

**Oracle® Warehouse Builder**

Scripting Reference

10g Release 1 (10.1)

**Part No. B12152-01**

November 2003

Oracle Warehouse Builder Scripting Reference, 10g Release 1 (10.1)

Part No. B12152-01

Copyright © 2001, 2003, Oracle. All rights reserved.

Contributing Author: Michelle Bird, Christine Cheng, Justin Ho, Padmaja Potineni, John Potter, Robert Velisar

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

---

---

# Contents

<b>Send Us Your Comments</b> .....	ix
<b>Preface</b> .....	xi
Purpose .....	xi
Audience .....	xii
How This Guide Is Organized.....	xii
New in 10g Release 1 (10.1).....	xiii
Added in Release 2 (9.0.4) .....	xv
Conventions .....	xix
Related Publications.....	xix
Documentation Accessibility .....	xx
Contacting Oracle.....	xx
<b>1 Introduction</b>	
<b>About the OMB Scripting Language</b> .....	1-1
Using OMB Plus .....	1-1
<b>OMB Plus Commands</b> .....	1-4
Metadata Manipulation Language (MML) Commands.....	1-4
Examples .....	1-5
Shell Commands .....	1-6
Administrative Commands .....	1-7
Navigation Commands .....	1-7
Service Commands .....	1-8
<b>How to Read Syntax Diagrams</b> .....	1-9
Required Keywords and Parameters .....	1-10
Syntax Loops.....	1-10
<b>Sample OMB Plus Scripts</b> .....	1-10
Updating a Design Repository .....	1-11
Reporting on Repository Objects .....	1-11
Finding Invalid Objects .....	1-12
Using OMB Plus to Navigate Repositories.....	1-12
<b>2 OMB Plus Commands</b>	
<b>OMBCC</b> .....	2-2
<b>OMBCOMMIT</b> .....	2-3

OMBCOMPARE_SNAPSHOT .....	2-4
OMBCOMPILE.....	2-7
OMBCONN.....	2-9
OMBCONNECT_RUNTIME.....	2-11
OMBCONNECT.....	2-13
OMBCOPY.....	2-15
OMBDCC.....	2-17
OMBDEPLOY .....	2-18
OMBDESCRIBE_CLASS_DEFINITION.....	2-20
OMBDISC.....	2-24
OMBDISCONNECT.....	2-25
OMBENV .....	2-26
OMBEXPORT.....	2-27
OMBEXPORT_MDL_FILE .....	2-31
OMBHELP .....	2-35
OMBIMPORT .....	2-36
OMBIMPORT_MDL_FILE.....	2-39
OMBLIST.....	2-42
OMBLIST_SNAPSHOT.....	2-44
OMBLOCK .....	2-46
OMBMOVE.....	2-48
OMBRECONCILE.....	2-50
OMBREDEFINE_CLASS_DEFINITION.....	2-53
OMBREGISTER_LOCATION.....	2-57
OMBRESTORE_SNAPSHOT.....	2-64
OMBROLLBACK.....	2-67
OMBUNLOCK.....	2-68
OMBVALIDATE.....	2-70

### 3 OMBALTER

OMBALTER.....	3-2
OMBALTER_ADVANCED_QUEUE.....	3-7
OMBALTER_COLLECTION .....	3-11
OMBALTER_CONNECTOR.....	3-15
OMBALTER_CUBE_TABLE.....	3-18
OMBALTER_DEPLOYMENT_ACTION_PLAN.....	3-30
OMBALTER_EXTERNAL_TABLE.....	3-35
OMBALTER_FLAT_FILE.....	3-44
OMBALTER_FLAT_FILE_MODULE .....	3-54
OMBALTER_FUNCTION .....	3-58
OMBALTER_LOCATION .....	3-64
OMBALTER_MAPPING .....	3-67
OMBALTER_MATERIALIZED_VIEW.....	3-146
OMBALTER_OBJECT_TYPE.....	3-165
OMBALTER_ORACLE_MODULE.....	3-169
OMBALTER_PACKAGE .....	3-179
OMBALTER_PROCEDURE.....	3-182

OMBALTER PROCESS_FLOW.....	3-188
OMBALTER PROCESS_FLOW_MODULE.....	3-200
OMBALTER PROCESS_FLOW_PACKAGE.....	3-203
OMBALTER PROJECT.....	3-206
OMBALTER RUNTIME_REPOSITORY_CONNECTIONS.....	3-209
OMBALTER SEQUENCE.....	3-213
OMBALTER SNAPSHOT.....	3-217
OMBALTER TABLE.....	3-221
OMBALTER VIEW.....	3-240

#### 4 OMBCREATE

OMBCREATE.....	4-2
OMBCREATE ADVANCED_QUEUE.....	4-6
OMBCREATE COLLECTION.....	4-10
OMBCREATE CONNECTOR.....	4-13
OMBCREATE CUBE_TABLE.....	4-16
OMBCREATE DEPLOYMENT_ACTION_PLAN.....	4-24
OMBCREATE EXTERNAL_TABLE.....	4-27
OMBCREATE FLAT_FILE.....	4-35
OMBCREATE FLAT_FILE_MODULE.....	4-43
OMBCREATE FUNCTION.....	4-46
OMBCREATE LOCATION.....	4-51
OMBCREATE MAPPING.....	4-54
OMBCREATE MATERIALIZED_VIEW.....	4-131
OMBCREATE OBJECT_TYPE.....	4-145
OMBCREATE ORACLE_MODULE.....	4-148
OMBCREATE PACKAGE.....	4-157
OMBCREATE PROCEDURE.....	4-160
OMBCREATE PROCESS_FLOW.....	4-164
OMBCREATE PROCESS_FLOW_MODULE.....	4-173
OMBCREATE PROCESS_FLOW_PACKAGE.....	4-176
OMBCREATE PROJECT.....	4-179
OMBCREATE RUNTIME_REPOSITORY_CONNECTION.....	4-182
OMBCREATE SEQUENCE.....	4-185
OMBCREATE SNAPSHOT.....	4-188
OMBCREATE TABLE.....	4-192
OMBCREATE VIEW.....	4-207

#### 5 OMBDROP

OMBDROP.....	5-2
OMBDROP ADVANCED_QUEUE.....	5-4
OMBDROP COLLECTION.....	5-5
OMBDROP CONNECTOR.....	5-6
OMBDROP CUBE_TABLE.....	5-7
OMBDROP DEPLOYMENT_ACTION_PLAN.....	5-8
OMBDROP EXTERNAL_TABLE.....	5-9

OMBDROP FLAT_FILE .....	5-10
OMBDROP FLAT_FILE_MODULE .....	5-11
OMBDROP FUNCTION .....	5-12
OMBDROP GATEWAY_MODULE .....	5-13
OMBDROP LOCATION .....	5-14
OMBDROP MAPPING .....	5-15
OMBDROP MATERIALIZED_VIEW .....	5-16
OMBDROP OBJECT_TYPE .....	5-17
OMBDROP ORACLE_MODULE .....	5-18
OMBDROP PACKAGE .....	5-19
OMBDROP PROCEDURE .....	5-20
OMBDROP PROCESS_FLOW .....	5-21
OMBDROP PROCESS_FLOW_MODULE .....	5-22
OMBDROP PROCESS_FLOW_PACKAGE .....	5-23
OMBDROP PROJECT .....	5-24
OMBDROP RUNTIME_REPOSITORY_CONNECTION .....	5-25
OMBDROP SAP_MODULE .....	5-26
OMBDROP SEQUENCE .....	5-27
OMBDROP SNAPSHOT .....	5-28
OMBDROP TABLE .....	5-29
OMBDROP VIEW .....	5-30

## 6 OMBRETRIEVE

OMBRETRIEVE .....	6-2
OMBRETRIEVE ADVANCED_QUEUE .....	6-4
OMBRETRIEVE COLLECTION .....	6-8
OMBRETRIEVE CONNECTOR .....	6-12
OMBRETRIEVE CUBE_TABLE .....	6-14
OMBRETRIEVE DEPLOYMENT_ACTION_PLAN .....	6-23
OMBRETRIEVE EXTERNAL_TABLE .....	6-25
OMBRETRIEVE FLAT_FILE .....	6-30
OMBRETRIEVE FLAT_FILE_MODULE .....	6-38
OMBRETRIEVE FUNCTION .....	6-40
OMBRETRIEVE GATEWAY_MODULE .....	6-45
OMBRETRIEVE LOCATION .....	6-48
OMBRETRIEVE MAPPING .....	6-51
OMBRETRIEVE MATERIALIZED_VIEW .....	6-126
OMBRETRIEVE OBJECT_TYPE .....	6-138
OMBRETRIEVE ORACLE_MODULE .....	6-141
OMBRETRIEVE PACKAGE .....	6-149
OMBRETRIEVE PROCEDURE .....	6-152
OMBRETRIEVE PROCESS_FLOW .....	6-157
OMBRETRIEVE PROCESS_FLOW_MODULE .....	6-162
OMBRETRIEVE PROCESS_FLOW_PACKAGE .....	6-164
OMBRETRIEVE PROJECT .....	6-166
OMBRETRIEVE RUNTIME_REPOSITORY_CONNECTION .....	6-168
OMBRETRIVE SAP_MODULE .....	6-171

OMBRETRIEVE SEQUENCE.....	6-173
OMBRETRIEVE SNAPSHOT .....	6-176
OMBRETRIEVE TABLE .....	6-178
OMBRETRIEVE VIEW .....	6-190

**A Running OMB Plus in Oracle JDeveloper**

Installing OMB Plus in Oracle JDeveloper.....	A-1
Invoking Keyword Auto Completion .....	A-2
Invoking the OMBPlus Interpreter .....	A-2
Viewing the OMBPlus Console.....	A-2
Viewing Help Documentation.....	A-2

**B Default Groups and Parameters**

Default Group Names and Attribute Names.....	B-1
--	-----





---

---

# Send Us Your Comments

## **Oracle Warehouse Builder Scripting Reference, 10g Release 1 (10.1)**

**Part No. B12152-01**

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, please indicate the title and part number of the documentation and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: [dwhdoc\\_us@oracle.com](mailto:dwhdoc_us@oracle.com)
- Postal service:

Oracle Corporation  
Oracle Warehouse Builder Documentation  
500 Oracle Parkway, Mailstop 2op10  
Redwood Shores, CA 94065  
U.S. A.

If you would like a reply, please give your name, address, telephone number, and electronic mail address.

If you have problems with the software, please contact your local Oracle Support Services.



---

---

# Preface

This preface includes the following topics:

- [Purpose](#) on page xi
- [Audience](#) on page xii
- [How This Guide Is Organized](#) on page xii
- [New in 10g Release 1 \(10.1\)](#) on page xiii
- [Added in Release 2 \(9.0.4\)](#) on page xv
- [Conventions](#) on page xix
- [Related Publications](#) on page xix
- [Documentation Accessibility](#) on page xx
- [Contacting Oracle](#) on page xx

## Purpose

Oracle Warehouse Builder is a comprehensive toolset for practitioners who move and transform data, develop and implement business intelligence systems, perform metadata management, or create and manage Oracle databases and metadata. This guide describes how to use Warehouse Builder Oracle MetaBase (OMB) Scripting Language to:

- Create a definition of a data warehouse.
- Configure the definitions for a physical instance of the data warehouse.
- Validate the set of definitions and their configurations.
- Generate a set of scripts to create and populate the data warehouse instance.
- Generate data transformation scripts.
- Deploy and initially load the data warehouse instance.
- Maintain the physical instance by conditionally refreshing it with generated scripts.
- Integrate Warehouse Builder metadata with other Business Intelligence products.
- Populate Oracle Discoverer EULs and OLAP catalogs for analyzing the data warehouse.

## Audience

This guide is intended for data warehouse practitioners who want to access Warehouse Builder functionality programmatically:

- Business Intelligence application developers
- Warehouse architects, designers, and developers—especially SQL and PL/SQL developers
- Developers of large-scale products based on data warehouses
- Warehouse administrators
- System administrators
- Other MIS professionals

In order to use the information in this guide, you need to be comfortable with the concepts of Relational Database Management Systems and Data Warehouse design. For information on data warehousing, refer to the Oracle Database *Data Warehousing Guide*. Also, you need to be familiar with Oracle's relational database software products such as Oracle Database, SQL\*Plus, SQL\*Loader, Oracle Enterprise Manager, and Oracle Workflow.

## How This Guide Is Organized

The Oracle Warehouse Builder Scripting Reference contains the following chapters and appendixes.

- [Chapter 1, "Introduction"](#), provides introductory information about the Warehouse Builder scripting language, OMB Plus. It describes the types of commands and how to use them. It also explains how to read the syntax diagrams included in this reference.
- [Chapter 2, "OMB Plus Commands"](#), includes an alphabetical listing of the navigation, service, administrative, and shell commands you can use in OMB Plus. Commands for metadata manipulation are contained in separate chapters.
- [Chapter 3, "OMBALTER"](#), provides a reference for using the metadata manipulation command OMBALTER with Warehouse Builder objects.
- [Chapter 4, "OMBCREATE"](#), provides a reference for using the metadata manipulation command OMBCREATE with Warehouse Builder objects.
- [Chapter 5, "OMBDROP"](#), provides a reference for using the metadata manipulation command OMBDROP with Warehouse Builder objects.
- [Chapter 6, "OMBRETRIEVE"](#), provides a reference for using the metadata manipulation command OMBRETRIEVE with Warehouse Builder objects.
- [Appendix A, "Running OMB Plus in Oracle JDeveloper"](#), gives instructions on how to use OMB Plus in JDeveloper.
- [Appendix B, "Default Groups and Parameters"](#), lists the default groups and parameter for mappings.

## New in 10g Release 1 (10.1)

### **Enhancements to the Mapping Editor: Mapping Debugger**

Warehouse Builder now provides you with extensive debugging capabilities for your mappings from within the Mapping Editor. Use the Mapping Debugger to locate logical design errors in your mappings. The new features allow you to step through the data flow of a mapping using comprehensive debugging functions such as setting breakpoints and watches and interactively changing test data.

### **Enhanced Support of Multiple Targets: Correlated Commit**

This release introduces a new commit strategy for mappings with multiple targets. In previous releases, Warehouse Builder performed independent commits. That is, Warehouse Builder committed and rolled back each target separately and independently of other targets. In addition to this option, Warehouse Builder now also performs correlated commits. Warehouse Builder considers all targets collectively and commits or rolls back data uniformly across all targets. Use the correlated commit when it is important to ensure that every row in the source impacts all affected targets uniformly.

### **Direct Partition Exchange Loading**

In previous releases, Warehouse Builder by default created a temporary table for mappings that required additional processing of source data before exchanging partitions. This occurred when the mapping contained remote sources or multiple sources joined together. Beginning in this release, you can now by-pass the creation of a temporary table and directly swap a source into a target. Use Direct PEL in a mapping to instantaneously publish fact tables that you loaded in a previously executed mapping.

### **Data Quality Features**

- **Multiple Name and Address Software Providers:** Beginning in this release, Warehouse Builder is compatible with multiple certified Name and Address software providers. Third-party vendors can license Name and Address software directly to you for use with Warehouse Builder. This allows you to choose a name and address provider whose offering is the most appropriate for your project.
- **Name-Address Operator Wizard:** In previous releases, you defined the Name-Address operator using the mapping canvas and the operator Configuration Properties sheet. For improved usability, Warehouse Builder now enables you to use a wizard and Operator editor to create and edit the Name-Address operator.
- **Match-Merge Operator:** Warehouse Builder incorporates the data quality functionality formerly available in Oracle Pure Integrate. You can use the Match-Merge operator available in the Mapping Editor to define business rules for matching and merging records. The Match-Merge operator together with the Name-Address operator support householding, the process of identifying unique households in name and address data.

### **Metadata Change Management**

In a previous release, you could perform metadata change management using the OMB Plus scripting utility. Beginning in this release, you can also access these functions from the Warehouse Builder client user interface. Metadata change management enables you to take snapshots of metadata objects and use them for backup and history management. Snapshots are supported for any object on the

navigation tree and can store information about an object alone (such as a table or module), or the objects within it as well (such as the tables within a module).

### Extending Oracle Warehouse Builder Functionality

- **Security:** Warehouse Builder now provides advanced repository security and auditing options that you can implement according to your security requirements. The advanced security options include the following:

**Proactive Security:** Warehouse Builder enables you to plug in a customized security PL/SQL implementation package in the Warehouse Builder repository to provide tailored access control to users according to the security rules defined by your organization.

**Reactive Security:** Warehouse Builder enables you to track audit information based on the metadata history and to determine security policies from such audit information.

**Data Stewardship:** Warehouse Builder enables an individual or a group of individuals to "own" portions of the metadata rather than the technical administrators. Metadata ownership thus becomes an important component of metadata security management.

- **RAC Support:** With the 9.2 release, Warehouse Builder provides increased support for RAC features. Warehouse Builder now supports the use of net service names in the runtime. This enables you to plan maintenance of nodes in a cluster without having to reconfigure the runtime environment. Warehouse Builder also provides an increased availability in the runtime service. For example, if either the service instance or its associated node fails or is taken out of service, then the runtime service instance on a different node can take over. While the Warehouse Builder design repository can also be used in a RAC cluster, it will not take advantage of any failover features of RAC for this release.

### Enhancements to Flat File Support

- **ZONED Data Type Support:** Warehouse Builder now allows you to load fixed format data files containing ZONED decimal data. In the Flat File Sample Wizard, specify the ZONED data type for a flat file you import. The format for ZONED data is a string of decimal digits, one per byte, with the sign included in the last byte. (In COBOL, this is a SIGN TRAILING field.) The length of this field is equal to the precision (number of digits) that you specify. You may also specify a scale, which is the number of digits to the right of the decimal point.
- **DECIMAL Data Type Support:** DECIMAL data is in packed decimal format; two digits per byte, except for the last byte, which contains a digit and sign. The DECIMAL data type includes precision and scale and therefore can represent fractional values.

### Enhancements in Database Connectivity

Warehouse Builder now enables you to create public database links that can be shared across a database. Public database links can be created by repository owners, as well as any user with the `CREATE PUBLIC DATABASE LINK` privilege.

### Warehouse Builder Available on HP-UX and AIX

Starting with this release, Warehouse Builder is available on HP-UX and AIX platforms. This new availability is an addition to the UNIX (Solaris and Linux), and Windows (NT, 2000, and XP) platforms, which have been available from previous releases. (Note that the MITI Bridges feature is only available on Windows platforms

and the Name and Address Server is only available on Windows and Solaris platforms.)

### Public Application Programming Interface

Starting in this release, Warehouse Builder now includes a Public Application Programming Interface (API). To access the API, unzip and extract the following file to a folder on your local machine:

```
<owb home directory>\owb\lib\int\pubapi_javadoc.jar
```

Double click on the file `index.html`. Select the Help link for information on how to use the API.

## Added in Release 2 (9.0.4)

The following new features were introduced in Oracle Warehouse Builder :

### Changes in the Warehouse Builder Console

- **Enhanced Navigation Tree:** The navigation tree that displays in the Warehouse Builder console has been enhanced to improve navigation between projects and facilitate direct access to metadata repository objects. All projects are now visible from the tree, whereas previously you could only see one project at a time. Now you can expand a project node to display the contents of the active project. The module tree no longer appears in a separate window.
- **Wizards, Editors, and Properties Sheets:** All Warehouse Builder wizards, editors, and properties sheets are now launched from the navigation tree.
- **Business Areas Renamed to Collections:** In previous releases, you could create business areas in warehouse modules to organize objects in Warehouse Builder and to export metadata to tools such as Oracle Discoverer. Starting in this release, *collections* replace *business areas* in all functions and introduce enhancements, such as the ability to import metadata into and export metadata from a collection.
- **Fact Tables Renamed to Cubes:** The terms *fact* and *fact table* have been replaced with *cube* in this release to be in line with OLAP industry standards.
- **Logical Names Renamed to Business Names:** All references to *logical names* of objects have changed to *business names* in this release.
- **Toolbars in the Warehouse Builder Console:** The utility drawer has been removed and the side and top toolbars in the Warehouse Builder console have been merged at the top to consolidate the most important functionality in one place.

### Enhancements to Deployment

- **Addition of Deployment Management Objects:** This release introduces three object types to assist in managing connections to deployment sources and targets: Locations, Connectors, and Runtime Repository Connections. Locations define the physical location of the deployment. Connectors define relationships between locations. Runtime Repository Connections provide information about Runtime Repositories. Using these objects, you can create multiple deployment targets for the same target design.
- **Single Deployment Management Interface:** The Deployment Manager provides a single interface for managing deployments of all objects, and executions of deployed mappings, transformations, and process flows. It also provides

immediate access to the history of previously deployed objects. Not only does the Deployment Manager enable you to perform all these tasks from one interface, but Warehouse Builder now keeps track of runtime metadata, providing you with the history of what has previously been deployed.

### Enhancements to Warehouse Builder Metadata Browser

- **Design Metadata Browsing:** The Warehouse Builder Design Browser has been enhanced to include all new exposed objects, such as external tables, locations and connectors. In addition, you can now launch the Design Browser as a standalone executable; it no longer requires Oracle DatabaseAS to be installed for a single-user usage.
- **Runtime Metadata Browsing:** The Warehouse Builder Runtime Audit Viewer has been replaced by the Runtime Audit Browser, which provides web-based reporting. The Runtime Audit Browser provides a more extensive set of deployment and execution audit reports than was available in previous releases. This audit data comes from information stored in the Runtime Repository and includes both deployment and execution data.

### Enhancements to Warehouse Builder Programmatic Access

- **Warehouse Builder Public APIs:** Starting with this release, Warehouse Builder offers this alternative for programmatic access to Oracle Warehouse Builder features: a full set of Java public APIs for application programmers who want to embed Warehouse Builder features and services in their own applications.
- **Warehouse Builder Scripting Language:** Oracle MetaBase (OMB) Scripting Language provides access to all Warehouse Builder functions without accessing the Warehouse Builder graphical user interface. Users can access Warehouse Builder metadata and functionality by using OMB Plus, Warehouse Builder's scripting utility. This gives developers the power of using Warehouse Builder programmatically and extending its functionality where required. For more information on OMB Scripting Language, refer to the Oracle Warehouse Builder Scripting Reference .

### Enhancements to Metadata Management

- **Security:** Warehouse Builder now provides an optional repository security and auditing system that you can implement according to your security requirements. You can create a multiple user account system where multiple identifiable users can access the same Warehouse Builder repository. Warehouse Builder also enables you to plug in a customized security PL/SQL implementation package in the Warehouse Builder repository to provide tailored access control to users according to the security rules defined by your organization.
- **Metadata Change Management (Metadata Snapshots):** Starting in this release, you can take snapshots of metadata objects and use them for backup and history management. Snapshots are supported for any object on the navigation tree and can store information about an object alone (such as a table or module), or the objects within it as well (such as the tables within a module).
- **Multiple Language Support (MLS):** With this feature, you can store the displayed business names and descriptions in languages other than the base language of the repository. Your different translations of business names and descriptions can be used to deploy to an EUL in the language of the target user population.
- **Extensibility Through User-Defined Properties:** Users can define additional properties for any Warehouse Builder objects using the Warehouse Builder OMB



Plus scripting utility. After you define user-defined properties through scripting, you can access them in the user interface, the Oracle MetaBase (OMB) Scripting Language, Oracle Warehouse Builder Java API Reference s, and Warehouse Builder Design Browser. This enhances the extensibility of Warehouse Builder and makes it easier to integrate it with other Business Intelligence products.

- **Metadata Loader (Import and Export) Flexibility Enhancements:** Two new features were added to enhance this area of the product. The first is the ability for you to export metadata directly from Collections. The second feature is available from the Metadata Loader command line utility. It provides you with flexibility to specify the type of actions you want to apply when you import a first-class object.

### Process Flow Editor

Starting in this release, you can use the Process Flow Editor in Warehouse Builder to create and define process flows. External process operators that you previously defined in mappings are upgraded to user-defined processes and are contained within a process flow module. Process flows now integrate in the same Warehouse Builder design environment and no longer require you to use Oracle Workflow design client to perform these functions. The Warehouse Builder process flow modeler natively understands the semantic of your mappings and enables you to model activities such as FTP, email, etc.

### Performance Improvements

- **Mapping User Interface:** A new pre-defined display set, named Mapping, was added in this release. Selecting this display set causes the Mapping to only display columns that effectively are mapped, or used.
- **Mapping Compression:** This feature automatically detects unused connections between operators and attributes in any given mapping and eliminates them from the repository. This dramatically enhances the performance of loading and storing large mappings that represent significant data flows.
- **Metadata Loader (Import and Export):** Import and export functionality now takes advantage of the new compression feature available for each mapping. This means that the Metadata Loader now exports and imports only those mapping objects that are actually used.

### Oracle Database Integration

- **OLAP Integration:** Warehouse Builder enables you to design, deploy, and load multidimensional OLAP objects as ROLAP or MOLAP models from different data sources. After the data is loaded, you can use BI tools and applications to run complex analytical queries that answer your business questions. Using Warehouse Builder, you can now create and manage both your relational and multidimensional objects from the same cube and dimension designs.
- **Advanced Queue (AQ) Integration:** Warehouse Builder enables you to import Advanced Queue definitions and to use AQs as data sources and targets while designing your data warehouse. Through Advanced Queue functionality coupled with the Messaging Gateways, Warehouse Builder enables you to support messaging applications on MQ Series and Tibco as Warehouse Builder data sources. AQs also enable you to propagate change data capture from your source system to your target. The ability to integrate AQs lays the foundation for providing real time data warehousing in the future.
- **External Tables:** Starting in this release, you can use external tables to represent data from non-relational file sources in a relational, read-only format. You can

import an existing external table from an Oracle Database database. Or you can create an external table in Warehouse Builder based on a flat file definition. Warehouse Builder will generate the right DDL for you to deploy your external table to an Oracle Database database.

- **Oracle Database Multiple Table Inserts:** Warehouse Builder takes advantage of Oracle Database database functionality and generates a multiple-table insert statement when the target database is Oracle Database. This enables you to optimize mappings to insert data into multiple tables in one operation.
- **Oracle Database Table Functions:** Warehouse Builder introduces the Table Function operator that enables you to improve performance when loading your target system. Use this operator to develop custom code that can manipulate a set of input rows and return another set of rows possibly of different cardinality. Unlike conventional functions, table functions output a set of rows that can be queried like a physical table.

### Enhanced Support for Flat Files

- **Unbound Flat Files as Targets:** In this release, you can create a new, unbound flat file object as you create your mapping. Warehouse Builder creates a new comma-separated, single-record-type flat file in the specified location. This feature makes it easier to load the contents of a relational object into a flat file.
- **Outbound Reconcile for Flat Files:** Outbound reconciliation makes it possible to create a new repository object from a mapping flat file. This results in a new, comma-separated file to be created where specified, provided that the flat file is new to that repository. This feature makes it easier to "quickly dump" the contents of a relational object to a flat file.
- **Logical Records for Delimited Files:** The Flat File Sample Wizard has also been enhanced to display an improved user interface that allows you to define logical records for delimited files.
- **Position-Based Master-Detail Loading:** Position-based master-detail flat files are now easier to load with the use of additional mapping operators.
- **SQL Property Extensions:** You can now specify SQL properties for flat files you import into Warehouse Builder. This enables you to pre-define SQL property values for each flat file field. Thus, if mapping a flat file source to a relational target, the target column will default to these pre-defined SQL property values. These values will be used when building a relational target column or when creating an external table column.

### Mapping Editor Enhancements

- **Mapping User Interface:** A new set of property tabs is now available for you to quickly create and edit mapping operators and attribute properties.
- **Pivot and Unpivot Operators:** Starting in this release, you can add a pivot operator or an unpivot operator to a mapping. The pivot operator allows you to transform a single row of attributes into multiple rows. The unpivot operator converts multiple input rows into one output row.
- **Name and Address Operator Enhancements:** The Name and Address operator has been enhanced to include new input roles and output attributes. The United States Postal Service Code Accuracy Support System (CASS) reporting is also supported starting with this release.

## Warehouse Builder Is Now Available on UNIX Platforms

Starting with this release, Warehouse Builder is available on UNIX (Solaris, and Linux), as well as Windows (NT, 2000, and XP) platforms. This applies to all the components of Warehouse Builder, with the exception of the Name and Address libraries, which are not available on Linux in this release. (Note that the OLAP Bridges feature is only available on Windows platforms and the Name and Address Server is only available on Windows and Solaris platforms.)

## Conventions

In this manual, Windows refers to the Windows NT, Windows 2000, and Windows XP operating systems. The SQL\*Plus interface to Oracle Database may be referred to as SQL.

In the examples, an implied carriage return occurs at the end of each line, unless otherwise noted. You must press the Return key at the end of a line of input.

The following conventions are also used in this manual:

Convention	Meaning
.	Vertical ellipsis points in an example mean that information not directly related to the example has been omitted.
...	Horizontal ellipsis points in statements or commands mean that parts of the statement or command not directly related to the example have been omitted.
<b>boldface text</b>	Boldface type in text refers to interface buttons and links. Boldface type also serves as emphasis to set apart main ideas.
<i>italicized text</i>	Italicized text applies to new terms introduced for the first time. Italicized text also serves as an emphasis on key concepts.
<code>unicode text</code>	Unicode text denotes exact code, file directories and names, and literal commands.
<i>italicized unicode text</i>	Italicized unicode text refers to parameters whose value is specified by the user.
[]	Brackets enclose optional clauses from which you can choose one or none.

## Related Publications

The Warehouse Builder documentation set includes these manuals:

- Oracle Warehouse Builder User's Guide
- Oracle Warehouse Builder Installation and Configuration Guide
- Oracle Warehouse Builder Transformation Guide
- Oracle Warehouse Builder Scripting Reference
- Oracle Warehouse Builder Release Notes

In addition to the Warehouse Builder documentation, you can refer to other documents listed below:

- Oracle Database *Data Warehousing Guide*

Oracle provides additional information sources, including other documentation, training, and support services that can enhance your understanding and knowledge of Oracle Warehouse Builder .

- For more information on Oracle Warehouse Builder technical support, contact Oracle World Wide Support services at:

<http://www.oracle.com/support>

- For the latest information on, and downloads of, software and documentation updates to Oracle Warehouse Builder , visit MetaLink at:

<http://metalink.oracle.com>

- You can order other Oracle documentation at:

<http://oraclestore.oracle.com>

## Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at

<http://www.oracle.com/accessibility/>

### Accessibility of Code Examples in Documentation

JAWS, a Windows screen reader, may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, JAWS may not always read a line of text that consists solely of a bracket or brace.

## Contacting Oracle

### Oracle Metalink

Metalink is the Oracle support web site where you can find the latest product information, including documentation, patch information, BUG reports, and TAR entries. Once registered, you can access email, phone and web resources for all Oracle products. Metalink is located at:

<http://metalink.oracle.com>

Check Metalink regularly for Warehouse Builder information and updates.

### Documentation

You can order Oracle product documentation by phone or through the World Wide Web:

- **Phone:** Call 800-252-0303 to order documentation or request a fax listing of available Oracle documentation.
- **Oracle Documentation Sales Web site:** <http://oraclestore.oracle.com>

- **Oracle Customer Support Web site:** <http://www.oracle.com/support>



---

---

# Introduction

OMB Plus is a flexible, high-level command line metadata access tool for Oracle Warehouse Builder . Use OMB Plus to create, modify, delete, and retrieve object metadata in Warehouse Builder design and runtime repositories.

This chapter contains the following topics:

- [About the OMB Scripting Language](#) on page 1-1
- [OMB Plus Commands](#) on page 1-4
- [How to Read Syntax Diagrams](#) on page 1-9
- [Sample OMB Plus Scripts](#) on page 1-10

## About the OMB Scripting Language

The Warehouse Builder scripting language, known as OMB Plus, is an extension of the Tcl programming language. With OMB Plus, you can write the syntactic constructs such as variable support, conditional and looping control structures, error handling, and standard library procedures.

Use OMB Plus to create, modify, delete, and retrieve object metadata in Warehouse Builder design and runtime repositories.

OMB Plus enables you to edit Warehouse Builder repository metadata using a scripting interface. You can use this scripting interface to:

- Perform complex actions directly in Warehouse Builder, without launching the client user interface.
- Define sets of routine operations to be executed in Warehouse Builder.
- Perform batch operations in Warehouse Builder.
- Automate a series of conditional operations in Warehouse Builder.

## Using OMB Plus

To use OMB Plus, first launch OMB Plus and then connect to a repository. Type all commands and keywords in OMB Plus in uppercase.

### Launching OMB Plus

To launch the OMB Plus console, follow the instructions specific to your operating system.

- **UNIX:** At the command prompt, enter:

```
<OWB_HOME>/bin/unix/OMBPlus.sh
```

- **Windows:** From the Start Menu, navigate to Warehouse Builder, located within Oracle-OUI . Select OWB OMB Plus from the list of menu items.

## Connecting to a Repository

From the OMB Plus console, enter:

```
OMBCONNECT <repos>/<password>@<host>:<port>:<service_name>
```

In the expression above, <repos> is the name of the design-time repository, <host> is the machine on which the repository is installed, and <servicename> is the name of the database that contains the repository.

## Writing OMB Plus Commands

An OMB Plus command can span multiple lines. In a multiple line OMB Plus command, you must use only a backward slash (\) at the end of each line.

Keep in mind the following points when you execute OMB Plus commands:

- Do not use a number sign(#) in the middle of an OMB Plus command.
- Do not use a backward slash (\) when you specify the full path for the commands that use the full path, such as, OMBIMPORT, OMBVALIDATE, OMBLOG, and so on. For example, in the following commands, the log file is not created.

```
set OMBLOG c:\my_project\omb_logfile.log (On Windows)
set OMBLOG \home\my_project\omb_logfile.log (On Unix)
```

On Unix, use a slash as the path separator. For example, the following command creates a log file.

```
set OMBLOG /home/my_project/omb_logfile.log
```

On Windows, you can use either a forward slash(/) or two backward slashes (\\) as a path separator. Alternately, you can use a backward slash in the path, but in this case, enclose the entire filename in curly braces. The following are examples of commands that you can use to create a log file.

```
set OMBLOG c:/my_project/omb_logfile.log
set OMBLOG c:\\my_project\\omb_logfile.log
set OMBLOG {c:\my_project\omb_logfile.log}
```

- Do not use a semi-colon(;) in a quoted string. A semi-colon is special character that is used to separate two commands. Thus, using a semi-colon in a quoted string results in an error. The work around you can use for this is to escape the semi-colon character by putting a backward slash (\) in front of the semi-colon. For example,

```
OMBCREATE FLATFILE 'FF_DSR_RLE' \
SET PROPERTIES(DATA_FILE_NAME, IS_DEMILITED, CHARACTER_SET, RECORD_DELIMITER, \
FIELD_DELIMITER, FIELD_LEFT_ENCLOSURE, FIELD_RIGHT_ENCLOSURE) \
VALUES('DSR_RLE.dat', 'TRUE', 'WE8MSWIN1252', '\n', '\;', '','','')
```

- You can set the value of a Boolean configuration property, for example the IS\_DELIMITED property in the example, using any one of the following values: TRUE, FALSE, YES, NO, 1, or 0. When you set the value to TRUE, FALSE, YES, or NO, enclose the value in single quotes. When you specify numeric values for a configuration property, do not enclose the values in single quotes.



## Running Scripts in OMB Plus

You can write scripts and run them in OMB Plus. For examples of scripts you can write, see ["Sample OMB Plus Scripts"](#) on page 1-10.

Inside the interactive shell, type `source test.tcl` where 'test' is the name of the script you want to run.

At the command line, type `OMBPlus.sh test.tcl` for scripts on UNIX and `OMBPlus.bat test.tcl` for scripts on Windows operating systems.

## Locating Errors in Scripts and Multi-line Commands

OMB Plus reports only the first error it encounters while executing a command. As soon as it encounters the first error, it stops processing the command and exits reporting the error.

When an error occurs during the execution a multi-line OMB Plus command, the error message that is displayed does not specify the exact line at which the error occurred. To determine the line at which the error occurred, use the following command immediately after you encounter an error:

```
OMB+> puts $errorInfo
```

## Predefined Tcl Procedures

You can use the predefined Tcl procedures in OMB Plus:

- **OMBToSettableString:** Use this procedure when setting string values that contain single quotes that need to be escaped. The input for this procedure is a Tcl string and the output is a Tcl string with all single-quotes escaped.
- **OMBToTypeObjListString:** This procedure converts an input two-dimensional list to a comma-separated string. For example, the procedure converts input in the form of

```
{{<object_type> <name>} ... }  
to  
"<object_type> <name>,..."
```

- **OMBPageBreak:** This procedure displays the input string as a sequence of pages, with a pause after each page. When the output of a command is more than the page height, it may be difficult for screen reading software (used for accessibility) to read the whole text. This procedure may be used to break the output of a command into pages.

The two inputs to the OMBPageBreak command are the number of lines to be displayed in a page and the string that is to be split into pages. The string may be the output of an OMB Plus command. For example, the following command displays the output of the OMBHELP OMBCREATE command with 10 lines per page.

```
OMBPageBreak 10 [OMBHELP OMBCREATE]
```

To display the next 10 lines of the output, press <Enter> on your keyboard.

The OMPageBreak procedure is available for every OMB Plus session.

## OMB Plus Commands

The sections that follow describe the types of commands that comprise the OMB Scripting Language.

- **Metadata Manipulation Language (MML) Commands:** Includes commands for creating, altering, deleting, and retrieving metadata objects.
- **Shell Commands:** Includes help and environment support such as `OMBDCC` and `OMBHELP`. Although these commands enable you to control the scripting environment, you cannot use them to edit the metadata.
- **Administrative Commands:** Fits the MML to the Warehouse Builder backend. For example, the commands `OMBCONNECT`, `OMBDISCONNECT`, `OMBCOMMIT`, or `OMBROLLBACK`.
- **Navigation Commands:** Enable you to navigate the Warehouse Builder repository just as you would navigate a UNIX file system.
- **Service Commands:** Enable you to start Warehouse Builder metadata services such as validation, compilation, deployment, and import or export.

### Metadata Manipulation Language (MML) Commands

OMB Plus enables you to create, modify, delete, and retrieve object metadata in Warehouse Builder design and runtime repositories. OMB Plus commands work within the context of a first class object. For a list of first class objects, see "[Warehouse Builder Metadata Objects](#)" on page 1-5.

Table 1–1 lists the standard command names for MML.

**Table 1–1 Metadata Manipulation Language Commands**

Metadata Manipulation Language (MML)	Description
OMBCREATE	Creates a first class object.
OMBDROP	Deletes a first class object.
OMBCREATE	Modifies a first class object.
OMBRETRIEVE	Retrieves information from a first class object.

OMB Plus executes commands like `OMBCREATE`, `OMBCREATE`, and `OMBDROP` within a nested transaction.

OMB Plus interprets clauses within a single command one by one, as illustrated by the following example:

```
OMBCREATE TABLE 'T1' \
  MODIFY COLUMN 'C1' RENAME TO 'C1_NEW' \
  ADD UNIQUE_KEY 'UK1' \
  SET REF COLUMNS ('C1_NEW', 'C2')
```

In the above example, OMB Plus renames column `C1` to `C1_NEW` when parsing the `modify_column` clause. In the last line, use the new name for the column, `C1_NEW`, to specify the referenced columns for the new unique key. For more details about synchronization of cached data, see "[Synchronizing Cached Data with Repository Objects](#)"

The OMBCREATE and OMBRETRIEVE commands synchronize only the first content object that they are currently working on. The OMBCREATE command synchronizes only the parent folder.

## Examples

The following example lists the high-level scripting command syntax definitions for the OMBCREATE command:

```
OMBCREATE <fco_type> <fco_name> ( [ rename_clause ] [ properties_clause ] [ [ sco_
add_clause_for_alter ] | [ sco_modify_clause ] | [ sco_delete_clause ] ]* )1
rename_clause ::= RENAME TO <new_name>
sco_add_clause_for_alter ::= ADD <sco_type> <sco_name> [ OF parent_sco_clause ] [
AT POSITION <position> ] [ properties_clause ] [ references_clause ]*
sco_modify_clause ::= MODIFY <sco_type> <sco_name> [ OF parent_sco_clause ] ( [
rename_clause ] [ move_to_clause ] [ properties_clause ] [ references_clause ]* )1
move_to_clause ::= MOVE TO POSITION <position>
sco_delete_clause ::= DELETE <sco_type> <sco_name> [ OF parent_sco_clause ]
```

In the above example, the number 1 following a group of clauses enclosed by ( ) brackets indicates that you must specify at least one of the clauses.

You can specify a particular Warehouse Builder object by tracing the aggregation relationship from its parent first class object. You can also capture the association relationships by the references clauses. For example, `getSCOClause`, where `sco_type` is the second class object type.

Each action, create, alter, drop, or retrieve works only on the properties and the immediate children of the currently specified object. For example, the retrieve command on a table only enables you to access the properties of the table and the lists of column and constraint names owned by that table. To drill down to the detailed descriptions of the columns and constraints, you can call retrieve on these objects respectively.

The following statement retrieves the datatype and length for a column in a view:

```
OMBRETRIEVE VIEW 'V1' COLUMN 'COL1' \
GET PROPERTIES (DATATYPE, LENGTH)
```

When you set and retrieve properties using the `set_properties_clause` and the `get_properties_clause`, you can type the property names in any order.

Physical names are used as object identifiers in scripting. Business names represent an object property. Business names are not used to identify objects. You can identify a cross-component first class object by a path notation.

```
/<project_name>/<module_name>/<fco_name>
or
../<module_name> <fco_name>
```

String values, including object names and string property values, must be enclosed in single quotes.

### Warehouse Builder Metadata Objects

Use OMB Plus to access and manipulate the following Warehouse Builder objects, also known as first class objects:

**Table 1–2 Warehouse Builder Metadata Objects**

■ Advanced Queues	■ Functions	■ Process Flows
■ Collections	■ Gateway_Modules	■ Process Flow Modules
■ Connectors	■ Locations	■ Process Flow Packages
■ Cube Tables	■ Mappings	■ Projects
■ Deployment Action Plans	■ Materialized Views	■ Runtime Repository Connections
■ Dimension Tables	■ Object Types	■ SAP Modules
■ External Tables	■ Oracle Modules	■ Sequences
■ Flat File	■ Packages	■ Snapshots
■ Flat File Modules	■ Procedures	■ Tables

For Oracle Modules, you can access only those Oracle modules designated as warehouse modules. You cannot access Oracle source modules using OMB Plus.

## Shell Commands

Shell commands provide you with an interactive interface to run all Warehouse Builder scripts and standard Tcl commands. OMB Plus shell commands include: OMBHELP, OMBCC, OMBDCC, and OMBENV.

### OMBHELP

Use the OMBHELP command to display help on Warehouse Builder commands. The help describes the purpose of the command, the syntax in BNF format, and each of the keywords or options. For complex commands such as OMBCREATE, OMBCREATE, and OMBRETRIEVE, you can specify an optional `fcc_type` parameter. OMBHELP then displays the detailed syntax for that particular parameter type. Each command also provides specific options that enable you to display sub-sections of the help page.

The syntax for OMBHELP is:

```
help ::= OMBHELP <command_name> [ <command_specific_options> ] [DETAIL]
```

For example, OMBHELP OMBCONNECT displays the following:

```
OMBCONNECT
```

### Purpose

To connect to OWB repository.

### Syntax

```
OMBCONNECT <user>/<password>@<host:port:SID>
```

where

`<user>` is the OWB repository user name

`<password>` is the OWB repository user password

`<host>` is the name or IP address of the OWB repository host machine

`<port>` is the numeric port for OWB repository database listener

`<SID>` is the unique database identifier for OWB repository database

Notes:

The connection to OWB repository will be established in single user mode.

If you type `OMBHELP <command_name>` followed by `[DETAIL]`, OMB Plus displays the command purpose, prerequisites, syntax, descriptions for each keyword and parameter, and examples of how to use the command.

The `OMBHELP` command synchronizes only the only the FCO that you are currently working on.

## OMBENV

The syntax for `OMBENV` is:

```
environment ::= OMBENV
```

This command lists the values for all Warehouse Builder-specific environment variables. [Table 1-3](#) lists the environmental variables. To set an environmental variable, use the `Tcl set` command. Use `unset` to `unset` an environmental variable.

**Table 1-3 Warehouse Builder Environment Variables**

Environment Variable	Meaning	Possible Values
OMBTIMER	Enables timing on each Warehouse Builder scripting command. The time is logged to a log file and to the console or shell.	A Tcl boolean value.
OMBLOG	Stores the filename for Warehouse Builder log file.	A valid filename including its path.
OMBPROMPT	Indicates whether OMB Plus will update the command prompt each time the you call <code>OMBCC</code> .	A Tcl boolean value.
OMBCONTINUE_ON_ERROR	Ignores errors that occur in any command that is part of a script and moves to the next command in the script.	A Tcl boolean value.

## Administrative Commands

Use these commands to perform administrative jobs on a Warehouse Builder repository. The following commands are available: `OMBCONNECT`, `OMBDISCONNECT`, `OMBCOMMIT`, and `OMBROLLBACK`.

```
connect ::= OMBCONNECT <username>/<password>@<host>:<port>:<sid>
disconnect ::= OMBDISCONNECT
commit ::= OMBCOMMIT
rollback ::= OMBROLLBACK
```

## Navigation Commands

You can use the following commands to navigate the Warehouse Builder repository in the same way you navigate a UNIX file system.

### OMBCC

This command enables users to change context Up and Down the Warehouse Builder navigation tree. For example, when you type `...` the current context changes to the parent context. However, if the current context is a modified project, an error message prompts you to commit or rollback your changes.

## OMBDCC

This command shows you the current context and the context type. The syntax for OMBDCC is:

```
display_current_context ::= OMBDCC
```

## OMBLIST

The child first class objects for folders are listed under OMBLIST. Using this command on folders describes only the folder properties. Note also that the list command allows name matching by regular expression. If you do not include the regular expression, then OMBLIST displays all objects sorted alphabetically.

The generic syntax for OMBLIST in a folder context is:

```
list_folder ::= OMBLIST ( <child_type1_plural> | ... | <child_typeN_plural> ) [ name_
in_regexp ]
name_in_regexp ::= a name in regular expression.
```

For example, under the root context you have:

```
list_root ::= OMBLIST PROJECTS [ name_in_regexp ]
```

The OMBLIST command synchronizes all parent-child relations in the navigation tree.

## Service Commands

Service commands perform services like batch operations on Warehouse Builder metadata. [Table 1–4](#) contains a list of service commands and their descriptions.

**Table 1–4 Service Commands**

Command	Description
OMBCOMPILE	Use this command to compile folders or first class objects such as tables, views, sequences, dimensions, and cubes.
OMBDEPLOY	This command provides deployment service.
OMBIMPORT	This command provides the metadata import service. You can only invoke the OMBIMPORT command from the root context.  The four available modes are: CREATE_MODE (CREATE), REPLACE_MODE (REPLACE), UPDATE_MODE (UPDATE), and MERGE_MODE (INCREMENTALUPDATE). The default mode, if not specified in the command, is CREATE_MODE.
OMBVALIDATE	Use this command to validate folders or first class objects such as tables, views, sequences, dimensions, and cubes.

### Synchronizing Cached Data with Repository Objects

When you start an OMB Plus session, data about the objects is fetched from the OWB repository and cached in the OMB Plus session. The cached data is synchronized with the data from the repository at certain predefined intervals.

The extent to which the objects are synchronized depends on the OMB Plus command that you execute. For example, some commands synchronize all the parent-child relationships in the navigation tree, whereas some commands synchronize only the first class object that they are currently working on.

- The OMBLIST command synchronizes all the parent-child relationships in the navigation tree.

- The OMBCREATE and OMBRETRIEVE commands synchronize only the first class object that they are currently working on.
- The OMBCREATE command synchronizes only the parent folder.

Consider the following example on synchronization of cached data. You open an OMB Plus session as well as the OWB Client. In the OWB Client, you delete a flat file module called FFM1. You then undelete FFM1 and commit the changes. In the OMB Plus session, you perform the sequence of operations listed below. The details of the result of the operation and the logic behind the result is explained below.

1. OMBCC FFM1

The context is changed to the module FFM1.

2. OMBCREATE FLATFILE

The flat file is not created because OMBCREATE synchronizes the parent folder. When the synchronization is performed, the parent folder is not found in the cache.

3. OMBLIST FLAT\_FILE\_MODULE

FFM1 is listed as one of the modules. This is because OMBLIST synchronizes all the parent-child relationships in the navigation tree.

