" "-TOFILE=./MDSLS2.XML" -XML_ELEMENT=EDI
"-CHARSET_ENCODING=UTF-8" -IF_FILE_EXISTS=overwrite -KEEP_XML_PROLOGUE=xml
"-HEADER= <EDI_BATCH>" "-FOOTER=</EDI_BATCH>"
```

OdiXMLSplit

This tool extracts all instances of a given element stored in a source XML file and splits it over several target XML files. This tool parses and generates well formed XML. It does not modify or generate a DTD for the generated files. A reference to an existing DTD can be specified in the `-HEADER` parameter or preserved from the original files using `-KEEP_XML_PROLOGUE`.

Note: XML namespaces are not supported by this tool. Provide the local part of the element name (without the namespace or prefix value) in the `-ELEMENT_NAME` parameter.

Usage

```
OdiXMLSplit -FILE=<file> -TOFILE=<file_pattern> -XML_ELEMENT=<element_name>
[-CHARSET_ENCODING=<encoding>] [-IF_FILE_EXISTS=<overwrite|skip|error>]
[-KEEP_XML_PROLOGUE=<all|xml|doctype|none>] [-HEADER=<header>]
[-FOOTER=<footer>]
```

Parameters

Parameters	Mandatory	Description
<code>-FILE=<file></code>	Yes	Source XML file to split.

Parameters	Mandatory	Description
-TOFILE=<file_pattern>	Yes	<p>File pattern for the target files. Each file is named after a pattern containing a mask representing a generated number sequence or the value of an attribute of the XML element used to perform the split:</p> <ul style="list-style-type: none"> ■ Number Sequence Mask: Use the * (star) value to indicate the place of the file number value. For example, if the <file_pattern> is equal to target_*.xml, the files created are named target_1.xml, target_2.xml, and so on. ■ Attribute Value Mask: Between square brackets, specify the name of the attribute of <element_name> whose value should be pushed to create the file name. For example, customer_[CUSTID].xml creates files named customer_041.xml, customer_123.xml, and so on, depending on the value of the attribute CUSTID of the element used to split. Note that if a value repeats over several successive elements, target files may be overwritten according to the value of the -OVERWRITE parameter. <p>Note that the pattern can be used for creating different files within a directory or files in different directories. The following patterns are valid:</p> <ul style="list-style-type: none"> ■ /tmp/files_*/customer.xml ■ /tmp/files_[CUSTID]/customer.xml ■ /tmp/files/customer_*.xml ■ /tmp/files/customer_[CUSTID].xml
-XML_ELEMENT=<element_name>	Yes	<p>Local name of the XML element (without enclosing <> characters, prefix, or namespace information) to be extracted with its content and child elements from the source files.</p> <p>Note that this element detection is not recursive. If a given instance of <element_name> contains other instances of <element_name>, only the element of higher level is taken into account and child elements are only extracted as a part of the top element's content.</p>
-CHARSET_ENCODING=<encoding>	No	<p>Target files encoding. The default value is ISO-8859-1. For the list of supported encodings, see: http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html</p>
-IF_FILE_EXISTS=<overwrite skip error>	No	<p>Define behavior when the target file exists.</p> <ul style="list-style-type: none"> ■ overwrite: Overwrite the target file if it exists. ■ skip: Do nothing for this file. ■ error: Raise an error.

Parameters	Mandatory	Description
-KEEP_XML_PROLOGUE=<all xml doctype none>	No	<p>Copies the source file XML prologue in the target file. Depending on this parameter's value, the following parts of the XML prologue are preserved:</p> <ul style="list-style-type: none"> ▪ all: Copies all of the prologue (XML and document type declaration). ▪ xml: Copies only the XML declaration <?xml . . . ?> and not the document type declaration. ▪ doctype: Copies only the document type declaration and not the XML declaration. ▪ none: Does not copy the prologue from the source file. <p>Note: If all or part of the prologue is not preserved, it should be specified in the -HEADER parameter.</p>
-HEADER=<header>	No	String that is appended after the prologue (if any) in each target file. You can use this parameter to create a customized XML prologue or root element.
-FOOTER=<footer>	No	String that is appended at the end of each target file. You can use this parameter to close a root element added in the header.

Examples

Split the file MDSLS.TXT.XML into several files. The files ord1.xml, ord2.xml, and so on are created and contain each instance of the IDOC element contained in the source file.

```
OdiXMLSplit "-FILE= ./MDSLS.TXT.XML" "-TOFILE= ./ord_i/ord*.xml" -XML_ELEMENT=IDOC
"-CHARSET_ENCODING=UTF-8" -IF_FILE_EXISTS=overwrite -KEEP_XML_PROLOGUE=xml
"-HEADER= <WMMBID02>" "-FOOTER= </WMMBID02>"
```

Split the file MDSLS.TXT.XML the same way as in the previous example except name the files using the value of the BEGIN attribute of the IDOC element that is being split. The XML prologue is not preserved in this example but entirely generated in the header.

```
OdiXMLSplit "-FILE= ./MDSLS.TXT.XML" "-TOFILE= ./ord_i/ord[BEGIN].xml"
-XML_ELEMENT=IDOC "-CHARSET_ENCODING=UTF-8" -IF_FILE_EXISTS=overwrite -KEEP_XML
PROLOGUE=none "-HEADER= <?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<WMMBID02>"
"-FOOTER=</WMMBID02>"
```

OdiZip

Use this command to create a ZIP file from a directory or several files.

Usage

```
OdiZip -DIR=<directory> -FILE=<file> -TOFILE=<target_file> [-OVERWRITE=<yes|no>]
[-RECURSE=<yes|no>] [-CASESENS=<yes|no>]
[-ENCODING=<file_name_encoding>]
```

Parameters

Parameters	Mandatory	Description
-DIR=<directory>	Yes if -FILE is omitted	Base directory (or folder) that will be the future root in the ZIP file to generate. If only -DIR and not -FILE is specified, all files under this directory are archived.
-FILE=<file>	Yes if -DIR is omitted	Path from the base directory of the file(s) to archive. If only -FILE and not -DIR is specified, the default directory is the current work directory if the -FILE path is relative. Use * to specify the generic characters. Examples: /var/tmp/*.log (all files with the log extension of the directory /var/tmp) arch_*.lst (all files starting with arch_ and with the extension lst)
-TOFILE=<target_file>	Yes	Target ZIP file.
-OVERWRITE=<yes no>	No	Indicates whether the target ZIP file must be overwritten (Yes) or simply updated if it already exists (No). By default, the ZIP file is updated if it already exists.
-RECURSE=<yes no>	No	Indicates if the archiving is recursive in the case of a directory that contains other directories. The value No indicates that only the files contained in the directory to copy (without the subfolders) are archived.
-CASESENS=<yes no>	No	Indicates if file search is case-sensitive. By default, Oracle Data Integrator searches files in uppercase (set to No).
-ENCODING=<file_name_encoding>	No	Character encoding to use for file names inside the archive file. For the list of supported encodings, see: http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html This defaults to the platform's default character encoding.

Examples

Create an archive of the directory C:\Program files\odi.

```
OdiZip "-DIR=C:\Program Files\odi" -FILE=*. * -TOFILE=C:\TEMP\odi_archive.zip
```

Create an archive of the directory C:\Program files\odi while preserving the odi directory in the archive.

```
OdiZip "-DIR=C:\Program Files" -FILE=odi\*. * -TOFILE=C:\TEMP\odi_archive.zip
```