4. OMBCREATE FLATFILE

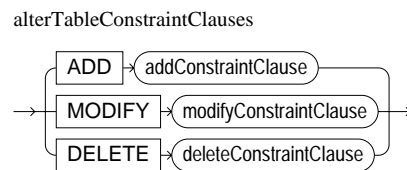
The flat file is created. This is because the undelete is now reflected in the cache because of the OMBLIST command.

## How to Read Syntax Diagrams

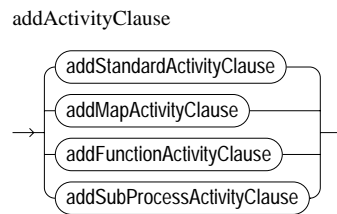
Syntax diagrams are drawings that illustrate valid SQL syntax. To read a diagram, trace it from left to right, in the direction shown by the arrows. Commands and other keywords appear in UPPERCASE inside rectangles. Parameters appear in lowercase inside ovals. Variables are used for the parameters. Punctuation, operators, delimiters, and terminators appear inside circles.

If the syntax diagram has more than one path, you can choose any path to travel. For example, [Figure 1-1](#) shows a syntax diagram that indicates you can specify either ADD, MODIFY, or DELETE:

**Figure 1-1** Syntax Diagram with Multiple Paths



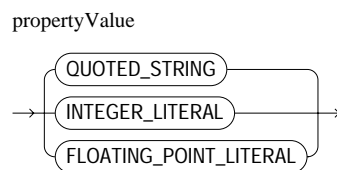
If you have the choice of more than one keyword, operator, or parameter, syntax diagrams display the options in a vertical list. For example, in the syntax diagram shown in [Figure 1-2](#), you can specify one or more of the multiple parameters in the stack:

**Figure 1–2 Syntax Diagram with Multiple Parameters**

## Required Keywords and Parameters

Required keywords and parameters can appear singly or in a vertical list of alternatives. Single required keywords and parameters appear on the *main path* – that is, on the horizontal line you are currently traveling.

If multiple keywords or parameters appear in a vertical list that intersects the main path, one of them is required. You must choose one of the keywords or parameters, but not necessarily the one that appears on the main path. In the example shown in [Figure 1–3](#), select one of the displayed settings:

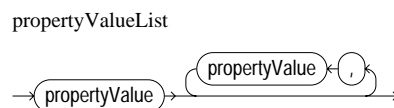
**Figure 1–3 Syntax Diagram with Multiple Choices for a Required Parameter**

## Optional Keywords and Parameters

If keywords and parameters appear in a vertical list *above* the main path, they are optional.

## Syntax Loops

Loops enable you to repeat the syntax within them as many times as you like. In the example in [Figure 1–4](#), you can choose one property value and repeatedly to choose another. Separate your selections by commas.

**Figure 1–4 Syntax Diagram with a Syntax Loop**

## Sample OMB Plus Scripts

The body of the Scripting Reference contains examples for almost every reference topic. This section contains lengthy examples that are appropriate in the context of a single OMB command statement. These examples are intended to provide uninterrupted the series of steps that you would use to take advantage of particular Warehouse Builder functionality. They do not replace the syntax diagrams and semantics found for each individual OMB command statement in the body of the reference.



This section includes sample scripts for the following tasks:

- [Updating a Design Repository](#)
- [Reporting on Repository Objects](#)
- [Finding Invalid Objects](#)
- [Using OMB Plus to Navigate Repositories](#)

## Updating a Design Repository

One possible use case is to perform mass update on repository metadata. Users can write the following script to add a primary key with local column ID for each table with name beginning in EDW inside the module MY\_MODULE:

```
OMBCC MY_MODULE;
foreach tableName [OMBLIST TABLE EDW*] { \
  OMBCREATE TABLE '$tableName' \
  ADD PRIMARY_KEY '$tableName_PK' SET REFERENCE COLUMNS ('ID');}
```

We can build even more powerful and useful script using if-then-else:

```
foreach tableName [OMBLIST TABLE EDW*] { \
  set columnList [OMBRETRIEVE TABLE '$tableName' GET COLUMNS]; # Use lsearch to
  search for a name in a list
  if {[lsearch $columnList 'ID'] == -1} {
    OMBCREATE TABLE '$tableName' \
      ADD COLUMN 'ID' \
        SET PROPERTIES (DATATYPE, LENGTH, NOT_NULL) VALUES \
          ('NUMBER', 10, 'true');
  }
}
```

The above script checks the list of tables which name begins with EDW whether each of them contains an ID column. If not, it will create an ID column for that table. Hence, executing the above script will guarantee that all tables with names beginning in EDW will have the ID column defined.

## Reporting on Repository Objects

Another common use is for reporting purpose. The following script displays the properties of the table T1 and its column definitions on standard output:

```
#Displaying metadata of a table
puts -nonewline "Please enter the table name: " gets stdin tableName
puts ""
puts "Report on $tableName"
puts "====="
puts "Physical name = $tableName"
puts "Logical name = [lindex [OMBRETRIEVE TABLE '$tableName' GET \
PROPERTIES(BUSINESS_NAME)] 0]"
puts "Description = [lindex [OMBRETRIEVE TABLE '$tableName' GET \
PROPERTIES(DESCRIPTION)] 0]"
puts "-----"
set columnList [OMBRETRIEVE TABLE '$tableName' GET COLUMNS]
set i 1
foreach colName $columnList {
  set dt [lindex [OMBRETRIEVE TABLE '$tableName' COLUMN '$colName' GET \
PROPERTIES(DATATYPE)] 0]
  if { $dt == "VARCHAR2" } {
    set prop [OMBRETRIEVE TABLE '$tableName' COLUMN '$colName' GET
PROPERTIES(LENGTH, NOT_NULL)]
```

```

        puts "Column $i: $colName datatype=VARCHAR2 length=[lindex $prop 0] \
            not_null=[lindex $prop 1]"
    } elseif { $dt == "NUMBER" } {
        set prop [OMBRETRIEVE TABLE '$tableName' COLUMN '$colName' \
            GET PROPERTIES(PRECISION, SCALE, NOT_NULL)]
        puts "Column $i: $colName datatype=NUMBER precision=[lindex $prop 0] \
            scale=[lindex $prop 1] not_null=[lindex $prop 2]"
    } elseif { $dt == "DATE" } {
        set prop [OMBRETRIEVE TABLE '$tableName' COLUMN '$colName' GET \
            PROPERTIES(NOT_NULL)]
        puts "Column $i: $colName datatype=DATE not_null=[lindex $prop 0]"
    } # end else
    incr i
}

```

A sample output is like the following:

```

Physical name = T1
Logical name = Table 1
Description = This is my first table.
=====
Column: ID datatype=NUMBER precision=0 scale=0 not_null=1
Column: NAME datatype=VARCHAR2 length=30 not_null=1
Column: VALUE datatype=VARCHAR2 length=100 not_null=0

```

## Finding Invalid Objects

Users can also take advantage of the validation service provided by scripting, like this:

```

set tableList [OMBLIST TABLES];
foreach tableName $tableList {
    if { [OMBCOMPILE TABLE '$tableName' == "Invalid." ] {
        set context [OMBDCC];
        puts "Table $context/$tableName is invalid.";
    }
}

```

The above script will tell users which table is invalid under the current module.

## Using OMB Plus to Navigate Repositories

Another scenario we present below is for a disabled user that relies on OMB Plus interactive shell (and also some ADA screen reading software) to navigate through a Warehouse Builder repository:

```

OMB> OMBCONNECT owb/owb@localhost:1521:dev901
Connected.
OMB> OMBLIST PROJECTS
DIM_TEST_PROJ MY_PROJECT PROJ_ENDTOEND PROJ_RELATIONAL TEST_DIM_PROJ
OMB> OMBLIST PROJECTS .*RELATION.*
PROJ_RELATIONAL
OMB> OMBCC 'PROJ_RELATIONAL'
Context changed.
OMB> OMBDCC
PROJECT /PROJ_RELATIONAL
OMB> set OMBPROMPT ON
ON
OMB> OMBDCC
PROJECT /PROJ_RELATIONAL
/PROJ_RELATIONAL>
/PROJ_RELATIONAL> OMBLIST ORACLE_MODULES
WH
/PROJ_RELATIONAL> OMBCC 'WH'

```

```
Context changed.  
/PROJ_RELATIONAL/WH> OMBLIST TABLES  
PRODUCT PO  
/PROJ_RELATIONAL/WH> OMBRETRIEVE TABLE 'PO' GET COLUMNS  
OID PROD_ID ORDER_DATE CUSTNAME  
/PROJ_RELATIONAL/WH> OMBCC `..`  
Context changed.  
/PROJ_RELATIONAL> OMBCC `..`  
Context changed.  
/>  
/> OMBDISCONNECT  
Disconnected.
```



---



---

## OMB Plus Commands

This chapter contains an alphabetical listing of the navigation, service, administrative, and shell commands you can use in OMB Plus. Commands for metadata manipulation are contained in separate chapters.

This chapter includes the following topics listed in a columnar table that reads down the columns from left to right to conserve space:

---

<a href="#">OMBCC</a> on page 2-2	<a href="#">OMBDISC</a> on page 2-26	<a href="#">OMBLIST_SNAPSHOT</a> on page 2-46
<a href="#">OMBCOMMIT</a> on page 2-3	<a href="#">OMBDISCONNECT</a> on page 2-27	<a href="#">OMBLOCK</a> on page 2-49
<a href="#">OMBCOMPARE_SNAPSHOT</a> on page 2-4	<a href="#">OMBENV</a> on page 2-28	<a href="#">OMBMOVE</a> on page 2-52
<a href="#">OMBCOMPILE</a> on page 2-8	<a href="#">OMBEXPORT</a> on page 2-29	<a href="#">OMBRECONCILE</a> on page 2-55
<a href="#">OMBCONN</a> on page 2-10	<a href="#">OMBEXPORT_MDL_FILE</a> on page 2-41	<a href="#">OMBREDEFINE_CLASS_DEFINITION</a> on page 2-55
<a href="#">OMBCONNECT_RUNTIME</a> on page 2-12	<a href="#">OMBHELP</a> on page 2-37	<a href="#">OMBREGISTER_LOCATION</a> on page 2-63
<a href="#">OMBCONNECT</a> on page 2-14	<a href="#">OMBIMPORT</a> on page 2-38	<a href="#">OMBRESTORE_SNAPSHOT</a> on page 2-70
<a href="#">OMBCOPY</a> on page 2-16	<a href="#">OMBIMPORT_MDL_FILE</a> on page 2-41	<a href="#">OMBROLLBACK</a> on page 2-74
<a href="#">OMBDCC</a> on page 2-19	<a href="#">OMBLIST</a> on page 2-44	<a href="#">OMBUNLOCK</a> on page 2-75
<a href="#">OMBDEPLOY</a> on page 2-20	<a href="#">OMBLIST_SNAPSHOT</a> on page 2-46	<a href="#">OMBVALIDATE</a> on page 2-78

---

## OMBCC

### Purpose

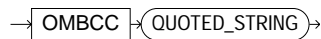
OMBCC - Change Context command allows users to change the current context to the desired location in OWB tree. The target context can be specified either as an absolute path starting from the root ('/') or as a relative path starting from the current context. Also, the path can contain '..', which allows to navigate "up" to the parent context.

### Prerequisites

Must be connected to a OWB repository.

### Syntax Diagrams

parseChangeContextCommand



### Syntax

```
parseChangeContextCommand = OMBCC "QUOTED_STRING" ;
```

### Keywords and Parameters

parseChangeContextCommand

Specify change context command.

QUOTED\_STRING

The target context to switch to.

### Examples

```
OMBCC '/'
```

changes the context to the root.

```
OMBCC '/MY_PROJECT/ORACLE_1'
```

changes the context to Oracle module 'ORACLE\_1', within project 'MY\_PROJECT'.

OMBCC '..' changes the context to the parent of current context (to the project level, if the current context is an Oracle module, for example).

### See Also

OMBDCC

## OMBCOMMIT

### Purpose

OMBCOMMIT - Perform commit action on the repository.

### Prerequisites

Must be connected to a OWB repository.

### Syntax Diagrams

parseCommitCommand



### Syntax

```
parseCommitCommand = OMBCOMMIT ;
```

### Keywords and Parameters

parseCommitCommand  
Specify commit command.

### Examples

```
OMBCOMMIT
```

### See Also

OMBROLLBACK

# OMBCOMPARE\_SNAPSHOT

## Purpose

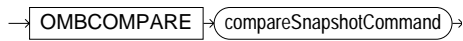
OMBCOMPARE SNAPSHOT - Change management is a key piece of metadata management. This command provides comparison services of any complex object model in the repository. This command writes the diff between snapshot/component to XML file.

## Prerequisites

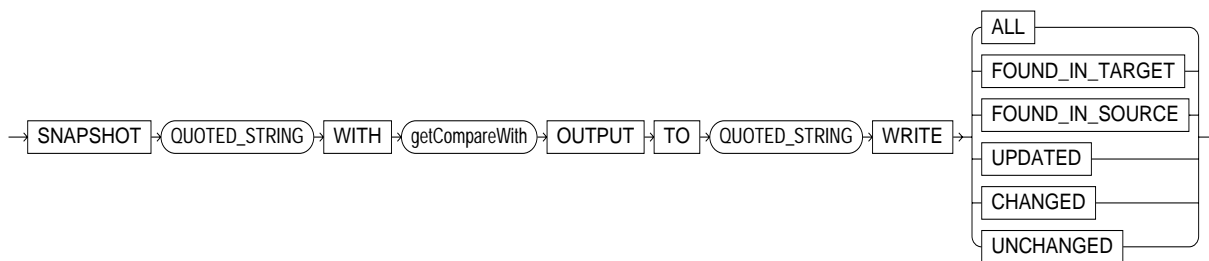
Snapshots can be compared from any context and either with another snapshot or the current repository objects.

## Syntax Diagrams

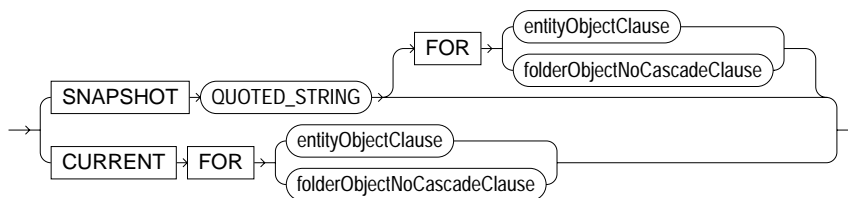
parseCompareCommand



compareSnapshotCommand

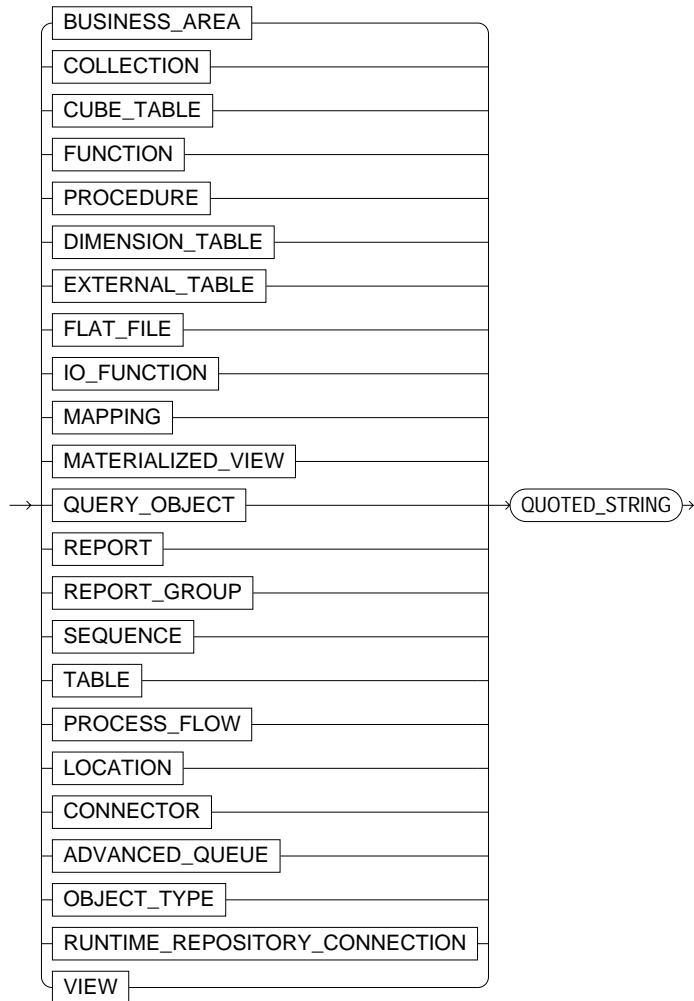


getCompareWith

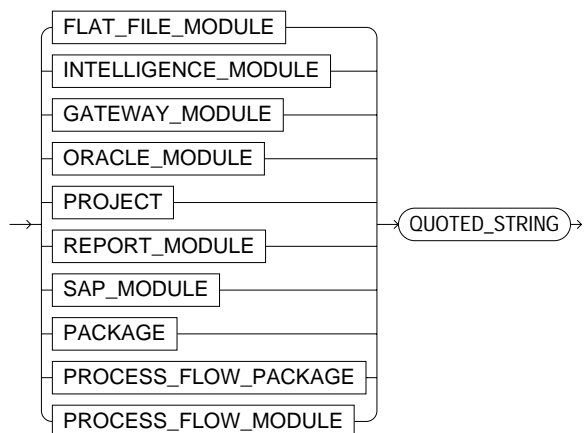




entityObjectClause



folderObjectNoCascadeClause



### Syntax

```
parseCompareCommand = OMBCOMPARE "compareSnapshotCommand";
```

```
compareSnapshotCommand = (( SNAPSHOT "QUOTED_STRING" ) WITH  
"getCompareWith" OUTPUT TO "QUOTED_STRING" WRITE ( ALL | FOUND_IN_  
TARGET | FOUND_IN_SOURCE | UPDATED | CHANGED | UNCHANGED ));  
getCompareWith = SNAPSHOT "QUOTED_STRING" [ FOR ( "entityObjectClause" |  
"folderObjectNoCascadeClause" ) ] | ( CURRENT FOR ( "entityObjectClause" |  
"folderObjectNoCascadeClause" ) );  
entityObjectClause = ( ( BUSINESS_AREA | COLLECTION | CUBE_TABLE |  
FUNCTION | PROCEDURE | DIMENSION_TABLE | EXTERNAL_TABLE | FLAT_  
FILE | IO_FUNCTION | MAPPING | MATERIALIZED_VIEW | QUERY_OBJECT |  
REPORT | REPORT_GROUP | SEQUENCE | TABLE | PROCESS_FLOW |  
LOCATION | CONNECTOR | ADVANCED_QUEUE | OBJECT_TYPE | RUNTIME_  
REPOSITORY_CONNECTION | VIEW ) "QUOTED_STRING" );  
folderObjectNoCascadeClause = ( FLAT_FILE_MODULE | INTELLIGENCE_  
MODULE | GATEWAY_MODULE | ORACLE_MODULE | PROJECT | REPORT_  
MODULE | SAP_MODULE | PACKAGE | PROCESS_FLOW_PACKAGE |  
PROCESS_FLOW_MODULE ) "QUOTED_STRING";
```

## Keywords and Parameters

**parseCompareCommand**

Root production of OMBCOMPARE SNAPSHOT.

**compareSnapshotCommand**

To compare components of snapshots.

**QUOTED\_STRING**

Name of source snapshot which needs to be compared with the target snapshot.

**OUTPUT**

Specifies output filename where the XML comparison result will written.

**WRITE**

Specifies filter clause which will make the diff engine only write specified objects of a certain diff state.

**getCompareWith**

Target of the comparison.

**SNAPSHOT**

Target snapshot which will be compared with the source snapshot.

**FOR**

Specifies component which exists in the current repository.

**CURRENT**

Indicates current component's definition as the target of compare action.

**entityObjectClause**

Clause that can either refer to the relative path of the component in the current repository or the fully-qualified path of an object from a snapshot. If both source and target are snapshots, then it has to be the fully-qualified path of the component within one of the snapshots. If the target is the current repository, then it has to be either the relative path of the component within the repository or the fully qualified path in the source snapshot.

**folderObjectNoCascadeClause**

Clause that can either refer to the relative path of the folder in the current repository or the fully-qualified path of a folder from a snapshot. If both source and target are snapshots, then it has to be the fully-qualified path of the folder within one of the snapshots. If the target is the current repository, then it has to be either the relative path of the folder within the repository or the fully qualified path in the source snapshot.

**Examples**

OMBCOMPARE SNAPSHOT 'S1' WITH SNAPSHOT 'S2' OUTPUT TO 'd:diff.xml'  
WRITE CHANGED This command compares snapshot S1 with S2, and writes objects with CHANGED state into diff.xml, Though the diff engine allows to compare any two snapshot, even two unrelate snapshots with totally different components in them. CHANGED filter writes FOUND\_IN\_SOURCE or FOUND\_IN\_TARGET or UPDATED components, CHANGED state represents whole diff, OMBCOMPARE SNAPSHOT 'S1' WITH SNAPSHOT 'S2' OUTPUT TO 'd:diff.xml' WRITE FOUND\_IN\_SOURCE This command writes objects which are only found in snapshot S1.  
OMBCOMPARE SNAPSHOT 'S1' WITH CURRENT FOR TABLE '/Project1/WH1/T1' OUTPUT TO 'd:diff.xml' WRITE ALL This command writes all table objects with any diff state.

**See Also**

OMBCREATE SNAPSHOT, OMBALTER SNAPSHOT, OMBDROP SNAPSHOT,  
OMBRESTORE SNAPSHOT, OMBLIST SNAPSHOT, OMBRETRIEVE SNAPSHOT

# OMBCOMPILE

## Purpose

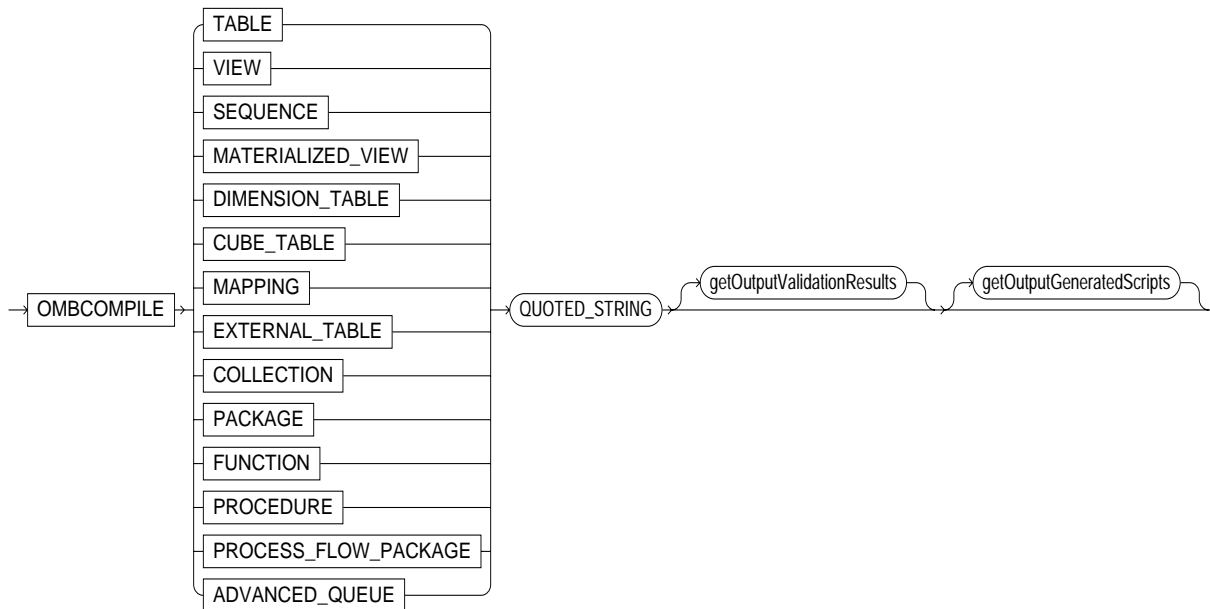
OMBCOMPILE - This command compiles a repository object. The results are generated in a file in a user defined directory.

## Prerequisites

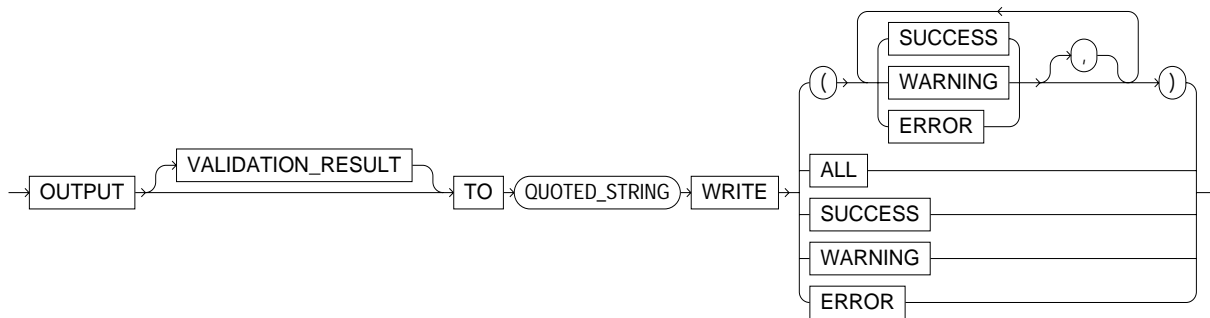
In the context of a Oracle Module.

## Syntax Diagrams

parseCompileCommand



getOutputValidationResults



getOutputGeneratedScripts



## Syntax

```

parseCompileCommand = OMBCOMPILE (( TABLE | VIEW | SEQUENCE |
MATERIALIZED_VIEW | DIMENSION_TABLE | CUBE_TABLE | MAPPING |
EXTERNAL_TABLE | COLLECTION | PACKAGE | FUNCTION | PROCEDURE |
PROCESS_FLOW_PACKAGE | ADVANCED_QUEUE ) "QUOTED_STRING" [
"getOutputValidationResults" ] [ "getOutputGeneratedScripts" ] );

getOutputValidationResults = OUTPUT [ VALIDATION_RESULT ] TO "QUOTED_
STRING" WRITE (( "(" (( SUCCESS | WARNING | ERROR ) [ "," ] )+ ")" ) | ALL |
SUCCESS | WARNING | ERROR );

getOutputGeneratedScripts = OUTPUT GENERATION_SCRIPTS TO "QUOTED_
STRING";

```

## Keywords and Parameters

`parseCompileCommand`

This command compiles a repository object.

`QUOTED_STRING`

The name of the object.

`getOutputValidationResults`

This clause outputs the validation results to one or more files in the specified folder.

`QUOTED_STRING`

A directory where validation results are stored.

`getOutputGeneratedScripts`

This clause outputs the generated scripts for an object to one or more files in specified folder.

`QUOTED_STRING`

A directory where generated scripts are stored.

## Examples

```

OMBCOMPILE TABLE 'T1' OUTPUT VALIDATION_RESULT TO '/tmp' WRITE
SUCCESS OUTPUT GENERATION_SCRIPTS TO '/tmp'. OMBCOMPILE TABLE 'T1'
OUTPUT GENERATION_SCRIPTS TO '/tmp'

```

The first example gets the validation results and generated scripts for the table, whereas the second example gets only the generated scripts.

## See Also

OMBVALIDATE

## OMBCONN

### Purpose

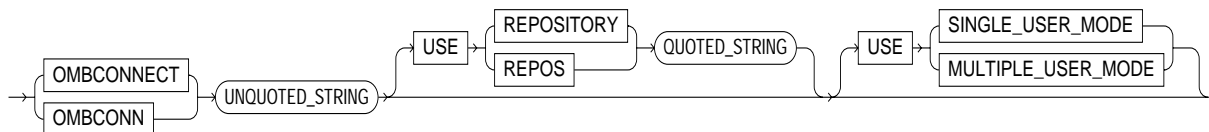
OMBCONN - To connect to a OWB repository.

### Prerequisites

Must not be connected to another OWB repository. If connected to another repository, use OMBDISCONNECT to disconnect first.

### Syntax Diagrams

parseConnectCommand



### Syntax

```

parseConnectCommand = ( ( OMBCONNECT | OMBCONN ) "UNQUOTED_
STRING" [ USE ( REPOSITORY | REPOS ) "QUOTED_STRING" ] [ USE ( SINGLE_
USER_MODE | MULTIPLE_USER_MODE ) ] );
  
```

### Keywords and Parameters

parseConnectCommand

Specify connect command.

UNQUOTED\_STRING

Specify the connection string to the database, in the format:  
username/password@host:port:service name

QUOTED\_STRING

Optionally, specify the name of a repository to work on. If not provided, the default repository will be used.

SINGLE\_USER\_MODE

If specified, the user will use the repository exclusively.

MULTIPLE\_USER\_MODE

If specified, more than one session can work on the same repository at the same time. This is the default mode.

**Examples**

OMBCONNECT owb\_normal\_user/welcome@dwsun42:1521:dev817 USE  
REPOSITORY 'owb\_repos' USE SINGLE\_USER\_MODE will connect a normal user  
owb\_normal\_user to database and work on repository named 'owb\_repos' in single  
user mode.

**See Also**

OMBDISCONNECT

# OMBCONNECT\_RUNTIME

## Purpose

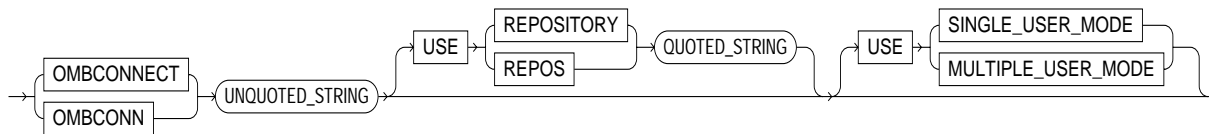
OMBCONNECT - To connect to a OWB repository.

## Prerequisites

Must not be connected to another OWB repository. If connected to another repository, use OMBDISCONNECT to disconnect first.

## Syntax Diagrams

parseConnectCommand



## Syntax

```

parseConnectCommand = ( ( OMBCONNECT | OMBCONN ) "UNQUOTED_
STRING" [ USE ( REPOSITORY | REPOS ) "QUOTED_STRING" ] [ USE ( SINGLE_
USER_MODE | MULTIPLE_USER_MODE ) ] );
  
```

## Keywords and Parameters

parseConnectCommand

Specify connect command.

UNQUOTED\_STRING

Specify the connection string to the database, in the format:  
username/password@host:port:service name

QUOTED\_STRING

Optionally, specify the name of a repository to work on. If not provided, the default repository will be used.

SINGLE\_USER\_MODE

If specified, the user will use the repository exclusively.

MULTIPLE\_USER\_MODE

If specified, more than one session can work on the same repository at the same time. This is the default mode.



**Examples**

OMBCONNECT owb\_normal\_user/welcome@dwsun42:1521:dev817 USE  
REPOSITORY 'owb\_repos' USE SINGLE\_USER\_MODE will connect a normal user  
owb\_normal\_user to database and work on repository named 'owb\_repos' in single  
user mode.

**See Also**

OMBDISCONNECT

# OMBCONNECT

## Purpose

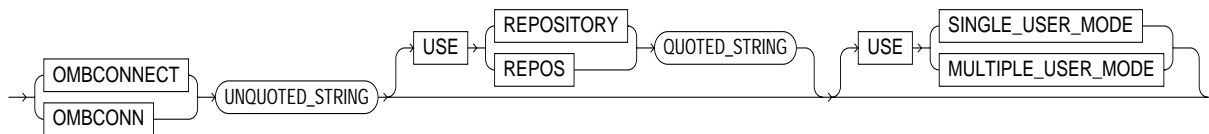
OMBCONNECT - To connect to a OWB repository.

## Prerequisites

Must not be connected to another OWB repository. If connected to another repository, use OMBDISCONNECT to disconnect first.

## Syntax Diagrams

parseConnectCommand



## Syntax

```

parseConnectCommand = ( ( OMBCONNECT | OMBCONN ) "UNQUOTED_
STRING" [ USE ( REPOSITORY | REPOS ) "QUOTED_STRING" ] [ USE ( SINGLE_
USER_MODE | MULTIPLE_USER_MODE ) ] );
  
```

## Keywords and Parameters

parseConnectCommand

Specify connect command.

UNQUOTED\_STRING

Specify the connection string to the database, in the format:  
username/password@host:port:service name

QUOTED\_STRING

Optionally, specify the name of a repository to work on. If not provided, the default repository will be used.

SINGLE\_USER\_MODE

If specified, the user will use the repository exclusively.

MULTIPLE\_USER\_MODE

If specified, more than one session can work on the same repository at the same time. This is the default mode.

### Examples

OMBCONNECT owb\_normal\_user/welcome@dwsun42:1521:dev817 USE REPOSITORY 'owb\_repos' USE SINGLE\_USER\_MODE will connect a normal user owb\_normal\_user to database and work on repository named 'owb\_repos' in single user mode.

### See Also

OMBDISCONNECT

## OMBCOPY

### Purpose

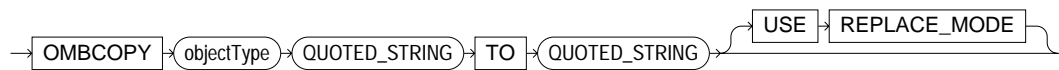
OMBCOPY - Copy one or more objects of the same object type. The replace option allows you to overwrite.

### Prerequisites

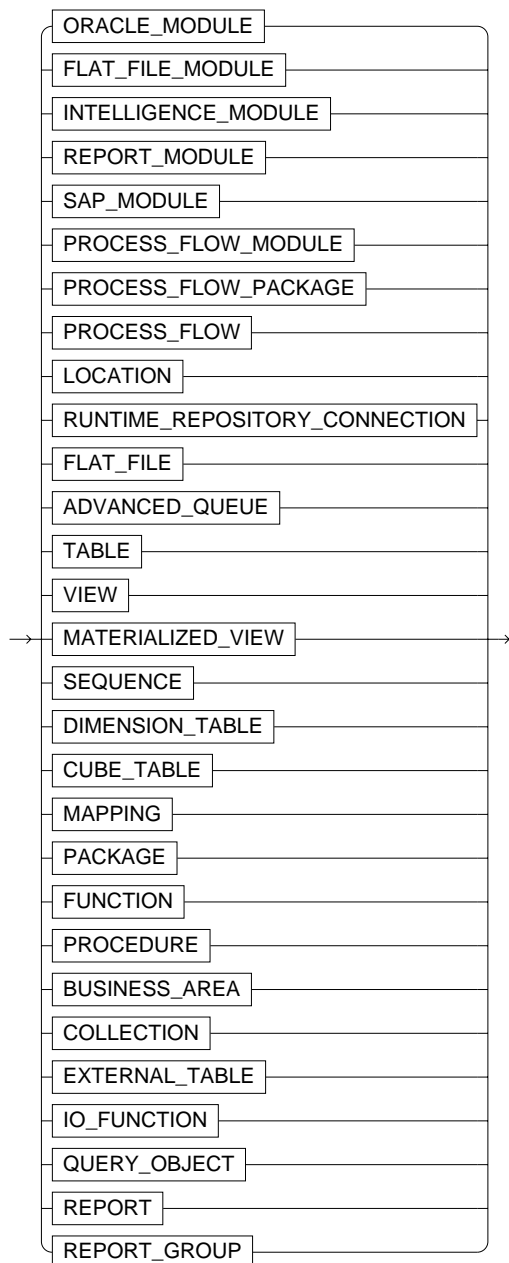
Use of relative path specifications requires awareness of the current context.

### Syntax Diagrams

parseCopyCommand



objectType



## Syntax

```
parseCopyCommand = OMBCOPY "objectType" "QUOTED_STRING" TO "QUOTED_STRING" [ USE REPLACE_MODE ];
```

```
objectType = ( ORACLE_MODULE | FLAT_FILE_MODULE | INTELLIGENCE_MODULE | REPORT_MODULE | SAP_MODULE | PROCESS_FLOW_MODULE | PROCESS_FLOW_PACKAGE | PROCESS_FLOW | LOCATION | RUNTIME_REPOSITORY_CONNECTION | FLAT_FILE | ADVANCED_QUEUE | TABLE | VIEW | MATERIALIZED_VIEW | SEQUENCE | DIMENSION_TABLE | CUBE_TABLE | MAPPING | PACKAGE | FUNCTION | PROCEDURE | BUSINESS_AREA | COLLECTION | EXTERNAL_TABLE | IO_FUNCTION | QUERY_OBJECT | REPORT | REPORT_GROUP );
```

## Keywords and Parameters

`parseCopyCommand`

Specifies the source object type, source path, and target path for the object to copy.

`QUOTED_STRING`

Source and target path specifications can be absolute or relative. To copy multiple objects, include a regular expression as the final step of the source path. If you are copying multiple objects, the final step of the target path must be the folder to which the objects are being copied. If you are only copying one object, you can specify the object's original name or a new name as the final step of the target path.

`REPLACE_MODE`

Use this option to overwrite existing target objects.

`objectType`

Copying objects is subject to the following restrictions: 1. You cannot copy an entire project. 2. When copying objects between projects, you can only copy objects into the current project; you cannot copy objects out to other projects.

## Examples

```
OMBCOPY TABLE 'MY_TABLE1' TO 'MY_TABLE2' USE REPLACE_MODE
OMBCOPY ORACLE_MODULE '/MY_PROJECT/WH1' TO '/MY_PROJECT/WH2'
OMBCOPY TABLE 'MY_.*' TO '/MY_PROJECT/WH2'
```

## See Also

`OMBMOVE`

---

## OMBDC

### Purpose

OMBDC - Display Current Context command displays the current context (location) and, for contexts other than the root ('/'), the type of the current folder.

### Prerequisites

Must be connected to a OWB repository.

### Syntax Diagrams

parseDisplayCurrentContextCommand

→ OMBDC →

### Syntax

parseDisplayCurrentContextCommand = OMBDC ;

### Keywords and Parameters

parseDisplayCurrentContextCommand

Specify display current context command.

### Examples

OMBDC will display / if the current context is the root. OMBDC will display PROJECT /MY\_PROJECT if the current context is the project 'MY\_PROJECT'.

### See Also

OMBCC

## OMBDEPLOY

### Purpose

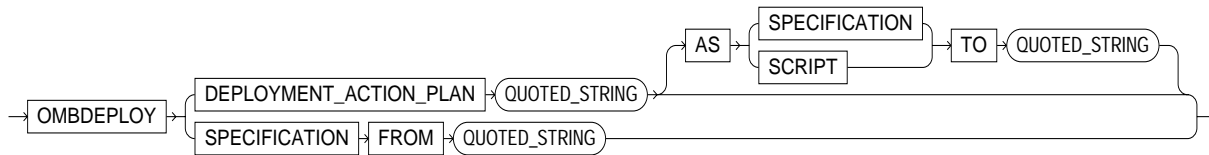
OMBDEPLOY - To Deploy Action Plans to Runtime Platforms or File Systems.

### Prerequisites

A Runtime Platform connection and a named Deployment Action Plan are required. Also, the current Context must be either an Oracle Module, a Process Flow Module or a Location.

### Syntax Diagrams

parseDeployCommand



### Syntax

```
parseDeployCommand = OMBDEPLOY ( ( DEPLOYMENT_ACTION_PLAN
"QUOTED_STRING" [ AS ( SPECIFICATION | SCRIPT ) TO "QUOTED_STRING" ] ) |
( SPECIFICATION FROM "QUOTED_STRING" ) );
```

### Keywords and Parameters

parseDeployCommand  
Specify Deploy command.

DEPLOYMENT\_ACTION\_PLAN

Deploy a Deployment Action Plan.

QUOTED\_STRING

The Deployment Action Plan name. Or, the directory (if AS SCRIPT) or file name (if AS SPECIFICATION) into which the deployment is performed. Or, the source XML filename (if SPECIFICATION FROM).

AS

Perform the Deployment to a File System.



**SPECIFICATION**

Deploy as a Deployment Specification XML file.

**SCRIPT**

Deploy as a set of Oracle Script files.

**FROM**

Perform the Deployment from a file on a File System.

**Examples**

```
OMBDEPLOY DEPLOYMENT_ACTION_PLAN 'MY_DEPLOY_PLAN'
```

# OMBDESCRIBE\_CLASS\_DEFINITION

## Purpose

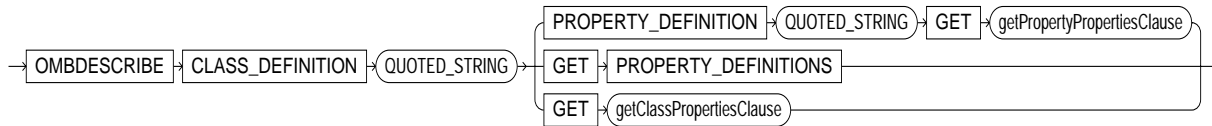
OMBDESCRIBE CLASS\_DEFINITION - To describe a class definition or its property definitions.

## Prerequisites

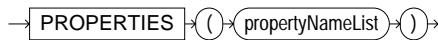
Class definition must be already exist. This command can be executed for any class definition regardless of current context.

## Syntax Diagrams

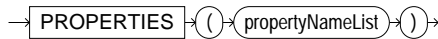
parseDescribeClassCommand



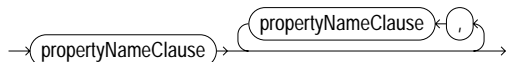
getPropertyPropertiesClause



getClassPropertiesClause



propertyNameList



propertyNameClause



## Syntax

```

parseDescribeClassCommand = OMBDESCRIBE CLASS_DEFINITION "QUOTED_STRING" ( ( PROPERTY_DEFINITION "QUOTED_STRING" GET "getPropertyPropertiesClause" ) | ( GET PROPERTY_DEFINITIONS ) | GET "getClassPropertiesClause" );

getPropertyPropertiesClause = PROPERTIES "(" "propertyNameList" ")";

getClassPropertiesClause = PROPERTIES "(" "propertyNameList" ")";

propertyNameList = "propertyNameClause" { "," "propertyNameClause" };

propertyNameClause = ( "UNQUOTED_STRING" );
    
```

## Keywords and Parameters

parseDescribeClassCommand

Describe a class definition.

QUOTED\_STRING

Name of the class definition.

PROPERTY\_DEFINITIONS

Get the list of property definitions for the class definition.

getPropertyPropertiesClause

Get properties for the property definition.

PROPERTIES

Get the properties.

getClassPropertiesClause

Get properties for the class definition.

Basic properties for PROPERTY\_DEFINITION:

Name: TYPE

Type: STRING(200)

Valid Values: N/A

Default: N/A

Name of the type of the property definition.

Name: DEFAULT\_VALUE

Type: STRING(4000)

Valid Values: Depends on the type of the property definition.

Default: N/A

Default value of the property definition.

Basic properties for CLASS\_DEFINITION:

Name: SCRIPT\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Scripting name of the class definition.

Name: STEREO TYPE

Type: STRING

Valid Values: interface/class

Default: class

Stereotype of the class definition.

Name: IS\_ABSTRACT

Type: BOOLEAN

Valid Values: true/false

Default: false

If class is abstract, return true.

Name: DESCRIPTION

Type: STRING

Valid Values: N/A

Default: N/A

Description of the class definition.

PROPERTIES

Get the properties on the class definition.

propertyNameList

List of the names of the properties.

propertyNameClause

Name of the property.

UNQUOTED\_STRING

Name of the property.

## Examples

OMBDESCRIBE CLASS\_DEFINITION 'TABLE' GET PROPERTY\_DEFINITIONS This will list all property definitions in this class definition. OMBDESCRIBE CLASS\_

DEFINITION 'TABLE' GET PROPERTIES (STEREOTYPE, IS\_ABSTRACT, DESCRIPTION) This will give the information about this class. OMBDESCRIBE CLASS\_DEFINITION 'TABLE' PROPERTY\_DEFINITION 'tbl\_udp' GET PROPERTIES (TYPE, DEFAULT\_VALUE, BUSINESS\_NAME) This will give the information about the property definition.

### See Also

OMBREDEFINE CLASS\_DEFINITION

## OMBDISC

### Purpose

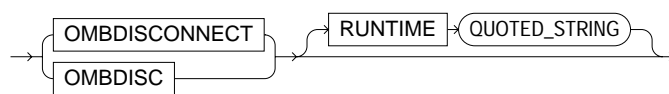
OMBDISC - To disconnect from a OWB repository or the named Runtime Platform.

### Prerequisites

Currently need to be connected to a OWB repository or the named Runtime Platform.

### Syntax Diagrams

parseDisconnectCommand



### Syntax

```
parseDisconnectCommand = ( ( OMBDISCONNECT | OMBDISC ) [ RUNTIME
"QUOTED_STRING" ] );
```

### Keywords and Parameters

parseDisconnectCommand

Specify disconnect command.

QUOTED\_STRING

Specify the run time repository name to disconnect.

### Examples

```
OMBDISCONNECT
```

### See Also

OMBCONNECT

---

## OMBDISCONNECT

### Purpose

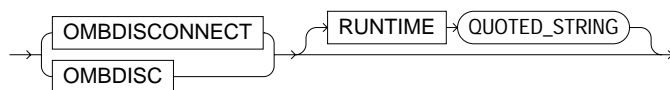
OMBDISCONNECT - To disconnect from a OWB repository or the named Runtime Platform.

### Prerequisites

Currently need to be connected to a OWB repository or the named Runtime Platform.

### Syntax Diagrams

parseDisconnectCommand



### Syntax

```
parseDisconnectCommand = ( ( OMBDISCONNECT | OMBDISC ) [ RUNTIME
"QUOTED_STRING" ] );
```

### Keywords and Parameters

parseDisconnectCommand  
Specify disconnect command.

QUOTED\_STRING  
Specify the run time repository name to disconnect.

### Examples

OMBDISCONNECT

### See Also

OMBCONNECT

## OMBENV

### Purpose

OMBENV - This command will list the values for all set OMBPlus environment variables. OMBPlus environment variables are regular Tcl variables, so they can be set using standard Tcl "set" command, and unset using standard "unset" command.

### Prerequisites

None.

### Syntax Diagrams

parseEnvironmentCommand

→ OMBENV →

### Syntax

parseEnvironmentCommand = OMBENV ;

### Keywords and Parameters

parseEnvironmentCommand

Specify environment command. The environment variables recognized by OMBPlus are: OMBPROMPT if set, will display the current context as the prompt OMBLOG if set with a file name, will log the output to that file OMBTIMER if set, will display the time taken by the executed command. Note that the elapsed time will be appended to the command result, therefore the user should not set this variable when the result of the command is expected to be processed OMBCONTINUE\_ON\_ERROR if set, will continue the execution of the script, even when some OMB commands fail. Note that if this variable is set, OMB commands no longer return Tcl errors, so any enclosing 'catch' command will not be effective.

### Examples

OMBENV



# OMBEXPORT

## Purpose

OMBEXPORT - Exports current metadata or snapshot metadata to a metadata loader file and metadata definitions to a metadata definition file.

## Prerequisites

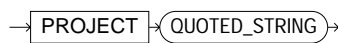
Connection must be established to repository to be exported from.

## Syntax Diagrams

ExportCommand



projectClause



componentsClause



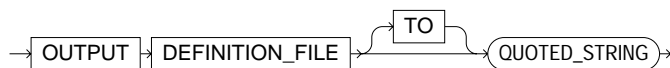
useFieldSeparatorClause



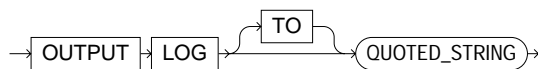
controlFileClause



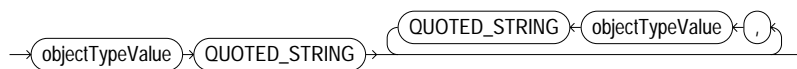
definitionFileClause



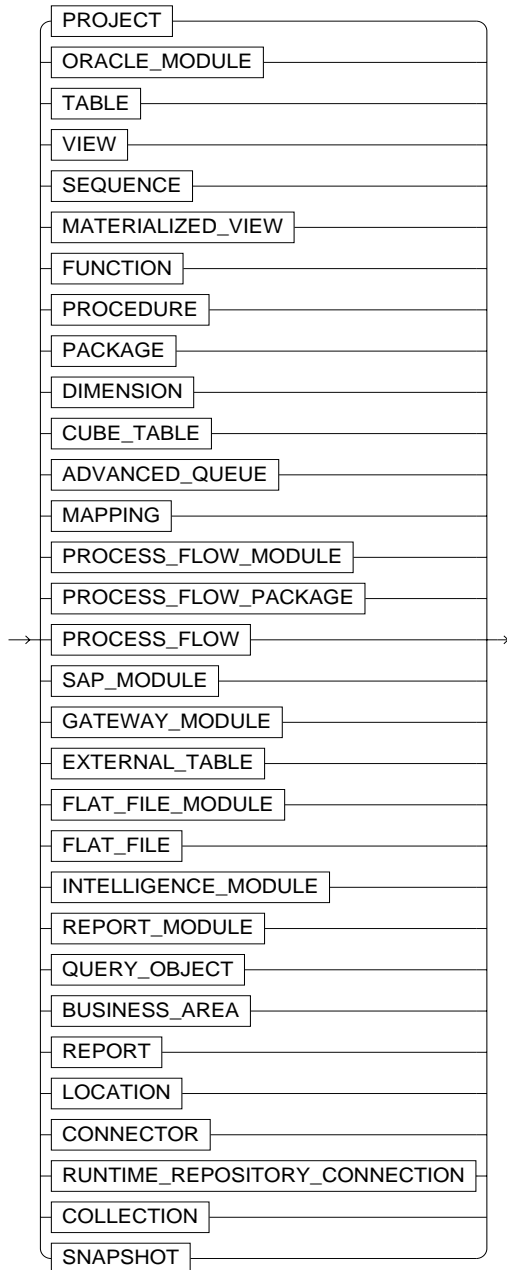
outputLogClause



componentsList



objectTypeValue



## Syntax

```
ExportCommand = OMBEXPORT ( ( [ TO ] MDL_FILE "QUOTED_STRING" [ FROM
] [ "projectClause" ] [ "componentsClause" ] [ "useFieldSeparatorClause" ] [
"controlFileClause" ] [ "definitionFileClause" ] "outputLogClause" ) );
projectClause = PROJECT "QUOTED_STRING";
componentsClause = COMPONENTS "(" "componentsList" ")";
useFieldSeparatorClause = [ USE ] FIELD_SEPARATOR "UNQUOTED_STRING";
controlFileClause = CONTROL_FILE "QUOTED_STRING";
```

```

definitionFileClause = OUTPUT DEFINITION_FILE [ TO ] "QUOTED_STRING";
outputLogClause = OUTPUT LOG [ TO ] "QUOTED_STRING";
componentsList = "objectTypeValue" "QUOTED_STRING" { "," "objectTypeValue"
"QUOTED_STRING" };
objectTypeValue = ( PROJECT | ORACLE_MODULE | TABLE | VIEW | SEQUENCE
| MATERIALIZED_VIEW | FUNCTION | PROCEDURE | PACKAGE |
DIMENSION | CUBE_TABLE | ADVANCED_QUEUE | MAPPING | PROCESS_
FLOW_MODULE | PROCESS_FLOW_PACKAGE | PROCESS_FLOW | SAP_
MODULE | GATEWAY_MODULE | EXTERNAL_TABLE | FLAT_FILE_MODULE |
FLAT_FILE | INTELLIGENCE_MODULE | REPORT_MODULE | QUERY_OBJECT |
BUSINESS_AREA | REPORT | LOCATION | CONNECTOR | RUNTIME_
REPOSITORY_CONNECTION | COLLECTION | SNAPSHOT );

```

## Keywords and Parameters

ExportCommand

ExportCmdParser\$ExportCommand??

QUOTED\_STRING

Enclose the name of the export metadata file in single quotes.

projectClause

Specify the project to be exported.

componentsClause

List components to be exported.

useFieldSeparatorClause

Specify character to be used as the field separator in the metadata file.

UNQUOTED\_STRING

Metadata file field separator. Use BAR or CARAT. The default is BAR.

controlFileClause

Specify a control file with export options not directly supported by OMBEXPORT command.

QUOTED\_STRING

Enclose the control file name in single quotes.

definitionFileClause

Export metadata definitions to a metadata definition (.mdd) file. If metadata definitions are not defined, a file is not created.

**QUOTED\_STRING**

Enclose the name of the metadata definition file to be exported in single quotes.

**outputLogClause**

Export log file for export messages and statistics.

**QUOTED\_STRING**

Enclose the log file name in single quotes.

**componentsList**

Comma separated list of components to be exported.

**QUOTED\_STRING**

Absolute or relative path name of an object (e.g. 'MODULE\_X/TABLE\_Y').

**objectTypeValue**

Object type to be exported (e.g. TABLE, VIEW etc.).

**Examples**

```
OMBEXPORT TO MDL_FILE 'd:/mdl/exp1.mdl' FROM PROJECT 'MY_PROJECT'  
OUTPUT LOG TO 'd:/mdl/exp1.log'
```

```
OMBEXPORT TO MDL_FILE 'd:/mdl/exp1.mdl' FROM PROJECT 'MY_PROJECT'  
USE FIELD_SEPARATOR CARAT OUTPUT LOG TO 'd:/mdl/exp1.log'
```

```
OMBEXPORT MDL_FILE 'd:/mdl/exp1.mdl' PROJECT 'MY_PROJECT'  
COMPONENTS (ORACLE_MODULE 'DW1', TABLE 'EMP', VIEW 'DEPT_VW')  
OUTPUT LOG 'd:/mdl/exp1.log'
```

**See Also**

OMBIMPORT

## OMBEXPORT\_MDL\_FILE

### Purpose

OMBEXPORT MDL\_FILE - Exports current metadata or snapshot metadata to a metadata loader file and metadata definitions to a metadata definition file.

### Prerequisites

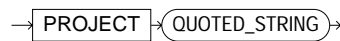
Connection must be established to repository to be exported from.

### Syntax Diagrams

ExportCommand



projectClause



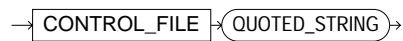
componentsClause



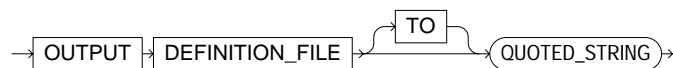
useFieldSeparatorClause



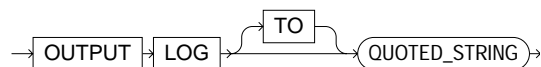
controlFileClause



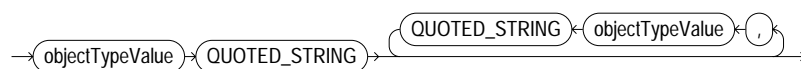
definitionFileClause



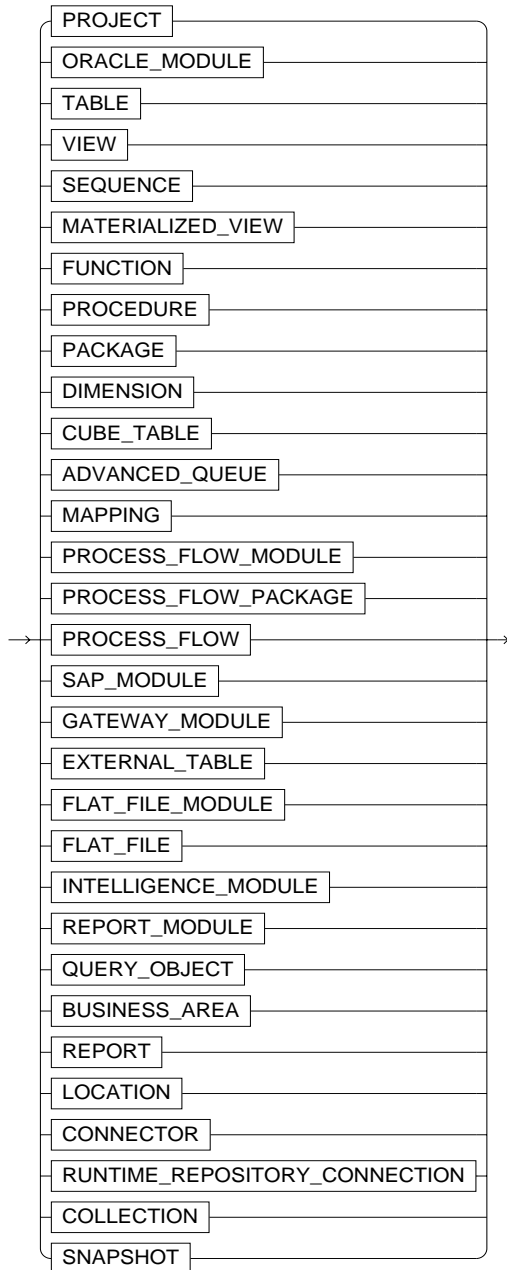
outputLogClause



componentsList



objectTypeValue



## Syntax

```
ExportCommand = OMBEXPORT ( ([ TO ] MDL_FILE "QUOTED_STRING" [ FROM
] [ "projectClause" ] [ "componentsClause" ] [ "useFieldSeparatorClause" ] [
"controlFileClause" ] [ "definitionFileClause" ] "outputLogClause" ) );
```

```
projectClause = PROJECT "QUOTED_STRING";
```

```
componentsClause = COMPONENTS "(" "componentsList" ")";
```

```
useFieldSeparatorClause = [ USE ] FIELD_SEPARATOR "UNQUOTED_STRING";
```

```
controlFileClause = CONTROL_FILE "QUOTED_STRING";
```

```
definitionFileClause = OUTPUT DEFINITION_FILE [ TO ] "QUOTED_STRING";
```

```

outputLogClause = OUTPUT LOG [ TO ] "QUOTED_STRING";
componentsList = "objectTypeValue" "QUOTED_STRING" { "," "objectTypeValue"
"QUOTED_STRING" };
objectTypeValue = ( PROJECT | ORACLE_MODULE | TABLE | VIEW | SEQUENCE
| MATERIALIZED_VIEW | FUNCTION | PROCEDURE | PACKAGE |
DIMENSION | CUBE_TABLE | ADVANCED_QUEUE | MAPPING | PROCESS_
FLOW_MODULE | PROCESS_FLOW_PACKAGE | PROCESS_FLOW | SAP_
MODULE | GATEWAY_MODULE | EXTERNAL_TABLE | FLAT_FILE_MODULE |
FLAT_FILE | INTELLIGENCE_MODULE | REPORT_MODULE | QUERY_OBJECT |
BUSINESS_AREA | REPORT | LOCATION | CONNECTOR | RUNTIME_
REPOSITORY_CONNECTION | COLLECTION | SNAPSHOT );

```

## Keywords and Parameters

ExportCommand

ExportCmdParser\$ExportCommand??

QUOTED\_STRING

Enclose the name of the export metadata file in single quotes.

projectClause

Specify the project to be exported.

componentsClause

List components to be exported.

useFieldSeparatorClause

Specify character to be used as the field separator in the metadata file.

UNQUOTED\_STRING

Metadata file field separator. Use BAR or CARAT. The default is BAR.

controlFileClause

Specify a control file with export options not directly supported by OMBEXPORT command.

QUOTED\_STRING

Enclose the control file name in single quotes.

definitionFileClause

Export metadata definitions to a metadata definition (.mdd) file. If metadata definitions are not defined, a file is not created.

**QUOTED\_STRING**

Enclose the name of the metadata definition file to be exported in single quotes.

**outputLogClause**

Export log file for export messages and statistics.

**QUOTED\_STRING**

Enclose the log file name in single quotes.

**componentsList**

Comma separated list of components to be exported.

**QUOTED\_STRING**

Absolute or relative path name of an object (e.g. 'MODULE\_X/TABLE\_Y').

**objectTypeValue**

Object type to be exported (e.g. TABLE, VIEW etc.).

**Examples**

```
OMBEXPORT TO MDL_FILE 'd:/mdl/exp1.mdl' FROM PROJECT 'MY_PROJECT'  
OUTPUT LOG TO 'd:/mdl/exp1.log'
```

```
OMBEXPORT TO MDL_FILE 'd:/mdl/exp1.mdl' FROM PROJECT 'MY_PROJECT'  
USE FIELD_SEPARATOR CARAT OUTPUT LOG TO 'd:/mdl/exp1.log'
```

```
OMBEXPORT MDL_FILE 'd:/mdl/exp1.mdl' PROJECT 'MY_PROJECT'  
COMPONENTS (ORACLE_MODULE 'DW1', TABLE 'EMP', VIEW 'DEPT_VW')  
OUTPUT LOG 'd:/mdl/exp1.log'
```

**See Also**

OMBIMPORT MDL\_FILE



---

## OMBHELP

### Purpose

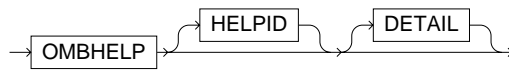
OMBHELP - Displays the manual page for OMB commands.

### Prerequisites

None.

### Syntax Diagrams

parseHelpCommand



### Syntax

```
parseHelpCommand = OMBHELP [ HELPID ] [ DETAIL ] ;
```

### Keywords and Parameters

parseHelpCommand

Specifies the command for which to invoke help.

HELPID

An unquoted string that represents either the command name and the optional additional parameter on the command.

DETAIL

Use this keyword to display the manual page in long format.

### Examples

The following statement shows the manual page for OMBCREATE TABLE in short format

```
OMBHELP OMBCREATE TABLE
```

To show the manual page in long format (including parameters and example sections), use the DETAIL keyword, like this:

```
OMBHELP OMBCREATE TABLE DETAIL
```

# OMBIMPORT

## Purpose

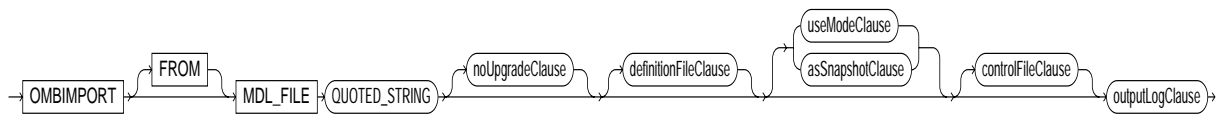
OMBIMPORT - Imports metadata from a Metadata Loader file.

## Prerequisites

Must be connected to the repository where the import is to be performed.

## Syntax Diagrams

ImportCommand



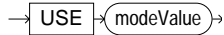
noUpgradeClause



definitionFileClause



useModeClause



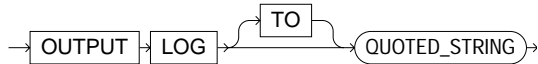
asSnapshotClause



controlFileClause



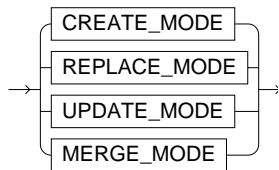
outputLogClause



noUpgradeValue



modeValue



## Syntax

```

ImportCommand = OMBIMPORT ( ( [ FROM ] MDL_FILE "QUOTED_STRING" [
"noUpgradeClause" ] [ "definitionFileClause" ] [ "useModeClause" |
"asSnapshotClause" ] [ "controlFileClause" ] "outputLogClause" ) );
noUpgradeClause = "noUpgradeValue";
definitionFileClause = DEFINITION_FILE "QUOTED_STRING";
useModeClause = USE "modeValue";
asSnapshotClause = AS SNAPSHOT "QUOTED_STRING";
controlFileClause = CONTROL_FILE "QUOTED_STRING";
outputLogClause = OUTPUT LOG [ TO ] "QUOTED_STRING";
noUpgradeValue = ( NO_UPGRADE );
modeValue = ( CREATE_MODE | REPLACE_MODE | UPDATE_MODE | MERGE_
MODE );

```

## Keywords and Parameters

### ImportCommand

Import metadata from a file.

### QUOTED\_STRING

Enclose the import metadata file name in single quotes.

### noUpgradeClause

Specify that MDL files that are not compatible with current repository version should not be automatically upgraded. The default is to automatically upgrade unless this clause is used.

### definitionFileClause

Import metadata definitions from a file.

### QUOTED\_STRING

Enclose the import metadata definition file name in single quotes.

### useModeClause

Specify import mode for importing metadata file.

### asSnapshotClause

Specify a new snapshot name into which the contents of the MDL file are to be imported. Use this option to import old archive files as snapshots.

**QUOTED\_STRING**

Enclose the new snapshot name in single quotes.

**controlFileClause**

Specify a control file with import options not directly supported by the OMBIMPORT command.

**QUOTED\_STRING**

Enclose the control file name in single quotes.

**outputLogClause**

Log file for import messages and statistics.

**QUOTED\_STRING**

Enclose the log file name in single quotes.

**modeValue**

Import mode. Use CREATE\_MODE, REPLACE\_MODE, UPDATE\_MODE, or MERGE\_MODE. The default is CREATE\_MODE.

**Examples**

```
OMBIMPORT FROM MDL_FILE 'd:/mdl/exp1.mdl' OUTPUT LOG TO  
'd:/mdl/exp1_imp.log'
```

```
OMBIMPORT MDL_FILE 'd:/mdl/exp1.mdl' USE UPDATE_MODE OUTPUT LOG  
TO 'd:/mdl/exp1_imp.log'
```

**See Also**

OMBEXPORT

# OMBIMPORT\_MDL\_FILE

## Purpose

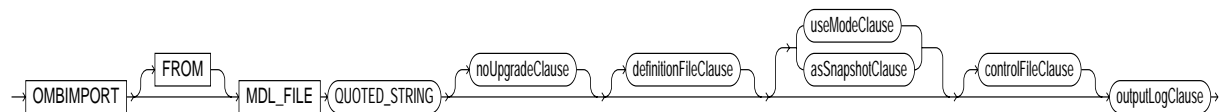
OMBIMPORT MDL\_FILE - Imports metadata from a Metadata Loader file.

## Prerequisites

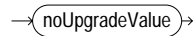
Must be connected to the repository where the import is to be performed.

## Syntax Diagrams

ImportCommand



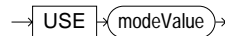
noUpgradeClause



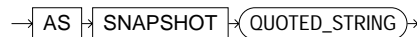
definitionFileClause



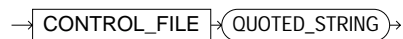
useModeClause



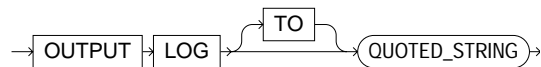
asSnapshotClause



controlFileClause



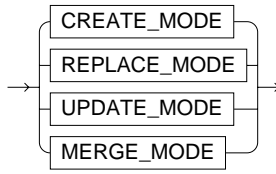
outputLogClause



noUpgradeValue



modeValue



## Syntax

```

ImportCommand = OMBIMPORT ( ( [ FROM ] MDL_FILE "QUOTED_STRING" [
"noUpgradeClause" ] [ "definitionFileClause" ] [ "useModeClause" |
"asSnapshotClause" ] [ "controlFileClause" ] "outputLogClause" ) );
noUpgradeClause = "noUpgradeValue";
definitionFileClause = DEFINITION_FILE "QUOTED_STRING";
useModeClause = USE "modeValue";
asSnapshotClause = AS SNAPSHOT "QUOTED_STRING";
controlFileClause = CONTROL_FILE "QUOTED_STRING";
outputLogClause = OUTPUT LOG [ TO ] "QUOTED_STRING";
noUpgradeValue = ( NO_UPGRADE );
modeValue = ( CREATE_MODE | REPLACE_MODE | UPDATE_MODE | MERGE_
MODE );
  
```

## Keywords and Parameters

**ImportCommand**

Import metadata from a file.

**QUOTED\_STRING**

Enclose the import metadata file name in single quotes.

**noUpgradeClause**

Specify that MDL files that are not compatible with current repository version should not be automatically upgraded. The default is to automatically upgrade unless this clause is used.

**definitionFileClause**

Import metadata definitions from a file.

**QUOTED\_STRING**

Enclose the import metadata definition file name in single quotes.

**useModeClause**

Specify import mode for importing metadata file.

**asSnapshotClause**

Specify a new snapshot name into which the contents of the MDL file are to be imported. Use this option to import old archive files as snapshots.

**QUOTED\_STRING**

Enclose the new snapshot name in single quotes.

**controlFileClause**

Specify a control file with import options not directly supported by the OMBIMPORT command.

**QUOTED\_STRING**

Enclose the control file name in single quotes.

**outputLogClause**

Log file for import messages and statistics.

**QUOTED\_STRING**

Enclose the log file name in single quotes.

**modeValue**

Import mode. Use CREATE\_MODE, REPLACE\_MODE, UPDATE\_MODE, or MERGE\_MODE. The default is CREATE\_MODE.

**Examples**

```
OMBIMPORT FROM MDL_FILE 'd:/mdl/exp1.mdl' OUTPUT LOG TO  
'd:/mdl/exp1_imp.log'
```

```
OMBIMPORT MDL_FILE 'd:/mdl/exp1.mdl' USE UPDATE_MODE OUTPUT LOG  
TO 'd:/mdl/exp1_imp.log'
```

**See Also**

OMBEXPORT MDL\_FILE

## OMBLIST

### Purpose

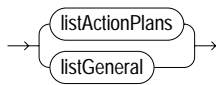
OMBLIST - This command lists the specified Warehouse Builder objects under the current folder. You can also provide a regular expression to list objects in a different folder and/or to list objects with names matching a pattern. Before displaying the objects, this command first refreshes the object list from the repository. This means that the list of objects returned also reflects any additions or deletions by other users.

### Prerequisites

Must be connected to a OWB repository.

### Syntax Diagrams

parseListCommand



listActionPlans



listGeneral



### Syntax

```

parseListCommand = "listActionPlans" | "listGeneral" ;
listActionPlans = OMBLIST DEPLOYMENT_ACTION_PLANS [ "QUOTED_STRING"
];
listGeneral = OMBLIST "UNQUOTED_STRING" [ "QUOTED_STRING" ];
  
```

### Keywords and Parameters

**parseListCommand**

Specify the list command.

**listActionPlans**

Deployment action plans are listed.

**listGeneral**

OWB objects of the specified type are listed.



## UNQUOTED\_STRING

The type (plural) of the objects to be listed. Valid object types are: PROJECTS, ORACLE\_MODULES, FLAT\_FILE\_MODULES, PROCESS\_FLOW\_MODULES, SAP\_MODULES, INTELLIGENCE\_MODULES, REPORT\_MODULES, TRANSFORMATION\_MODULES, PACKAGES, ADVANCED\_QUEUES, BUSINESS\_AREAS, COLLECTIONS, CONNECTORS, CUBE\_TABLES, DIMENSION\_TABLES, EXTERNAL\_TABLES, FLAT\_FILES, FUNCTIONS, IO\_FUNCTIONS, MAPPINGS, MATERIALIZED\_VIEWS, OBJECT\_TYPES, PROCEDURES, QUERY\_OBJECTS, REPORTS, REPORT\_GROUPS, SEQUENCES, TABLES, PROCESS\_FLOWS, PROCESS\_FLOW\_PACKAGES, LOCATIONS, RUNTIME\_REPOSITORY\_CONNECTIONS, VIEWS.

## QUOTED\_STRING

Optionally, specify a regular expression, which is used to filter the results. The syntax of the regular expressions follow the syntax from TCL. Here are some of the quantifiers: \* -- represents a sequence of 0 or more matches of the atom. + -- represents a sequence of 1 or more matches of the atom. ? -- represents a sequence of 0 or 1 matches of the atom. {m} -- represents a sequence of exactly m matches of the atom. Here are some of the atoms: . -- matches any single character {} -- matches an empty string, noted for possible reporting. k -- (where k is a non-alphanumeric character) matches that character taken as an ordinary character, e.g. \\ matches a backslash character. c -- where c is alphanumeric (possibly followed by other characters), an escape. (re) -- (where re is any regular expression) matches a match for re, with the match noted for possible reporting. [char] -- a bracket expression, matching any one of the chars. Further documentation on the syntax can be found at the TCL manual page under the command 'regexp'.

## Examples

### OMBLIST TABLES

will list the tables under the current Oracle module folder context (if the current context is not an Oracle module, an error will be produced). OMBLIST ORACLE\_MODULES '/MY\_PROJECT/O.\*' will list Oracle modules starting with letter 'O', within project 'MY\_PROJECT'. OMBLIST ORACLE\_MODULES '/MY\_PROJECT/[ABC].\*' will list Oracle modules starting with either letter A or B or C, within project 'MY\_PROJECT'. </BODY> </HTML>

# OMBLIST\_SNAPSHOT

## Purpose

OMBLIST SNAPSHOT - This command lists all the snapshots existing or all snapshots for a specific component.

## Prerequisites

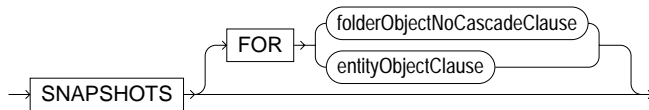
Snapshots can be listed from any context.

## Syntax Diagrams

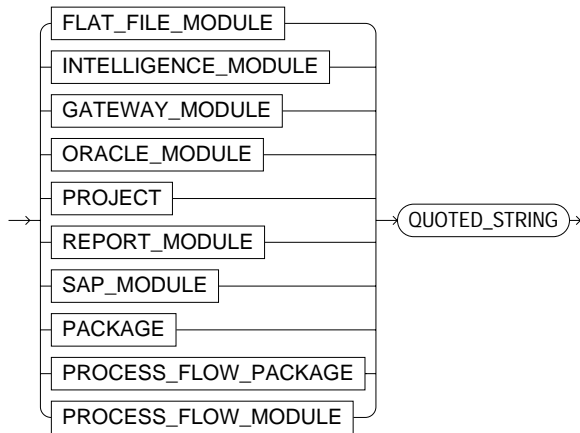
parseListCommand



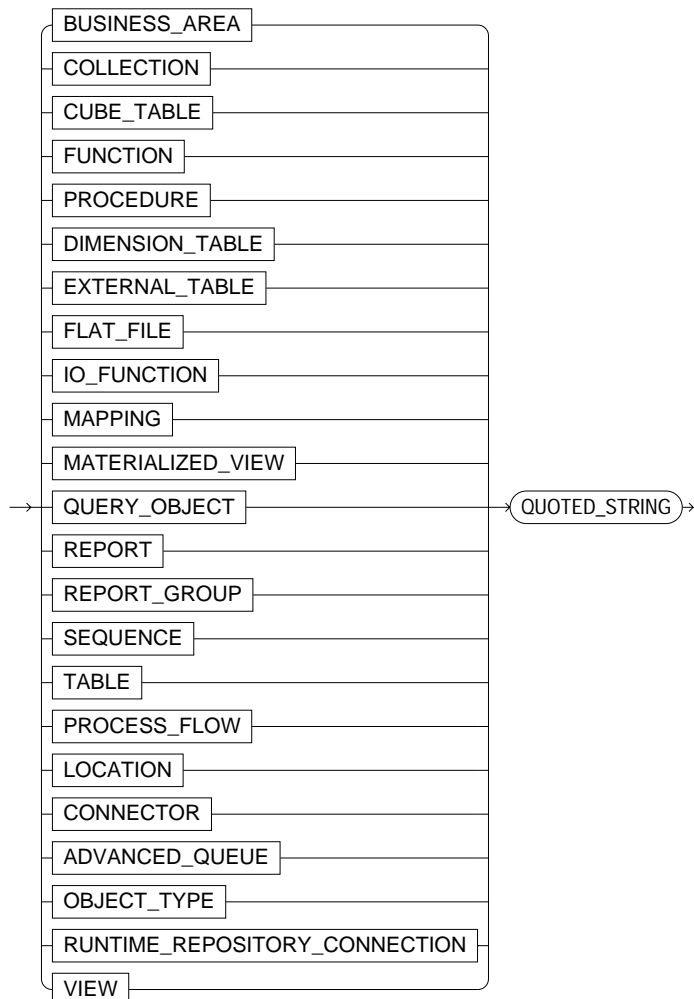
listSnapshotCommand



folderObjectNoCascadeClause



entityObjectClause



## Syntax

```
parseListCommand = OMBLIST "listSnapshotCommand";
```

```
listSnapshotCommand = SNAPSHOTS [ FOR ( "folderObjectNoCascadeClause" |
"entityObjectClause" ) ] ;
```

```
folderObjectNoCascadeClause = ( FLAT_FILE_MODULE | INTELLIGENCE_
MODULE | GATEWAY_MODULE | ORACLE_MODULE | PROJECT | REPORT_
MODULE | SAP_MODULE | PACKAGE | PROCESS_FLOW_PACKAGE |
PROCESS_FLOW_MODULE ) "QUOTED_STRING";
```

```
entityObjectClause = ( ( BUSINESS_AREA | COLLECTION | CUBE_TABLE |
FUNCTION | PROCEDURE | DIMENSION_TABLE | EXTERNAL_TABLE | FLAT_
FILE | IO_FUNCTION | MAPPING | MATERIALIZED_VIEW | QUERY_OBJECT |
REPORT | REPORT_GROUP | SEQUENCE | TABLE | PROCESS_FLOW |
LOCATION | CONNECTOR | ADVANCED_QUEUE | OBJECT_TYPE | RUNTIME_
REPOSITORY_CONNECTION | VIEW ) "QUOTED_STRING" );
```

## Keywords and Parameters

parseListCommand

Root production of OMBLIST SNAPSHOT.

listSnapshotCommand

To list existing snapshots.

SNAPSHOTS

Lists all the snapshots existing in the repository.

FOR

Lists all the snapshots existing for a specific component or folder.

folderObjectNoCascadeClause

Folder clause whose snapshots the user wants to list.

entityObjectClause

Component clause whose snapshots the user wants to list.

## Examples

OMBLIST SNAPSHOTS This command lists all the snapshots ever taken in the lifetime of the repository. OMBLIST SNAPSHOTS FOR TABLE '/Project/WH/T1' This command lists all the snapshots for table T1. It can be seen as a version tree of table T1.

## See Also

OMBCREATE SNAPSHOT, OMBALTER SNAPSHOT, OMBDROP SNAPSHOT, OMBRESTORE SNAPSHOT, OMBCOMPARE SNAPSHOT, OMBRETRIEVE SNAPSHOT

---

## OMBLOCK

### Purpose

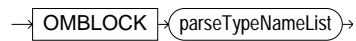
OMBLOCK - Lock one or more objects. If a list of objects is specified, the objects will be locked one by one in the given order. The command will behave atomically, i.e. if it fails to lock any of the objects in the list, then it will lock none of them.

### Prerequisites

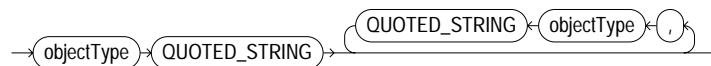
No other user should have a lock on any of the objects.

### Syntax Diagrams

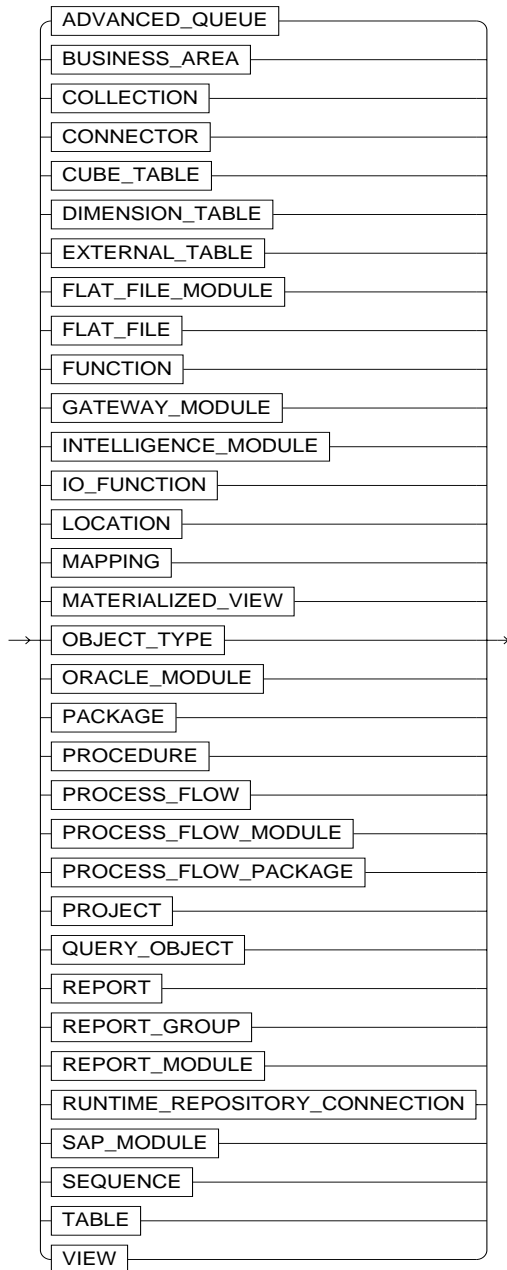
parseLockCommand



parseTypeNameList



objectType



### Syntax

```

parseLockCommand = OMBLOCK "parseTypeNameList" ;
parseTypeNameList = "objectType" "QUOTED_STRING" { "," "objectType" "QUOTED_
STRING" };

objectType = ( ADVANCED_QUEUE | BUSINESS_AREA | COLLECTION |
CONNECTOR | CUBE_TABLE | DIMENSION_TABLE | EXTERNAL_TABLE |
FLAT_FILE_MODULE | FLAT_FILE | FUNCTION | GATEWAY_MODULE |
INTELLIGENCE_MODULE | IO_FUNCTION | LOCATION | MAPPING |
MATERIALIZED_VIEW | OBJECT_TYPE | ORACLE_MODULE | PACKAGE |

```

---

PROCEDURE | PROCESS\_FLOW | PROCESS\_FLOW\_MODULE | PROCESS\_FLOW\_PACKAGE | PROJECT | QUERY\_OBJECT | REPORT | REPORT\_GROUP | REPORT\_MODULE | RUNTIME\_REPOSITORY\_CONNECTION | SAP\_MODULE | SEQUENCE | TABLE | VIEW );

### Keywords and Parameters

parseLockCommand

Specify lock command.

parseTypeNameList

Specify the object or the list of objects to be locked.

QUOTED\_STRING

Name of the object to be locked. Can be specified as an absolute path or as a path relative to the current context. However, there is the restriction that all objects to be locked must be in the current project.

objectType

Type of the object to be locked.

### Examples

```
OMBLOCK TABLE 'T1', VIEW '/MY_PROJECT/ORACLE_1/V1'
```

will lock table 'T1' in the current module, and view 'V1' in Oracle module 'ORACLE\_1' from project 'MY\_PROJECT'.

### See Also

OMBUNLOCK

## OMBMOVE

### Purpose

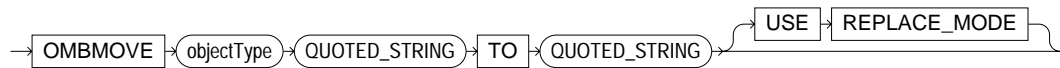
OMBMOVE - Move one or more objects of the same object type. The replace option allows you to overwrite.

### Prerequisites

Use of relative path specifications requires awareness of the current context.

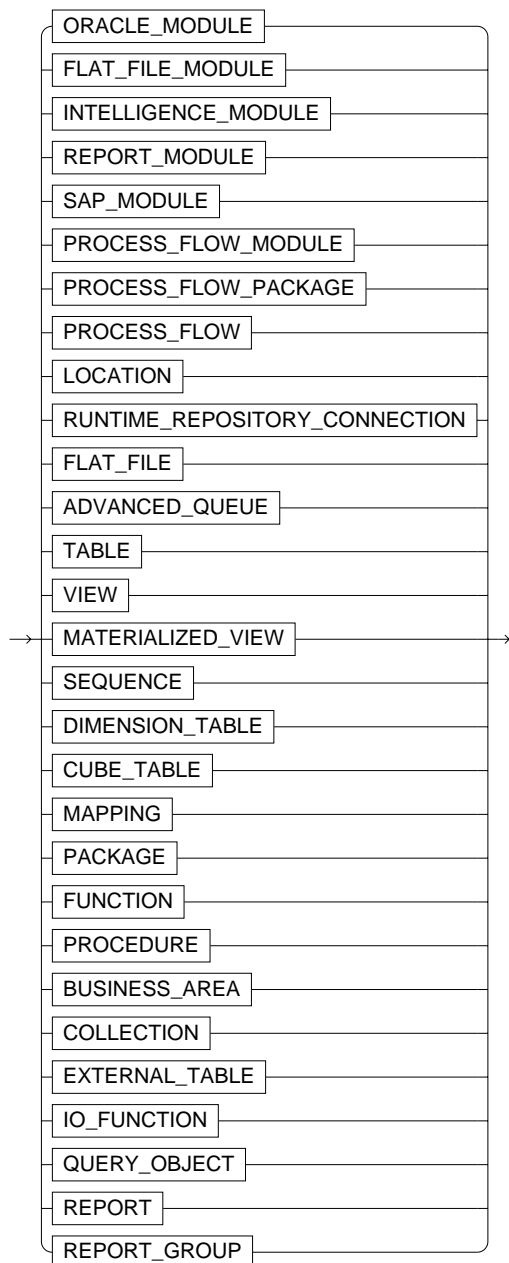
### Syntax Diagrams

parseMoveCommand





objectType



## Syntax

```
parseMoveCommand = OMBMOVE "objectType" "QUOTED_STRING" TO
"QUOTED_STRING" [ USE REPLACE_MODE ];
```

```
objectType = ( ORACLE_MODULE | FLAT_FILE_MODULE | INTELLIGENCE_
MODULE | REPORT_MODULE | SAP_MODULE | PROCESS_FLOW_MODULE |
PROCESS_FLOW_PACKAGE | PROCESS_FLOW | LOCATION | RUNTIME_
REPOSITORY_CONNECTION | FLAT_FILE | ADVANCED_QUEUE | TABLE |
VIEW | MATERIALIZED_VIEW | SEQUENCE | DIMENSION_TABLE | CUBE_
TABLE | MAPPING | PACKAGE | FUNCTION | PROCEDURE | BUSINESS_AREA
```

| COLLECTION | EXTERNAL\_TABLE | IO\_FUNCTION | QUERY\_OBJECT |  
REPORT | REPORT\_GROUP );

## Keywords and Parameters

parseMoveCommand

Specifies the source object type, source path, and target path for the object to move.

QUOTED\_STRING

Source and target path specifications can be absolute or relative. To move multiple objects, include a regular expression as the final step of the source path. If you are moving multiple objects, the final step of the target path must be the folder to which the objects are being copied. If you are only moving one object, you can specify the object's original name or a new name as the final step of the target path.

REPLACE\_MODE

Use this option to overwrite existing target objects.

objectType

Moving objects is subject to the following restrictions: 1. You can only move objects within the current project; you cannot move objects between projects. 2. You cannot move an object within the same parent folder; if your purpose is to rename the object, use the 'OMBALTER...RENAME TO...' command.

## Examples

```
OMBMOVE TABLE 'MY_TABLE1' TO '../WH2/MY_TABLE1' USE REPLACE_MODE
```

```
OMBMOVE VIEW 'MY_VIEW1' TO '../WH2/MY_VIEW1'
```

```
OMBMOVE TABLE 'MY_.*' TO '/MY_PROJECT1/WH2'
```

## See Also

OMBCOPY

# OMBRECONCILE

## Purpose

OMBRECONCILE - Reconcile the target metadata definition with the source metadata definition.

## Prerequisites

1. The current context of scripting must be a project atleast.
2. The target and source objects should exist in the current project.
3. No concurrent user is operating on the target.

## Syntax Diagrams

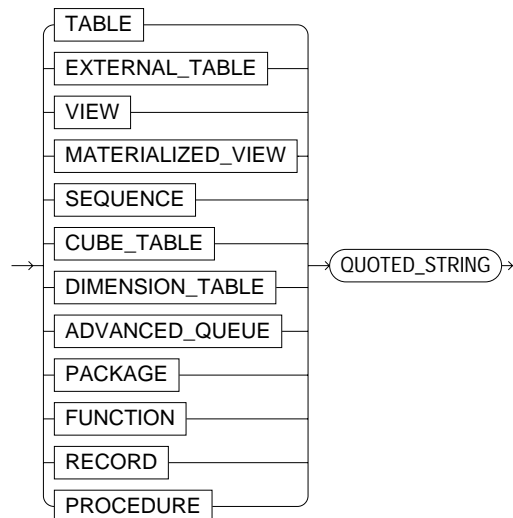
ReconcileCommand



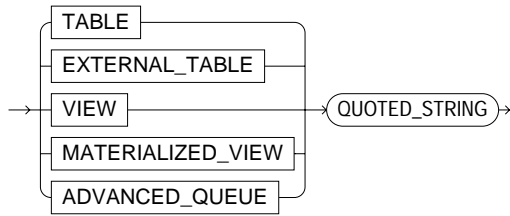
parseFCOSCO



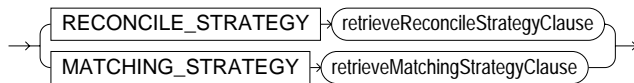
parseSourceFCO



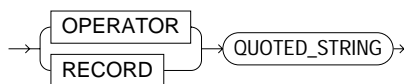
parseTargetFCO



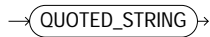
setStrategyClause



retrieveSCOClaue



retrieveReconcileStrategyClause



retrieveMatchingStrategyClause



## Syntax

```
ReconcileCommand = OMBRECONCILE ( "parseFCOSCO" | "parseSourceFCO" ) TO
( "parseFCOSCO" | "parseTargetFCO" ) USE ( "setStrategyClause" | "setStrategyClause" );
```

```
parseFCOSCO = ( MAPPING | FLAT_FILE ) "QUOTED_STRING"
"retrieveSCOClaue";
```

```
parseSourceFCO = ( TABLE | EXTERNAL_TABLE | VIEW | MATERIALIZED_VIEW
| SEQUENCE | CUBE_TABLE | DIMENSION_TABLE | ADVANCED_QUEUE |
PACKAGE | FUNCTION | RECORD | PROCEDURE ) "QUOTED_STRING";
```

```
parseTargetFCO = ( TABLE | EXTERNAL_TABLE | VIEW | MATERIALIZED_VIEW
| ADVANCED_QUEUE ) "QUOTED_STRING";
```

```
setStrategyClause = RECONCILE_STRATEGY "retrieveReconcileStrategyClause" |
MATCHING_STRATEGY "retrieveMatchingStrategyClause";
```

```
retrieveSCOClaue = ( OPERATOR | RECORD ) "QUOTED_STRING";
```

```
retrieveReconcileStrategyClause = "QUOTED_STRING";
```

```
retrieveMatchingStrategyClause = "QUOTED_STRING";
```

## Keywords and Parameters

ReconcileCommand

Reconciles the target metadata definition with the source metadata definition.

**parseFCOSCO**

The source object in the source component or the target object in the target component for reconcile.

**parseSourceFCO**

The source component for reconcile.

**parseTargetFCO**

The target component for reconcile.

**setStrategyClause**

The strategy to be used for reconcile. Must be one of RECONCILE\_STRATEGY OR MATCHING\_STRATEGY

**RECONCILE\_STRATEGY**

MERGE or REPLACE. MERGE : Updates the matching objects in the target with the metadata definition in the source object and creates new objects in the target for source objects that do not match. REPLACE : Updates the matching objects in the target with the metadata definition in the source object, creates a new object in the target for source objects that do not match, and deletes objects in the target that have no match in the source.

**MATCHING\_STRATEGY**

Indicates the matching strategy to be used between the object types in source and target. Currently, the list of available matching strategies for a source and target combination are listed below. For reconciliation between RECORD and EXTERNAL\_TABLE, the valid matching strategies are MATCH\_BY\_OBJECT\_ID, MATCH\_BY\_OBJECT\_NAME, MATCH\_BY\_OBJECT\_POSITION. For reconciliation involving OPERATOR of MAP, the available matching strategies are MATCH\_BY\_OBJECT\_ID, MATCH\_BY\_OBJECT\_NAME, MATCH\_BY\_OBJECT\_POSITION, MATCH\_BY\_OBJECT\_ID\_AND\_NAME, MATCH\_BY\_OBJECT\_ID\_AND\_POSITION, MATCH\_BY\_OBJECT\_NAME\_AND\_POSITION, MATCH\_BY\_OBJECT\_ID\_AND\_NAME\_AND\_POSITION. Note that the source map might be modified during outbound reconcile of maps, which would require locking both the source and the target before reconcile invocation.

**retrieveSCOClause**

Retrieves the source or the target object in the component specified.

**retrieveReconcileStrategyClause**

Must of one of 'MERGE' or 'REPLACE'

**retrieveMatchingStrategyClause**

String representing the matching paradigm between the source and the target object. The list varies for every source and target object type.

## Examples

```
OMBRECONCILE EXTERNAL_TABLE 'et1' TO EXTERNAL_TABLE 'et2' USE  
(RECONCILE_STRATEGY 'MERGE', MATCHING_STRATEGY 'MATCH_BY_  
OBJECT_NAME')
```

```
OMBRECONCILE MAPPING 'm1' OPERATOR 'o1' TO TABLE 't1' USE (RECONCILE_  
STRATEGY 'REPLACE', MATCHING_STRATEGY 'MATCH_BY_OBJECT_NAME')
```

```
OMBRECONCILE MAPPING 'm1' OPERATOR 'o1' TO MAPPING 'm1' OPERATOR  
'o2' USE (RECONCILE_STRATEGY 'REPLACE', MATCHING_STRATEGY 'MATCH_  
BY_OBJECT_ID') </BODY> </HTML>
```

# OMBREDEFINE\_CLASS\_DEFINITION

## Purpose

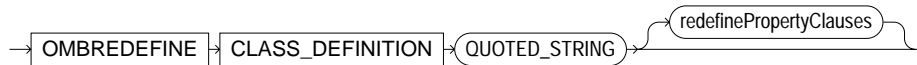
OMBREDEFINE CLASS\_DEFINITION - To redefine a class.

## Prerequisites

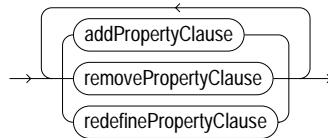
Class definition to be redefined should already exist. This command can be executed for any class definition regardless of current context. User must be the repository owner to run this command, and user has to connect in single user mode.

## Syntax Diagrams

parseRedefineClassCommand



redefinePropertyClauses



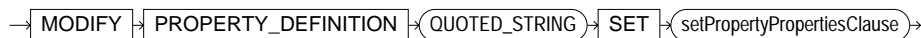
addPropertyClause



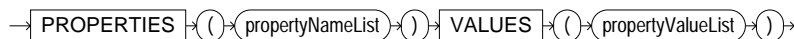
removePropertyClause



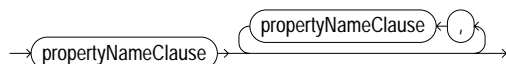
redefinePropertyClause



setPropertyPropertiesClause



propertyNameList



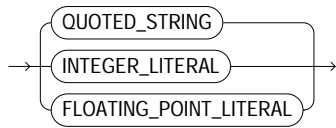
propertyValueList



propertyNameClause



propertyValue



## Syntax

```
parseRedefineClassCommand = OMBREDEFINE CLASS_DEFINITION "QUOTED_
STRING" [ "redefinePropertyClauses" ];

redefinePropertyClauses = ( "addPropertyClause" | "removePropertyClause" |
"redefinePropertyClause" );

addPropertyClause = ( ADD PROPERTY_DEFINITION "QUOTED_STRING" SET
"setPropertyPropertiesClause" );

removePropertyClause = ( DELETE PROPERTY_DEFINITION "QUOTED_STRING" );

redefinePropertyClause = ( MODIFY PROPERTY_DEFINITION "QUOTED_STRING"
SET "setPropertyPropertiesClause" );

setPropertyPropertiesClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("
"propertyValueList" ")";

propertyNameList = "propertyNameClause" { "," "propertyNameClause" };

propertyValueList = "propertyValue" { "," "propertyValue" };

propertyNameClause = ( "UNQUOTED_STRING" );

propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );
```

## Keywords and Parameters

`parseRedefineClassCommand`

Redefine a class definition.

`CLASS_DEFINITION`

Redefine a class definition.

`QUOTED_STRING`

Name of the class definition.

`redefinePropertyClauses`

Add, delete or modify user defined property definitions on class definition. The name of the property definition must always start with prefix 'UDP\_'.



addPropertyClause

Add property definition to class definition.

QUOTED\_STRING

Name of the property definition to be added.

removePropertyClause

Remove property definition from class definition.

QUOTED\_STRING

Name of the property definition to be deleted.

redefinePropertyClause

Modify property definition on class definition.

QUOTED\_STRING

Name of the property definition to be modified.

setPropertyPropertiesClause

Set the properties for property definition. Valid properties are TYPE, DEFAULT\_VALUE, BUSINESS\_NAME.

propertyNameList

The list of properties for the class definition.

propertyValueList

The list of values provided for the class definition.

propertyNameClause

The name of the property.

UNQUOTED\_STRING

The name of the property for the class definition.

propertyValue

The value of the property.

**QUOTED\_STRING**

The value in string format of the property for the class definition.

**INTEGER\_LITERAL**

The integer value of the property for the class definition.

**FLOATING\_POINT\_LITERAL**

The float value of the property for the class definition.

**Examples**

```
OMBREDEFINE CLASS_DEFINITION 'TABLE' ADD PROPERTY_DEFINITION
'UDP_TBL_1' SET PROPERTIES (TYPE, DEFAULT_VALUE) VALUES ('INTEGER',
'100') This will add an User-defined property definition to class definition 'TABLE'.
OMBREDEFINE CLASS_DEFINITION 'TABLE' DELETE PROPERTY_DEFINITION
'UDP_TBL_1' This will delete property definition 'UDP_TBL_1' from class definition
'TABLE'. Property definition must exist before deleting it. OMBREDEFINE CLASS_
DEFINITION 'TABLE' MODIFY PROPERTY_DEFINITION 'UDP_TBL_1' SET
PROPERTIES (DEFAULT_VALUE, BUSINESS_NAME) VALUES ('99', 'UDP_TBL_2')
This will change the name of property definition to 'UDP_TBL_2' and default value to
99. Property definition must exist before modifying it. TYPE can not be changed for
property definition.
```

**See Also**

OMBDESCRIBE CLASS\_DEFINITION

## OMBREGISTER\_LOCATION

### Purpose

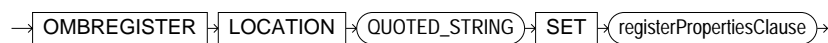
OMBREGISTER LOCATION - Register a location with a runtime repository.

### Prerequisites

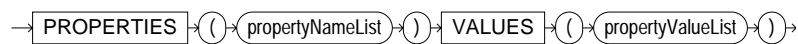
Must be in the context of a project and connected to a runtime repository.

### Syntax Diagrams

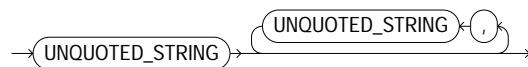
registerLocationCommand



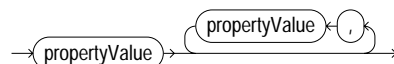
registerPropertiesClause



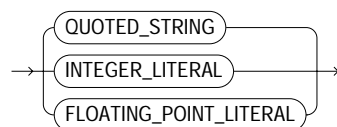
propertyNameList



propertyValueList



propertyValue



### Syntax

```
registerLocationCommand = OMBREGISTER LOCATION "QUOTED_STRING" SET
"registerPropertiesClause";
```

```
registerPropertiesClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("
"propertyValueList" ")";
```

```
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
```

```
propertyValueList = "propertyValue" { "," "propertyValue" };
```

```
propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );
```

## Keywords and Parameters

registerLocationCommand

Register a location.

QUOTED\_STRING

The name of the location to register.

registerPropertiesClause

Set the values you wish to register the location with.

propertyNameList

The names of the properties whose values you want to set.

Properties for registering an ORACLE DATABASE LOCATION, ORACLE WORKFLOW LOCATION, or SAP LOCATION:

Name: HOST

Type: STRING

Valid Values: N/A

Default: N/A

The machine name.

Name: PORT

Type: NUMBER

Valid Values: 0 - 65535

Default: N/A

The port number of a database listener.

Name: SERVICE

Type: STRING

Valid Values: N/A

Default: N/A

The database service name.

Name: SCHEMA

Type: STRING

Valid Values: N/A

Default: N/A

The database schema name.

Name: PASSWORD

Type: STRING

Valid Values: N/A

Default: N/A

The password.

Name: REUSE

Type: STRING

Valid Values: 'true', 'yes'

Default: N/A

Set this property to either 'true' or 'yes' to signify to the runtime platform that you want to register the current location name using the current UOID, even if the location was previously registered with a different UOID. If you do not specify REUSE, this action will result in an error to prevent data inconsistencies in the Runtime Repository.

Properties for registering a GATEWAY LOCATION:

Name: HOST

Type: STRING

Valid Values: N/A

Default: N/A

The machine name.

Name: PORT

Type: NUMBER

Valid Values: 0 - 65535

Default: N/A

The port number of a database listener.

Name: SERVICE

Type: STRING

Valid Values: N/A

Default: N/A

The database service name.

Name: SCHEMA

Type: STRING

Valid Values: N/A

Default: N/A

The database schema name.

Name: CONNECTAS

Type: STRING

Valid Values: N/A

Default: N/A

The user name.

Name: PASSWORD

Type: STRING

Valid Values: N/A

Default: N/A

The password for the CONNECTAS user.

Name: REUSE

Type: STRING

Valid Values: 'true', 'yes'

Default: N/A

Set this property to either 'true' or 'yes' to signify to the runtime platform that you want to register the current location name using the current UOID, even if the location was previously registered with a different UOID. If you do not specify REUSE, this action will result in an error to prevent data inconsistencies in the Runtime Repository.

Properties for registering a FILE LOCATION:

Name: HOST

Type: STRING

Valid Values: N/A

Default: N/A

The machine name.

Name: ROOTPATH

Type: STRING

Valid Values: N/A

Default: N/A

The file system directory.

Name: USER

Type: STRING

Valid Values: N/A

Default: N/A

The user name.

Name: PASSWORD

Type: STRING

Valid Values: N/A

Default: N/A

The password.

Name: REUSE

Type: STRING

Valid Values: 'true', 'yes'

Default: N/A

Set this property to either 'true' or 'yes' to signify to the runtime platform that you want to register the current location name using the current UOID, even if the location was previously registered with a different UOID. If you do not specify REUSE, this action will result in an error to prevent data inconsistencies in the Runtime Repository.

Properties for registering an OEM LOCATION:

Name: DOMAIN

Type: STRING

Valid Values: N/A

Default: N/A

The address of a machine running the Oracle Management Service.

Name: AGENT

Type: STRING

Valid Values: N/A

Default: N/A

The name of an Oracle Enterprise Manager (OEM) node running an OEM Agent. This name must be entered exactly as shown under the nodes in the Oracle Management Service.

Name: USER

Type: STRING

Valid Values: N/A

Default: N/A

The user name.

Name: PASSWORD

Type: STRING

Valid Values: N/A

Default: N/A

The password.

Name: REUSE

Type: STRING

Valid Values: 'true', 'yes'

Default: N/A

Set this property to either 'true' or 'yes' to signify to the runtime platform that you want to register the current location name using the current UOID, even if the location was previously registered with a different UOID. Without specifying REUSE, this action will result in an error to prevent data inconsistencies in the Runtime Repository.

propertyValueList

The values for the named properties.

propertyValue

A property value.

## Examples

```
OMBREGISTER LOCATION 'MY_ORACLE_LOCATION' SET PROPERTIES (HOST,  
PORT, SERVICE, SCHEMA, PASSWORD) VALUES ('localhost', 1521, 'orcl', 'scott',  
'tiger')
```

```
OMBREGISTER LOCATION 'MY_FILE_LOCATION' SET PROPERTIES (HOST,  
ROOTPATH, USER, PASSWORD) VALUES ('localhost', 'C:\temp\', 'usr1', 'password')
```

```
OMBREGISTER LOCATION 'MY_GATEWAY_LOCATION' SET PROPERTIES (HOST,  
PORT, SERVICE, SCHEMA, CONNECTAS, PASSWORD) VALUES ('localhost', 1521,  
'orcl', 'scott', 'scott', 'tiger')
```



```
OMBREGISTER LOCATION 'MY_SAP_LOCATION' SET PROPERTIES (HOST, PORT,  
SERVICE,  
SCHEMA, PASSWORD) VALUES ('localhost', 1521, 'sap', 'scott', 'tiger')
```

```
OMBREGISTER LOCATION 'MY_OEM_LOCATION' SET PROPERTIES (DOMAIN,  
AGENT, USER,  
PASSWORD) VALUES ('domain', 'agent', 'scott', 'tiger')
```

```
OMBREGISTER LOCATION 'MY_WORKFLOW_LOCATION' SET PROPERTIES  
(HOST, PORT, SERVICE, SCHEMA, PASSWORD) VALUES ('localhost', 1521, 'orcl',  
'scott', 'tiger') </BODY> </HTML>
```

# OMBRESTORE\_SNAPSHOT

## Purpose

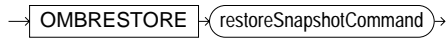
OMBRESTORE SNAPSHOT - A snapshot is a history point of individual or group of components. The user can recover the components' previously captured states by using the snapshot restore functionality.

## Prerequisites

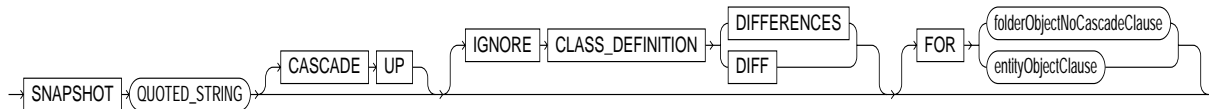
A component can be restored from a snapshot into the current repository.

## Syntax Diagrams

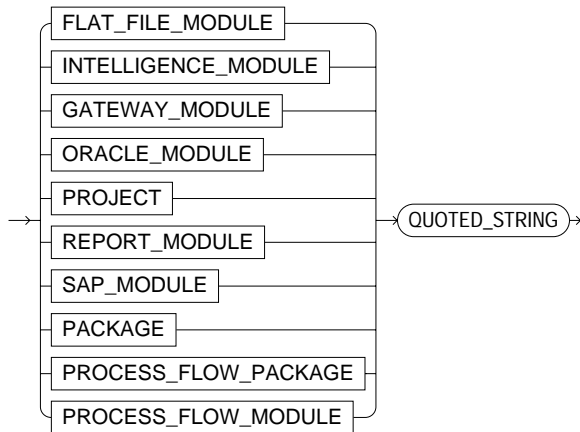
parseRestoreCommand



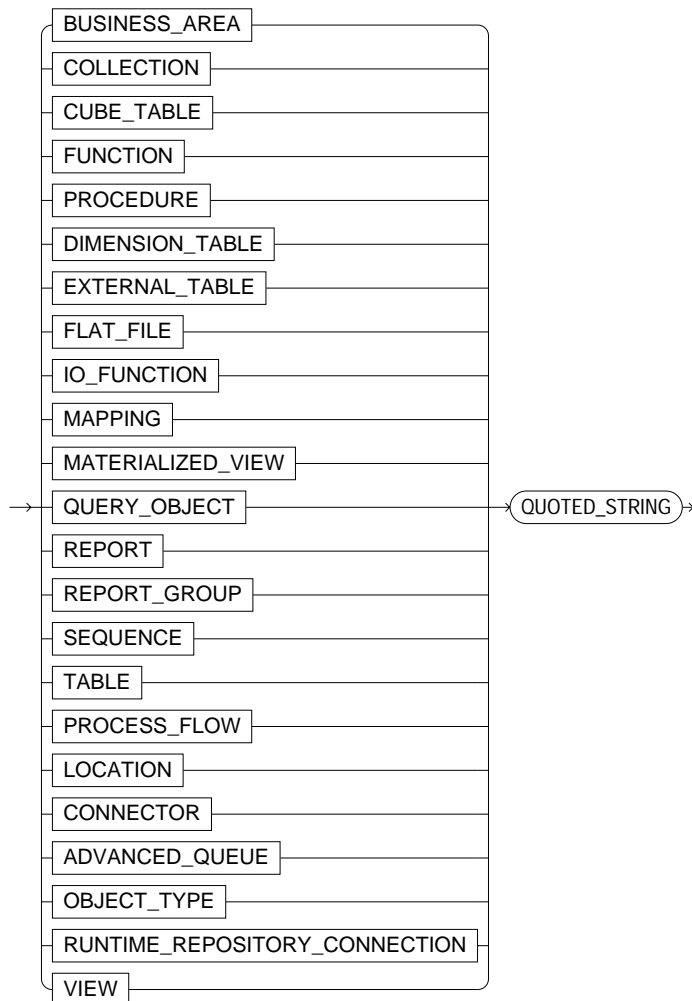
restoreSnapshotCommand



folderObjectNoCascadeClause



entityObjectClause



## Syntax

```
parseRestoreCommand = OMBRESTORE "restoreSnapshotCommand";
```

```
restoreSnapshotCommand = ( SNAPSHOT "QUOTED_STRING" [ CASCADE UP ] [
  IGNORE CLASS_DEFINITION ( DIFFERENCES | DIFF ) ] [ FOR (
    "folderObjectNoCascadeClause" | "entityObjectClause" ) ] );
```

```
folderObjectNoCascadeClause = ( FLAT_FILE_MODULE | INTELLIGENCE_
  MODULE | GATEWAY_MODULE | ORACLE_MODULE | PROJECT | REPORT_
  MODULE | SAP_MODULE | PACKAGE | PROCESS_FLOW_PACKAGE |
  PROCESS_FLOW_MODULE ) "QUOTED_STRING";
```

```
entityObjectClause = ( ( BUSINESS_AREA | COLLECTION | CUBE_TABLE |
  FUNCTION | PROCEDURE | DIMENSION_TABLE | EXTERNAL_TABLE | FLAT_
  FILE | IO_FUNCTION | MAPPING | MATERIALIZED_VIEW | QUERY_OBJECT |
  REPORT | REPORT_GROUP | SEQUENCE | TABLE | PROCESS_FLOW |
  LOCATION | CONNECTOR | ADVANCED_QUEUE | OBJECT_TYPE | RUNTIME_
  REPOSITORY_CONNECTION | VIEW ) "QUOTED_STRING" );
```

## Keywords and Parameters

parseRestoreCommand

Root production for OMBRESTORE SNAPSHOT.

restoreSnapshotCommand

To restore snapshot components into the repository.

QUOTED\_STRING

Name of the snapshot from which components are to be restored.

CASCADE

CASCADE UP - Optional clause for letting the user restore a component even if its parent does not exist in the current repository.

CLASS\_DEFINITION

IGNORE CLASS\_DEFINITION DIFF - Optional clause for letting the user restore a snapshot whether or not the meta-model of the snapshot is different from that of the current repository.

FOR

Optional component clause for partial restore. This can be used to specify which components of a snapshot are to be restored.

folderObjectNoCascadeClause

Folder component with no cascade clause, It refers to the folder object itself, its properties and none of its contained components. This will be the absolute path of the folder within the snapshot.

entityObjectClause

Component clause. This will be the absolute path of the component within the snapshot.

## Examples

OMBRESTORE SNAPSHOT 'S1' This command restores all components from the snapshot into the repository. If the corresponding components are not found in the repository, then they appear as newly recovered components from history.  
OMBRESTORE SNAPSHOT 'S1' FOR TABLE '/Project/WH1/T1' This commands replaces the current definition of the component in the repository with the snapshot component.  
OMBRESTORE SNAPSHOT 'S1' CASCADE UP This command restores all objects of the snapshot into the repository whether or not their parents exist in the current repository. If not, components along with their corresponding parents are

restored. OMBRESTORE SNAPSHOT 'S1' IGNORE CLASS\_DEFINITION DIFF This command restores the snapshot regardless of whether or not the meta-model of the snapshot is different to that of the current repository.

### See Also

OMBCREATE SNAPSHOT, OMBALTER SNAPSHOT, OMBDROP SNAPSHOT,  
OMBCOMPARE SNAPSHOT, OMBLIST SNAPSHOT, OMBRETRIEVE SNAPSHOT

## OMBROLLBACK

### Purpose

OMBROLLBACK - Perform rollback action on the repository.

### Prerequisites

Must be connected to a OWB repository.

### Syntax Diagrams

parseRollbackCommand

→ OMBROLLBACK →

### Syntax

```
parseRollbackCommand = OMBROLLBACK ;
```

### Keywords and Parameters

parseRollbackCommand  
Specify rollback command.

### Examples

```
OMBROLLBACK
```

### See Also

OMBCOMMIT

---

## OMBUNLOCK

### Purpose

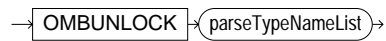
OMBUNLOCK - Unlock one or more objects, previously locked by OMBLOCK command. Note that if the object(s) have been modified, a commit is also required in order for the lock(s) to be released.

### Prerequisites

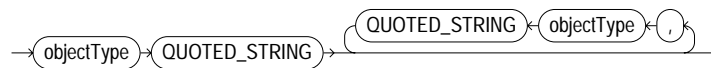
The object(s) should have been locked previously using OMBLOCK command.

### Syntax Diagrams

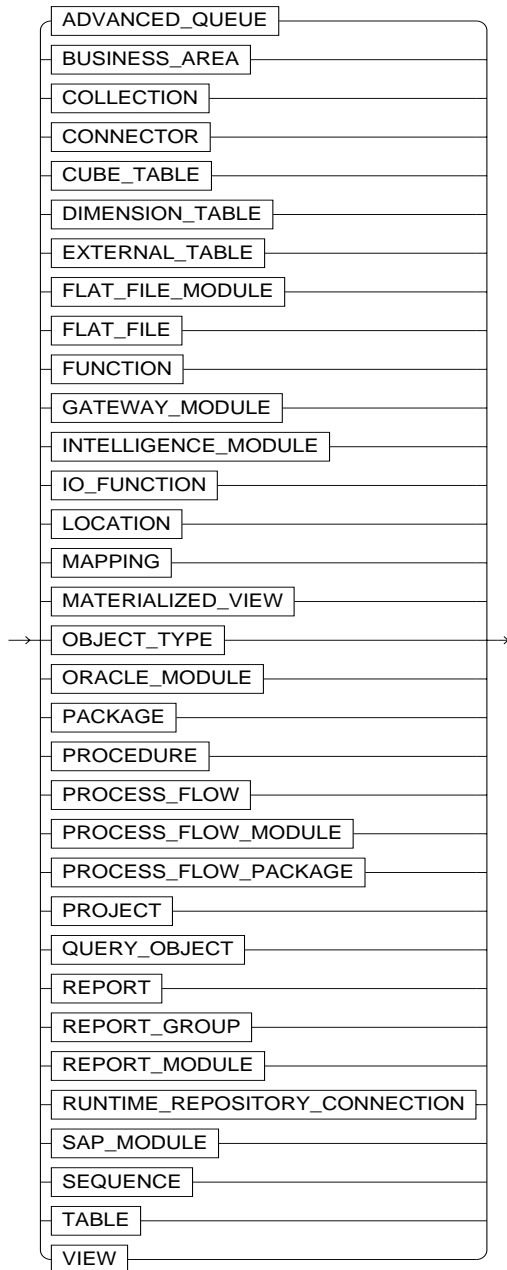
parseUnLockCommand



parseTypeNameList



objectType



### Syntax

```

parseUnLockCommand = OMBUNLOCK "parseTypeNameList" ;
parseTypeNameList = "objectType" "QUOTED_STRING" { "," "objectType" "QUOTED_
STRING" };

objectType = ( ADVANCED_QUEUE | BUSINESS_AREA | COLLECTION |
CONNECTOR | CUBE_TABLE | DIMENSION_TABLE | EXTERNAL_TABLE |
FLAT_FILE_MODULE | FLAT_FILE | FUNCTION | GATEWAY_MODULE |
INTELLIGENCE_MODULE | IO_FUNCTION | LOCATION | MAPPING |
MATERIALIZED_VIEW | OBJECT_TYPE | ORACLE_MODULE | PACKAGE |

```



PROCEDURE | PROCESS\_FLOW | PROCESS\_FLOW\_MODULE | PROCESS\_FLOW\_PACKAGE | PROJECT | QUERY\_OBJECT | REPORT | REPORT\_GROUP | REPORT\_MODULE | RUNTIME\_REPOSITORY\_CONNECTION | SAP\_MODULE | SEQUENCE | TABLE | VIEW );

### Keywords and Parameters

parseUnLockCommand

Specify unlock command.

parseTypeNameList

Specify the object or the list of objects to be unlocked.

QUOTED\_STRING

Name of the object to be unlocked. Can be specified as an absolute path or as a path relative to the current context. However, there is the restriction that all objects to be unlocked must be in the current project.

objectType

Type of the object to be unlocked.

### Examples

```
OMBUNLOCK TABLE 'T1', VIEW '/MY_PROJECT/ORACLE_1/V1'
```

will unlock table 'T1' in the current module, and view 'V1' in Oracle module 'ORACLE\_1' from project 'MY\_PROJECT'.

### See Also

OMBLOCK

# OMBVALIDATE

## Purpose

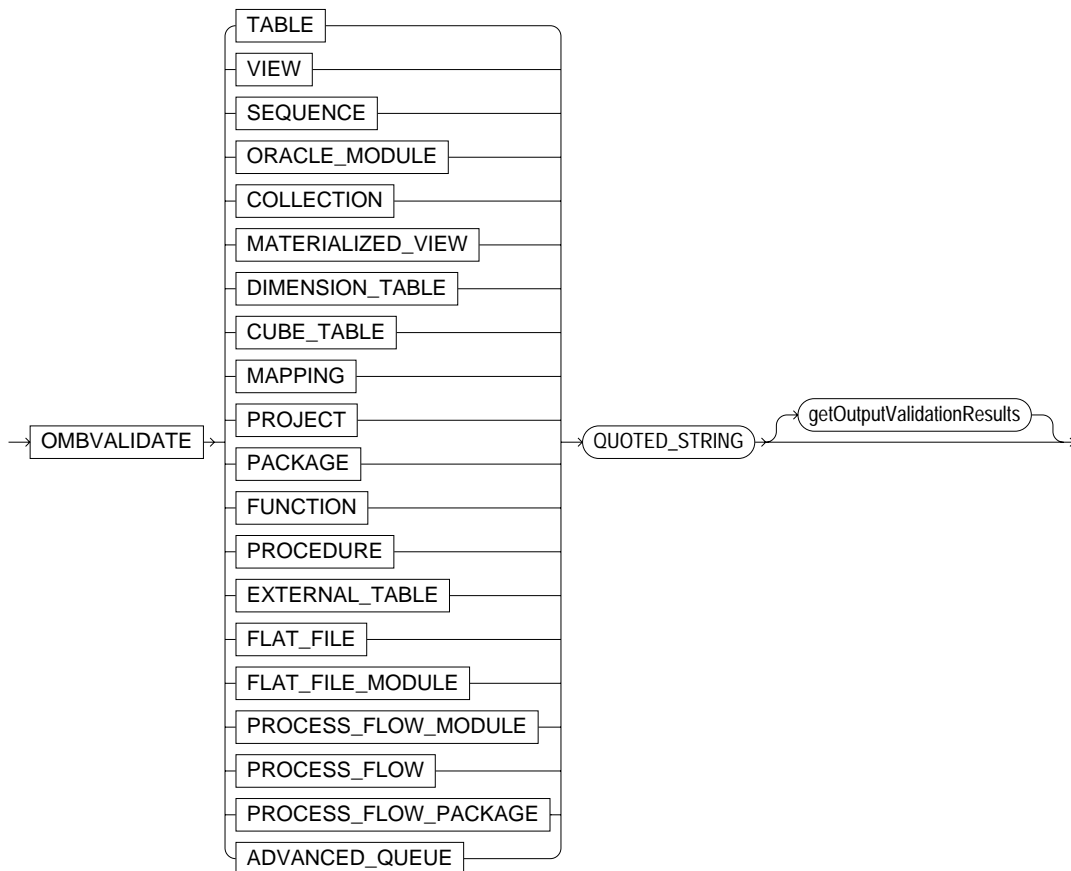
OMBVALIDATE - This command validates an repository object. The results are generated in a file in a user defined directory.

## Prerequisites

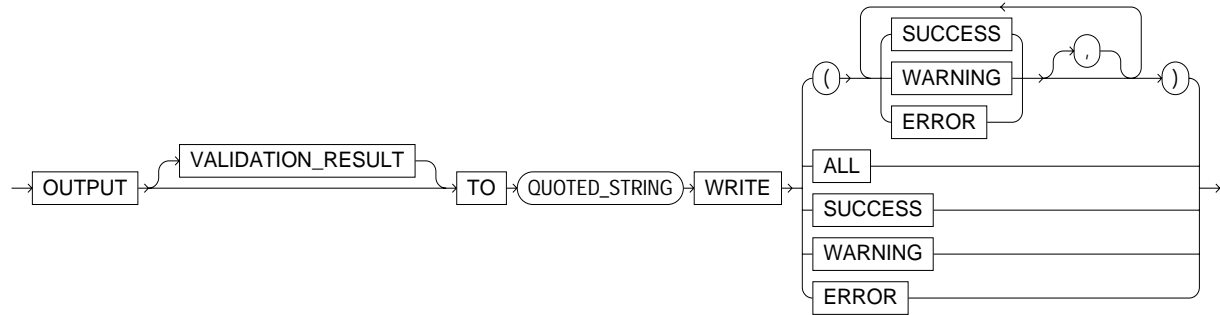
In the context of a Oracle Module except when validating Project and Oracle Module. To validate a Project the user needs to be in the Root context. To validate Oracle Module the user needs to be in Project context.

## Syntax Diagrams

parseValidateCommand



getOutputValidationResults



## Syntax

```

parseValidateCommand = OMBVALIDATE (( TABLE | VIEW | SEQUENCE |
ORACLE_MODULE | COLLECTION | MATERIALIZED_VIEW | DIMENSION_
TABLE | CUBE_TABLE | MAPPING | PROJECT | PACKAGE | FUNCTION |
PROCEDURE | EXTERNAL_TABLE | FLAT_FILE | FLAT_FILE_MODULE |
PROCESS_FLOW_MODULE | PROCESS_FLOW | PROCESS_FLOW_PACKAGE |
ADVANCED_QUEUE ) "QUOTED_STRING" [ "getOutputValidationResults" ] );
  
```

```

getOutputValidationResults = OUTPUT [ VALIDATION_RESULT ] TO "QUOTED_
STRING" WRITE (( "(" (( SUCCESS | WARNING | ERROR ) [ "," ] ) + " ) ) | ALL |
SUCCESS | WARNING | ERROR );
  
```

## Keywords and Parameters

parseValidateCommand

This command validates a repository object.

QUOTED\_STRING

The name of the object.

getOutputValidationResults

This clause outputs the validation results to one or more files in the specified folder.

QUOTED\_STRING

A directory where validation results are stored.

## Examples

```

OMBVALIDATE TABLE 'T1' OUTPUT VALIDATION_RESULT TO '/tmp' WRITE
(SUCCESS, ERROR)
  
```

## See Also

OMBCOMPILE



---



---

# OMBALTER

This chapter contains the following topics:

---

<a href="#">OMBALTER</a> on page 3-2	<a href="#">OMBALTER FUNCTION</a> on page 3-63	<a href="#">OMBALTER PROCESS_FLOW_MODULE</a> on page 3-207
<a href="#">OMBALTER ADVANCED_QUEUE</a> on page 3-8	<a href="#">OMBALTER LOCATION</a> on page 3-69	<a href="#">OMBALTER PROCESS_FLOW_PACKAGE</a> on page 3-210
<a href="#">OMBALTER COLLECTION</a> on page 3-12	<a href="#">OMBALTER MAPPING</a> on page 3-72	<a href="#">OMBALTER PROJECT</a> on page 3-213
<a href="#">OMBALTER CONNECTOR</a> on page 3-17	<a href="#">OMBALTER MATERIALIZED_VIEW</a> on page 3-151	<a href="#">OMBALTER RUNTIME_REPOSITORY_CONNECTIONS</a> on page 3-216
<a href="#">OMBALTER CUBE_TABLE</a> on page 3-21	<a href="#">OMBALTER OBJECT_TYPE</a> on page 3-171	<a href="#">OMBALTER SEQUENCE</a> on page 3-220
<a href="#">OMBALTER DEPLOYMENT_ACTION_PLAN</a> on page 3-34	<a href="#">OMBALTER ORACLE_MODULE</a> on page 3-175	<a href="#">OMBALTER SNAPSHOT</a> on page 3-224
<a href="#">OMBALTER EXTERNAL_TABLE</a> on page 3-40	<a href="#">OMBALTER PACKAGE</a> on page 3-185	<a href="#">OMBALTER TABLE</a> on page 3-229
<a href="#">OMBALTER FLAT_FILE</a> on page 3-49	<a href="#">OMBALTER PROCEDURE</a> on page 3-188	<a href="#">OMBALTER VIEW</a> on page 3-207
<a href="#">OMBALTER FLAT_FILE_MODULE</a> on page 3-59	<a href="#">OMBALTER PROCESS_FLOW</a> on page 3-194	

---

# OMBALTER

## Purpose

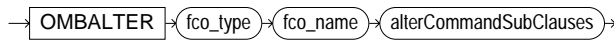
OMBALTER - Alter metadata for a component.

## Prerequisites

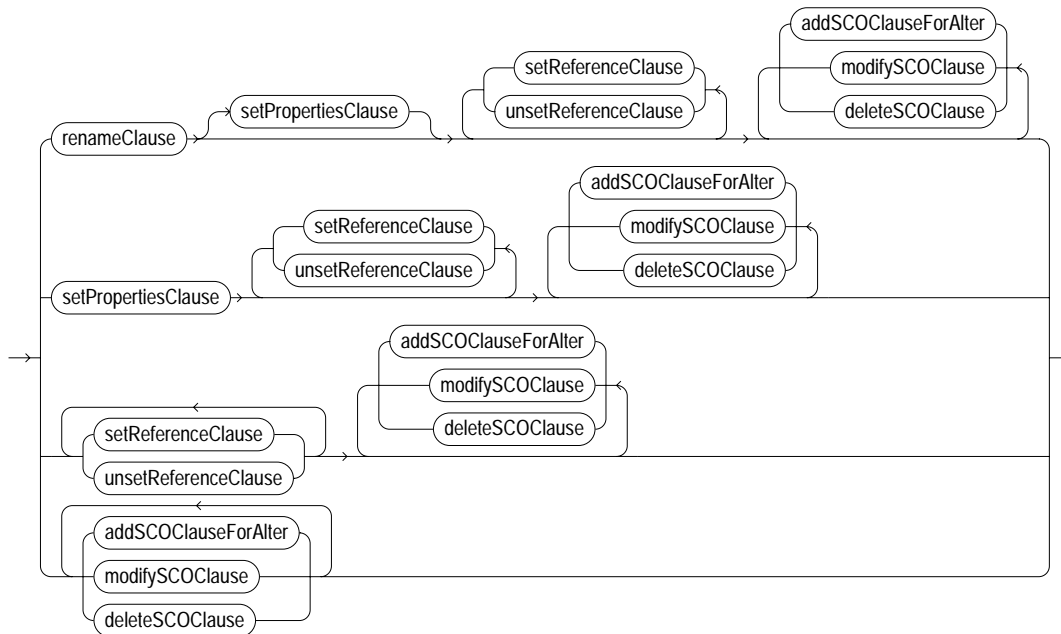
Should be in the parent context of the component to alter.

## Syntax Diagrams

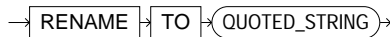
alterCommand



alterCommandSubClauses



renameClause



setPropertiesClause



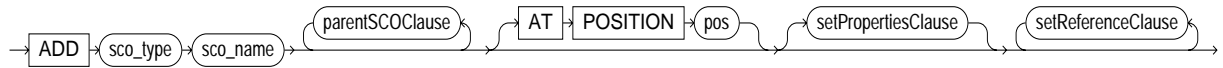
setReferenceClause



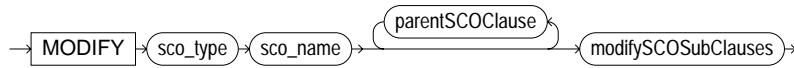
unsetReferenceClause



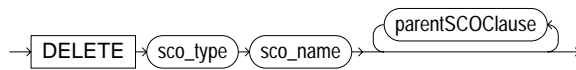
addSCOClauseForAlter



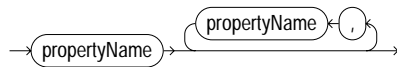
modifySCOClause



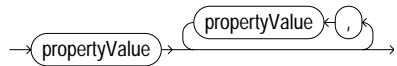
deleteSCOClause



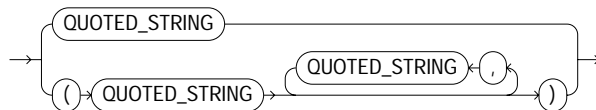
propertyNameList



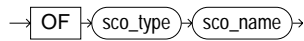
propertyValueList



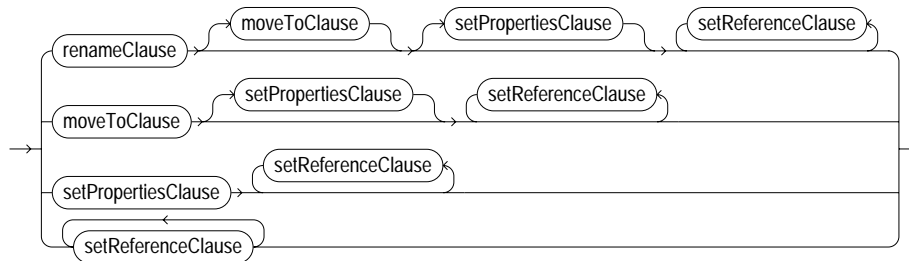
quotedNameList



parentSCOClause



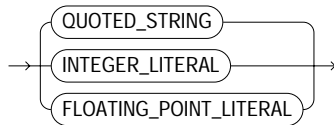
modifySCOSubClauses



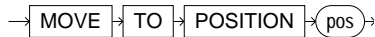
propertyName



propertyValue



moveToClause



## Syntax

```

alterCommand = OMBALTER "fco_type" "fco_name" "alterCommandSubClauses";
alterCommandSubClauses = ( "renameClause" [ "setPropertyClause" ] {
  "setReferenceClause" | "unsetReferenceClause" } { "addSCOClauseForAlter" |
  "modifySCOClause" | "deleteSCOClause" } ) | ( "setPropertyClause" {
  "setReferenceClause" | "unsetReferenceClause" } { "addSCOClauseForAlter" |
  "modifySCOClause" | "deleteSCOClause" } ) | ( ( "setReferenceClause" |
  "unsetReferenceClause" )+ { "addSCOClauseForAlter" | "modifySCOClause" |
  "deleteSCOClause" } ) | ( "addSCOClauseForAlter" | "modifySCOClause" |
  "deleteSCOClause" )+;
renameClause = RENAME TO "QUOTED_STRING";
setPropertyClause = SET PROPERTIES "(" "propertyNameList" ")" VALUES "("
  "propertyValueList" ")";
setReferenceClause = SET ( REF | REFERENCE ) [ "qualifier" ] "type"
  "quotedNameList" [ { "parentSCOClause" } OF "fco_type" "fco_name" ];
unsetReferenceClause = UNSET ( REF | REFERENCE ) [ "qualifier" ] "type";
addSCOClauseForAlter = ADD "sco_type" "sco_name" { "parentSCOClause" } [ AT
  POSITION "pos" ] [ "setPropertyClause" ] { "setReferenceClause" };
modifySCOClause = MODIFY "sco_type" "sco_name" { "parentSCOClause" }
  "modifySCOSubClauses";
deleteSCOClause = DELETE "sco_type" "sco_name" { "parentSCOClause" };
propertyNameList = "propertyName" { "," "propertyName" };
propertyValueList = "propertyValue" { "," "propertyValue" };
quotedNameList = "QUOTED_STRING" | "(" "QUOTED_STRING" { "," "QUOTED_
  STRING" } ")";
parentSCOClause = OF "sco_type" "sco_name";
modifySCOSubClauses = ( "renameClause" [ "moveToClause" ] [
  "setPropertyClause" ] { "setReferenceClause" } ) | ( "moveToClause" [
  "setPropertyClause" ] { "setReferenceClause" } ) | ( "setPropertyClause" {
  "setReferenceClause" } ) | "setReferenceClause"+;
propertyName = "UNQUOTED_STRING";
propertyValue = "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
  POINT_LITERAL";
moveToClause = MOVE TO POSITION "pos";

```



## Keywords and Parameters

`alterCommand`

Specify the component to alter.

`fco_type`

The type of the component.

`fco_name`

The physical name of the component in single quotes.

`alterCommandSubClauses`

Use this clause to rename the component, reset its properties, or modify the child objects of the component.

`renameClause`

Rename the component.

`setPropertyClause`

Set object properties.

`setReferenceClause`

Specify reference objects.

`qualifier`

Specify which reference to set, if there are more than one pointing to the same type.

`unsetReferenceClause`

Removes an existing reference.

`qualifier`

Specify which reference to set, if there are more than one pointing to the same type.

`addSCOClauseForAlter`

Add child objects under the component.

`modifySCOClause`

Change definition of a child object.

deleteSCOClause

Delete a child object.

propertyNameList

A list of property names.

propertyValueList

A list of property values.

quotedNameList

A list of single-quoted physical names.

parentSCOClause

Used to specify the path from a child object to the component

modifySCOSubClauses

Use this clause to rename a child object, reset its properties or references, or reorder it.

propertyName

An unquoted string representing the name of a property.

propertyValue

The value of a property.

moveToClause

Used to reorder child objects.

## Examples

This is an example for altering a table:

```
OMBALTER TABLE 'T1' SET PROPERTIES (DESCRIPTION) VALUES ('My First Table')
```

The following statement alters the column of a view:

```
OMBALTER VIEW 'V1'
```

MODIFY COLUMN 'COL1'  
SET PROPERTIES (DATATYPE) VALUES ('VARCHAR2')

**See Also**

OMBCREATE, OMBDROP

# OMBALTER ADVANCED\_QUEUE

## Purpose

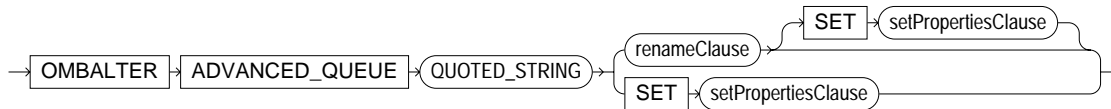
OMBALTER ADVANCED\_QUEUE - Alter the Advanced Queue by resetting its properties.

## Prerequisites

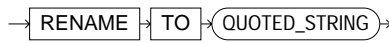
Should be in the context of an Oracle Module. The Object Type set as Payload Type should exist in the same Oracle Module.

## Syntax Diagrams

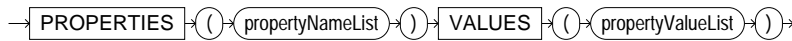
alterAQCommand



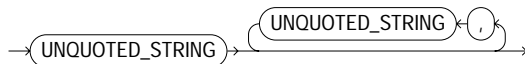
renameClause



setPropertyClause



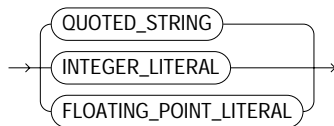
propertyNameList



propertyValueList



propertyValue



## Syntax

```
alterAQCommand = OMBALTER ( ADVANCED_QUEUE "QUOTED_STRING" (
    "renameClause" [ SET "setPropertyClause" ] | SET "setPropertyClause" ) );
renameClause = RENAME TO "QUOTED_STRING";
```

```

setPropertyClause = PROPERTIES "(" propertyNameList ")" VALUES "("
"propertyValueList" ")";
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
propertyValueList = "propertyValue" { "," "propertyValue" };
propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );

```

## Keywords and Parameters

**alterAQCommand**

Alters the Advanced Queue with the given name by either renaming it or by setting its properties or both.

**renameClause**

Renames the Advanced Queue to the given name.

**setPropertyClause**

Sets properties (core, logical, physical, user-defined) for Advanced Queue.

Valid properties are shown below:

Basic properties for **ADVANCED\_QUEUE**:

Name: **BUSINESS\_NAME**

Type: **STRING(200)**

Valid Values: **N/A**

Default: **"**

Business name of the Advanced Queue

Name: **DESCRIPTION**

Type: **STRING(4000)**

Valid Values: **N/A**

Default: **"**

Description of the Advanced Queue

Name: **PAYLOAD\_TYPE**

Type: **STRING(4000)**

Valid Values: **N/A**

Default: **"**

PayLoad Type of the Advanced Queue. This has to be the name of an Object Type(**OBJECT\_TYPE**) existing in the same Oracle Module.

**Properties for ADVANCED\_QUEUE:****Name:** DEPLOYABLE**Type:** BOOLEAN**Valid Values:** true, false**Default:** true

Warehouse Builder generates a set of scripts to create an object only for those objects marked as Deployable = true.

**Name:** GENERATE\_ADVANCED\_QUEUE**Type:** BOOLEAN**Valid Values:** true, false**Default:** true

Generate the code to create the Advanced Queue.

**Name:** GENERATE\_OBJECT\_TYPE**Type:** BOOLEAN**Valid Values:** true, false**Default:** true

Generate a script to create the Object Type.

**Name:** GENERATE\_QUEUE\_TABLE**Type:** BOOLEAN**Valid Values:** true, false**Default:** true

Generate code to create the queue table that will persist the messages of this Advanced Queue.

**Name:** GENERATE\_TEMPORARY\_TABLE**Type:** BOOLEAN**Valid Values:** true, false**Default:** true

Generate the code to create the temporary table.

**Name:** QUEUE\_TABLE\_NAME**Type:** STRING(30)**Valid Values:** N/A

Default: "

The name of the queue table that is used to persist the messages in this Advanced Queue.

Name: TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Use the Tablespace parameter to specify the name of tablespace.

Note:

1. N/A means any valid character in supported character set.
2. " represents an empty string

propertyNameList

The list of properties.

propertyValueList

The list of property values.

propertyValue

This clause adds the property values.

## Examples

```
OMBALTER ADVANCED_QUEUE 'SOME_ADVANCED_QUEUE' SET PROPERTIES  
(TABLESPACE, QUEUE_TABLE_NAME) VALUES ('TABLESPACE', 'QUEUE_TABLE')  
This will set its property Tablespace Name to "TABLESPACE" and Queue Table to  
'QUEUE_TABLE'.
```

## See Also

OMBALTER, OMBCREATE ADVANCED\_QUEUE, OMBRETRIEVE ADVANCED\_QUEUE, OMBDROP ADVANCED\_QUEUE

# OMBALTER COLLECTION

## Purpose

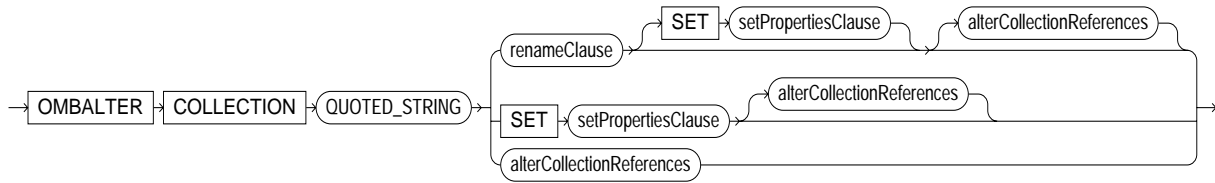
OMBALTER COLLECTION - Alter the collection by adding, removing or modifying shortcuts.

## Prerequisites

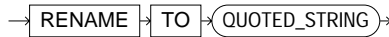
Should be in the context of a project, before altering a collection.

## Syntax Diagrams

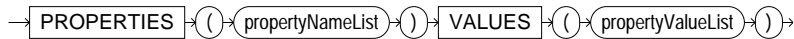
alterCollectionCommand



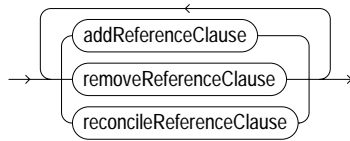
renameClause



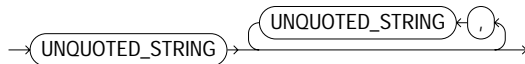
setPropertiesClause



alterCollectionReferences



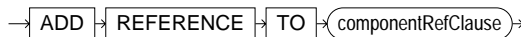
propertyNameList



propertyValueList



addReferenceClause





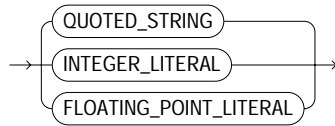
removeReferenceClause



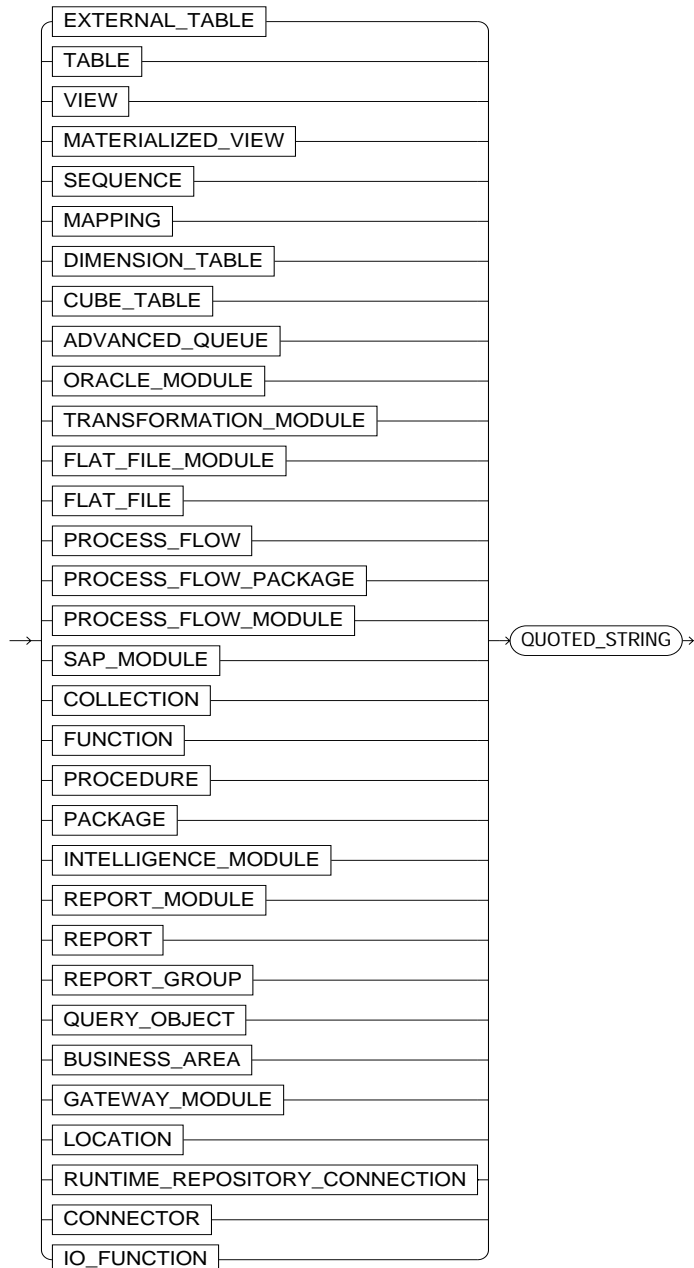
reconcileReferenceClause



propertyValue



componentRefClause



## Syntax

```
alterCollectionCommand = OMBALTER ( COLLECTION "QUOTED_STRING" (
"renameClause" [ SET "setPropertyClause" ] [ "alterCollectionReferences" ] | SET
"setPropertyClause" [ "alterCollectionReferences" ] | "alterCollectionReferences" ));
renameClause = RENAME TO "QUOTED_STRING";
setPropertyClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("
"propertyValueList" ");
alterCollectionReferences = ( "addReferenceClause" | "removeReferenceClause" |
"reconcileReferenceClause" );
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
propertyValueList = "propertyValue" { "," "propertyValue" };
addReferenceClause = ADD REFERENCE TO "componentRefClause";
removeReferenceClause = REMOVE REFERENCE TO "componentRefClause";
reconcileReferenceClause = RECONCILE;
propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );
componentRefClause = ( EXTERNAL_TABLE | TABLE | VIEW | MATERIALIZED_
VIEW | SEQUENCE | MAPPING | DIMENSION_TABLE | CUBE_TABLE |
ADVANCED_QUEUE | ORACLE_MODULE | TRANSFORMATION_MODULE |
FLAT_FILE_MODULE | FLAT_FILE | PROCESS_FLOW | PROCESS_FLOW_
PACKAGE | PROCESS_FLOW_MODULE | SAP_MODULE | COLLECTION |
FUNCTION | PROCEDURE | PACKAGE | INTELLIGENCE_MODULE | REPORT_
MODULE | REPORT | REPORT_GROUP | QUERY_OBJECT | BUSINESS_AREA |
GATEWAY_MODULE | LOCATION | RUNTIME_REPOSITORY_CONNECTION |
CONNECTOR | IO_FUNCTION ) "QUOTED_STRING";
```

## Keywords and Parameters

**alterCollectionCommand**

Alter a collection of objects.

**QUOTED\_STRING**

Name of the existing collection in quotes.

**renameClause**

Rename a collection.

**setPropertyClause**

Associate a set of properties with a collection.

Basic properties for COLLECTION:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the collection

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the collection

Note:

1. N/A means any valid character in supported character set.

2. " represents an empty string

alterCollectionReferences

Alter the collections references, includes adding, dropping and reconciling the collection.

propertyNameList

Comma separated list of property names. Property names are unquoted.

propertyValueList

Comma separated list of property values.

addReferenceClause

Add a reference to the collection.

removeReferenceClause

Remove a reference from the collection.

reconcileReferenceClause

Reconcile the collection, deleting references which now refer to deleted objects.

propertyValue

Value of a property.

`componentRefClause`

Specify the type of the object to reference.

### Examples

```
OMBALTER COLLECTION 'PURCHASING_WAREHOUSE' SET PROPERTIES  
(DESCRIPTION) VALUES ('Group for purchasing usage.') ADD REFERENCE TO  
TABLE 'PURCHASING/PRODUCT'
```

### See Also

OMBALTER, OMBCREATE COLLECTION

# OMBALTER CONNECTOR

## Purpose

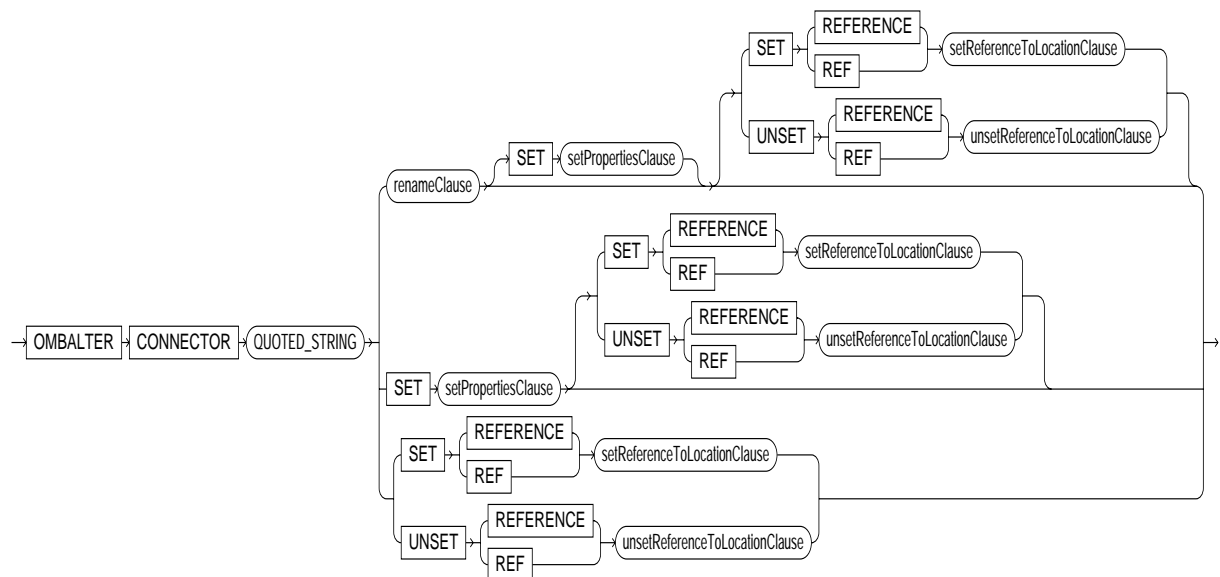
OMBALTER CONNECTOR - Alter the connector by renaming it, and/or reset its properties.

## Prerequisites

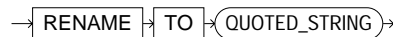
Should be in the context of the connector's owning location.

## Syntax Diagrams

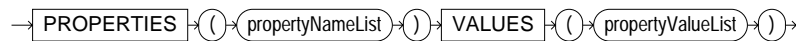
alterConnectorCommand



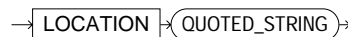
renameClause



setPropertiesClause



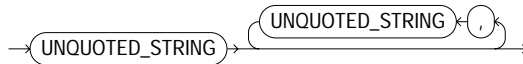
setReferenceToLocationClause



unsetReferenceToLocationClause



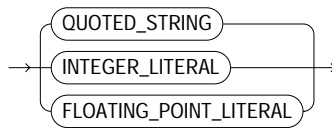
propertyNameList



propertyValueList



propertyValue



## Syntax

```

alterConnectorCommand = OMBALTER ( CONNECTOR "QUOTED_STRING" ( (
"renameClause" [ SET "setPropertyClause" ] [ SET ( REFERENCE | REF )
"setReferenceToLocationClause" | UNSET ( REFERENCE | REF )
"unsetReferenceToLocationClause" ] ) | ( SET "setPropertyClause" [ SET (
REFERENCE | REF ) "setReferenceToLocationClause" | UNSET ( REFERENCE | REF )
) "unsetReferenceToLocationClause" ] ) | ( SET ( REFERENCE | REF )
"setReferenceToLocationClause" | UNSET ( REFERENCE | REF )
"unsetReferenceToLocationClause" ) ) );

renameClause = RENAME TO "QUOTED_STRING";

setPropertyClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("
"propertyValueList" ")";

setReferenceToLocationClause = LOCATION "QUOTED_STRING";

unsetReferenceToLocationClause = LOCATION;

propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };

propertyValueList = "propertyValue" { "," "propertyValue" };

propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );
    
```

## Keywords and Parameters

**alterConnectorCommand**

Alter the connector specified by the quoted string.

**renameClause**

Rename the connector.

**setPropertyClause**

Set specified properties of the connector.

`setReferenceToLocationClause`

Set the name of the location which the connector references.

`unsetReferenceToLocationClause`

Unset the referenced location.

`propertyNameList`

The names of the properties whose values you want to set.

Properties for CONNECTOR:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those objects marked as Deployable = true

Note:

1. N/A means any valid character in supported character set.
2. " " represents an empty string

`propertyValueList`

The values for the named properties.

`propertyValue`

A property value.

## Examples

```
OMBALTER CONNECTOR 'OLD_CONNECTOR' RENAME TO 'NEW_
CONNECTOR' SET PROPERTIES
```

(DESCRIPTION, BUSINESS\_NAME) VALUES ('This becomes a new connector.', 'new connector') This will rename the connector "OLD\_CONNECTOR" to "NEW\_CONNECTOR", and set its description to "This becomes a new connector", set its business name to "new connector".

```
OMBALTER CONNECTOR 'A_CONNECTOR' SET REFERENCE LOCATION 'MY_
LOCATION'
```

OMBALTER CONNECTOR 'A\_CONNECTOR' UNSET REF LOCATION

**See Also**

OMBALTER, OMBCREATE CONNECTOR, OMBDROP CONNECTOR



# OMBALTER CUBE\_TABLE

## Purpose

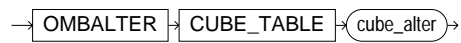
OMBALTER CUBE\_TABLE - This command alters a cube.

## Prerequisites

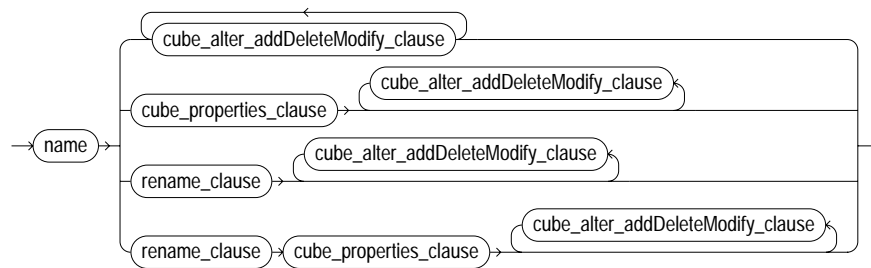
Should be in Oracle Module context.

## Syntax Diagrams

OMBAAlterCube



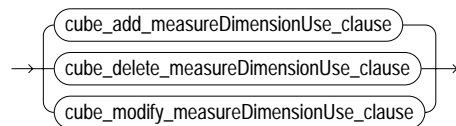
cube\_alter



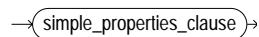
name



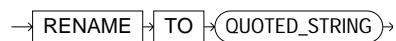
cube\_alter\_addDeleteModify\_clause



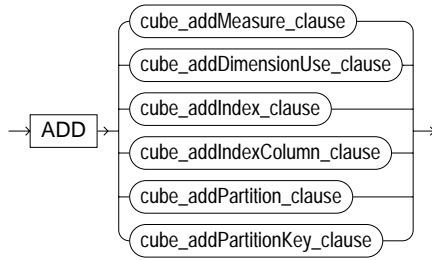
cube\_properties\_clause



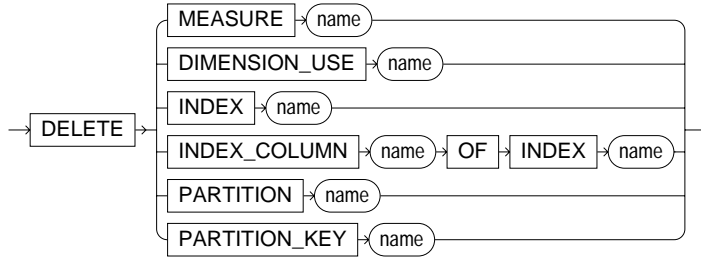
rename\_clause



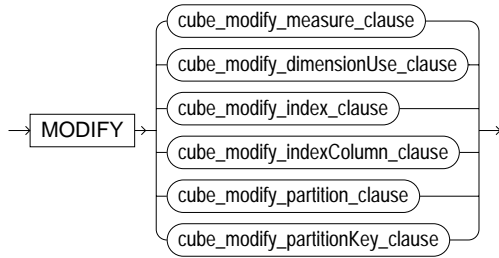
cube\_add\_measureDimensionUse\_clause



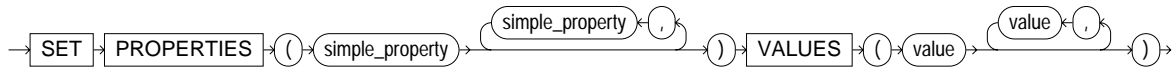
cube\_delete\_measureDimensionUse\_clause



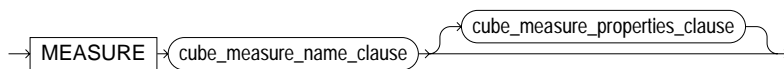
cube\_modify\_measureDimensionUse\_clause



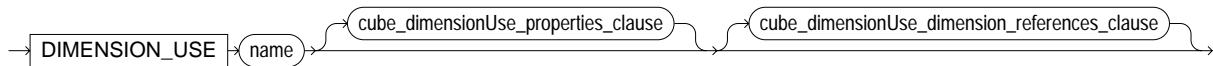
simple\_properties\_clause



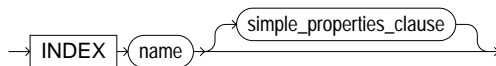
cube\_addMeasure\_clause



cube\_addDimensionUse\_clause



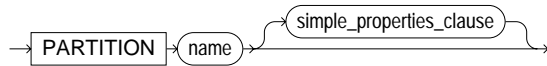
cube\_addIndex\_clause



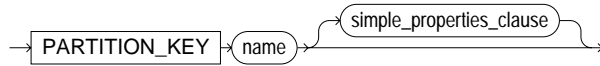
cube\_addIndexColumn\_clause



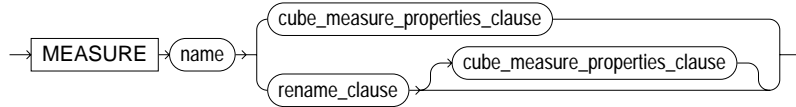
cube\_addPartition\_clause



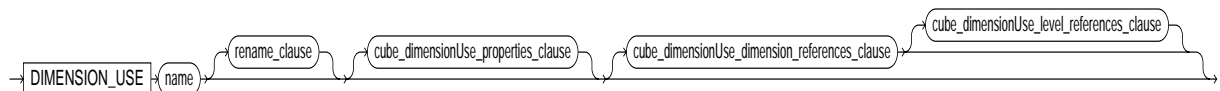
cube\_addPartitionKey\_clause



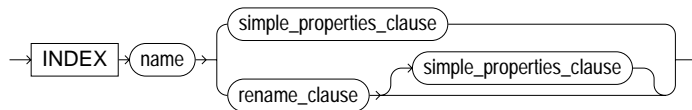
cube\_modify\_measure\_clause



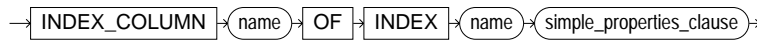
cube\_modify\_dimensionUse\_clause



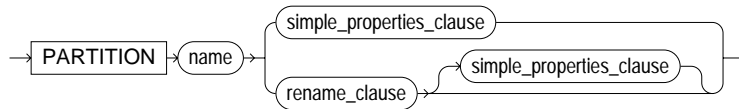
cube\_modify\_index\_clause



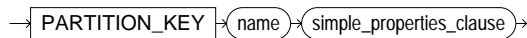
cube\_modify\_indexColumn\_clause



cube\_modify\_partition\_clause



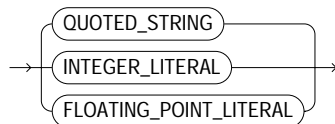
cube\_modify\_partitionKey\_clause



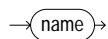
simple\_property



value



cube\_measure\_name\_clause



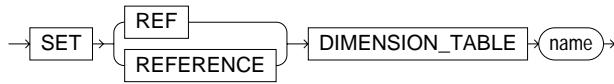
cube\_measure\_properties\_clause



cube\_dimensionUse\_properties\_clause



cube\_dimensionUse\_dimension\_references\_clause



cube\_dimensionUse\_level\_references\_clause



## Syntax

```
OMBAlterCube = OMBALTER CUBE_TABLE "cube_alter";
```

```
cube_alter = "name" ( "cube_alter_addDeleteModify_clause"+ | "cube_properties_
clause" { "cube_alter_addDeleteModify_clause" } | "rename_clause" { "cube_alter_
addDeleteModify_clause" } | ( "rename_clause" "cube_properties_clause" { "cube_
alter_addDeleteModify_clause" } ) );
```

```
name = ("QUOTED_STRING");
```

```
cube_alter_addDeleteModify_clause = "cube_add_measureDimensionUse_clause" |
"cube_delete_measureDimensionUse_clause" | "cube_modify_
measureDimensionUse_clause";
```

```
cube_properties_clause = "simple_properties_clause";
```

```
rename_clause = RENAME TO "QUOTED_STRING";
```

```
cube_add_measureDimensionUse_clause = ADD ( "cube_addMeasure_clause" |
"cube_addDimensionUse_clause" | "cube_addIndex_clause" | "cube_
addIndexColumn_clause" | "cube_addPartition_clause" | "cube_addPartitionKey_
clause" );
```

```
cube_delete_measureDimensionUse_clause = DELETE ( MEASURE "name" |
DIMENSION_USE "name" | INDEX "name" | INDEX_COLUMN "name" OF INDEX
"name" | PARTITION "name" | PARTITION_KEY "name" );
```

```
cube_modify_measureDimensionUse_clause = MODIFY ( "cube_modify_measure_
clause" | "cube_modify_dimensionUse_clause" | "cube_modify_index_clause" |
"cube_modify_indexColumn_clause" | "cube_modify_partition_clause" | "cube_
modify_partitionKey_clause" );
```

```
simple_properties_clause = SET PROPERTIES (" "simple_property" { "," "simple_
property" } )" VALUES (" "value" { "," "value" } )";
```

```
cube_addMeasure_clause = MEASURE "cube_measure_name_clause" [ "cube_
measure_properties_clause" ];
```

```
cube_addDimensionUse_clause = DIMENSION_USE ( "name" [ "cube_
dimensionUse_properties_clause" ] [ "cube_dimensionUse_dimension_references_
clause" ] );
```

```
cube_addIndex_clause = INDEX "name" [ "simple_properties_clause" ];
```

```

cube_addIndexColumn_clause = INDEX_COLUMN "name" OF INDEX "name" [
"simple_properties_clause" ];
cube_addPartition_clause = PARTITION "name" [ "simple_properties_clause" ];
cube_addPartitionKey_clause = PARTITION_KEY "name" [ "simple_properties_
clause" ];
cube_modify_measure_clause = MEASURE "name" ( "cube_measure_properties_
clause" | ( "rename_clause" [ "cube_measure_properties_clause" ] ) );
cube_modify_dimensionUse_clause = DIMENSION_USE "name" [ "rename_clause" ] [
"cube_dimensionUse_properties_clause" ] [ "cube_dimensionUse_dimension_
references_clause" [ "cube_dimensionUse_level_references_clause" ] ];
cube_modify_index_clause = INDEX "name" ( "simple_properties_clause" | (
"rename_clause" [ "simple_properties_clause" ] ) );
cube_modify_indexColumn_clause = INDEX_COLUMN "name" OF INDEX "name"
"simple_properties_clause";
cube_modify_partition_clause = PARTITION "name" ( "simple_properties_clause" | (
"rename_clause" [ "simple_properties_clause" ] ) );
cube_modify_partitionKey_clause = PARTITION_KEY "name" "simple_properties_
clause";
simple_property = "UNQUOTED_STRING";
value = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_POINT_
LITERAL" );
cube_measure_name_clause = "name";
cube_measure_properties_clause = "simple_properties_clause";
cube_dimensionUse_properties_clause = "simple_properties_clause";
cube_dimensionUse_dimension_references_clause = SET ( REF | REFERENCE ) (
DIMENSION_TABLE "name" );
cube_dimensionUse_level_references_clause = SET ( REF | REFERENCE ) ( LEVEL
"name" );

```

## Keywords and Parameters

OMBAAlterCube

This command alter a cube\_table

cube\_alter

This clause alter a cube\_table by adding, deleting, and modifying clauses.

name

The name has to be a quoted string or an integer, or a decimal number.

cube\_alter\_addDeleteModify\_clause

This clause adds, deletes, and modifies measures, dimensionUses, indexes, indexColumns, partition, and partitionKeys.

cube\_properties\_clause

This clause uses the simple properties.

rename\_clause

This clause renames to another name

cube\_add\_measureDimensionUse\_clause

This clause adds measures, dimensionUses, indexes, indexColumns, partition, and partitionKeys.

cube\_delete\_measureDimensionUse\_clause

This clause deletes measures, dimensionUses, indexes, indexColumns, partition, and partitionKeys.

cube\_modify\_measureDimensionUse\_clause

This clause modifies measures, dimensionUses, indexes, indexColumns, partition, and partitionKeys.

simple\_properties\_clause

Sets properties and their values.

Basic properties for TABLE:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the table

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the table

Basic properties for INDEX, PARTITION, PARTITION\_KEY, INDEX\_COLUMN:

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the Index, Partition, PartitionKey, IndexColumn

Properties for CUBE\_TABLE:

Name: ANALYZE\_TABLE\_ESTIMATE\_PERCENT

Type: NUMBER

Valid Values: 0 - 100

Default: 99

Value represents the sample size as a percentage of total rows. When set to a nonzero value, Builder generates a DDL script to analyze the table.

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: HASH\_PARTITION\_TABLESPACE\_LIST

Type: STRING

Valid Values: N/A

Default: "

A comma separated list of tablespaces to use for [sub]partition storage.

Name: HASH\_SUBPARTITION\_NUMBER

Type: NUMBER

Valid Values: 2 - 63999

Default: 2

To create Hash partition, specify the number of Hash subpartition.

Name: LOGGING\_MODE

Type: STRING

Valid Values: NOLOGGING, LOGGING

Default: LOGGING

Recovery requirements for a data warehouse : Logging or not logging to Redo Log File.

Name: PARALLEL\_ACCESS\_MODE

Type: STRING

Valid Values: NOPARALLEL, PARALLEL

Default: PARALLEL

Enables or disables parallel processing when the table is created. Also enables or disables parallel processing or access.

Name: TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Use the Tablespace parameter to specify the name of tablespace.

Properties for INDEX:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: INDEX\_TYPE

Type: STRING

Valid Values: BITMAP, UNIQUE, NO\_INDEX

Default: UNIQUE

The types of Indexes created on Dimension are BITMAP, UNIQUE or a non-specific index.



Name: LOCAL\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

A local index is constructed so that it reflects the structure of the underlying table. It is equipartitioned with the underlying table, meaning that it is partitioned on the same columns as the underlying table, creates the same number of partitions or subpartitions, and gives them the same partition bounds as corresponding partitions of the underlying table.

Name: LOGGING\_MODE

Type: STRING

Valid Values: NOLOGGING, LOGGING

Default: LOGGING

Recovery requirements for a data warehouse : Logging or not logging to Redo Log File.

Name: PARALLEL\_ACCESS\_MODE

Type: STRING

Valid Values: NOPARALLEL, PARALLEL

Default: PARALLEL

Enables or disables parallel processing when the table is created. Also enables or disables parallel processing or access.

Name: TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Use the Tablespace parameter to specify the name of tablespace.

Name: USING\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

Use of Index on Dimension Constraint

Properties for PARTITION:

Name: DATE\_LESS\_THAN

Type: STRING

Valid Values: N/A

Default: TO\_DATE('01-JAN-1999', 'DD-MON-YYYY')

Value that represents upper bound of partition stored in warehouse key column for the Days Dimension.

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: EMPTY\_STRING

Use the Tablespace parameter to specify the name of tablespace.

Name: VALUES\_LESS\_THAN

Type: STRING

Valid Values: N/A

Default: TO\_DATE('01-JAN-1999', 'DD-MON-YYYY')

Noninclusive upper bound for the current partition. Type a comma separated list in the same order as the columns in range partitioned key.

Name: VALUE\_LESS\_THAN

Type: STRING

Valid Values: N/A

Default: "

Noninclusive upper bound for the current partition. Type a comma separated list in the same order as the columns in range partitioned key.

Properties for PARTITION\_KEY:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: TYPE

Type: STRING

Valid Values: HASH, RANGE

Default: RANGE

Oracle partitions the storage space and stores rows according to a Hash Algorithm or specified ranges.

Note:

1. N/A means any valid character in supported character set.
2. " represents an empty string

cube\_addMeasure\_clause

This clause adds the measure name and its properties.

cube\_addDimensionUse\_clause

This clause adds the dimensionUse and its properties.

cube\_addIndex\_clause

This clause adds Index name and its properties.

cube\_addIndexColumn\_clause

This clause adds IndexColumn of an index and its properties.

cube\_addPartition\_clause

This clause adds partition and its properties.

cube\_addPartitionKey\_clause

This clause adds a partitionKey and its properties.

cube\_modify\_measure\_clause

This clause modifies measure by renaming it and changing its properties.

cube\_modify\_dimensionUse\_clause

This clause modifies dimensionUse.

cube\_modify\_index\_clause

This clause modifies an index.

cube\_modify\_indexColumn\_clause

This clause modifies an indexColumn.

cube\_modify\_partition\_clause

This clause modifies an partition.

cube\_modify\_partitionKey\_clause

cube\_modify\_partition\_clause = This clause modifies an partitionKey.

simple\_property

gets the simple property.

UNQUOTED\_STRING

Name of the simple property.

value

The quoted string name

cube\_measure\_name\_clause

name of the measure.

cube\_measure\_properties\_clause

alters simple properties of a measure.

cube\_dimensionUse\_properties\_clause

alters simple properties of a dimensionUse.

cube\_dimensionUse\_dimension\_references\_clause

The dimensionUse references to the first level of the dimension.

cube\_dimensionUse\_level\_references\_clause

The dimensionUse references to the user specified level of the dimension.

**Examples**

```
OMBALTER CUBE_TABLE 'CUBE1' ADD MEASURE 'MEASURE_2' DELETE  
MEASURE 'MEASURE_1'
```

**See Also**

```
OMBCREATE CUBE_TABLE, OMBDROP CUBE_TABLE, OMBRETRIEVE CUBE_  
TABLE
```

# OMBALTER DEPLOYMENT\_ACTION\_PLAN

## Purpose

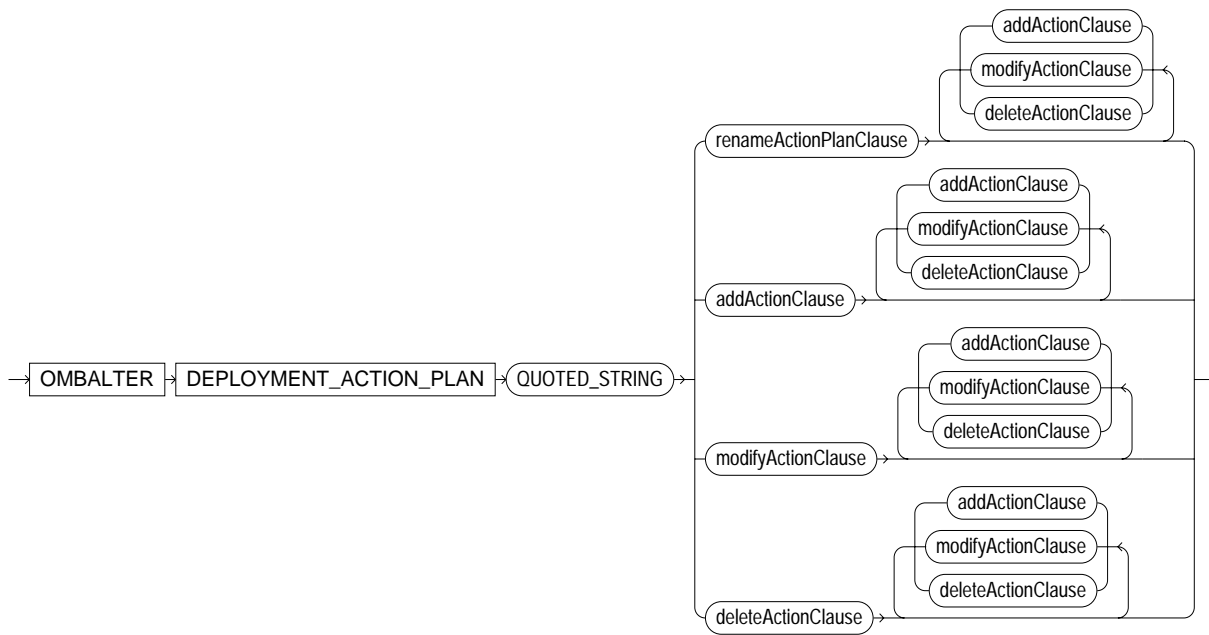
OMBALTER DEPLOYMENT\_ACTION\_PLAN - Modify an existing deployment action plan.

## Prerequisites

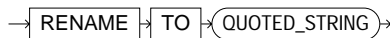
There must be a current working project.

## Syntax Diagrams

AlterActionPlanCommand



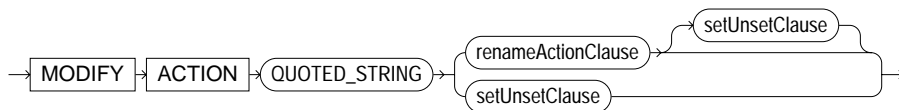
renameActionPlanClause



addActionClause



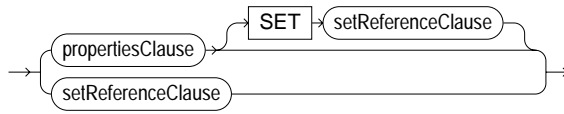
modifyActionClause



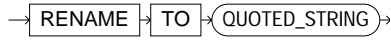
deleteActionClause



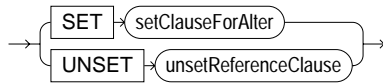
setClause



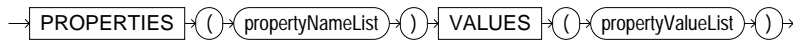
renameActionClause



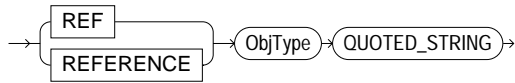
setUnsetClause



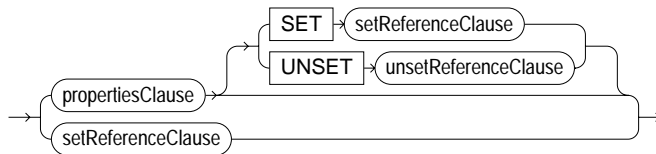
propertiesClause



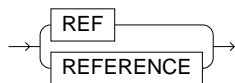
setReferenceClause



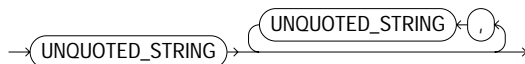
setClauseForAlter



unsetReferenceClause



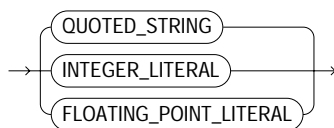
propertyNameList



propertyValueList



propertyValue



## Syntax

```
AlterActionPlanCommand = ( OMBALTER ( DEPLOYMENT_ACTION_PLAN )
"QUOTED_STRING" ( ( "renameActionPlanClause" { "addActionClause" |
"modifyActionClause" | "deleteActionClause" } ) | ( "addActionClause" {
"addActionClause" | "modifyActionClause" | "deleteActionClause" } ) | (
"modifyActionClause" { "addActionClause" | "modifyActionClause" |
"deleteActionClause" } ) | ( "deleteActionClause" { "addActionClause" |
"modifyActionClause" | "deleteActionClause" } ) ) );
renameActionPlanClause = RENAME TO "QUOTED_STRING";
addActionClause = ADD ACTION "QUOTED_STRING" [ SET "setClause" ];
modifyActionClause = MODIFY ACTION "QUOTED_STRING" ( (
"renameActionClause" [ "setUnsetClause" ] ) | "setUnsetClause" );
deleteActionClause = DELETE ACTION "QUOTED_STRING";
setClause = ( "propertiesClause" [ SET "setReferenceClause" ] ) | "setReferenceClause";
renameActionClause = RENAME TO "QUOTED_STRING";
setUnsetClause = ( ( SET "setClauseForAlter" ) | ( UNSET "unsetReferenceClause" ) );
propertiesClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("
"propertyValueList" ")";
setReferenceClause = ( REF | REFERENCE ) "ObjType" "QUOTED_STRING";
setClauseForAlter = ( "propertiesClause" [ SET "setReferenceClause" | UNSET
"unsetReferenceClause" ] ) | "setReferenceClause";
unsetReferenceClause = ( REF | REFERENCE );
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
propertyValueList = "propertyValue" { "," "propertyValue" };
propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );
```

## Keywords and Parameters

**AlterActionPlanCommand**

Modify an existing deployment action plan.

**renameActionPlanClause**

Rename an action plan.

**addActionClause**

Add an action to an action plan.

**modifyActionClause**

Modify an action of an action plan.



**deleteActionClause**

Remove an action from an action plan.

**setClause**

Set the properties of an action and/or associate an object with an action.

**renameActionClause**

Rename an action.

**setUnsetClause**

Set the properties and/or associate/disassociate an object with an action.

**propertiesClause**

Associate a set of properties with an action.

**PROPERTIES**

The only valid property is OPERATION, which specifies the type of action to be taken.

**setReferenceClause**

Associate an object with an action.

**ObjType**

Object type. Valid values are ADVANCED\_QUEUE, CUBE\_TABLE, DIMENSION\_TABLE,

EXTERNAL\_TABLE, CONNECTOR, FUNCTION, MAPPING, MATERIALIZED\_VIEW, PROCEDURE,

PROCESS\_FLOW\_PACKAGE, SEQUENCE, TABLE, and VIEW.

**QUOTED\_STRING**

Absolute or relative path name of an object (e.g. '/MY\_PROJECT/MODULE\_X/TABLE\_Y').

**setClauseForAlter**

Set the properties and/or associate/disassociate an object with an action.

**unsetReferenceClause**

Disassociate a previously associated object from an action.

propertyNameList

Comma separated list of property names. Property names are unquoted.

propertyValueList

Comma separated list of property values.

propertyValue

Value of a property. Valid values for OPERATION are DROP and CREATE.

## Examples

```
OMBALTER DEPLOYMENT_ACTION_PLAN 'MY_PLAN' RENAME TO 'MY_
PLAN_2'
```

```
OMBALTER DEPLOYMENT_ACTION_PLAN 'MY_PLAN' ADD ACTION 'MY_
ALTER_TABLE'
```

```
SET PROPERTIES (OPERATION) VALUES ('CREATE') SET REFERENCE TABLE
'TABLE_X'
```

```
OMBALTER DEPLOYMENT_ACTION_PLAN 'MY_PLAN' MODIFY ACTION 'MY_
VIEW_CREATE'
```

```
RENAME TO 'MY_VIEW_DROP' SET PROPERTIES (OPERATION) VALUES ('DROP')
```

```
OMBALTER DEPLOYMENT_ACTION_PLAN 'MY_PLAN' DELETE ACTION 'MY_
TABLE_DEPLOY'
```

```
OMBALTER DEPLOYMENT_ACTION_PLAN 'MY_PLAN'
```

```
ADD ACTION 'MY_ALTER_VIEW' SET PROPERTIES (OPERATION) VALUES
('CREATE')
```

```
SET REFERENCE VIEW '/MY_PROJECT/MY_MODULE/VIEW_Y'
```

```
MODIFY ACTION 'MY_TABLE_DEPLOY' SET REFERENCE TABLE 'MY_
MODULE/TABLE_Z'
```

```
DELETE ACTION 'MY_VIEW_CREATE'
```

## See Also

OMBCREATE DEPLOYMENT\_ACTION\_PLAN, OMBDEPLOY



# OMBALTER EXTERNAL\_TABLE

## Purpose

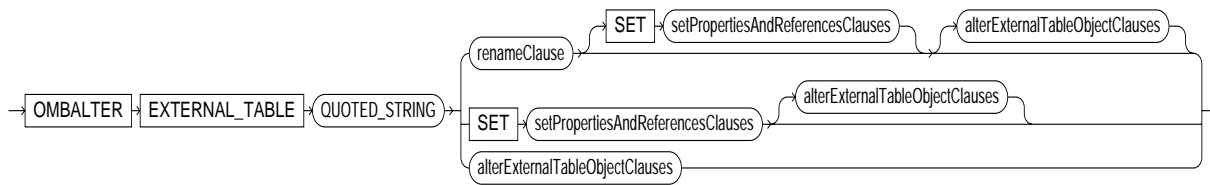
OMBALTER EXTERNAL\_TABLE - Alter the external table by renaming it, and/or reset its properties.

## Prerequisites

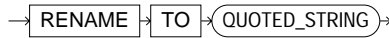
Should be in the context of an Oracle Module.

## Syntax Diagrams

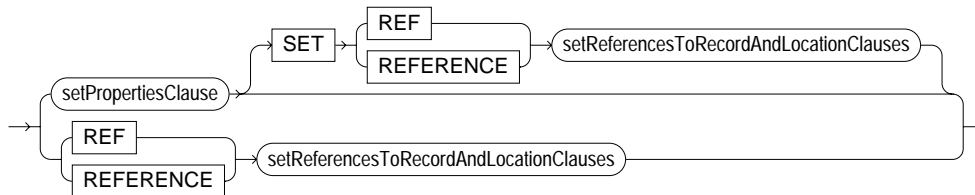
alterExternalTableCommand



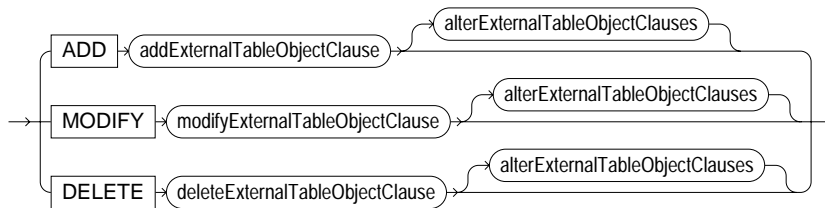
renameClause



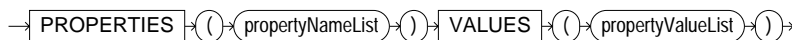
setPropertiesAndReferencesClauses



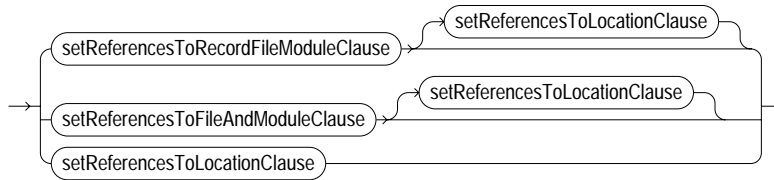
alterExternalTableObjectClauses



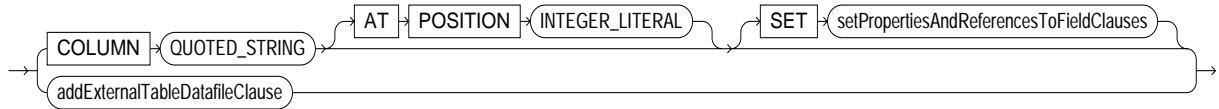
setPropertiesClause



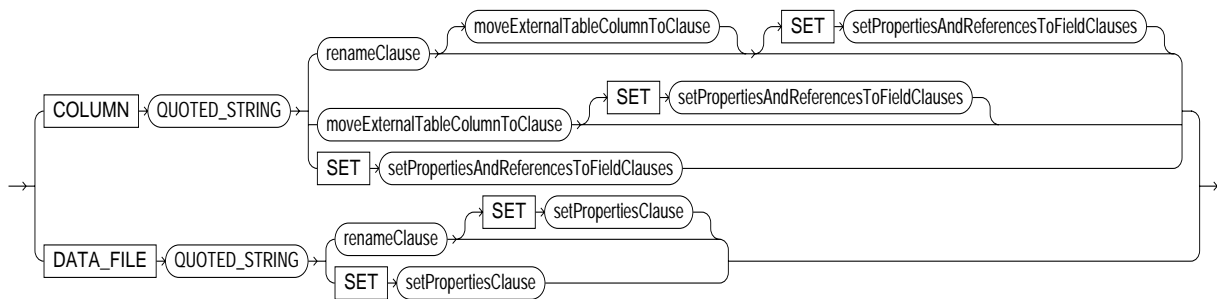
setReferencesToRecordAndLocationClauses



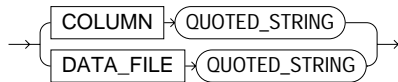
addExternalTableObjectClause



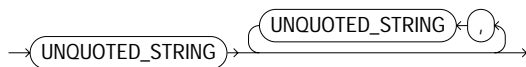
modifyExternalTableObjectClause



deleteExternalTableObjectClause



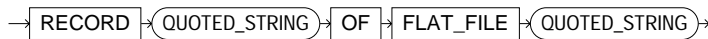
propertyNameList



propertyValueList



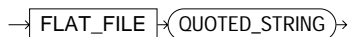
setReferencesToRecordFileModuleClause



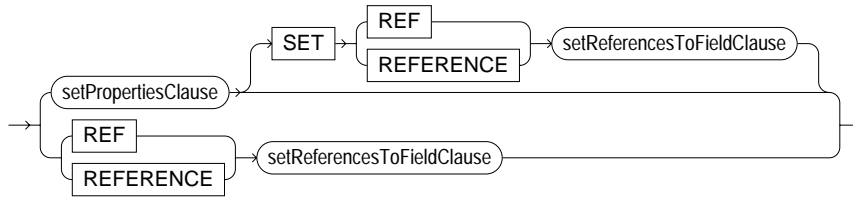
setReferencesToLocationClause



setReferencesToFileAndModuleClause



setPropertiesAndReferencesToFieldClauses



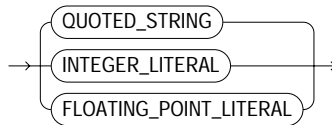
addExternalTableDatafileClause



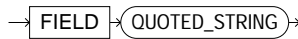
moveExternalTableColumnToClause



propertyValue



setReferencesToFieldClause



## Syntax

```
alterExternalTableCommand = OMBALTER ( EXTERNAL_TABLE "QUOTED_
STRING" ("renameClause" [ SET "setPropertiesAndReferencesClauses" ] [
"alterExternalTableObjectClauses" ] | SET "setPropertiesAndReferencesClauses" [
"alterExternalTableObjectClauses" ] | "alterExternalTableObjectClauses" ) );
```

```
renameClause = RENAME TO "QUOTED_STRING";
```

```
setPropertiesAndReferencesClauses = "setPropertiesClause" [ SET ( REF |
REFERENCE ) "setReferencesToRecordAndLocationClauses" ] | ( REF | REFERENCE
) "setReferencesToRecordAndLocationClauses";
```

```
alterExternalTableObjectClauses = ADD "addExternalTableObjectClause" [
"alterExternalTableObjectClauses" ] | MODIFY "modifyExternalTableObjectClause" [
"alterExternalTableObjectClauses" ] | DELETE "deleteExternalTableObjectClause" [
"alterExternalTableObjectClauses" ];
```

```
setPropertiesClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("
"propertyValueList" ")";
```

```
setReferencesToRecordAndLocationClauses =
"setReferencesToRecordFileModuleClause" [ "setReferencesToLocationClause" ] |
"setReferencesToFileAndModuleClause" [ "setReferencesToLocationClause" ] |
"setReferencesToLocationClause";
```

```
addExternalTableObjectClause = COLUMN "QUOTED_STRING" [ AT POSITION
"INTEGER_LITERAL" ] [ SET "setPropertiesAndReferencesToFieldClauses" ] |
"addExternalTableDatafileClause";
```

```

modifyExternalTableObjectClause = COLUMN "QUOTED_STRING" (
"renameClause" [ "moveExternalTableColumnToClause" ] [ SET
"setPropertiesAndReferencesToFieldClauses" ] |
"moveExternalTableColumnToClause" [ SET
"setPropertiesAndReferencesToFieldClauses" ] | SET
"setPropertiesAndReferencesToFieldClauses" ) | DATA_FILE "QUOTED_STRING" (
"renameClause" [ SET "setPropertiesClause" ] | SET "setPropertiesClause");

deleteExternalTableObjectClause = COLUMN "QUOTED_STRING" | DATA_FILE
"QUOTED_STRING";

propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };

propertyValueList = "propertyValue" { "," "propertyValue" };

setReferencesToRecordFileModuleClause = RECORD "QUOTED_STRING" OF FLAT_
FILE "QUOTED_STRING";

setReferencesToLocationClause = DEFAULT_LOCATION "QUOTED_STRING";

setReferencesToFileAndModuleClause = FLAT_FILE "QUOTED_STRING";

setPropertiesAndReferencesToFieldClauses = "setPropertiesClause" [ SET ( REF |
REFERENCE ) "setReferencesToFieldClause" ] | ( REF | REFERENCE )
"setReferencesToFieldClause";

addExternalTableDatafileClause = DATA_FILE "QUOTED_STRING" [ SET
"setPropertiesClause" ];

moveExternalTableColumnToClause = MOVE TO POSITION "INTEGER_LITERAL";

propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );

setReferencesToFieldClause = FIELD "QUOTED_STRING";

```

## Keywords and Parameters

**alterExternalTableCommand**

Alter the existing external table.

**QUOTED\_STRING**

The name of the external table to alter.

**renameClause**

Rename the external table to the value of the following quoted string.

**setPropertiesAndReferencesClauses**

Set the properties and/or flat file reference of the external table.

**alterExternalTableObjectClauses**

Add, modify, or delete columns or data files.

**setPropertiesClause**

Set specified properties of the external table.

`setReferencesToRecordAndLocationClauses`

Set the referenced record and/or default location.

`addExternalTableObjectClause`

Add a column to the external table. The name of the new column will be the quoted string.

`modifyExternalTableObjectClause`

Modify the properties of a column or data file or move a column to a new position.

`deleteExternalTableObjectClause`

Delete a column or data file.

`propertyNameList`

The names of the properties whose values you want to set.

Properties for EXTERNAL\_TABLE:

Name: BAD\_FILE\_LOCATION

Type: STRING

Valid Values: N/A

Default: USE\_DEFAULT\_LOCATION

Location to use when creating the bad file. When the Access Parameters property is specified for the External Table, this configuration parameter is ignored.

Name: BAD\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: "

Name to use when creating the bad file. When the Access Parameters property is specified for the External Table, this configuration parameter is ignored.

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true



When deployable is set to true, a script to create an External Table is generated.

Name: DISCARD\_FILE\_LOCATION

Type: STRING

Valid Values: N/A

Default: USE\_DEFAULT\_LOCATION

Location to use when creating the discard file. When the Access Parameters property is specified for the External Table, this configuration parameter is ignored.

Name: DISCARD\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: "

Name to use when creating the discard file. When the Access Parameters property is specified for the External Table, this configuration parameter is ignored.

Name: ENDIAN

Type: STRING

Valid Values: BIG, LITTLE, PLATFORM

Default: PLATFORM

Data endian should be platform default, little or big. When the Access Parameters property is specified for the External Table, this configuration parameter is ignored.

Name: LOAD\_NULLS\_WHEN\_MISSING\_VALUES

Type: BOOLEAN

Valid Values: true, false

Default: false

If TRUE, then NULLs are loaded for any missing values in the record. If FALSE, then records with missing values are rejected. When the Access Parameters property is specified for the External Table, this configuration parameter is ignored.

Name: LOG\_FILE\_LOCATION

Type: STRING

Valid Values: N/A

Default: USE\_DEFAULT\_LOCATION

Location to use when creating the log file. When the Access Parameters property is specified for the External Table, this configuration parameter is ignored.

Name: LOG\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: "

Name to use when creating the log file. When the Access Parameters property is specified for the External Table, this configuration parameter is ignored.

Name: NUMBER\_OF\_REJECTS\_ALLOWED

Type: NUMBER

Valid Values: 0 - 2147483647

Default: 0

The number of rejects allowed before processing is terminated.

Name: PARALLEL\_ACCESS\_DRIVERS

Type: NUMBER

Valid Values: 1 - 63999

Default: 1

The number of parallel access drivers to enable.

Name: PARALLEL\_ACCESS\_MODE

Type: BOOLEAN

Valid Values: true, false

Default: false

Enable or disable parallel processing.

Name: REJECTS\_ARE\_UNLIMITED

Type: BOOLEAN

Valid Values: true, false

Default: true

Enable or disable limiting the number of rejected records.

Name: STRING\_SIZES\_IN

Type: STRING

Valid Values: CHARACTERS, BYTES

Default: BYTES

String sizes are in bytes or characters. When the Access Parameters property is specified for the External Table, this configuration parameter is ignored.

Name: TRIM

Type: STRING

Valid Values: LEFT, NONE, SQL\*LOADER, BOTH, RIGHT

Default: NONE

Specification from trim option on input fields. When the Access Parameters property is specified for the External Table, this configuration parameter is ignored.

Properties for DATA\_FILE:

Name: DATA\_FILE\_LOCATION

Type: STRING

Valid Values: N/A

Default: USE\_DEFAULT\_LOCATION

The location of this data file for the external table.

Name: DATA\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: "

Name of this data file.

Note:

1. N/A means any valid character in supported character set.
2. " represents an empty string

propertyValueList

The values for the named properties.

setReferencesToRecordFileModuleClause

Specify the record and full path to the flat file for the external table to reference.

setReferencesToLocationClause

Specify the default location for the external table.

setReferencesToFileAndModuleClause

Specify the full path to the flat file for the external table to reference.

**setPropertyAndReferencesToFieldClauses**

Set the properties and/or field reference of the external table column.

**addExternalTableDatafileClause**

Add a new data file to the external table. The name of the new data file will be the quoted string. You may also set the properties of the new data file.

**moveExternalTableColumnToClause**

Move a column of the external table.

**propertyValue**

A property value.

**setReferencesToFieldClause**

Set the name of the field which the external table column references.

**Examples**

```
OMBALTER EXTERNAL_TABLE 'SRC_TABLE' RENAME TO 'MY_TABLE' SET
PROPERTIES (DESCRIPTION, BUSINESS_NAME) VALUES ('a new description', 'My
Table') This will rename the external table "SRC_TABLE" to "MY_TABLE", and set its
description to "a new description", and set its business name to "My Table".
```

**See Also**

OMBALTER, OMBCREATE EXTERNAL\_TABLE, OMBDROP EXTERNAL\_TABLE

# OMBALTER FLAT\_FILE

## Purpose

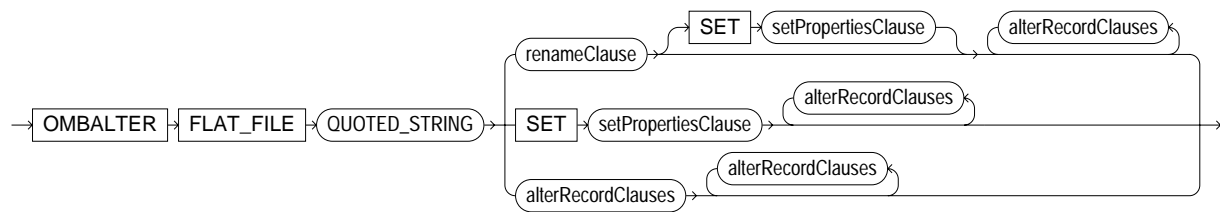
OMBALTER FLAT\_FILE - Alter the flat file by renaming it, and/or reset its properties.

## Prerequisites

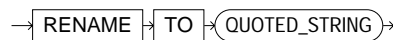
Should be in the context of a flat file module.

## Syntax Diagrams

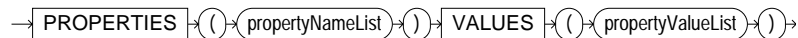
alterFlatFileCommand



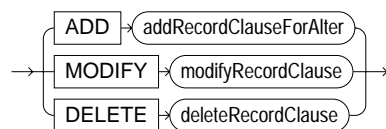
renameClause



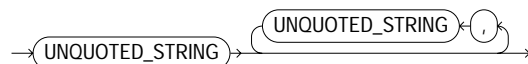
setPropertiesClause



alterRecordClauses



propertyNameList



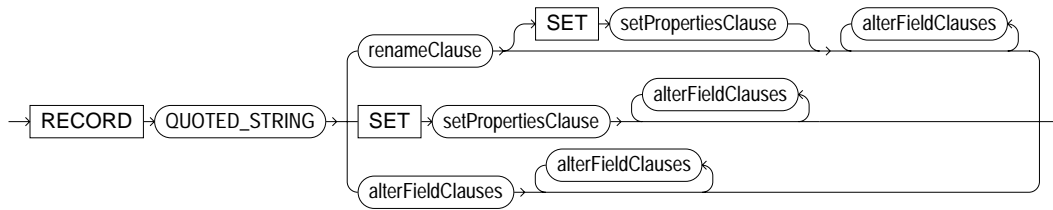
propertyValueList



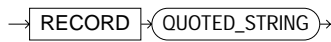
addRecordClauseForAlter



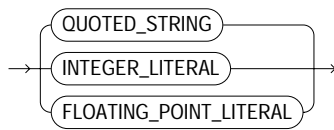
modifyRecordClause



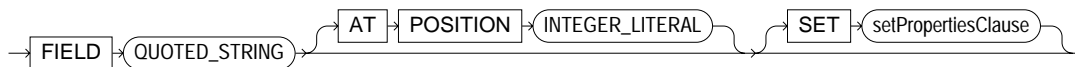
deleteRecordClause



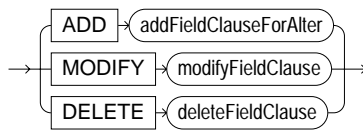
propertyValue



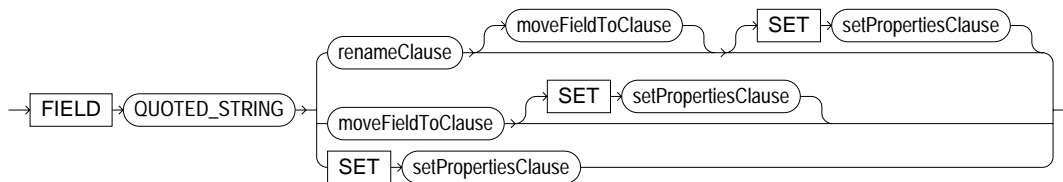
addFieldClauseForAlter



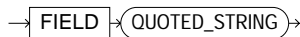
alterFieldClauses



modifyFieldClause



deleteFieldClause



moveFieldToClause



## Syntax

```

alterFlatFileCommand = OMBALTER ( FLAT_FILE "QUOTED_STRING" (
  "renameClause" [ SET "setPropertiesClause" ] { "alterRecordClauses" } | SET
  "setPropertiesClause" { "alterRecordClauses" } | "alterRecordClauses" {
  "alterRecordClauses" } ) );
  
```

```

renameClause = RENAME TO "QUOTED_STRING";
  
```

```

setPropertyClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("
"propertyValueList" ")";

alterRecordClauses = ADD "addRecordClauseForAlter" | MODIFY
"modifyRecordClause" | DELETE "deleteRecordClause";

propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };

propertyValueList = "propertyValue" { "," "propertyValue" };

addRecordClauseForAlter = RECORD "QUOTED_STRING" [ SET
"setPropertyClause" ] { ADD "addFieldClauseForAlter" };

modifyRecordClause = RECORD "QUOTED_STRING" ( "renameClause" [ SET
"setPropertyClause" ] { "alterFieldClauses" } | SET "setPropertyClause" {
"alterFieldClauses" } | "alterFieldClauses" { "alterFieldClauses" } );

deleteRecordClause = RECORD "QUOTED_STRING";

propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );

addFieldClauseForAlter = FIELD "QUOTED_STRING" [ AT POSITION "INTEGER_
LITERAL" ] [ SET "setPropertyClause" ];

alterFieldClauses = ADD "addFieldClauseForAlter" | MODIFY "modifyFieldClause"
| DELETE "deleteFieldClause";

modifyFieldClause = FIELD "QUOTED_STRING" ( "renameClause" [
"moveFieldToClause" ] [ SET "setPropertyClause" ] | "moveFieldToClause" [ SET
"setPropertyClause" ] | SET "setPropertyClause" );

deleteFieldClause = FIELD "QUOTED_STRING";

moveFieldToClause = MOVE TO POSITION "INTEGER_LITERAL";

```

## Keywords and Parameters

**alterFlatFileCommand**

Alter a flat file.

**QUOTED\_STRING**

The name of the flat file to alter.

**renameClause**

Rename the flat file to the following quoted string.

**setPropertyClause**

Set the properties of the flat file, record, or field.

**alterRecordClauses**

Add, modify, or drop a record of the flat file.

**propertyNameList**

The names of the properties whose values you want to set.

Properties for FLAT\_FILE:

Name: DATA\_FILE\_NAME

Type: STRING

Valid Values: N/A Default " The name of the "sampled" file. Also the default data file value used in SQL\*Loader maps and External Tables.

Name: IS\_DELIMITED

Type: BOOLEAN

Valid Values: true, false, 1, 0

Default: true

True indicates that this flat file is delimited. False indicates that its fields are defined by fixed lengths

Name: CHARACTERSET

Type: STRING

Valid Values:

AL24UTFSS,AR8ARABICMAC,AR8ARABICMACS,AR8ISO8859P6,AR8MSAWIN,A  
R8MSWIN1256,BLT8CP921,BLT8EBCDIC1112,BLT8MSWIN1257,BLT8PC775,CDN8PC  
863,CL8EBCDIC1025,CL8EBCDIC1025X,CL8ISO8859P5,CL8KOI8R,CL8MACCYRILLI  
C,CL8MACCYRILLICS,CL8MSWIN1251,D8EBCDIC273,DK8EBCDIC277,EE8EBCDIC  
870,EE8ISO8859P2,EE8MACCE,EE8MACCES,EE8MACCROATIAN,EE8MACCROATI  
ANS,EE8MSWIN1250,EE8PC852,EL8EBCDIC875,EL8ISO8859P7,EL8MACGREEK,EL8  
MACGREEKS,EL8MSWIN1253,EL8PC437S,EL8PC737,EL8PC869,F8EBCDIC297,I8EBC  
DIC280,IS8MACICELANDIC,IS8MACICELANDICS,IS8PC861,IW8EBCDIC424,IW8IS  
O8859P8,IW8MACHEBREW,IW8MACHEBREWS,IW8MSWIN1255,JA16EBCDIC930,J  
A16EUC,JA16EUCYEN,JA16MACSJIS,JA16SJIS,JA16SJISYEN,JA16VMS,KO16KSC560  
1,LT8MSWIN921,N8PC865,NEE8ISO8859P4,RU8PC855,RU8PC866,S8EBCDIC278,SE8I  
SO8859P3,TH8MACTHAI,TH8MACTHAIS,TH8TISASCII,TR8EBCDIC1026,TR8MAC  
TURKISH,TR8MACTURKISHS,TR8MSWIN1254,TR8PC857,US7ASCII,US8PC437,UTF  
8,WE8EBCDIC284,WE8EBCDIC285,WE8EBCDIC37,WE8EBCDIC37C,WE8EBCDIC500  
,WE8EBCDIC500C,WE8EBCDIC871,WE8ISO8859P1,WE8ISO8859P9,WE8MACROMA  
N8,WE8MACROMAN8S,WE8MSWIN1252,WE8PC850,WE8PC860,ZHS16CGB231280,  
ZHS16GBK,ZHS16MACCGB231280,ZHT16BIG5,ZHT16MSWIN950,ZHT32EUC

Default: WE8MSWIN1252

The character set of the data file.

Name: RECORD\_DELIMITER

Type: STRING

Valid Values: N/A

Default: "



---

The character(s) which denote the end of a physical record in a data file. (Please note that this is not the FIELD\_DELIMITER.

Name: RECORD\_LENGTH

Type: NUMBER

Valid Values: 0+

Default: 0 (Records are delimited by default)

The length (in characters) of the records in the data file.

Name: RECORD\_TYPE\_COLUMN\_NUMBER

Type: NUMBER

Valid Values: 0+

Default: 1

The column which contains the record type values for a delimited, multi-record type file.

Name: RECORD\_TYPE\_START\_POSITION

Type: NUMBER

Valid Values: 0+

Default: 0

The starting position of the field (relative to 1) which contains the record type values for a fixed-length, multi-record type file.

Name: RECORD\_TYPE\_END\_POSITION

Type: NUMBER

Valid Values: 0+

Default: 0

The ending position of the field (relative to 1) which contains the record type values for a fixed-length, multi-record type file.

Name: NUMBER\_OF\_RECORDS\_TO\_SKIP

Type: NUMBER

Valid Values: 0 - 1000

Default: 0

The default number of records to skip when loading this file.

Name: FIELD\_DELIMITER

Type: STRING

Valid Values: Any single character

Default: ',' (Comma)

The character to divide the fields in a delimited file.

Name: FIELD\_LEFT\_ENCLOSURE

Type: STRING

Valid Values: Any single character

Default: None

A character to enclose fields which may contain the field delimiter.

Name: FIELD\_RIGHT\_ENCLOSURE

Type: STRING

Valid Values: Any single character

Default: None

A character to enclose fields which may contain the field delimiter.

Name: NUMBER\_OF\_PHYSICAL\_RECORDS\_PER\_LOGICAL

Type: Number

Valid Values: 0+

Default: 0

Set this value if you wish to concatenate a fixed number of physical records to form a single logical record.

Name: CONTINUE\_IF\_ENDS\_WITH

Type: STRING

Valid Values: Any single character

Default: None

Set this value if you wish to concatenate a variable number of physical records to form a single logical record, determined by records ending with this character.

Name: CONTINUE\_IF\_STARTS\_WITH

Type: STRING

Valid Values: N/A

Default: None

Set this value if you wish to concatenate a variable number of physical records to form a single logical record, determined by records beginning with this character.

Properties for RECORD:

Name: RECORD\_TYPE\_VALUE

Type: STRING

Valid Values: N/A

Default: None

This is a mandatory property for each record of a multi-record type file. It is the string which will identify this record type in the data file.

Properties for FIELD:

Name: DATATYPE

Type: STRING

Valid Values: CHAR, DATE, DECIMAL EXTERNAL, FLOAT EXTERNAL, INTEGER EXTERNAL, ZONED EXTERNAL, ZONED

Default: CHAR

This is the SQL\*Loader data type for the field.

Name: LENGTH

Type: NUMBER

Valid Values: 1+

Default: 1

This is the length of the field in a fixed length file. This is the max length of the field in a delimited file.

Name: PRECISION

Type: NUMBER

Valid Values: Depends on data type

Default: 0

Precision of the field.

Name SCALE

Type: NUMBER

Valid Values: Depends on data type

Default: 0

Scale of the field

Name: START\_POSITION

Type: NUMBER

Valid Values: 0+

Default: 0

The starting position of a field for a fixed length file.

Name: SQL\_DATATYPE

Type: STRING

Valid Values: CHAR, DATE, FLOAT, NUMBER, VARCHAR, VARCHAR2, DEFAULT

Default: DEFAULT. This will derive the SQL\_DATATYPE from the value of

DATATYPE. The data type which the field will be treated as in mapping and for External Tables.

Name: SQL\_LENGTH

Type: NUMBER

Valid Values: for 'CHAR' : 1 - 2000 for 'VARCHAR' and 'VARCHAR2' : 1 - 4000

Default: Depends on Sql data type.

Name: SQL\_PRECISION

Type: NUMBER

Valid Values: 1 - 38

Default: 1

Name: SQL\_SCALE

Type: NUMBER

Valid Values: -84 - 127

Default: 0

Name: MASK

Type: STRING

Valid Values: N/A

Default: None

This is the mask used to define the format of DATE fields in the data file.

Name: NULL\_IF

Type: STRING

---

Valid Values: Either = or != followed by either 'BLANKS', a single quoted string, or a hexadecimal string

Default: None

If this condition is true for a field, the value loaded will be NULL.

Name: DEFAULT\_IF

Type: STRING

Valid Values: Either = or != followed by either 'BLANKS', a single quoted string, or a hexadecimal string

Default: None

If this condition is true for a field, the value loaded will be either NULL or 0, dependent on data type.

propertyValueList

The values for the named properties.

addRecordClauseForAlter

Add a record named by the following quoted string.

modifyRecordClause

Modify a record specified by the following quoted string.

deleteRecordClause

Delete a record specified by the following quoted string.

propertyValue

A property value.

addFieldClauseForAlter

Add a field to the record.

alterFieldClauses

Add, modify, or drop a field.

modifyFieldClause

Modify the properties of a field specified by the following quoted string.

deleteFieldClause

Delete a field specified by the following quoted string.

`moveFieldToClause`

Move a field to a new position in the record.

### Examples

```
OMBALTER FLAT_FILE 'OLD_NAME' RENAME TO 'NEW_NAME' SET
PROPERTIES (DESCRIPTION, BUSINESS_NAME) VALUES ('This becomes a target
file.', 'target file') This will rename the flat file "OLD_NAME" to "NEW_NAME", set its
description to "This becomes a target file", and set its business name to "target file".
```

### See Also

OMBALTER, OMBCREATE FLAT\_FILE, OMBDROP FLAT\_FILE

# OMBALTER FLAT\_FILE\_MODULE

## Purpose

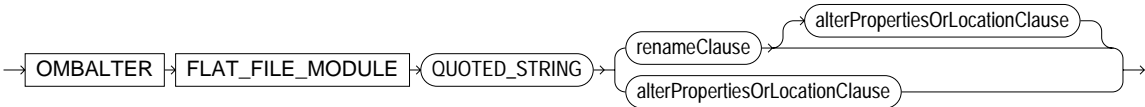
OMBALTER FLAT\_FILE\_MODULE - Alter the flat file module by renaming it, and/or resetting its properties.

## Prerequisites

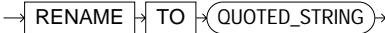
Should be in the context of a project.

## Syntax Diagrams

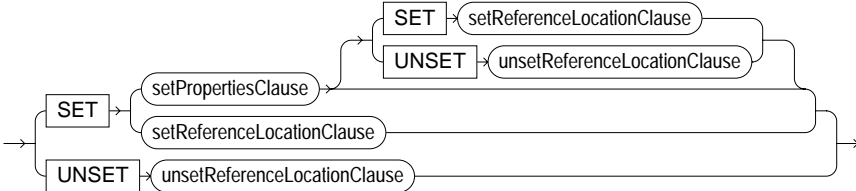
alterFlatFileModuleCommand



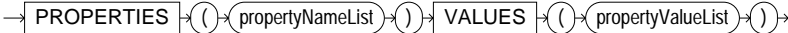
renameClause



alterPropertiesOrLocationClause



setPropertiesClause



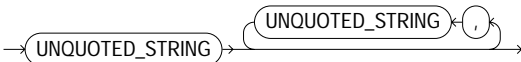
setReferenceLocationClause



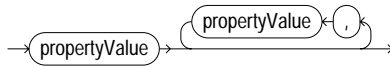
unsetReferenceLocationClause



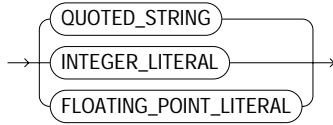
propertyNameList



propertyValueList



propertyValue



## Syntax

```
alterFlatFileModuleCommand = OMBALTER ( FLAT_FILE_MODULE "QUOTED_
STRING" ("renameClause" [ "alterPropertiesOrLocationClause" ] |
"alterPropertiesOrLocationClause" ) );
```

```
renameClause = RENAME TO "QUOTED_STRING";
```

```
alterPropertiesOrLocationClause = SET ( "setPropertiesClause" [ SET
"setReferenceLocationClause" | UNSET "unsetReferenceLocationClause" ] |
"setReferenceLocationClause" ) | UNSET "unsetReferenceLocationClause";
```

```
setPropertiesClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("
"propertyValueList" ")";
```

```
setReferenceLocationClause = ( REFERENCE | REF ) LOCATION "QUOTED_
STRING";
```

```
unsetReferenceLocationClause = ( REFERENCE | REF ) LOCATION "QUOTED_
STRING";
```

```
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
```

```
propertyValueList = "propertyValue" { "," "propertyValue" };
```

```
propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );
```

## Keywords and Parameters

**alterFlatFileModuleCommand**

Alter a flat file module.

**QUOTED\_STRING**

The name of the flat file module to alter.

**renameClause**

Rename the flat file module.

**QUOTED\_STRING**

The new name for the flat file module.



**alterPropertiesOrLocationClause**

Alter either the properties of the flat file module, the location of the flat file module, or both.

**setPropertyClause**

Set some properties of the flat file module.

**setReferenceLocationClause**

Set the location for the flat file module to the location specified by the quoted string.

**unsetReferenceLocationClause**

Unset the location for the flat file module.

**propertyNameList**

The names of the properties whose values you want to set.

**Basic properties for FLAT\_FILE\_MODULE:**

**Name:** BUSINESS\_NAME

**Type:** STRING(200)

**Valid Values:** N/A

**Default:** NAME

Business name of the flat file module.

**Name:** DESCRIPTION

**Type:** STRING(4000)

**Valid Values:** N/A

**Default:** "

Description of the flat file module.

**Name:** UOID

**Type:** STRING(40)

**Valid Values:** N/A

**Default:** N/A

UOID of the flat file module.

propertyValueList

The values for the named properties.

propertyValue

A property value.

## Examples

```
OMBALTER FLAT_FILE_MODULE 'src_module' RENAME TO 'tgt_module' SET
PROPERTIES (DESCRIPTION, BUSINESS_NAME) VALUES ('This becomes a target
module.', 'target module') This will rename the flat file module "src_module" to "tgt_
module", and set its description to "This becomes a target module", set its business
name to "target module".
```

## See Also

OMBALTER, OMBCREATE FLAT\_FILE\_MODULE, OMBDROP FLAT\_FILE\_
MODULE

# OMBALTER FUNCTION

## Purpose

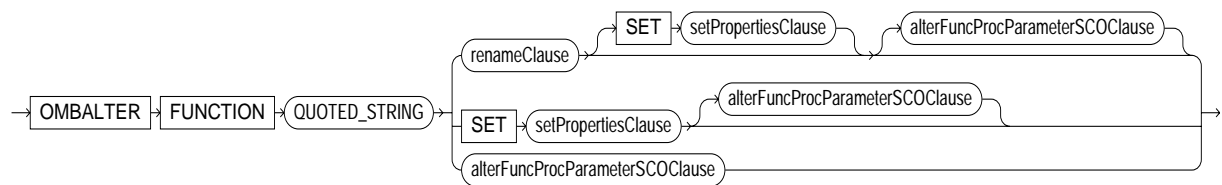
OMBALTER FUNCTION - Alter the Function by renaming it, and/or reset its properties.

## Prerequisites

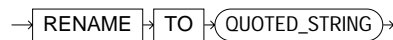
Should be in the context of a Oracle Module or Package or Transformation Module.

## Syntax Diagrams

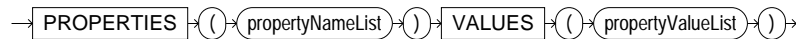
alterFunctionCommand



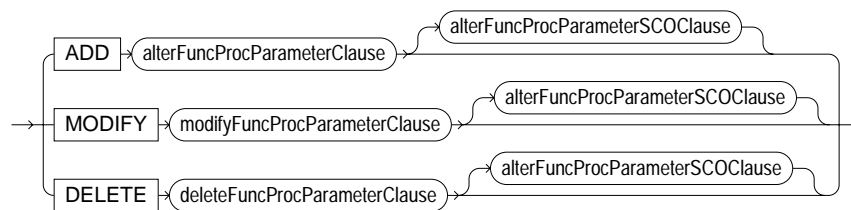
renameClause



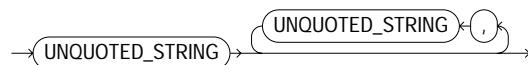
setPropertiesClause



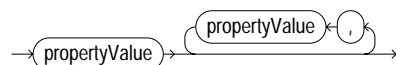
alterFuncProcParameterSCOClause



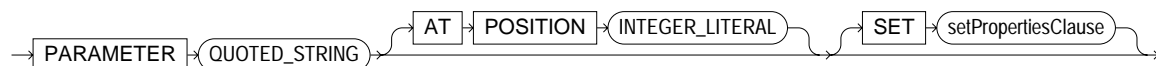
propertyNameList



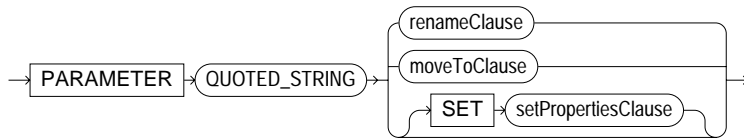
propertyValueList



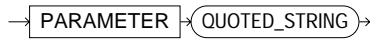
alterFuncProcParameterClause



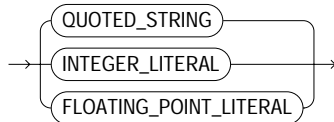
modifyFuncProcParameterClause



deleteFuncProcParameterClause



propertyValue



moveToClause



## Syntax

```

alterFunctionCommand = OMBALTER ( FUNCTION "QUOTED_STRING" (
"renameClause" [ SET "setPropertiesClause" ] [ "alterFuncProcParameterSCOClause" ]
| SET "setPropertiesClause" [ "alterFuncProcParameterSCOClause" ] |
"alterFuncProcParameterSCOClause" ));

renameClause = RENAME TO "QUOTED_STRING";

setPropertiesClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("
"propertyValueList" ")";

alterFuncProcParameterSCOClause = ( ADD "alterFuncProcParameterClause" [
"alterFuncProcParameterSCOClause" ] | MODIFY "modifyFuncProcParameterClause"
[ "alterFuncProcParameterSCOClause" ] | DELETE "deleteFuncProcParameterClause"
[ "alterFuncProcParameterSCOClause" ] );

propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };

propertyValueList = "propertyValue" { "," "propertyValue" };

alterFuncProcParameterClause = PARAMETER "QUOTED_STRING" [ AT POSITION
"INTEGER_LITERAL" ] [ SET "setPropertiesClause" ];

modifyFuncProcParameterClause = ( PARAMETER "QUOTED_STRING" (
"renameClause" | "moveToClause" | [ SET "setPropertiesClause" ] ));

deleteFuncProcParameterClause = ( PARAMETER "QUOTED_STRING" );

propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );

moveToClause = MOVE TO POSITION "INTEGER_LITERAL";
  
```

## Keywords and Parameters

alterFunctionCommand

This command modifies an existing Function.

**QUOTED\_STRING**

Name of the existing Function in single quotes.

**renameClause**

Rename a Function.

**setPropertyClause**

Used to set properties (core, user-defined) for function. Valid properties are shown below:

Basic properties for FUNCTION:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the Function

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the Function

Name: RETURN\_TYPE

Type: STRING

Valid Values: PLS\_INTEGER, BINARY\_INTEGER, BOOLEAN, NUMBER, FLOAT, CHAR, VARCHAR, VARCHAR2, DATE

Default: NUMBER

Set the Return Type for Function

Name: IMPLEMENTATION

Type: STRING

Valid Values: N/A

Default: "

Set the code for Function which is included global variable declaration and code between BEGIN and END.

Name: IS\_DETERMINISTIC

Type: BOOLEAN

Valid Values: true, false

Default: false

This setting helps the optimizer avoid redundant function calls.

Name: IS\_PARALLEL\_ENABLE

Type: BOOLEAN

Valid Values: true, false

Default: false

This option sets flag to a stored function can be used safely in the slave sessions of parallel DML evaluations.

Basic properties for PARAMETER:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the Parameter

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the Parameter

Name: DATATYPE

Type: STRING

Valid Values: PLS\_INTEGER, BINARY\_INTEGER, BOOLEAN, NUMBER, FLOAT, CHAR, VARCHAR, VARCHAR2, DATE

Default: NUMBER

Set the data type for Parameter

Name: IN\_OUT

Type: STRING

Valid Values: IN, OUT, INOUT

Default: 'IN'

Set the parameter mode for Parameter

Name: DEFAULT\_VALUE

Type: STRING

Valid Values: N/A

Default: "

Set the default value for Parameter

Properties for FUNCTION:

Name: AUTHID

Type: STRING

Valid Values: None, Current\_User, Definer

Default: None

Generate the transformation with selected AUTHID option. Function will be executed with the permissions defined by the AUTHID clause rather than the function owner's permissions.

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Note:

1. N/A means any valid character in supported character set.
2. " represents an empty string

alterFuncProcParameterSCOClause

Second class object clause to modify, delete or add a Parameter for Function/Procedure.

propertyNameList

Comma separated list of property names. Property names are unquoted.

propertyValueList

Comma separated list of property values.

`alterFuncProcParameterClause`

This clause alters Parameter of a Function.

`modifyFuncProcParameterClause`

Modify one or more Parameters to this Function/Procedure.

`deleteFuncProcParameterClause`

Delete one or more Parameters to this Function/Procedure.

`propertyValue`

Value of a property.

`moveToClause`

Move a Parameters of this Function/Procedure.

## Examples

OMBALTER FUNCTION 'func' RENAME TO 'function\_1' SET PROPERTIES (DESCRIPTION, BUSINESS\_NAME) VALUES ('This becomes a function\_1', 'function\_1') This will rename the Function "func" to "function\_1", and set its description to "This becomes a function\_1", set its business name to "function\_1" If Packaged Function is overloaded, first find the Signature by using OMBLIST command, and then use OMBALTER command using appropriate signature. Example, if OMBLIST FUNCTIONS gives following two signatures, FUNC\_1 (NUMBER) RETURN NUMBER FUNC\_1 (VARCHAR2, NUMBER) RETURN NUMBER The OMBALTER syntax to modify the first one will be as follows OMBALTER FUNCTION 'FUNC\_1 \ (NUMBER\ ) RETURN NUMBER' SET PROPERTIES (DESCRIPTION, BUSINESS\_NAME) VALUES ('descri\_FUNC\_1', 'FUNC\_1')

## See Also

OMBALTER, OMBCREATE FUNCTION, OMBDROP FUNCTION



## OMBALTER LOCATION

### Purpose

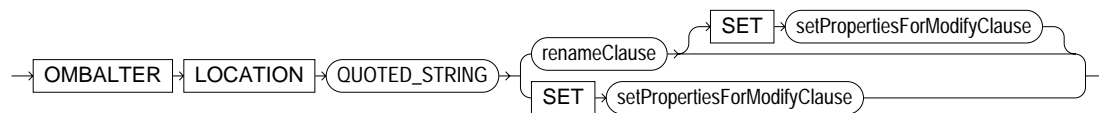
OMBALTER LOCATION - Alter the location by renaming it, and/or reset its properties.

### Prerequisites

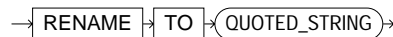
Should be in the context of a project.

### Syntax Diagrams

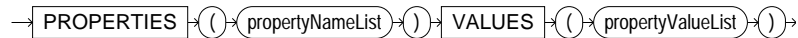
alterLocationCommand



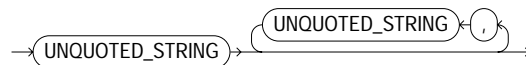
renameClause



setPropertyForModifyClause



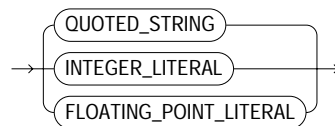
propertyNameList



propertyValueList



propertyValue



### Syntax

```
alterLocationCommand = OMBALTER ( LOCATION "QUOTED_STRING" (
  "renameClause" [ SET "setPropertyForModifyClause" ] | SET
  "setPropertyForModifyClause" ) );
```

```
renameClause = RENAME TO "QUOTED_STRING";
```

```

setPropertyForModifyClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("
"propertyValueList" ")";
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
propertyValueList = "propertyValue" { "," "propertyValue" };
propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );

```

## Keywords and Parameters

**alterLocationCommand**

Alter the location specified by the quoted string.

**renameClause**

Rename the location to the value of the following quoted string.

**setPropertyForModifyClause**

Set or alter specified properties of the location.

**propertyNameList**

The names of the properties whose values you want to set.

Properties for LOCATION:

Name: TYPE

Type: STRING

Valid Values: 'Oracle Gateway', 'File System', 'Oracle Database', 'OEM Agent', 'Oracle Workflow', 'SAP'

Default: N/A

The type of system the location represents.

Name: VERSION

Type: STRING

Valid Values:

for 'Oracle Gateway' : '0'

for 'File System' : '0'

for 'Oracle Database' : '9.2','9.0','8.1','8.0','7.3.4'

for 'OEM Agent' : '9.2','9.0'

for 'Oracle Workflow' : '2.6'

for 'SAP' : '4.x','3.x'

Default: N/A

---

The version of the system(s) the location represents.

Basic properties for LOCATION:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the location.

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the location.

Note:

1. N/A means any valid character in supported character set.
2. " represents an empty string

propertyValueList

The values for the named properties.

propertyValue

A property value.

## Examples

```
OMBALTER LOCATION 'OLD_LOCATION' RENAME TO 'NEW_LOCATION' SET
PROPERTIES (DESCRIPTION, BUSINESS_NAME) VALUES ('This becomes a new
location.', 'new location') This will rename the location "OLD_LOCATION" to "NEW_
LOCATION", and set its description to "This becomes a new location", set its business
name to "new location".
```

## See Also

OMBALTER, OMBCREATE LOCATION, OMBDROP LOCATION

# OMBALTER MAPPING

## Purpose

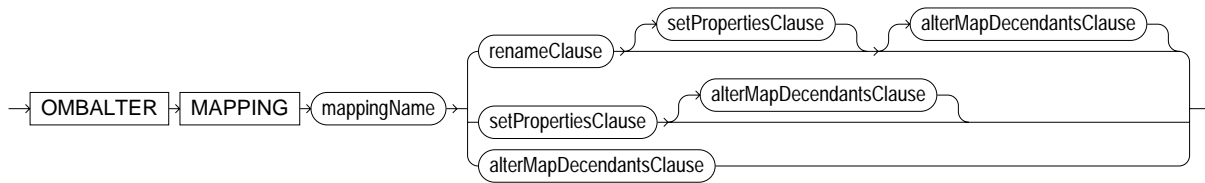
OMBALTER MAPPING - Alter the content of a mapping.

## Prerequisites

1. The current context of scripting must be an Oracle Module
2. No concurrent user should be modifying the mapping

## Syntax Diagrams

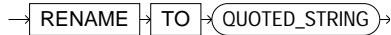
alterMappingCommand



mappingName



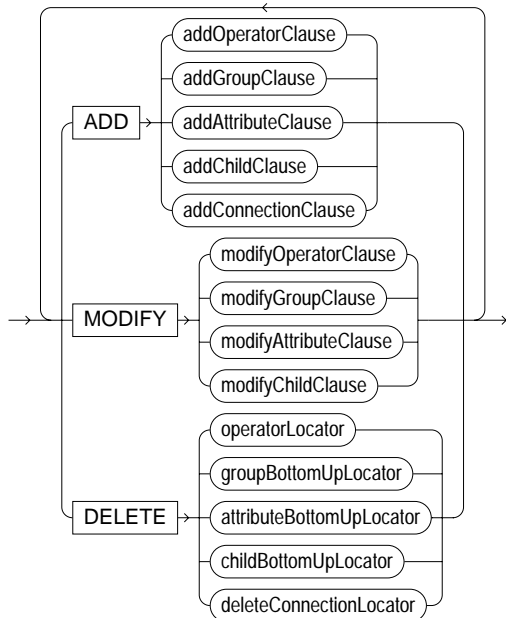
renameClause



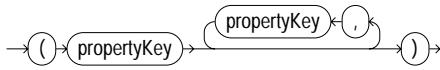
setPropertiesClause



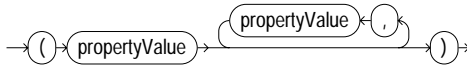
alterMapDecendantsClause



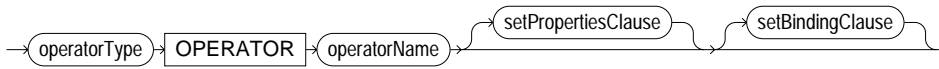
propertyKeyList



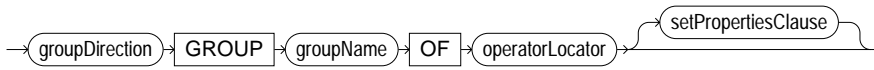
propertyValueList



addOperatorClause



addGroupClause



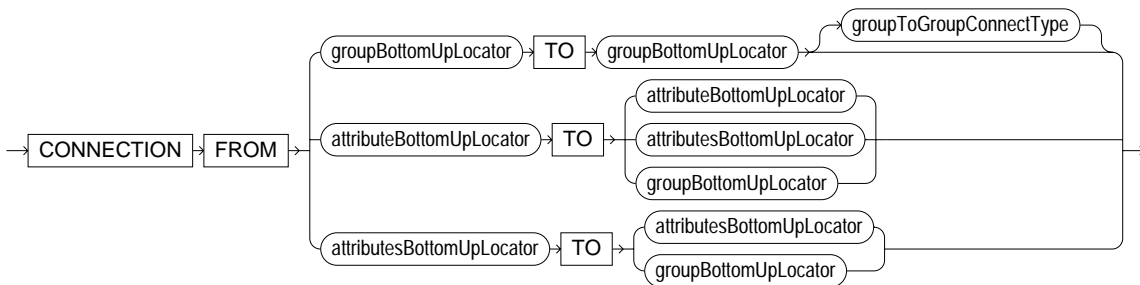
addAttributeClause



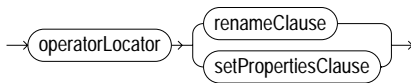
addChildClause



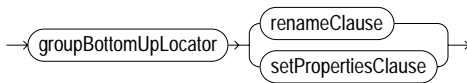
addConnectionClause



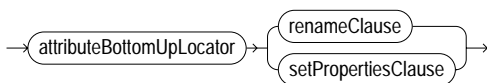
modifyOperatorClause



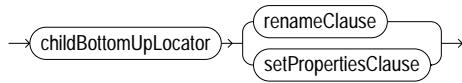
modifyGroupClause



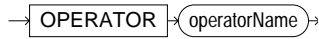
modifyAttributeClause



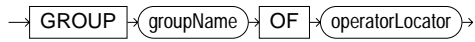
modifyChildClause



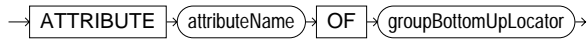
operatorLocator



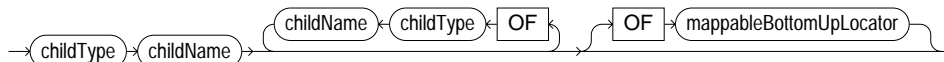
groupBottomUpLocator



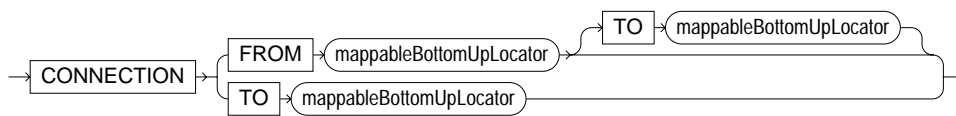
attributeBottomUpLocator



childBottomUpLocator



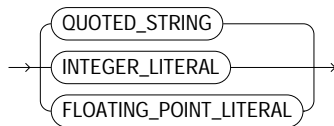
deleteConnectionLocator



propertyKey



propertyValue



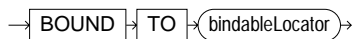
operatorType



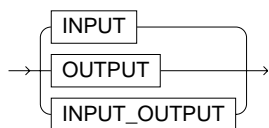
operatorName



setBindingClause



groupDirection



groupName



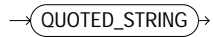
attributeName



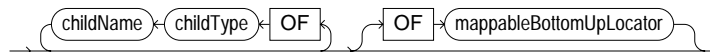
childType



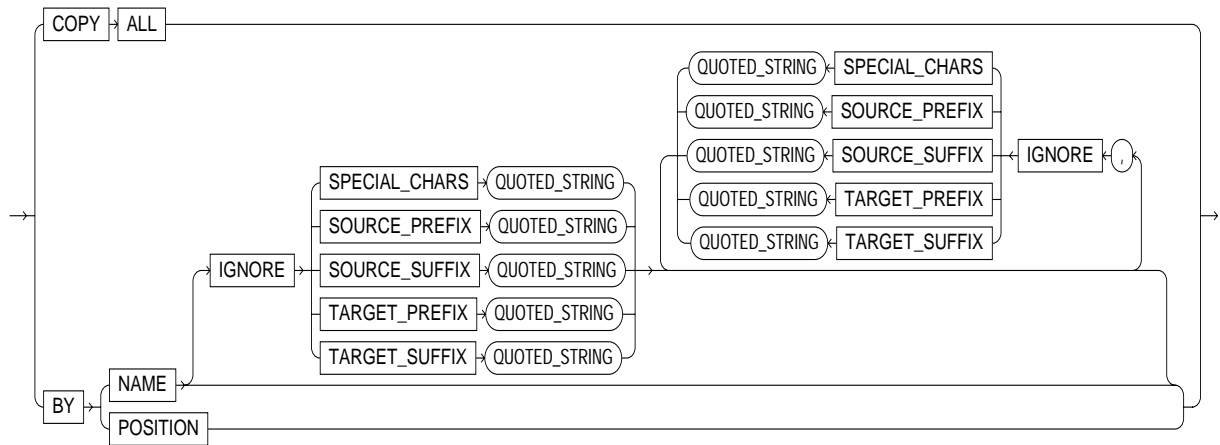
childName



childOwnerBottomUpLocator



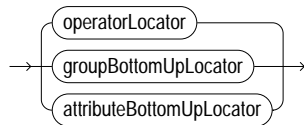
groupToGroupConnectType



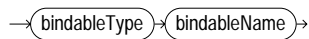
attributesBottomUpLocator



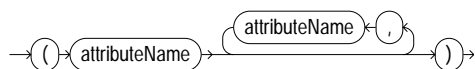
mappableBottomUpLocator



bindableLocator



attributeNameList



bindableType

→ UNQUOTED\_STRING →

bindableName

→ QUOTED\_STRING →

## Syntax

```

alterMappingCommand = OMBALTER MAPPING "mappingName" ( "renameClause"
[ "setPropertiesClause" ] [ "alterMapDecendantsClause" ] | "setPropertiesClause" [
"alterMapDecendantsClause" ] | "alterMapDecendantsClause" );

mappingName = "QUOTED_STRING";

renameClause = RENAME TO "QUOTED_STRING";

setPropertiesClause = SET PROPERTIES "propertyKeyList" VALUES
"propertyValueList";

alterMapDecendantsClause = ( ADD ( "addOperatorClause" | "addGroupClause" |
"addAttributeClause" | "addChildClause" | "addConnectionClause" ) | MODIFY (
"modifyOperatorClause" | "modifyGroupClause" | "modifyAttributeClause" |
"modifyChildClause" ) | DELETE ( "operatorLocator" | "groupBottomUpLocator" |
"attributeBottomUpLocator" | "childBottomUpLocator" | "deleteConnectionLocator" )
)+;

propertyKeyList = "(" "propertyKey" { "," "propertyKey" } ")";

propertyValueList = "(" "propertyValue" { "," "propertyValue" } ")";

addOperatorClause = "operatorType" OPERATOR "operatorName" [
"setPropertiesClause" ] [ "setBindingClause" ];

addGroupClause = "groupDirection" GROUP "groupName" OF "operatorLocator" [
"setPropertiesClause" ];

addAttributeClause = ATTRIBUTE "attributeName" OF "groupBottomUpLocator" [
"setPropertiesClause" ];

addChildClause = "childType" "childName" "childOwnerBottomUpLocator" [
"setPropertiesClause" ];

addConnectionClause = CONNECTION FROM ( "groupBottomUpLocator" TO
"groupBottomUpLocator" [ "groupToGroupConnectType" ] |
"attributeBottomUpLocator" TO ( "attributeBottomUpLocator" |
"attributesBottomUpLocator" | "groupBottomUpLocator" ) |
"attributesBottomUpLocator" TO ( "attributesBottomUpLocator" |
"groupBottomUpLocator" ) );

modifyOperatorClause = "operatorLocator" ( "renameClause" | "setPropertiesClause"
);

modifyGroupClause = "groupBottomUpLocator" ( "renameClause" |
"setPropertiesClause" );

modifyAttributeClause = "attributeBottomUpLocator" ( "renameClause" |
"setPropertiesClause" );

```



```

modifyChildClause = "childBottomUpLocator" ( "renameClause" |
"setPropertyClause" );
operatorLocator = OPERATOR "operatorName";
groupBottomUpLocator = GROUP "groupName" OF "operatorLocator";
attributeBottomUpLocator = ATTRIBUTE "attributeName" OF
"groupBottomUpLocator";
childBottomUpLocator = "childType" "childName" { OF "childType" "childName" } [
OF "mappableBottomUpLocator" ];
deleteConnectionLocator = CONNECTION ( FROM "mappableBottomUpLocator" [
TO "mappableBottomUpLocator" ] | TO "mappableBottomUpLocator" );
propertyKey = "UNQUOTED_STRING";
propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );
operatorType = "UNQUOTED_STRING";
operatorName = "QUOTED_STRING";
setBindingClause = BOUND TO "bindableLocator";
groupDirection = INPUT | OUTPUT | INPUT_OUTPUT;
groupName = "QUOTED_STRING";
attributeName = "QUOTED_STRING";
childType = "UNQUOTED_STRING";
childName = "QUOTED_STRING";
childOwnerBottomUpLocator = { OF "childType" "childName" } [ OF
"mappableBottomUpLocator" ];
groupToGroupConnectType = COPY ALL | BY ( NAME [ IGNORE ( SPECIAL_
CHARS "QUOTED_STRING" | SOURCE_PREFIX "QUOTED_STRING" | SOURCE_
SUFFIX "QUOTED_STRING" | TARGET_PREFIX "QUOTED_STRING" | TARGET_
SUFFIX "QUOTED_STRING" ) { ", " IGNORE ( SPECIAL_CHARS "QUOTED_STRING"
| SOURCE_PREFIX "QUOTED_STRING" | SOURCE_SUFFIX "QUOTED_STRING" |
TARGET_PREFIX "QUOTED_STRING" | TARGET_SUFFIX "QUOTED_STRING" ) } ]
| POSITION );
attributesBottomUpLocator = ATTRIBUTES "attributeNameList" OF
"groupBottomUpLocator";
mappableBottomUpLocator = "operatorLocator" | "groupBottomUpLocator" |
"attributeBottomUpLocator";
bindableLocator = "bindableType" "bindableName";
attributeNameList = "(" "attributeName" { ", " "attributeName" } ")";
bindableType = "UNQUOTED_STRING";
bindableName = "QUOTED_STRING";

```

## Keywords and Parameters

alterMappingCommand

Alter the content of a mapping.

mappingName

Name of the mapping.

renameClause

Rename a mapping, mapping operator, mapping group, or mapping attribute.

setPropertyClause

Describe the keys of properties for the map or objects in the map.

alterMapDecendantsClause

Alter a map by adding, modifying or deleting its descendants.

propertyKeyList

The list of property keys

propertyValueList

A list of property values.

addOperatorClause

Adds a mapping operator to a map. When you add an operator, Warehouse Builder creates default groups and parameters for the operator. Please see the appendix section of the Scripting Reference.

The following is an example for creating a child object under a mapping (which is not an operator) OMBALTER MAPPING 'M1' ADD SOURCE\_DATA\_FILE 'FILE1'

The following is an example for creating an operator:

```
OMBALTER MAPPING 'M1' ADD TABLE OPERATOR 'T1'
```

In the second example, when user forgets to type "OPERATOR" "GROUP" "ATTRIBUTE" key word, instead of complaining the keywords are missing, OMBPlus will complain about error getting child objects. Here is an example: OMB+> OMBALTER MAPPING 'M1' ADD TABLE 'T1' OMB02932: Error getting child objects of type TABLE in M1

TO A USER: it looks like OMBPlus should complain they forget to type a keyword.

TO OMBPLUS: the syntax is actually for creating a non-operator child object

under the mapping. Therefore, it goes and tries to find type definition for non-operator child object "TABLE" and cannot find it. Therefore the exception is thrown.

#### addGroupClause

Add a mapping group to a mapping operator.

#### addAttributeClause

Add a mapping attribute to a mapping group.

#### addChildClause

Add a child to a mapping, mapping operator, mapping group or mapping attribute.

#### addConnectionClause

Add connections between mapping groups or mapping attributes.

#### modifyOperatorClause

Modify a mapping operator.

#### modifyGroupClause

Modify a mapping group.

#### modifyAttributeClause

Modify a mapping attribute.

#### modifyChildClause

Modify a child that belongs to a mapping, mapping operator, mapping group or mapping attribute.

#### operatorLocator

Location of a mapping operator.

#### groupBottomUpLocator

Location of a mapping group.

#### attributeBottomUpLocator

Location of a mapping attribute.

**childBottomUpLocator**

Location of the child that belongs to a map, mapping operator, mapping group or mapping attribute.

**deleteConnectionLocator**

Delete connections between mapping operators, mapping groups or mapping attributes.

**propertyKey**

A property key for an object.

**Basic properties for MAPPING:**

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the mapping

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the mapping

**Basic properties for OPERATOR:**

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the operator

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the operator

**Basic properties for GROUP:****Name:** BUSINESS\_NAME**Type:** STRING(200)**Valid Values:** N/A**Default:** "**Business name of the group****Name:** DESCRIPTION**Type:** STRING(4000)**Valid Values:** N/A**Default:** "**Description of the group****Basic properties for ATTRIBUTE:****Name:** BUSINESS\_NAME**Type:** STRING(200)**Valid Values:** N/A**Default:** "**Business name of the attribute****Name:** DESCRIPTION**Type:** STRING(4000)**Valid Values:** N/A**Default:** "**Description of the attribute****Name:** DATATYPE**Type:** STRING(20)**Valid Values:** NUMBER, VARCHAR2, VARCHAR, DATE, LONG**Default:** "**Datatype of the Attribute****Name:** LENGTH**Type:** NUMBER

Valid Values: N/A

Default: 0

Length of the attribute.

Name: PRECISION

Type: NUMBER

Valid Values: N/A

Default: 0

Precision of the attribute.

Name: SCALE

Type: NUMBER

Valid Values: N/A

Default: 0

Scale of the attribute.

Properties for MAPPING:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: STEP\_TYPE

Type: STRING

Valid Values: ABAP, SQLLOADER, PLSQL

Default: UNKNOWN

The step type used to generate this mapping

Properties for MAPPING ABAP STEP:

Name: CONTROL\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: owb.dat

---

**Control File Name**

Name: DATA\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: owb.dat

Data File Name

Name: FILE\_DELIMITER\_FOR\_STAGING\_FILE

Type: STRING

Valid Values: N/A

Default: ~

File Delimiter for Staging File

Name: LOG\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: owb.log

Log File Name

Name: NESTED\_LOOP

Type: STRING

Valid Values: TRUE\_DEFAULT, TRUE, FALSE

Default: TRUE\_DEFAULT

Nested Loop

Name: PRIMARY\_FOREIGN\_KEY\_FOR\_JOIN

Type: STRING

Valid Values: N/A

Default: DEFAULT

Primary Foreign Key for Join

Name: SAP\_SYSTEM\_VERSION

Type: STRING

Valid Values: SAP\_R3\_4X, SAP\_R3\_3X

Default: SAP\_R3\_4X

SAP System Version

Name: SQL\_JOIN\_COLLAPSING  
Type: STRING  
Valid Values: TRUE\_DEFAULT, TRUE, FALSE  
Default: TRUE\_DEFAULT  
Sql Join Collapsing

Name: STAGING\_FILE\_DIRECTORY  
Type: STRING  
Valid Values: N/A  
Default: C:\temp\  
Staging File Directory

Name: USE\_SELECT\_SINGLE  
Type: STRING  
Valid Values: TRUE\_DEFAULT, TRUE, FALSE  
Default: TRUE\_DEFAULT  
Use Select Single

Properties for MAPPING PLSQL STEP:

Name: ANALYZE\_TABLE\_SAMPLE\_PERCENTAGE  
Type: NUMBER  
Valid Values: N/A  
Default: 90

The default percentage of rows to be sampled when the target tables are analyzed for statistics to improve performance during insertion.

Name: ANALYZE\_TABLE\_STATEMENTS  
Type: BOOLEAN  
Valid Values: true, false  
Default: true  
Generate statistics collection statement if this is true.

Name: BULK\_PROCESSING\_CODE  
Type: BOOLEAN  
Valid Values: true, false



Default: true

Generate bulk processing code if this is true.

Name: BULK\_SIZE

Type: NUMBER

Valid Values: N/A

Default: 50

The default number of rows to be fetched in batch during cursor processing.

Name: COMMIT\_FREQUENCY

Type: NUMBER

Valid Values: N/A

Default: 1000

The default number of rows processed before a commit is issued.

Name: CORRELATED\_COMMIT

Type: BOOLEAN

Valid Values: true, false

Default: false

The mapping commits or rolls back correlated rows together.

Name: DEFAULT\_AUDIT\_LEVEL

Type: STRING

Valid Values: NONE, STATISTICS, ERROR\_DETAILS, COMPLETE

Default: ERROR\_DETAILS

The default audit level when the step is executed.

Name: DEFAULT\_OPERATING\_MODE

Type: STRING

Valid Values: SET\_BASED, ROW\_BASED, ROW\_BASED\_TARGET\_ONLY, SET\_BASED\_FAIL\_OVER\_TO\_ROW\_BASED, SET\_BASED\_FAIL\_OVER\_TO\_ROW\_BASED\_TARGET\_ONLY

Default: SET\_BASED\_FAIL\_OVER\_TO\_ROW\_BASED

The default operating mode.

Name: DEFAULT\_PURGE\_GROUP

Type: STRING

Valid Values: N/A

Default: WB

The default purge group to be used when the step is executed.

Name: MAXIMUM\_NUMBER\_OF\_ERRORS

Type: NUMBER

Valid Values: N/A

Default: 50

The default maximum number of errors encountered before aborting the step execution.

Name: OPTIMIZED\_CODE

Type: BOOLEAN

Valid Values: true, false

Default: true

Attempt to generate optimized code if this is true.

Name: PARALLEL\_ROW\_CODE

Type: BOOLEAN

Valid Values: true, false

Default: false

Generate parallel row code if this is true.

Properties for MAPPING SQLLOADER STEP:

Name: AUDIT

Type: BOOLEAN

Valid Values: true, false

Default: true

Perform audit when the step is executed.

Name: BIND\_SIZE

Type: NUMBER

Valid Values: N/A

Default: 50000

Bind Size

Name: CONTINUE\_LOAD

Type: BOOLEAN  
Valid Values: true, false  
Default: false  
Continue Load

Name: CONTROL\_FILE\_LOCATION  
Type: STRING  
Valid Values: N/A  
Default: N/A  
The Control File Location

Name: CONTROL\_FILE\_NAME  
Type: STRING  
Valid Values: N/A  
Default: N/A  
The control file name used in TCL generation

Name: DATABASE\_FILE\_NAME  
Type: STRING  
Valid Values: N/A  
Default: N/A  
Database file name from which to allocate extents

Name: DEFAULT\_PURGE\_GROUP  
Type: STRING  
Valid Values: N/A  
Default: WB  
The default purge group to be used when the step is executed.

Name: DELIMITED\_FILE\_RECORD\_TERMINATION  
Type: STRING  
Valid Values: N/A  
Default: N/A  
This property has been deprecated. Please set the record delimiter in the Flat File Sample Wizard or Property Sheet.

Name: DIRECT\_MODE

Type: BOOLEAN

Valid Values: true, false

Default: true

Direct Mode

Name: ERRORS\_ALLOWED

Type: NUMBER

Valid Values: N/A

Default: 50

Number of errors to allow

Name: LOG\_FILE\_LOCATION

Type: STRING

Valid Values: N/A

Default: N/A

The log file location.

Name: LOG\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The log file name.

Name: NLS\_CHARACTERSET

Type: STRING

Valid Values: AL24UTFFSS, AR8ARABICMAC, AR8ARABICMACS, AR8ISO8859P6, AR8MSAWIN, AR8MSWIN1256, BLT8CP921, BLT8EBCDIC1112, BLT8MSWIN1257, BLT8PC775, CDN8PC863, CL8EBCDIC1025, CL8EBCDIC1025X, CL8ISO8859P5, CL8KOI8R, CL8MACCYRILLIC, CL8MACCYRILLICS, CL8MSWIN1251, D8EBCDIC273, DK8EBCDIC277, EE8EBCDIC870, EE8ISO8859P2, EE8MACCE, EE8MACCES, EE8MACCROATIAN, EE8MACCROATIANS, EE8MSWIN1250, EE8PC852, EL8EBCDIC875, EL8ISO8859P7, EL8MACGREEK, EL8MACGREEKS, EL8MSWIN1253, EL8PC437S, EL8PC737, EL8PC869, F8EBCDIC297, I8EBCDIC280, IS8MACICELANDIC, IS8MACICELANDICS, IS8PC861, IW8EBCDIC424, IW8ISO8859P8, IW8MACHEBREW, IW8MACHEBREWS, IW8MSWIN1255, JA16EBCDIC930, JA16EUC, JA16EUCYEN, JA16MACSJIS, JA16SJIS, JA16SJISYEN, JA16VMS, KO16KSC5601, LT8MSWIN921, N8PC865, NEE8ISO8859P4, RU8PC855, RU8PC866, S8EBCDIC278, SE8ISO8859P3, TH8MACTHAI, TH8MACTHAIS, TH8TISASCII, TR8EBCDIC1026, TR8MACTURKISH, TR8MACTURKISHS, TR8MSWIN1254, TR8PC857, US7ASCII, US8PC437, UTF8, WE8EBCDIC284, WE8EBCDIC285, WE8EBCDIC37, WE8EBCDIC37C, WE8EBCDIC500, WE8EBCDIC500C, WE8EBCDIC871, WE8ISO8859P1, WE8ISO8859P9, WE8MACROMAN8, WE8MACROMAN8S, WE8MSWIN1252, WE8PC850,

---

WE8PC860, ZHS16CGB231280, ZHS16GBK, ZHS16MACCGB231280, ZHT16BIG5,  
ZHT16MSWIN950, ZHT32EUC

Default: WE8MSWIN1252

Nls Characterset

Name: OPERATION\_RECOVERABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Operation Recoverable

Name: PERFORM\_PARALLEL\_LOAD

Type: BOOLEAN

Valid Values: true, false

Default: false

Perform Parallel Load

Name: PRESERVE\_BLANKS

Type: BOOLEAN

Valid Values: true, false

Default: false

Preserve Blanks

Name: READ\_BUFFERS

Type: NUMBER

Valid Values: N/A

Default: 4

The Number of Buffers

Name: READ\_SIZE

Type: NUMBER

Valid Values: N/A

Default: 65536

The size of the read buffer

Name: RECORDS\_TO\_LOAD

Type: NUMBER

Valid Values: N/A

Default: N/A

Number of logical records to load. The default value of 0 indicates to load all records.

Name: RECORDS\_TO\_SKIP

Type: NUMBER

Valid Values: N/A

Default: N/A

Number of logical records to skip

Name: ROWS\_PER\_COMMIT

Type: NUMBER

Valid Values: N/A

Default: 200

Rows per Commit

Properties for MAPPING SQLLOADER STEP SOURCE\_DATA\_FILE CHILD:

Name: BAD\_FILE\_LOCATION

Type: STRING

Valid Values: N/A

Default: N/A

Bad file location.

Name: BAD\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Bad file name. If the Bad File Location is set then this must be a relative file name. Otherwise this should contain a fully qualified path.

Name: DATA\_FILE\_LOCATION

Type: STRING

Valid Values: N/A

Default: N/A

The location for this component.

Name: DATA\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The data file name for this component. If the Data File Location is set then this must be a relative file name. Otherwise this should contain a fully qualified path.

Name: DISCARD\_FILE\_LOCATION

Type: STRING

Valid Values: N/A

Default: N/A

Discard file location

Name: DISCARD\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Discard file name. If the Discard File Location is set then this must be a relative file name. Otherwise this should contain a fully qualified path.

Name: DISCARD\_MAX

Type: NUMBER

Valid Values: N/A

Default: 0

Discard Max

Properties for ADVANCED\_QUEUE OPERATOR:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Properties for AGGREGATOR OPERATOR:

Name: GROUP\_BY\_CLAUSE

Type: STRING

Valid Values: N/A

Default: N/A

The Group By clause for the aggregation

Name: HAVING\_CLAUSE

Type: STRING

Valid Values: N/A

Default: N/A

The Having clause for the aggregation

Properties for CUBE OPERATOR:

Name: ADVANCED\_MATCH\_BY\_CONSTRAINT

Type: STRING

Valid Values: N/A

Default: No constraints

This property is used by public API and scripting to influence how columns are used for UPDATE or DELETE DMLs. If this property is set with the name of a primary or unique key, all the columns in the key will be used for matching during UPDATE or DELETE; and all the columns not in the key are used for loading. The property can also be assigned the value "All constraints" or "No constraints". If the DML type is INSERT, TRUNCATE/INSERT, or CHECK/INSERT, setting this property causes no effect.

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DATABASE\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Database file name to allocate extents from

Name: DATABASE\_LINK



Type: STRING

Valid Values: N/A

Default: N/A

The database link used to access this entity during mapping.

Name: DATA\_COLLECTION\_FREQUENCY

Type: STRING

Valid Values: UNKNOWN, YEAR, QUARTER, MONTH, DAY, HOUR, MINUTE

Default: UNKNOWN

New Data Granularity

Name: DIRECT

Type: BOOLEAN

Valid Values: true, false

Default: false

Directly swap source into target as a partition without first creating a staging table.

Name: ENABLE\_CONSTRAINTS

Type: BOOLEAN

Valid Values: true, false

Default: true

Enable Constraints

Name: EXCEPTIONS\_TABLE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Exceptions Table Name

Name: EXTRACTION\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when extracting from this table using SQL

Name: LOADING\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when loading into this table using SQL

Name: LOADING\_TYPE

Type: STRING

Valid Values: INSERT, UPDATE, INSERT\_UPDATE, UPDATE\_INSERT, DELETE, NONE, TRUNCATE\_INSERT, DELETE\_INSERT, CHECK\_INSERT

Default: INSERT

The loading operation to be performed when this is a target.

Name: MATCH\_BY\_CONSTRAINT

Type: STRING

Valid Values: N/A

Default: YES

A property to indicate whether unique or primary key information on this target will override the matching criteria obtained from Match by constraint property on the attributes of this target.

Name: PARTITION\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Partition Name

Name: PEL\_ENABLED

Type: BOOLEAN

Valid Values: true, false

Default: false

PEL Enabled

Name: PRIMARY\_SOURCE

Type: STRING

Valid Values: YES, NO

Default: NO

A boolean value to indicate whether this is a primary source or not (only used in EDW).

Name: RECORDS\_TO\_SKIP

Type: NUMBER

Valid Values: N/A

Default: N/A

Number of records to skip

Name: REPLACE\_DATA

Type: BOOLEAN

Valid Values: true, false

Default: false

Replace existing data in target partition if there is any.

Name: SCHEMA

Type: STRING

Valid Values: N/A

Default: N/A

Schema

Name: SINGLEROW

Type: BOOLEAN

Valid Values: true, false

Default: false

Singlerow

Name: SORTED\_INDEXES\_CLAUSE

Type: STRING

Valid Values: N/A

Default: N/A

Sorted Indexes Clause

Name: SUBPARTITION\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Subpartition Name

Name: TARGET\_FILTER\_FOR\_DELETE

Type: STRING

Valid Values: N/A

Default: N/A

A condition on the rows in the target and if evaluated to true, that row will participate in the delete loading operation.

Name: TARGET\_FILTER\_FOR\_UPDATE

Type: STRING

Valid Values: N/A

Default: N/A

A condition on the rows in the target and if evaluated to true, that row will participate in the update loading operation.

Name: TRAILING\_NULLCOLS

Type: BOOLEAN

Valid Values: true, false

Default: false

Trailing Nullcols

Properties for DIMENSION OPERATOR:

Name: ADVANCED\_MATCH\_BY\_CONSTRAINT

Type: STRING

Valid Values: N/A

Default: No constraints

This property is used by public API and scripting to influence how columns are used for UPDATE or DELETE DMLs. If this property is set with the name of a primary or unique key, all the columns in the key will be used for matching during UPDATE or DELETE; and all the columns not in the key are used for loading. The property can also be assigned the value "All constraints" or "No constraints". If the DML type is INSERT, TRUNCATE/INSERT, or CHECK/INSERT, setting this property causes no effect.

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DATABASE\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Database file name to allocate extents from

Name: DATABASE\_LINK

Type: STRING

Valid Values: N/A

Default: N/A

The database link used to access this entity during mapping.

Name: DATA\_COLLECTION\_FREQUENCY

Type: STRING

Valid Values: UNKNOWN, YEAR, QUARTER, MONTH, DAY, HOUR, MINUTE

Default: UNKNOWN

New Data Granularity

Name: DIRECT

Type: BOOLEAN

Valid Values: true, false

Default: false

Directly swap source into target as a partition without first creating a staging table.

Name: ENABLE\_CONSTRAINTS

Type: BOOLEAN

Valid Values: true, false

Default: true

Enable Constraints

Name: EXCEPTIONS\_TABLE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Exceptions Table Name

Name: EXTRACTION\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when extracting from this table using SQL

Name: LOADING\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when loading into this table using SQL

Name: LOADING\_TYPE

Type: STRING

Valid Values: INSERT, UPDATE, INSERT\_UPDATE, UPDATE\_INSERT, DELETE, NONE, TRUNCATE\_INSERT, DELETE\_INSERT, CHECK\_INSERT

Default: INSERT

The loading operation to be performed when this is a target.

Name: MATCH\_BY\_CONSTRAINT

Type: STRING

Valid Values: N/A

Default: YES

A property to indicate whether unique or primary key information on this target will override the matching criteria obtained from Match by constraint property on the attributes of this target.

Name: PARTITION\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Partition Name

Name: PEL\_ENABLED

Type: BOOLEAN

Valid Values: true, false

Default: false

PEL Enabled

Name: PRIMARY\_SOURCE

Type: STRING

Valid Values: YES, NO

Default: NO

A boolean value to indicate whether this is a primary source or not (only used in EDW).

Name: RECORDS\_TO\_SKIP

Type: NUMBER

Valid Values: N/A

Default: N/A

Number of records to skip

Name: REPLACE\_DATA

Type: BOOLEAN

Valid Values: true, false

Default: false

Replace existing data in target partition if there is any.

Name: SCHEMA

Type: STRING

Valid Values: N/A

Default: N/A

Schema

Name: SINGLEROW

Type: BOOLEAN

Valid Values: true, false

Default: false

Singlerow

Name: SORTED\_INDEXES\_CLAUSE

Type: STRING

Valid Values: N/A

Default: N/A

Sorted Indexes Clause

Name: SUBPARTITION\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Subpartition Name

Name: TARGET\_FILTER\_FOR\_DELETE

Type: STRING

Valid Values: N/A

Default: N/A

A condition on the rows in the target and if evaluated to true, that row will participate in the delete loading operation.

Name: TARGET\_FILTER\_FOR\_UPDATE

Type: STRING

Valid Values: N/A

Default: N/A

A condition on the rows in the target and if evaluated to true, that row will participate in the update loading operation.

Name: TRAILING\_NULLCOLS

Type: BOOLEAN

Valid Values: true, false

Default: false

Trailing Nullcols

Properties for EXTERNAL\_TABLE OPERATOR:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DATABASE\_LINK

Type: STRING

Valid Values: N/A



Default: N/A

The database link used to access this entity during mapping.

Name: EXTRACTION\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when extracting from this table using SQL

Name: LOADING\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when loading into this table using SQL

Name: PRIMARY\_SOURCE

Type: STRING

Valid Values: YES, NO

Default: NO

A boolean value to indicate whether this is a primary source or not (only used in EDW).

Name: SCHEMA

Type: STRING

Valid Values: N/A

Default: N/A

Schema

Properties for FILTER OPERATOR:

Name: FILTER\_CONDITION

Type: STRING

Valid Values: N/A

Default: N/A

The boolean filtering condition that identifies what data is to be processed. Any row with a false condition will be ignored.

Properties for FLAT\_FILE OPERATOR:

Name: CONCATENATE\_RECORDS

Type: NUMBER

Valid Values: N/A

Default: 0

Number of Physical Records per Logical Record.

Name: CONTINUATION\_CHARACTER

Type: STRING

Valid Values: N/A

Default: N/A

Character that indicates the record is continued on the next line.

Name: CONTINUATION\_CHARACTER\_ON\_NEXT\_LINE

Type: BOOLEAN

Valid Values: true, false

Default: FALSE

If there is a continuation character, is it at the start of the line.

Name: FIELD\_ENCLOSURE\_CHARACTERS

Type: STRING

Valid Values: N/A

Default: N/A

Characters that wrap fields. Example ' or ".

Name: FIELD\_NAMES\_IN\_THE\_FIRST\_ROW

Type: BOOLEAN

Valid Values: true, false

Default: FALSE

Indicates whether to write the field names in the first row of the output file.

Name: FIELD\_TERMINATION\_CHARACTER

Type: STRING

Valid Values: N/A

Default: ,

Character that separates the fields of a delimited file.

Name: FILE\_FORMAT  
Type: STRING  
Valid Values: N/A  
Default: DELIMITED  
File Format (Fixed or Delimited).

Name: LOADING\_TYPE  
Type: STRING  
Valid Values: INSERT, UPDATE, NONE  
Default: INSERT  
The loading operation to be performed

Name: NLS\_CHARACTERSET  
Type: STRING  
Valid Values: N/A  
Default: WE8MSWIN1252  
NLS Characterset

Name: RECORD\_DELIMITER  
Type: STRING  
Valid Values: N/A  
Default: N/A  
Character that indicates the end of the record.

Name: RECORD\_SIZE  
Type: NUMBER  
Valid Values: N/A  
Default: 0  
Size of a fixed length record.

Name: RECORD\_TYPE\_LENGTH  
Type: NUMBER  
Valid Values: N/A  
Default: 0  
If this is a multi record file, this will indicate the length of the data that identifies the type of record. It is used with the Record Type Position.

Name: RECORD\_TYPE\_POSITION

Type: NUMBER

Valid Values: N/A

Default: 0

If this is a multi record file, this will indicate the position of the field that identifies the type of record.

Name: SAMPLED\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name of the file that was sampled to get the metadata for this file.

Name: TARGET\_DATA\_FILE\_LOCATION

Type: STRING

Valid Values: N/A

Default: N/A

The name of the location in which to open the target data file. Make sure the root path of this location, as registered in the Runtime Platform, is exactly specified in the initialization file (INIT.ORA) of your runtime database using the UTL\_FILE\_DIR parameter.

Name: TARGET\_DATA\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Name of the target data file, including extension (file type). This name should not include the file path. To specify where the target data file will be created/appended, set the Target Data File Location.

Properties for INPUT\_PARAMETER OPERATOR:

Properties for JOINER OPERATOR:

Name: JOIN\_CONDITION

Type: STRING

Valid Values: N/A

Default: N/A

The Join Condition for the join operator

**Properties for KEY\_LOOKUP OPERATOR:**

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DATABASE\_LINK

Type: STRING

Valid Values: N/A

Default: N/A

The database link used to access this entity during mapping.

Name: LOOKUP\_CONDITION

Type: STRING

Valid Values: N/A

Default: N/A

Key lookup condition based on the source inputs. This condition is used to lookup a value in the bound table. If the condition is not met, the default value expression will be returned. If a default expression is not defined, null is used.

Name: SCHEMA

Type: STRING

Valid Values: N/A

Default: N/A

Schema

**Properties for MATCHMERGE OPERATOR:**

Name: MATCH\_KEYS

Type: STRING

Valid Values: N/A

Default: N/A

Ordered list of attributes that control the set of records to be matched at any at any particular time.

Name: MERGED\_PREFIX

Type: STRING

Valid Values: N/A

Default: N/A

Set the prefix used for the merged attributes in the cross-reference group.

Properties for MATERIALIZED\_VIEW OPERATOR:

Name: ADVANCED\_MATCH\_BY\_CONSTRAINT

Type: STRING

Valid Values: N/A

Default: No constraints

This property is used by public API and scripting to influence how columns are used for UPDATE or DELETE DMLs. If this property is set with the name of a primary or unique key, all the columns in the key will be used for matching during UPDATE or DELETE; and all the columns not in the key are used for loading. The property can also be assigned the value "All constraints" or "No constraints". If the DML type is INSERT, TRUNCATE/INSERT, or CHECK/INSERT, setting this property causes no effect.

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DATABASE\_LINK

Type: STRING

Valid Values: N/A

Default: N/A

The database link used to access this entity during mapping.

Name: EXTRACTION\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when extracting from this table using SQL

Name: LOADING\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when loading into this table using SQL

Name: LOADING\_TYPE

Type: STRING

Valid Values: INSERT, UPDATE, INSERT\_UPDATE, UPDATE\_INSERT, DELETE, NONE, TRUNCATE\_INSERT, DELETE\_INSERT, CHECK\_INSERT

Default: INSERT

The loading operation to be performed when this is a target.

Name: MATCH\_BY\_CONSTRAINT

Type: STRING

Valid Values: N/A

Default: YES

A property to indicate whether unique or primary key information on this target will override the matching criteria obtained from Match by constraint property on the attributes of this target.

Name: PRIMARY\_SOURCE

Type: STRING

Valid Values: YES, NO

Default: NO

A boolean value to indicate whether this is a primary source or not (only used in EDW).

Name: SCHEMA

Type: STRING

Valid Values: N/A

Default: N/A

Schema

Name: TARGET\_FILTER\_FOR\_DELETE

Type: STRING

Valid Values: N/A

Default: N/A

A condition on the rows in the target and if evaluated to true, that row will participate in the delete loading operation.

Name: TARGET\_FILTER\_FOR\_UPDATE

Type: STRING

Valid Values: N/A

Default: N/A

A condition on the rows in the target and if evaluated to true, that row will participate in the update loading operation.

Properties for NAME\_AND\_ADDRESS OPERATOR:

Name: ADDRESS\_LINE\_1

Type: STRING

Valid Values: N/A

Default: N/A

Address line 1

Name: ADDRESS\_LINE\_2

Type: STRING

Valid Values: N/A

Default: N/A

Address line 2

Name: ADDRESS\_LINE\_3

Type: STRING

Valid Values: N/A

Default: N/A

Address line 3

Name: ADDRESS\_LINE\_4

Type: STRING

Valid Values: N/A

Default: N/A

Address line 4

Name: DUAL\_ADDRESS\_ASSIGNMENT



Type: STRING

Valid Values: NA\_DUALADDR\_STREET, NA\_DUALADDR\_POBOX, NA\_DUALADDR\_CLOSESTTOLASTLINE

Default: NA\_DUALADDR\_STREET

A dual address refers to two address lines for the same destination. For example, a record contains both a street address and a P.O. Box; this is common with business data. Select which of the two address lines should be assigned in these cases.

Name: GENERATE\_CASS\_REPORT

Type: STRING

Valid Values: NA\_YES, NA\_NO

Default: NA\_NO

Select "Yes" to generate CASS (Coding Accuracy Support System) report. CASS report is a text file specified by the United States Postal Service. The report is written to the [nas/bin/admin/reports] folder under the home folder of the name/address server.

Name: LIST\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The list name is optional and provides a reference for tracking multiple CASS reports.

Name: PARSING\_TYPE

Type: STRING

Valid Values: NA\_NAMEONLY, NA\_ADDRESSONLY, NA\_NAMEANDADDRESS

Default: NA\_NAMEANDADDRESS

Select a name-address parsing type to be performed on the input data

Name: PRIMARY\_COUNTRY

Type: STRING

Valid Values: NA\_ARG, NA\_AUS, NA\_BEL, NA\_BRA, NA\_CAN, NA\_CHL, NA\_COL, NA\_DNK, NA\_FRA, NA\_DEU, NA\_HKG, NA\_IND, NA\_IRL, NA\_ITA, NA\_MEX, NA\_MYS, NA\_NLD, NA\_NZL, NA\_PER, NA\_PHL, NA\_PRT, NA\_SGP, NA\_ZAF, NA\_ESP, NA\_SWE, NA\_CHE, NA\_ARE, NA\_GBR, NA\_USA, NA\_VEN

Default: NA\_USA

Select the primary parsing country which best represents the input data. Input addresses having the same country as the primary parsing country will only need to be parsed once. Input addresses having a different country than the primary parsing country may be reparsed by a different parser. For performance reasons, it is best to minimize the percentage of 2-pass parses by selecting the optimal parser.

Name: PROCESSOR\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The processor name is the name of the organization submitting the CASS report.

Properties for OUTPUT\_PARAMETER OPERATOR:

Properties for PIVOT OPERATOR:

Name: PIVOT\_GROUP\_SIZE

Type: NUMBER

Valid Values: 1 - 1000

Default: 2

A number specifying the pivot group size for the pivot operation. Pivot group size determines the number of output rows that are produced from each input row.

Properties for POSTMAPPING\_PROCESS OPERATOR:

Name: FUNCTION\_CALL

Type: STRING

Valid Values: N/A

Default: N/A

Expression template for procedure call

Name: FUNCTION\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Name of the transformation to be called.

Name: POSTMAPPING\_PROCESS\_RUN\_CONDITION

Type: STRING

Valid Values: ALWAYS, ON\_SUCCESS, ON\_ERROR, ON\_WARNING

Default: ON\_SUCCESS

Indicates under what condition of the mapping the post-mapping process will be run.

Name: ROWBASED\_ONLY

Type: BOOLEAN

Valid Values: true, false

Default: N/A

Indicates if this transformation must be used only Row Based mode. Some transformations can be used in SQL mode as well as Row Based mode.

Properties for PREMAPPING\_PROCESS OPERATOR:

Name: FUNCTION\_CALL

Type: STRING

Valid Values: N/A

Default: N/A

Expression template for procedure call

Name: FUNCTION\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Name of the transformation to be called.

Name: MAPPING\_RUN\_CONDITION

Type: STRING

Valid Values: ALWAYS, ON\_SUCCESS, ON\_ERROR

Default: ON\_SUCCESS

Indicates under what condition of the pre-mapping process the mapping will be run.

Name: ROWBASED\_ONLY

Type: BOOLEAN

Valid Values: true, false

Default: N/A

Indicates if this transformation must be used only Row Based mode. Some transformations can be used in SQL mode as well as Row Based mode.

Properties for SEQUENCE OPERATOR:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DATABASE\_LINK

Type: STRING

Valid Values: N/A

Default: N/A

The database link used to access this entity during mapping.

Name: SCHEMA

Type: STRING

Valid Values: N/A

Default: N/A

Schema

Properties for SET\_OPERATION OPERATOR:

Name: SET\_OPERATION

Type: STRING

Valid Values: UNION, UNIONALL, INTERSECT, MINUS

Default: UNION

Specifies the set operation that is to be performed by this operator.

Properties for SORTER OPERATOR:

Name: ORDER\_BY\_CLAUSE

Type: STRING

Valid Values: N/A

Default: N/A

The Order By Clause

Properties for TABLE OPERATOR:

Name: ADVANCED\_MATCH\_BY\_CONSTRAINT

Type: STRING

Valid Values: N/A

Default: No constraints

This property is used by public API and scripting to influence how columns are used for UPDATE or DELETE DMLs. If this property is set with the name of a primary or unique key, all the columns in the key will be used for matching during UPDATE or DELETE; and all the columns not in the key are used for loading. The property can also be assigned the value "All constraints" or "No constraints". If the DML type is INSERT, TRUNCATE/INSERT, or CHECK/INSERT, setting this property causes no effect.

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DATABASE\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Database file name to allocate extents from

Name: DATABASE\_LINK

Type: STRING

Valid Values: N/A

Default: N/A

The database link used to access this entity during mapping.

Name: DATA\_COLLECTION\_FREQUENCY

Type: STRING

Valid Values: UNKNOWN, YEAR, QUARTER, MONTH, DAY, HOUR, MINUTE

Default: UNKNOWN

New Data Granularity

Name: DIRECT

Type: BOOLEAN

Valid Values: true, false

Default: false

Directly swap source into target as a partition without first creating a staging table.

Name: ENABLE\_CONSTRAINTS

Type: BOOLEAN

Valid Values: true, false

Default: true

Enable Constraints

Name: EXCEPTIONS\_TABLE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Exceptions Table Name

Name: EXTRACTION\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when extracting from this table using SQL

Name: LOADING\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when loading into this table using SQL

Name: LOADING\_TYPE

Type: STRING

Valid Values: INSERT, UPDATE, INSERT\_UPDATE, UPDATE\_INSERT, DELETE, NONE, TRUNCATE\_INSERT, DELETE\_INSERT, CHECK\_INSERT

Default: INSERT

The loading operation to be performed when this is a target.

Name: MATCH\_BY\_CONSTRAINT

Type: STRING

Valid Values: N/A

Default: YES

A property to indicate whether unique or primary key information on this target will override the matching criteria obtained from Match by constraint property on the attributes of this target.

Name: PARTITION\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Partition Name

Name: PEL\_ENABLED

Type: BOOLEAN

Valid Values: true, false

Default: false

PEL Enabled

Name: PRIMARY\_SOURCE

Type: STRING

Valid Values: YES, NO

Default: NO

A boolean value to indicate whether this is a primary source or not (only used in EDW).

Name: RECORDS\_TO\_SKIP

Type: NUMBER

Valid Values: N/A

Default: N/A

Number of records to skip

Name: REPLACE\_DATA

Type: BOOLEAN

Valid Values: true, false

Default: false

Replace existing data in target partition if there is any.

Name: SCHEMA

Type: STRING

Valid Values: N/A

Default: N/A

Schema

Name: SINGLEROW

Type: BOOLEAN

Valid Values: true, false

Default: false

Singlerow

Name: SORTED\_INDEXES\_CLAUSE

Type: STRING

Valid Values: N/A

Default: N/A

Sorted Indexes Clause

Name: SUBPARTITION\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Subpartition Name

Name: TARGET\_FILTER\_FOR\_DELETE

Type: STRING

Valid Values: N/A

Default: N/A

A condition on the rows in the target and if evaluated to true, that row will participate in the delete loading operation.

Name: TARGET\_FILTER\_FOR\_UPDATE

Type: STRING

Valid Values: N/A

Default: N/A

A condition on the rows in the target and if evaluated to true, that row will participate in the update loading operation.

Name: TRAILING\_NULLCOLS

Type: BOOLEAN

Valid Values: true, false



Default: false  
Trailing Nullcols

Properties for TABLE\_FUNCTION OPERATOR:

Name: TABLE\_FUNCTION\_NAME  
Type: STRING  
Valid Values: N/A  
Default: N/A  
The name of the Table Function

Properties for TRANSFORMATION OPERATOR:

Name: FUNCTION\_NAME  
Type: STRING  
Valid Values: N/A  
Default: N/A  
Name of the transformation to be called.

Name: ROWBASED\_ONLY  
Type: BOOLEAN  
Valid Values: true, false  
Default: N/A

Indicates if this transformation must be used only Row Based mode. Some transformations can be used in SQL mode as well as Row Based mode.

Properties for VIEW OPERATOR:

Name: ADVANCED\_MATCH\_BY\_CONSTRAINT  
Type: STRING  
Valid Values: N/A  
Default: No constraints

This property is used by public API and scripting to influence how columns are used for UPDATE or DELETE DMLs. If this property is set with the name of a primary or unique key, all the columns in the key will be used for matching during UPDATE or DELETE; and all the columns not in the key are used for loading. The property can also be assigned the value "All constraints" or "No constraints". If the DML type is INSERT, TRUNCATE/INSERT, or CHECK/INSERT, setting this property causes no effect.

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DATABASE\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Database file name to allocate extents from

Name: DATABASE\_LINK

Type: STRING

Valid Values: N/A

Default: N/A

The database link used to access this entity during mapping.

Name: EXTRACTION\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when extracting from this table using SQL

Name: LOADING\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when loading into this table using SQL

Name: LOADING\_TYPE

Type: STRING

Valid Values: INSERT, UPDATE, INSERT\_UPDATE, UPDATE\_INSERT, DELETE, NONE, TRUNCATE\_INSERT, DELETE\_INSERT, CHECK\_INSERT

Default: INSERT

The loading operation to be performed when this is a target.

Name: MATCH\_BY\_CONSTRAINT

Type: STRING

Valid Values: N/A

Default: YES

A property to indicate whether unique or primary key information on this target will override the matching criteria obtained from Match by constraint property on the attributes of this target.

Name: PARTITION\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Partition Name

Name: PRIMARY\_SOURCE

Type: STRING

Valid Values: YES, NO

Default: NO

A boolean value to indicate whether this is a primary source or not (only used in EDW).

Name: RECORDS\_TO\_SKIP

Type: NUMBER

Valid Values: N/A

Default: N/A

Number of records to skip

Name: SCHEMA

Type: STRING

Valid Values: N/A

Default: N/A

Schema

Name: SINGLEROW

Type: BOOLEAN

Valid Values: true, false

Default: false

Singlerow

Name: SORTED\_INDEXES\_CLAUSE

Type: STRING

Valid Values: N/A

Default: N/A

Sorted Indexes Clause

Name: SUBPARTITION\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Subpartition Name

Name: TARGET\_FILTER\_FOR\_DELETE

Type: STRING

Valid Values: N/A

Default: N/A

A condition on the rows in the target and if evaluated to true, that row will participate in the delete loading operation.

Name: TARGET\_FILTER\_FOR\_UPDATE

Type: STRING

Valid Values: N/A

Default: N/A

A condition on the rows in the target and if evaluated to true, that row will participate in the update loading operation.

Name: TRAILING\_NULLCOLS

Type: BOOLEAN

Valid Values: true, false

Default: false

Trailing Nullcols

Properties for CUBE, DIMENSION, KEY\_LOOKUP, MATERIALIZED\_VIEW, TABLE, VIEW OPERATOR KEYS\_READONLY CHILD:

Name: KEY\_COLUMNS

Type: STRING

Valid Values: N/A

Default: N/A

Local columns that define this key (Comma separated if more than one).

Name: KEY\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Name of the primary, foreign or unique key (primary, foreign, or unique).

Name: KEY\_TYPE

Type: STRING

Valid Values: N/A

Default: UNIQUE

Type of key - primary, foreign or unique.

Name: REFERENCED\_KEYS

Type: STRING

Valid Values: N/A

Default: N/A

If the key is a foreign key, this will contain the key or keys used of the referenced object.

Properties for MATCHMERGE OPERATOR MATCH\_RULES CHILD:

Name: ADDRESS\_ADDRESS\_LINE\_SIMILARITY\_SCORE

Type: NUMBER

Valid Values: N/A

Default: N/A

Similarity score for address line in Address Match Rule.

Name: ADDRESS\_ALLOW\_DIFFERING\_SECONDARY\_ADDRESSES

Type: BOOLEAN

Valid Values: true, false

Default: false

Allow differing secondary addresses to match in Address Match Rule.

Name: ADDRESS\_LAST\_LINE\_SIMILARITY\_SCORE

Type: NUMBER

Valid Values: N/A

Default: N/A

Similarity score for last line in Address Match Rule.

Name: ADDRESS\_MATCH\_ON\_ADDRESS\_LINE\_SIMILARITY

Type: BOOLEAN

Valid Values: true, false

Default: false

Check address line similarity in Address Match Rule.

Name: ADDRESS\_MATCH\_ON\_BLANK\_SECONDARY\_ADDRESSES

Type: BOOLEAN

Valid Values: true, false

Default: false

Match on blank secondary address in Address Match Rule.

Name: ADDRESS\_MATCH\_ON\_LAST\_LINE\_SIMILARITY

Type: BOOLEAN

Valid Values: true, false

Default: false

Check last line similarity in Address Match Rule.

Name: ADDRESS\_MATCH\_ON\_STREET\_OR\_PO\_BOX

Type: BOOLEAN

Valid Values: true, false

Default: false

Match on Street or Post Office (PO) Box in Address Match Rule.

Name: CUSTOM\_RULE

Type: STRING

Valid Values: N/A

Default: N/A

Custom Merge Rule

Name: DESCRIPTION

Type: STRING

Valid Values: N/A

Default: N/A

Description of match rule.

Name: FIRM\_CROSS\_MATCH\_FIRM1\_AND\_FIRM2

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a firm match rule, should Warehouse Builder cross match firm 1 and firm 2?

Name: FIRM\_MATCH\_ON\_ABBREVIATIONS

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a firm match rule, should Warehouse Builder match on abbreviations?

Name: FIRM\_MATCH\_ON\_ACRONYMS

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a firm match rule, should Warehouse Builder match on acronyms?

Name: FIRM\_MATCH\_ON\_PARTIAL\_NAMES

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a firm match rule, should Warehouse Builder match on partial names?

Name: FIRM\_MATCH\_ON\_SIMILARITY

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a firm match rule, should Warehouse Builder check firm similarity?

Name: FIRM\_SIMILARITY\_SCORE

Type: NUMBER

Valid Values: N/A

Default: N/A

Similarity score for firm in Firm Match Rule.

Name: FIRM\_STRIP\_NOISE\_WORDS

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a firm match rule, should Warehouse Builder strip noise words?

Name: PERSON\_DETECT\_SWITCHED\_NAME\_ORDER

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person rule, should Warehouse Builder detect the switched name order.

Name: PERSON\_FN\_DETECT\_COMPOUND\_NAME

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder detect compound names?

Name: PERSON\_FN\_MATCH\_ON\_INITIALS

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder match on initials.

Name: PERSON\_FN\_MATCH\_ON\_SIMILARITY

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder match on first name similarity?

Name: PERSON\_FN\_MATCH\_ON\_SOUNDEX



Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder match on first name soundex?

Name: PERSON\_FN\_MATCH\_ON\_SUBSTRINGS

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder match on first name substrings?

Name: PERSON\_FN\_MRS\_MATCH

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder detect MRS? (For example, should Mrs John Smith match Mrs Smith)

Name: PERSON\_FN\_SIMILARITY\_SCORE

Type: NUMBER

Valid Values: N/A

Default: N/A

If matching in first name similarity, what is the similarity score?

Name: PERSON\_LN\_DETECT\_MISSING\_HYPHEN

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder detect missing hyphens?

Name: PERSON\_LN\_MATCH\_HYPHENATED\_NAMES

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder match on hyphenated last name?

Name: PERSON\_LN\_MATCH\_ON\_SIMILARITY

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder match on last name similarity?

Name: PERSON\_LN\_MATCH\_ON\_SOUNDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder match on last name soundex?

Name: PERSON\_LN\_SIMILARITY\_SCORE

Type: NUMBER

Valid Values: N/A

Default: N/A

If Warehouse Builder is matching on last name similarity, what is the similarity score?

Name: PERSON\_MN\_MATCH\_ON\_INITIALS

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder match on middle name initials?

Name: PERSON\_MN\_MATCH\_ON\_SIMILARITY

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder match on middle name similarity?

Name: PERSON\_MN\_MATCH\_ON\_SOUNDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

---

If this is a person match rule, should Warehouse Builder match on middle name soundex?

Name: PERSON\_MN\_MATCH\_ON\_SUBSTRINGS

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder match on middle name substrings?

Name: PERSON\_MN\_SIMILARITY\_SCORE

Type: NUMBER

Valid Values: N/A

Default: N/A

If Warehouse Builder is matching on middle name similarity, what is the similarity score?

Name: TOTAL\_WEIGHT\_SCORE

Type: NUMBER

Valid Values: N/A

Default: N/A

If this is a weight rule, the sum of all weights must be equal to or greater than this score for the records to pass the weight rule.

Name: TYPE

Type: STRING

Valid Values: MM\_TRUE, MM\_FALSE, MM\_WEIGHT, MM\_CONDITIONAL, MM\_CUSTOM, MM\_PERSON, MM\_FIRM, MM\_ADDRESS

Default: MM\_TRUE

What type of match rule is this? Possible types are: MM\_TRUE, MM\_FALSE, MM\_WEIGHT, MM\_CONDITIONAL, MM\_CUSTOM, MM\_PERSON, MM\_FIRM, MM\_ADDRESS

Name: USAGE

Type: STRING

Valid Values: MM\_ACTIVE, MM\_PASSIVE

Default: MM\_ACTIVE

Is this an active or passive rule? Possible usages are: MM\_ACTIVE, MM\_PASSIVE.

## Properties for MATCHMERGE OPERATOR MERGE\_RULES CHILD:

Name: ATTRIBUTE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Merge Attribute

Name: COPY\_ATTRIBUTE

Type: STRING

Valid Values: N/A

Default: N/A

Copy this merged attribute to merge rule target attribute.

Name: CUSTOM\_TEXT

Type: STRING

Valid Values: N/A

Default: N/A

Implementation text for custom merge rule. Include "BEGIN and END statements.

Name: DESCRIPTION

Type: STRING

Valid Values: N/A

Default: N/A

Description

Name: MATCH\_ID\_SEQUENCE\_MODULE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Physical sequence module name for match id merge rule. This sequence will be used to generate the match id.

Name: MATCH\_ID\_SEQUENCE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

---

Physical sequence name for match id merge rule. This sequence will be used to generate the match id.

Name: MIN\_MAX\_ATTRIBUTE

Type: STRING

Valid Values: N/A

Default: N/A

Selecting attribute for min/max merge rule

Name: MIN\_MAX\_TYPE

Type: STRING

Valid Values: MM\_MIN, MM\_MAX, MM\_SHORTEST, MM\_LONGEST

Default: MM\_MAX

Select record where attribute is min,max, shortest, longest. Possible values are MM\_MIN, MM\_MAX, MM\_SHORTEST, MM\_LONGEST.

Name: SEQUENCE\_MODULE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Physical sequence module name for sequence merge rule.

Name: SEQUENCE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Physical sequence name for sequence merge rule.

Name: TYPE

Type: STRING

Valid Values: MM\_ANY, MM\_MATCH\_ID, MM\_RANK, MM\_SEQUENCE, MM\_MIN\_MAX, MM\_COPY, MM\_CUSTOM, MM\_RECORD\_ANY, MM\_RECORD\_RANK, MM\_RECORD\_MIN\_MAX, MM\_RECORD\_CUSTOM

Default: MM\_ANY

Merge Rule Type. Possible values are:MM\_ANY,MM\_MATCH\_ID,MM\_RANK,MM\_SEQUENCE,MM\_MIN\_MAX,MM\_COPY,MM\_CUSTOM,MM\_RECORD\_ANY,MM\_RECORD\_RANK,MM\_RECORD\_MIN\_MAX,MM\_RECORD\_CUSTOM.

Properties for MATCHMERGE OPERATOR MATCH\_RULES CHILD ADDRESS\_ROLES CHILD:

Name: ATTRIBUTE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

What is the attribute for this role?

Name: TYPE

Type: STRING

Valid Values: MM\_PRIMARY\_ADDR, MM\_UNIT\_NUM, MM\_PO\_BOX, MM\_DUAL\_PRIMARY\_ADDR, MM\_DUAL\_UNIT\_NUM, MM\_DUAL\_PO\_BOX, MM\_CITY, MM\_STATE, MM\_POSTAL\_CODE, MM\_IS\_FOUND

Default: MM\_PRIMARY\_ADDR

What role is this attribute? Possible values are: MM\_PRIMARY\_ADDR, MM\_UNIT\_NUM, MM\_PO\_BOX, MM\_DUAL\_PRIMARY\_ADDR, MM\_DUAL\_UNIT\_NUM, MM\_DUAL\_PO\_BOX, MM\_CITY, MM\_STATE, MM\_POSTAL\_CODE, MM\_IS\_FOUND.

Properties for MATCHMERGE OPERATOR MATCH\_RULES CHILD CONDITIONS CHILD:

Name: ALGORITHM

Type: STRING

Valid Values: MM\_EXACT, MM\_STD\_EXACT, MM\_SOUNDEX, MM\_SIMILARITY, MM\_STD\_SIMILARITY, MM\_PARTIAL\_NAME, MM\_ABBREVIATION, MM\_ACRONYM

Default: MM\_EXACT

Algorithm of this condition. Possible values are: MM\_EXACT, MM\_STD\_EXACT, MM\_SOUNDEX, MM\_SIMILARITY, MM\_STD\_SIMILARITY, MM\_PARTIAL\_NAME, MM\_ABBREVIATION, MM\_ACRONYM.

Name: ATTRIBUTE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

To which attribute does this condition apply?

Name: BLANK\_MATCHING

Type: STRING

Valid Values: MM\_MATCH\_BOTH\_BLANK, MM\_MATCH\_EITHER\_BLANK, MM\_NO\_MATCH\_IF\_BLANK

Default: MM\_MATCH\_BOTH\_BLANK

How do you want blanks to be handled? Possible values are: MM\_MATCH\_BOTH\_BLANK, MM\_MATCH\_EITHER\_BLANK, MM\_NO\_MATCH\_IF\_BLANK.

Name: SIMILARITY\_SCORE

Type: NUMBER

Valid Values: N/A

Default: N/A

If this is a similarity condition, what score must the similarity equal or exceed for records to meet the condition?

Properties for MATCHMERGE OPERATOR MATCH\_RULES CHILD FIRM\_ROLES CHILD:

Name: ATTRIBUTE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

What is the attribute for this role?

Name: TYPE

Type: STRING

Valid Values: MM\_FIRM1, MM\_FIRM2

Default: MM\_FIRM1

What role is this attribute? Possible values are: MM\_FIRM1,MM\_FIRM2.

Properties for MATCHMERGE OPERATOR MATCH\_RULES CHILD PERSON\_ROLES CHILD:

Name: ATTRIBUTE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

What is the attribute for this role.

Name: TYPE

Type: STRING

Valid Values: MM\_PRENAME, MM\_FIRST\_NAME\_STD, MM\_MIDDLE\_NAME\_STD, MM\_MIDDLE\_NAME\_2\_STD, MM\_MIDDLE\_NAME\_3\_STD, MM\_LAST\_NAME, MM\_MATURITY\_POST\_NAME

Default: MM\_PRENAME

What role is this attribute? Possible values are: MM\_PRENAME, MM\_FIRST\_NAME\_STD, MM\_MIDDLE\_NAME\_STD, MM\_MIDDLE\_NAME\_2\_STD, MM\_MIDDLE\_NAME\_3\_STD, MM\_LAST\_NAME, MM\_MATURITY\_POST\_NAME.

Properties for MATCHMERGE OPERATOR MATCH\_RULES CHILD WEIGHTS CHILD:

Name: BLANK\_SCORE

Type: NUMBER

Valid Values: N/A

Default: N/A

Score if either attribute is blank

Name: WEIGHT\_ATTRIBUTE

Type: STRING

Valid Values: N/A

Default: N/A

Attribute

Name: WEIGHT\_SCORE

Type: NUMBER

Valid Values: N/A

Default: N/A

Score used for this weight if the attributes in the two match records are identical. Similarity will be used to generate this score.

Properties for MATCHMERGE OPERATOR MERGE\_RULES CHILD ATTRIBUTES CHILD:

Name: ATTRIBUTE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Record merge rule attribute



---

Properties for MATCHMERGE OPERATOR MERGE\_RULES CHILD RANK\_RULES CHILD:

Name: EXPRESSION

Type: STRING

Valid Values: N/A

Default: N/A

Rank expression.

Properties for FLAT\_FILE OPERATOR INOUT GROUP:

Name: RECORD\_TYPE\_VALUES

Type: STRING

Valid Values: N/A

Default: N/A

Record Type Values.

Properties for JOINER OPERATOR INPUT GROUP:

Properties for PIVOT OPERATOR OUTPUT GROUP:

Name: ROW\_LOCATOR

Type: STRING

Valid Values: N/A

Default: N/A

An expression indicating which attribute within the output group is the row locator.

Properties for SPLITTER OPERATOR OUTPUT GROUP:

Name: SPLIT\_CONDITION

Type: STRING

Valid Values: N/A

Default: N/A

Condition that defines when to perform the attribute maps for the attributes in this group.

Properties for TABLE\_FUNCTION OPERATOR INPUT GROUP:

Name: INPUT\_PARAMETER\_TYPE

Type: STRING

Valid Values: REF\_CURSOR, SCALAR

Default: REF\_CURSOR

This property specifies whether the input parameter is a scalar or a ref cursor type

Name: PARAMETER\_POSITION

Type: NUMBER

Valid Values: N/A

Default: N/A

The position of the argument in the table function signature corresponding to this parameter group

Properties for TABLE\_FUNCTION OPERATOR OUTPUT GROUP:

Name: RETURN\_TABLE\_OF\_SCALAR

Type: BOOLEAN

Valid Values: true, false

Default: false

This property specifies whether the return of the table function is a TABLE of SCALAR or not.

Properties for UNPIVOT OPERATOR INPUT GROUP:

Name: ROW\_LOCATOR

Type: STRING

Valid Values: N/A

Default: N/A

An expression indicating which attribute within the input group is the row locator.

Name: ROW\_LOCATOR\_VALUES

Type: STRING

Valid Values: N/A

Default: NULL, NULL

A comma-separated expressions that gives the possible values of the row locator within a unpivot group.

Properties for ADVANCED\_QUEUE OPERATOR INOUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Properties for CUBE OPERATOR INOUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: LOAD\_COLUMN\_WHEN\_INSERTING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will participate in the insert load operation.

Name: LOAD\_COLUMN\_WHEN\_UPDATING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will participate in the update load operation.

Name: MATCH\_COLUMN\_WHEN\_DELETING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will be used to construct the matching criteria between the incoming data and the existing data on the target during the delete load operation.

Name: MATCH\_COLUMN\_WHEN\_UPDATING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will be used to construct the matching criteria between the incoming data and the existing data on the target during the update load operation.

Name: UPDATE\_OPERATION

Type: STRING

Valid Values: =, +=, -=, =-, =| |, || =

Default: =

The computation to be performed on this attribute between the incoming data and the existing data on the target during the update load operation.

Properties for DIMENSION OPERATOR INOUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: LOAD\_COLUMN\_WHEN\_INSERTING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will participate in the insert load operation.

Name: LOAD\_COLUMN\_WHEN\_UPDATING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will participate in the update load operation.

Name: MATCH\_COLUMN\_WHEN\_DELETING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will be used to construct the matching criteria between the incoming data and the existing data on the target during the delete load operation.

Name: MATCH\_COLUMN\_WHEN\_UPDATING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will be used to construct the matching criteria between the incoming data and the existing data on the target during the update load operation.

Name: UPDATE\_OPERATION

Type: STRING

Valid Values: =, +=, -=, =-, =||, || =

Default: =

The computation to be performed on this attribute between the incoming data and the existing data on the target during the update load operation.

Properties for EXTERNAL\_TABLE OPERATOR INOUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Properties for INPUT\_PARAMETER OPERATOR OUTPUT GROUP ATTRIBUTE:

Name: DEFAULT\_VALUE

Type: STRING

Valid Values: N/A

Default: N/A

The default value applies when the map is not given a value for this attribute. The value can be specified only by the calling program.

Properties for KEY\_LOOKUP OPERATOR OUTPUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DEFAULT\_VALUE

Type: STRING

Valid Values: N/A

Default: N/A

The default value used for this attribute if no key lookup table row exists for a given input row.

Properties for MATCHMERGE OPERATOR OUTPUT GROUP ATTRIBUTE:

Name: MERGE\_ATTR

Type: STRING

Valid Values: N/A

Default: N/A

Related merge attribute

Properties for MATERIALIZED\_VIEW OPERATOR INOUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: LOAD\_COLUMN\_WHEN\_INSERTING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will participate in the insert load operation.

Name: LOAD\_COLUMN\_WHEN\_UPDATING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will participate in the update load operation.

Name: MATCH\_COLUMN\_WHEN\_DELETING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will be used to construct the matching criteria between the incoming data and the existing data on the target during the delete load operation.

Name: MATCH\_COLUMN\_WHEN\_UPDATING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will be used to construct the matching criteria between the incoming data and the existing data on the target during the update load operation.

Name: UPDATE\_OPERATION

Type: STRING

Valid Values: =, +=, -=, =-, =|, ||, |=

Default: =

The computation to be performed on this attribute between the incoming data and the existing data on the target during the update load operation.

Properties for NAME\_AND\_ADDRESS OPERATOR INPUT GROUP ATTRIBUTE:

Name: INPUT\_ROLE

Type: STRING

Valid Values: NA\_NONE, NA\_FIRSTNAME, NA\_MIDDLENAME, NA\_MIDDLENAME2, NA\_MIDDLENAME3, NA\_LASTNAME, NA\_FIRSTPARTNAME, NA\_LASTPARTNAME, NA\_PRENAME, NA\_POSTNAME, NA\_PERSON, NA\_

PERSON2, NA\_PERSON3, NA\_FIRMNAME, NA\_PRIMARYADDRESS, NA\_SECONDARYADDRESS, NA\_ADDRESS, NA\_ADDRESS2, NA\_NEIGHBORHOOD, NA\_LASTLINE, NA\_CITY, NA\_STATE, NA\_POSTALCODE, NA\_COUNTRYNAME, NA\_COUNTRYCODE, NA\_LINE1, NA\_LINE2, NA\_LINE3, NA\_LINE4, NA\_LINE5, NA\_LINE6, NA\_LINE7, NA\_LINE8, NA\_LINE9, NA\_LINE10

Default: NA\_NONE

Assigns a name-address input role to the selected input attribute

Properties for NAME\_AND\_ADDRESS OPERATOR OUTPUT GROUP ATTRIBUTE:

Name: ADDRESS\_TYPE

Type: STRING

Valid Values: NA\_ADDRTYPE\_NORMAL, NA\_ADDRTYPE\_DUAL

Default: NA\_ADDRTYPE\_NORMAL

You can designate an address type as Normal or Dual. For example, a dual address occurs when a record contains both a street address and a P.O. Box; this is common with business data. A normal address contains only one type of address.

Name: INSTANCE

Type: STRING

Valid Values: NA\_INSTANCE\_FIRST, NA\_INSTANCE\_SECOND, NA\_INSTANCE\_THIRD, NA\_INSTANCE\_FOURTH, NA\_INSTANCE\_FIFTH, NA\_INSTANCE\_SIXTH

Default: NA\_INSTANCE\_FIRST

The instance option is used when an address contains multiple names, you can specify which name in the group should be used. In addition, you can use this option to assign an address type to a miscellaneous address component.

Name: OUTPUT\_COMPONENT

Type: STRING

Valid Values: NA\_NONE, NA\_PRENAME, NA\_FIRSTNAMESTD, NA\_MIDDLENAMESTD, NA\_MIDDLENAME2STD, NA\_MIDDLENAME3STD, NA\_POSTNAME, NA\_OTHERPOSTNAME, NA\_NAMEDESIGNATOR, NA\_RELATIONSHIP, NA\_PERSON, NA\_FIRSTNAME, NA\_MIDDLENAME, NA\_MIDDLENAME2, NA\_MIDDLENAME3, NA\_LASTNAME, NA\_GENDER, NA\_PERSONCOUNT, NA\_FIRMNAME, NA\_FIRMCOUNT, NA\_ADDRESS, NA\_PRIMARYADDRESS, NA\_STREETNUMBER, NA\_PREDIRECTIONAL, NA\_STREETNAME, NA\_STREETTYPE, NA\_POSTDIRECTIONAL, NA\_SECONDARYADDRESS, NA\_UNITDESIGNATOR, NA\_UNITNUMBER, NA\_BOXNAME, NA\_BOXNUMBER, NA\_ROUTENAME, NA\_ROUTENUMBER, NA\_BUILDINGNAME, NA\_COMPLEX, NA\_MISCADDRESS, NA\_LASTLINE, NA\_NEIGHBORHOOD, NA\_CITY, NA\_STATE, NA\_POSTALCODE, NA\_POSTALCODEFORMATTED, NA\_DELIVERYPOINT, NA\_COUNTRYCODE, NA\_COUNTRYCODE3, NA\_COUNTRYNAME, NA\_ISGOODGROUP, NA\_ISPARSED, NA\_PARSESTATUS, NA\_PARSESTATUSDESC, NA\_ISGOODNAME, NA\_NAMEWARNING, NA\_ISGOODADDRESS, NA\_ISFOUND, NA\_CITYMATCH, NA\_



STREETNAMEMATCH, NA\_STREETNUMBERMATCH, NA\_STREETCOMPMATCH, NA\_NONAMBIGUOUSMATCH, NA\_CITYWARNING, NA\_STREETWARNING, NA\_ISADDRESSVERIFIABLE, NA\_ADDRESSCORRECTED, NA\_POSTALCODECORRECTED, NA\_CITYCORRECTED, NA\_STREETCORRECTED, NA\_STREETCOMPCORRECTED, NA\_ADDRESSTYPE, NA\_PARSINGCOUNTRY, NA\_INSTALLATIONTYPE, NA\_INSTALLATIONNAME, NA\_DELIVERYOFFICECODE, NA\_DELIVERYBEATCODE, NA\_ADDRESS2, NA\_LOCALITYCODE, NA\_LOCALITYNAME, NA\_COUNTYNAME, NA\_ZIP5, NA\_ZIP4, NA\_URBANIZATIONNAME, NA\_LACS, NA\_CART, NA\_CHECKDIGIT, NA\_MSA, NA\_MCD, NA\_LATITUDE, NA\_LONGITUDE, NA\_FIPSCOUNTY, NA\_FIPS, NA\_CENSUSID

Default: NA\_NONE

Assigns a Name and Address output component to the selected output attribute.

Properties for PIVOT OPERATOR INPUT GROUP ATTRIBUTE:

Name: GROUP\_KEY

Type: BOOLEAN

Valid Values: true, false

Default: false

A boolean value to indicate whether this input attribute is a part of the pivot group key.

Properties for PIVOT OPERATOR OUTPUT GROUP ATTRIBUTE:

Name: GROUP\_KEY

Type: BOOLEAN

Valid Values: true, false

Default: false

A boolean value to indicate whether this output attribute is a part of the pivot group key, which obtains its value from its corresponding input attribute.

Name: PIVOT\_EXPRESSION

Type: STRING

Valid Values: N/A

Default: N/A

A comma-separated expression that gives the input attribute to be used for each output row in the pivot group.

Properties for POSTMAPPING\_PROCESS OPERATOR INPUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DEFAULT\_VALUE

Type: STRING

Valid Values: N/A

Default: N/A

The Default Value for the function input parameter

Name: IS\_OPTIONAL

Type: BOOLEAN

Valid Values: true, false

Default: N/A

If true, the input is not required to be connected

Properties for POSTMAPPING\_PROCESS OPERATOR OUTPUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: FUNCTION\_RETURN

Type: BOOLEAN

Valid Values: true, false

Default: N/A

Specifies whether this output is the return value of this function

Properties for PREMAPPING\_PROCESS OPERATOR INPUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DEFAULT\_VALUE

Type: STRING

Valid Values: N/A

Default: N/A

The Default Value for the function input parameter

Name: IS\_OPTIONAL

Type: BOOLEAN

Valid Values: true, false

Default: N/A

If true, the input is not required to be connected

Properties for PREMAPPING\_PROCESS OPERATOR OUTPUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: FUNCTION\_RETURN

Type: BOOLEAN

Valid Values: true, false

Default: N/A

Specifies whether this output is the return value of this function

Properties for TABLE OPERATOR INOUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: LOAD\_COLUMN\_WHEN\_INSERTING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will participate in the insert load operation.

Name: LOAD\_COLUMN\_WHEN\_UPDATING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will participate in the update load operation.

Name: MATCH\_COLUMN\_WHEN\_DELETING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will be used to construct the matching criteria between the incoming data and the existing data on the target during the delete load operation.

Name: MATCH\_COLUMN\_WHEN\_UPDATING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will be used to construct the matching criteria between the incoming data and the existing data on the target during the update load operation.

Name: UPDATE\_OPERATION

Type: STRING

Valid Values: =, +=, -=, =-, =|, ||, |=

Default: =

The computation to be performed on this attribute between the incoming data and the existing data on the target during the update load operation.

---

Properties for TABLE\_FUNCTION OPERATOR INPUT GROUP ATTRIBUTE:

Name: PARAMETER\_POSITION

Type: NUMBER

Valid Values: N/A

Default: N/A

The position of the argument in the table function signature corresponding to this parameter

Properties for TABLE\_FUNCTION OPERATOR OUTPUT GROUP ATTRIBUTE:

Name: TYPE\_ATTRIBUTE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name of the field of the PLS Record or attribute of the Object Type or column of the ROWTYPE that corresponds to this attribute. This property is not applicable if the return type is TABLE or SCALAR.

Properties for TRANSFORMATION OPERATOR INPUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DEFAULT\_VALUE

Type: STRING

Valid Values: N/A

Default: N/A

The Default Value for the function input parameter

Name: IS\_OPTIONAL

Type: BOOLEAN

Valid Values: true, false

Default: N/A

If true, the input is not required to be connected

Properties for TRANSFORMATION OPERATOR OUTPUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: FUNCTION\_RETURN

Type: BOOLEAN

Valid Values: true, false

Default: N/A

Specifies whether this output is the return value of this function

Properties for TRANSFORMATION OPERATOR INOUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Properties for UNPIVOT OPERATOR INPUT GROUP ATTRIBUTE:

Name: GROUP\_KEY

Type: BOOLEAN

Valid Values: true, false

Default: false

A boolean value to indicate whether this input attribute is a part of the unpivot group key.

Properties for UNPIVOT OPERATOR OUTPUT GROUP ATTRIBUTE:

Name: GROUP\_KEY

Type: BOOLEAN

Valid Values: true, false

Default: false

A boolean value to indicate whether this output attribute is a part of the unpivot group key, which obtains its value from its corresponding input attribute.

Name: MATCHING\_ROW

Type: NUMBER

Valid Values: 1 - 1000

Default: 1

An positive integer to indicate from which row within the unpivot group this output attribute obtains its data.

Name: UNPIVOT\_EXPRESSION

Type: STRING

Valid Values: N/A

Default: NULL

An expression that gives the input attribute to be used as the output of this attribute.

Properties for VIEW OPERATOR INOUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: LOAD\_COLUMN\_WHEN\_INSERTING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will participate in the insert load operation.

Name: LOAD\_COLUMN\_WHEN\_UPDATING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will participate in the update load operation.

Name: MATCH\_COLUMN\_WHEN\_DELETING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will be used to construct the matching criteria between the incoming data and the existing data on the target during the delete load operation.

Name: MATCH\_COLUMN\_WHEN\_UPDATING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will be used to construct the matching criteria between the incoming data and the existing data on the target during the update load operation.

Name: UPDATE\_OPERATION

Type: STRING

Valid Values: =, +=, -=, =-, =| |, || =

Default: =

The computation to be performed on this attribute between the incoming data and the existing data on the target during the update load operation.

Note:

1. N/A means any valid character in supported character set.
2. " " represents an empty string

propertyValue

A single property value. It can be a number, float, boolean or single-quoted string.

operatorType

Type of a mapping operator. The following operator types are available: ADVANCED\_QUEUE, AGGREGATOR, CONSTANT, CUBE, DATA\_GENERATOR, DEDUPLICATOR, DIMENSION, EXPRESSION, EXTERNAL\_PROCESS, EXTERNAL\_TABLE, FILTER, FLAT\_FILE, INPUT\_PARAMETER, JOINER, KEY\_LOOKUP,



MATCHMERGE, MATERIALIZED\_VIEW, NAME\_AND\_ADDRESS, OUTPUT\_PARAMETER, PIVOT, POSTMAPPING\_PROCESS, PREMAPPING\_PROCESS, SEQUENCE, SET\_OPERATION, SORTER, SPLITTER, TABLE, TRANSFORMATION, UNPIVOT, VIEW.

operatorName

Name of a mapping operator.

setBindingClause

Set the binding during the creation of a mapping operator or mapping attribute.

groupDirection

Direction of a mapping group.

groupName

Name of a mapping group.

attributeName

Name of a mapping attribute.

childType

Type of a child that belongs to map, mapping operator, mapping group or mapping attribute.

childName

Name of a child that belongs to map, mapping operator, mapping group or mapping attribute.

childOwnerBottomUpLocator

Location of a child owner. A child owner can be a map, mapping operator, mapping group, mapping attribute or a child.

groupToGroupConnectType

Connecting from a mapping group in one mapping operator to a mapping group in another mapping operator.

attributesBottomUpLocator

Location of a list of mapping attributes.

`mappableBottomUpLocator`

Location of the object to be bound to a mapping mapping operator or mapping attribute.

`bindableLocator`

Location of the object to be bound to a mapping operator or mapping attribute.

`attributeNameList`

A list of attribute names.

`bindableType`

Type of object bound to a mapping operator or mapping attribute.

`bindableName`

Name of the object bound to a mapping operator or mapping attribute.

## Examples

```
OMBALTER MAPPING 'MAP1' RENAME TO 'MAP2'
```

```
OMBALTER MAPPING 'MAP1'  
ADD CONNECTION FROM GROUP 'INOUTGRP1' OF OPERATOR 'CUST_SRC'  
TO GROUP 'INOUTGRP1' OF OPERATOR 'CUST_LOOK_UP'
```

```
OMBALTER MAPPING 'MAP1' DELETE OPERATOR 'OP1'
```

## See Also

OMBALTER, OMBCREATE MAPPING, OMBRETRIEVE MAPPING, OMBDROP MAPPING

# OMBALTER MATERIALIZED\_VIEW

## Purpose

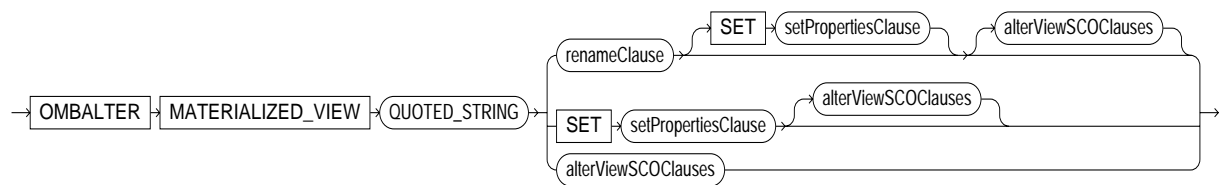
OMBALTER MATERIALIZED\_VIEW - To alter properties and definition of a materialized view.

## Prerequisites

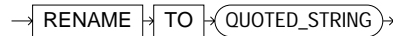
In the context of an Oracle Module.

## Syntax Diagrams

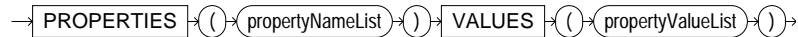
alterMaterializedViewCommand



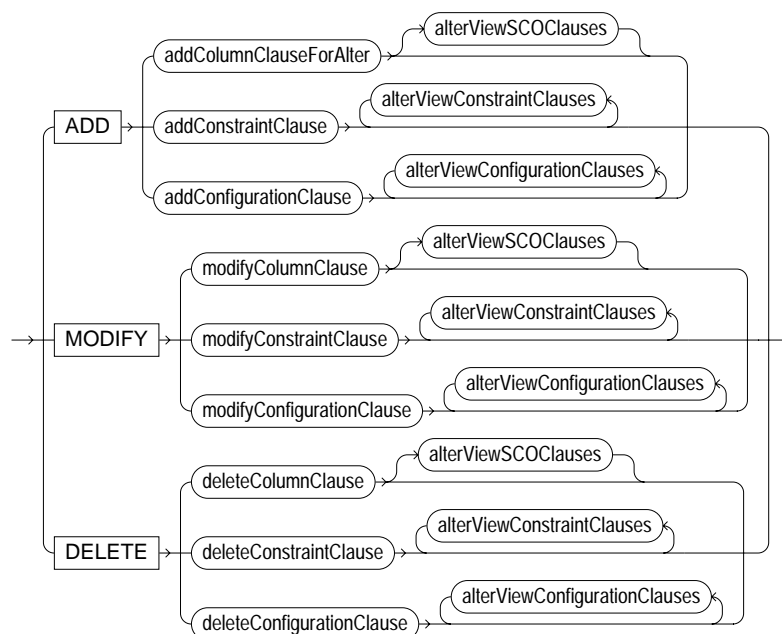
renameClause



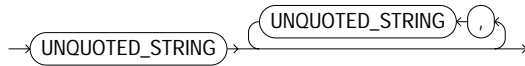
setPropertiesClause



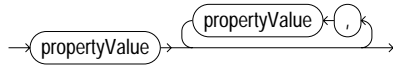
alterViewSCOClauses



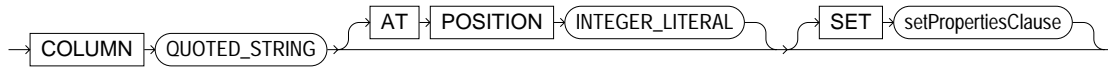
propertyNameList



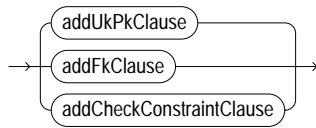
propertyValueList



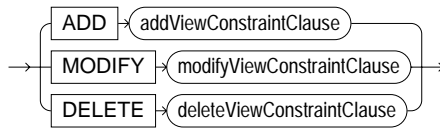
addColumnClauseForAlter



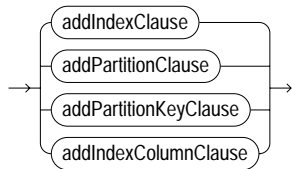
addConstraintClause



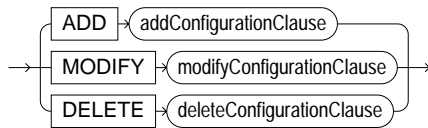
alterViewConstraintClauses



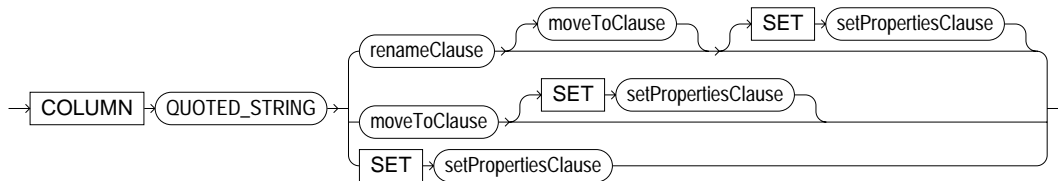
addConfigurationClause



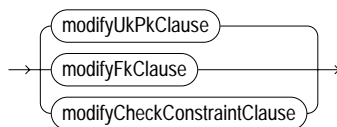
alterViewConfigurationClauses



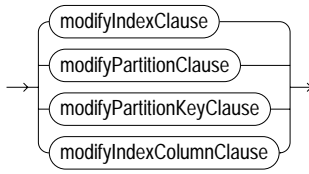
modifyColumnClause



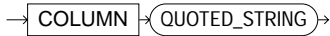
modifyConstraintClause



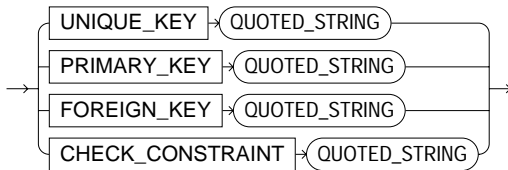
modifyConfigurationClause



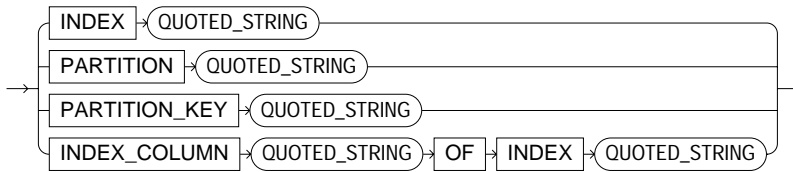
deleteColumnClause



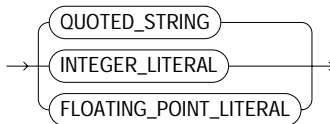
deleteConstraintClause



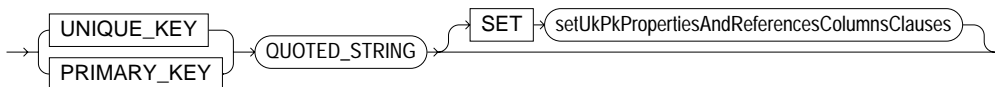
deleteConfigurationClause



property Value



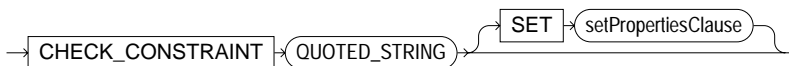
addUkPkClause



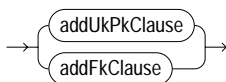
addFkClause



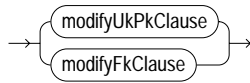
addCheckConstraintClause



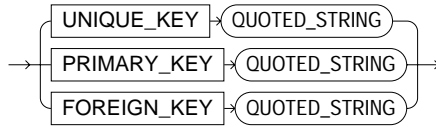
addViewConstraintClause



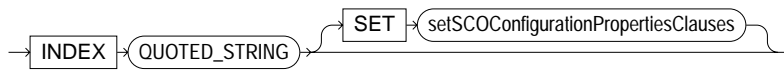
modifyViewConstraintClause



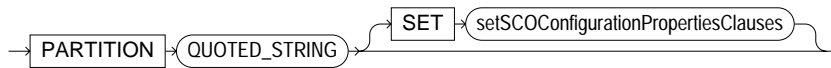
deleteViewConstraintClause



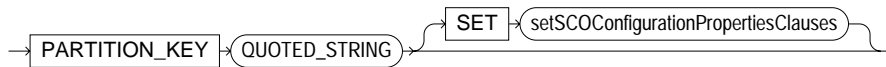
addIndexClause



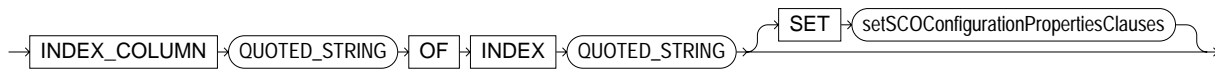
addPartitionClause



addPartitionKeyClause



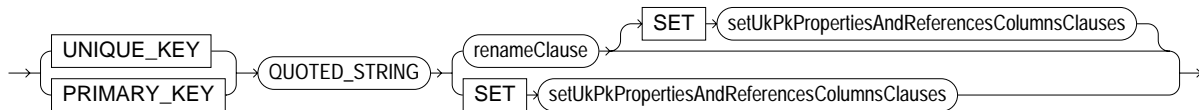
addIndexColumnClause



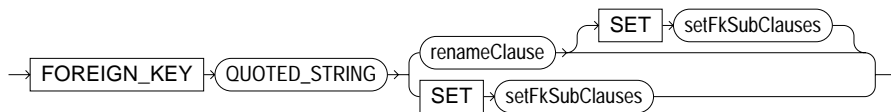
moveToClause



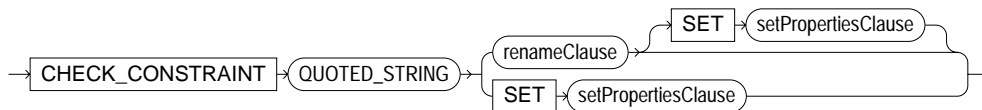
modifyUkPkClause



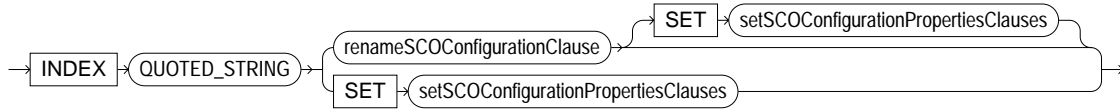
modifyFkClause



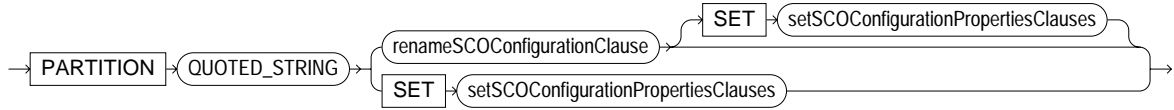
modifyCheckConstraintClause



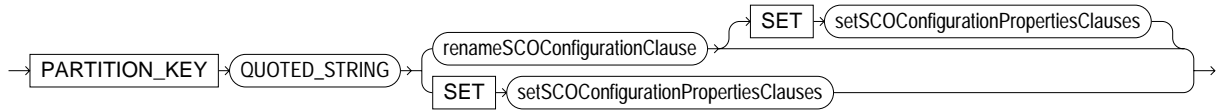
modifyIndexClause



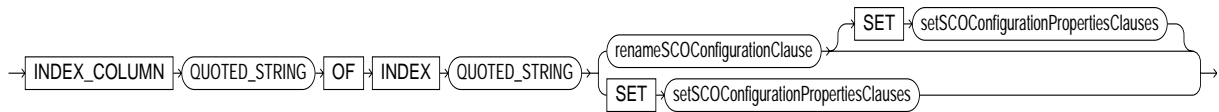
modifyPartitionClause



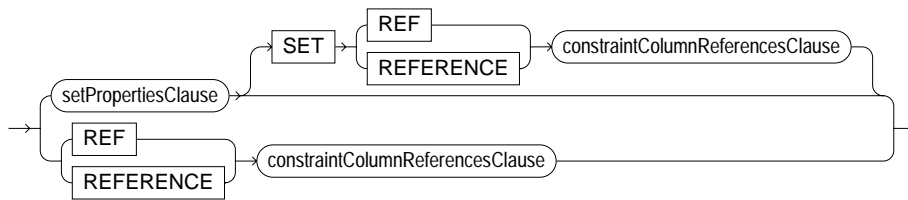
modifyPartitionKeyClause



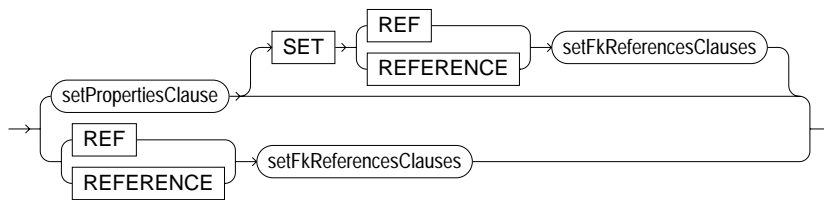
modifyIndexColumnClause



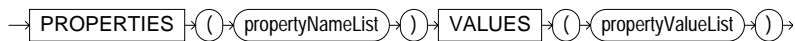
setUkPkPropertiesAndReferencesColumnsClauses



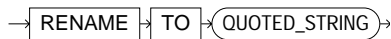
setFkSubClauses



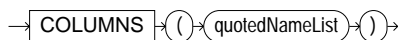
setSCOConfigurationPropertiesClauses



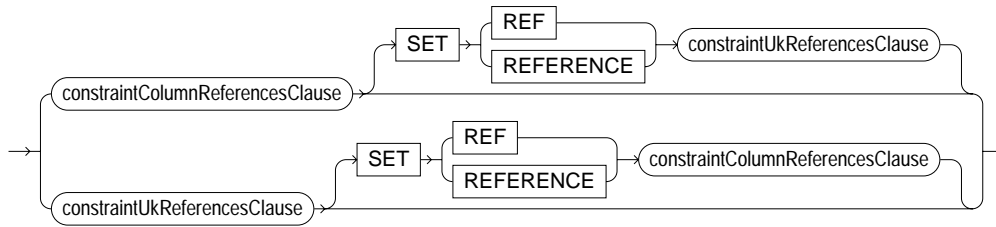
renameSCOConfigurationClause



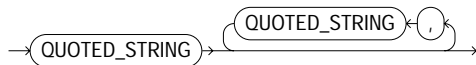
constraintColumnReferencesClause



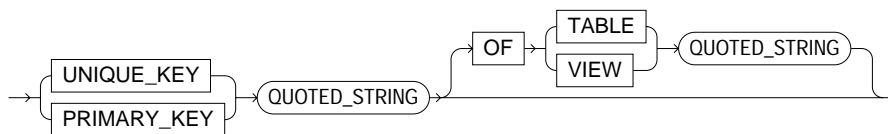
setFkReferencesClauses



quotedNameList



constraintUkReferencesClause



## Syntax

```

alterMaterializedViewCommand = OMBALTER ( MATERIALIZED_VIEW
"QUOTED_STRING" ( "renameClause" [ SET "setPropertiesClause" ] [
"alterViewSCOClauses" ] | SET "setPropertiesClause" [ "alterViewSCOClauses" ] |
"alterViewSCOClauses" ));

renameClause = RENAME TO "QUOTED_STRING";

setPropertiesClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("
"propertyValueList" ")";

alterViewSCOClauses = ADD ( "addColumnClauseForAlter" [ "alterViewSCOClauses"
] | "addConstraintClause" { "alterViewConstraintClauses" } |
"addConfigurationClause" { "alterViewConfigurationClauses" } ) | MODIFY (
"modifyColumnClause" [ "alterViewSCOClauses" ] | "modifyConstraintClause" {
"alterViewConstraintClauses" } | "modifyConfigurationClause" {
"alterViewConfigurationClauses" } ) | DELETE ( "deleteColumnClause" [
"alterViewSCOClauses" ] | "deleteConstraintClause" { "alterViewConstraintClauses" }
| "deleteConfigurationClause" { "alterViewConfigurationClauses" } );

propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };

propertyValueList = "propertyValue" { "," "propertyValue" };

addColumnClauseForAlter = COLUMN "QUOTED_STRING" [ AT POSITION
"INTEGER_LITERAL" ] [ SET "setPropertiesClause" ];

addConstraintClause = "addUkPkClause" | "addFkClause" |
"addCheckConstraintClause";

alterViewConstraintClauses = ADD "addViewConstraintClause" | MODIFY
"modifyViewConstraintClause" | DELETE "deleteViewConstraintClause";

addConfigurationClause = "addIndexClause" | "addPartitionClause" |
"addPartitionKeyClause" | "addIndexColumnClause";
    
```



```

alterViewConfigurationClauses = ADD "addConfigurationClause" | MODIFY
"modifyConfigurationClause" | DELETE "deleteConfigurationClause";

modifyColumnClause = COLUMN "QUOTED_STRING" ( "renameClause" [
"moveToClause" ] | SET "setPropertiesClause" ] | "moveToClause" [ SET
"setPropertiesClause" ] | SET "setPropertiesClause" );

modifyConstraintClause = "modifyUkPkClause" | "modifyFkClause" |
"modifyCheckConstraintClause";

modifyConfigurationClause = "modifyIndexClause" | "modifyPartitionClause" |
"modifyPartitionKeyClause" | "modifyIndexColumnClause";

deleteColumnClause = COLUMN "QUOTED_STRING";

deleteConstraintClause = UNIQUE_KEY "QUOTED_STRING" | PRIMARY_KEY
"QUOTED_STRING" | FOREIGN_KEY "QUOTED_STRING" | CHECK_
CONSTRAINT "QUOTED_STRING";

deleteConfigurationClause = INDEX "QUOTED_STRING" | PARTITION "QUOTED_
STRING" | PARTITION_KEY "QUOTED_STRING" | INDEX_COLUMN "QUOTED_
STRING" OF INDEX "QUOTED_STRING";

propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );

addUkPkClause = ( UNIQUE_KEY | PRIMARY_KEY ) "QUOTED_STRING" [ SET
"setUkPkPropertiesAndReferencesColumnsClauses" ];

addFkClause = FOREIGN_KEY "QUOTED_STRING" [ SET "setFkSubClauses" ];

addCheckConstraintClause = CHECK_CONSTRAINT "QUOTED_STRING" [ SET
"setPropertiesClause" ];

addViewConstraintClause = "addUkPkClause" | "addFkClause";

modifyViewConstraintClause = "modifyUkPkClause" | "modifyFkClause";

deleteViewConstraintClause = UNIQUE_KEY "QUOTED_STRING" | PRIMARY_
KEY "QUOTED_STRING" | FOREIGN_KEY "QUOTED_STRING";

addIndexClause = INDEX "QUOTED_STRING" [ SET
"setSCOConfigurationPropertiesClauses" ];

addPartitionClause = PARTITION "QUOTED_STRING" [ SET
"setSCOConfigurationPropertiesClauses" ];

addPartitionKeyClause = PARTITION_KEY "QUOTED_STRING" [ SET
"setSCOConfigurationPropertiesClauses" ];

addIndexColumnClause = INDEX_COLUMN "QUOTED_STRING" OF INDEX
"QUOTED_STRING" [ SET "setSCOConfigurationPropertiesClauses" ];

moveToClause = MOVE TO POSITION "INTEGER_LITERAL";

modifyUkPkClause = ( UNIQUE_KEY | PRIMARY_KEY ) "QUOTED_STRING" (
"renameClause" [ SET "setUkPkPropertiesAndReferencesColumnsClauses" ] | SET
"setUkPkPropertiesAndReferencesColumnsClauses" );

modifyFkClause = FOREIGN_KEY "QUOTED_STRING" ( "renameClause" [ SET
"setFkSubClauses" ] | SET "setFkSubClauses" );

modifyCheckConstraintClause = CHECK_CONSTRAINT "QUOTED_STRING" (
"renameClause" [ SET "setPropertiesClause" ] | SET "setPropertiesClause" );

```

```
modifyIndexClause = INDEX "QUOTED_STRING" (  
"renameSCOConfigurationClause" [ SET "setSCOConfigurationPropertiesClauses" ] |  
SET "setSCOConfigurationPropertiesClauses" );  
  
modifyPartitionClause = PARTITION "QUOTED_STRING" (  
"renameSCOConfigurationClause" [ SET "setSCOConfigurationPropertiesClauses" ] |  
SET "setSCOConfigurationPropertiesClauses" );  
  
modifyPartitionKeyClause = PARTITION_KEY "QUOTED_STRING" (  
"renameSCOConfigurationClause" [ SET "setSCOConfigurationPropertiesClauses" ] |  
SET "setSCOConfigurationPropertiesClauses" );  
  
modifyIndexColumnClause = INDEX_COLUMN "QUOTED_STRING" OF INDEX  
"QUOTED_STRING" ( "renameSCOConfigurationClause" [ SET  
"setSCOConfigurationPropertiesClauses" ] | SET  
"setSCOConfigurationPropertiesClauses" );  
  
setUkPkPropertiesAndReferencesColumnsClauses = "setPropertiesClause" [ SET (  
REF | REFERENCE) "constraintColumnReferencesClause" ] | ( REF | REFERENCE )  
"constraintColumnReferencesClause";  
  
setFkSubClauses = "setPropertiesClause" [ SET ( REF | REFERENCE )  
"setFkReferencesClauses" ] | ( REF | REFERENCE ) "setFkReferencesClauses";  
  
setSCOConfigurationPropertiesClauses = PROPERTIES "(" "propertyNameList" ")"  
VALUES "(" "propertyValueList" )";  
  
renameSCOConfigurationClause = RENAME TO "QUOTED_STRING";  
  
constraintColumnReferencesClause = COLUMNS "(" "quotedNameList" )";  
  
setFkReferencesClauses = "constraintColumnReferencesClause" [ SET ( REF |  
REFERENCE ) "constraintUkReferencesClause" ] | "constraintUkReferencesClause" [  
SET ( REF | REFERENCE ) "constraintColumnReferencesClause" ];  
  
quotedNameList = "QUOTED_STRING" { "," "QUOTED_STRING" };  
  
constraintUkReferencesClause = ( UNIQUE_KEY | PRIMARY_KEY ) "QUOTED_  
STRING" [ OF ( TABLE | VIEW ) "QUOTED_STRING" ];
```

## Keywords and Parameters

`alterMaterializedViewCommand`

This clause alters a materialized view.

`QUOTED_STRING`

name of the materialized view.

`renameClause`

renames a table with a different name.

`setPropertiesClause`

Used to set properties (core, logical, physical, user-defined) for tables, columns, unique keys, foreign keys, primary keys, and check constraints.

---

Basic properties for VIEW, MATERIALIZED\_VIEW:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the View, MaterializedView

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the View, MaterializedView

Name: VIEW\_QUERY

Type: STRING(4000)

Valid Values: N/A

Default: "

Sets the query definition in View and MaterializedView.

Name: UOID

Type: STRING

Valid Values: N/A

Default: N/A

Basic properties for INDEX, PARTITION, PARTITION\_KEY, INDEX\_COLUMN:

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the Index, Partition, PartitionKey, IndexColumn in a MaterializedView.

Properties for MATERIALIZED\_VIEW:

Name: BASE\_TABLES

Type: STRING

Valid Values: N/A

Default: "

Comma separated list of base tables.

Name: BUILD

Type: STRING

Valid Values: DEFERRED, IMMEDIATE

Default: IMMEDIATE

Immediate : populates the view when it is created. Deferred : delays population until the next refresh operation.

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: HASH\_PARTITION\_TABLESPACE\_LIST

Type: STRING

Valid Values: N/A

Default: "

A comma separated list of tablespaces to use for [sub]partition storage.

Name: HASH\_SUBPARTITION\_NUMBER

Type: NUMBER

Valid Values: 2 - 63999

Default: 2

Hash SubPartition Number

Name: LOGGING\_MODE

Type: STRING

Valid Values: NOLOGGING, LOGGING

Default: LOGGING

Recovery requirements for a data warehouse : Logging or not logging to Redo Log File.

Name: PARALLEL\_ACCESS\_MODE

Type: STRING

Valid Values: NOPARALLEL, PARALLEL

Default: PARALLEL

Enables or disables parallel processing when the table is created. Also enables or disables parallel processing or access.

Name: QUERY\_REWRITE

Type: STRING

Valid Values: DISABLE, ENABLE

Default: ENABLE

Enable marks the View eligible for query rewrite and disable marks the View ineligible for query rewrite

Name: REFRESH

Type: STRING

Valid Values: COMPLETE, FAST, ON\_COMMIT, ON\_DEMAND, FORCE

Default: COMPLETE

Complete : specifies the complete refresh method implemented by executing the query of the view. Fast : specifies the incremental refresh method which refreshes the view according to changes that have occurred to the master tables. Force : specifies that when a refresh occurs, Oracle performs a fast refresh if possible or a complete refresh otherwise.

Name: TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Use the Tablespace parameter to specify the name of tablespace.

Properties for UNIQUE\_KEY:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: USING\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

Use of Index on Dimension Constraint

Properties for FOREIGN\_KEY:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: USING\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

Use of Index on Dimension Constraint

Properties for CHECK\_CONSTRAINT:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: USING\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

Use of Index on Dimension Constraint

Properties for INDEX:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: INDEX\_TYPE

Type: STRING

Valid Values: BITMAP, UNIQUE, NO\_INDEX

Default: UNIQUE

The types of Indexes created on Dimension are BITMAP, UNIQUE or a non-specific index.

Name: LOCAL\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

A local index is constructed so that it reflects the structure of the underlying table. It is equipartitioned with the underlying table, meaning that it is partitioned on the same columns as the underlying table, creates the same number of partitions or subpartitions, and gives them the same partition bounds as corresponding partitions of the underlying table.

Name: LOGGING\_MODE

Type: STRING

Valid Values: NOLOGGING, LOGGING

Default: LOGGING

Recovery requirements for a data warehouse : Logging or not logging to Redo Log File.

Name: PARALLEL\_ACCESS\_MODE

Type: STRING

Valid Values: NOPARALLEL, PARALLEL

Default: PARALLEL

Enables or disables parallel processing when the table is created. Also enables or disables parallel processing or access.

Name: TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Use the Tablespace parameter to specify the name of tablespace.

Name: USING\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

Use of Index on Dimension Constraint

Properties for PARTITION:



Name: DATE\_LESS\_THAN

Type: STRING

Valid Values: N/A

Default: TO\_DATE('01-JAN-1999', 'DD-MON-YYYY')

Value that represents upper bound of partition stored in warehouse key column for the Days Dimension.

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: EMPTY\_STRING

Use the Tablespace parameter to specify the name of tablespace.

Name: VALUES\_LESS\_THAN

Type: STRING

Valid Values: N/A

Default: TO\_DATE('01-JAN-1999', 'DD-MON-YYYY')

Noninclusive upper bound for the current partition. Type a comma separated list in the same order as the columns in range partitioned key.

Name: VALUE\_LESS\_THAN

Type: STRING

Valid Values: N/A

Default: "

Noninclusive upper bound for the current partition. Type a comma separated list in the same order as the columns in range partitioned key.

Properties for PARTITION\_KEY:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: TYPE

Type: STRING

Valid Values: HASH, RANGE

Default: RANGE

Oracle partitions the storage space and stores rows according to a Hash Algorithm or specified ranges.

Note:

1. N/A means any valid character in supported character set.
2. " " represents an empty string

alterViewSCOClauses

This clause alters the view clause.

propertyNameList

The list of properties.

propertyValueList

The list of property values.

addColumnClauseForAlter

This clause adds a column at a particular position

QUOTED\_STRING

The column name.

addConstraintClause

Adds primary and unique key, and add check constraints.

alterViewConstraintClauses

This clause alters the view's constraint clause.

addConfigurationClause

---

This clause will add configuration objects.

`alterViewConfigurationClauses`

This clause alters the view's configuration clause.

`modifyColumnClause`

This clause renames, set properties, and move columns.

`modifyConstraintClause`

This clause modifies keys and check constraints

`modifyConfigurationClause`

This clause will modify configuration objects.

`deleteColumnClause`

This clause deletes a column.

`deleteConstraintClause`

This clause deletes a key or check constraint.

`deleteConfigurationClause`

This clause deletes a configuration object.

`QUOTED_STRING`

Either index, partition, partition\_key, or index column name.

`propertyValue`

This clause adds the property values.

`addUkPkClause`

This clause adds the adds unique key and primary keys.

`QUOTED_STRING`

name of the unique key or primary key.

`addFkClause`

This clause adds foreign key

QUOTED\_STRING

Name of the foreign key.

addCheckConstraintClause

add a check constraint.

QUOTED\_STRING

Name of the CheckConstraint.

addViewConstraintClause

This clause adds the view's configuration clause.

modifyViewConstraintClause

= This clause modifies the view's constraint clause.

deleteViewConstraintClause

This clause deletes the view's constraint.

addIndexClause

This clause adds an index.

QUOTED\_STRING

Name of the index.

addPartitionClause

This clause adds a partition.

QUOTED\_STRING

Name of the partition.

addPartitionKeyClause

This clause adds a partition key.

QUOTED\_STRING

Name of the partition key. This should be a column identifier.

**addIndexColumnClause**

This clause will add indexColumn to a specified index.

**QUOTED\_STRING**

Index name

**moveToClause**

This clause will move the column to given position.

**modifyUkPkClause**

It modifies unique or primary key.

**modifyFkClause**

This clause modifies the foreign key.

**modifyCheckConstraintClause**

This clause modifies the check constraint.

**modifyIndexClause**

This clause modifies the Index

**QUOTED\_STRING**

Name of the index.

**modifyPartitionClause**

This clause modifies a partition

**QUOTED\_STRING**

Name of the partition.

**modifyPartitionKeyClause**

This clause modifies a partition key.

**QUOTED\_STRING**

Name of the partition key.

**modifyIndexColumnClause**

Modifies the Index Column. The first quoted\_string in this clause denotes index column name, and the latter denotes index.

setUkPkPropertiesAndReferencesColumnsClauses

This clause adds properties and references to columns

setFkSubClauses

This clause set references to a foreign key.

setSCOConfigurationPropertiesClauses

Set the configuration properties for the following objects Index: LOGGING\_MODE, PARALLEL\_ACCESS\_MODE, TABLESPACE, INDEX\_TYPE, LOCAL\_INDEX, DEPLOYABLE Partition: DATE\_LESS\_THAN, TABLESPACE, DEPLOYABLE Partition\_key: TYPE, DEPLOYABLE RelationalCmdParser\$constraintColumnReferencesClause = This clause provides names of all columns.

renameSCOConfigurationClause

This clause renames configuration objects.

constraintColumnReferencesClause

RelationalCmdParser\$constraintColumnReferencesClause??

setFkReferencesClauses

This clause sets foreign key references.

quotedNameList

This clause gives column names.

constraintUkReferencesClause

The first QUOTED\_STRING denotes the UniqueKey or Primay key name, and the latter denotes the table's or view's name.

## Examples

```
OMBALTER MATERIALIZED_VIEW 'NEW_MATERIALIZED_VIEW' SET
PROPERTIES (DESCRIPTION, BUSINESS_NAME) VALUES ('this is an altered desc of
new materialized view', 'Altered New MaterializedView') This will alter a
materialized view named "NEW_MATERIALIZED_VIEW", its description is "this is an
altered desc of new materialized view", and business name is "Altered New
MaterializedView".
```

## See Also

OMBALTER, OMBCREATE MATERIALIZED\_VIEW, OMBDROP MATERIALIZED\_VIEW

# OMBALTER OBJECT\_TYPE

## Purpose

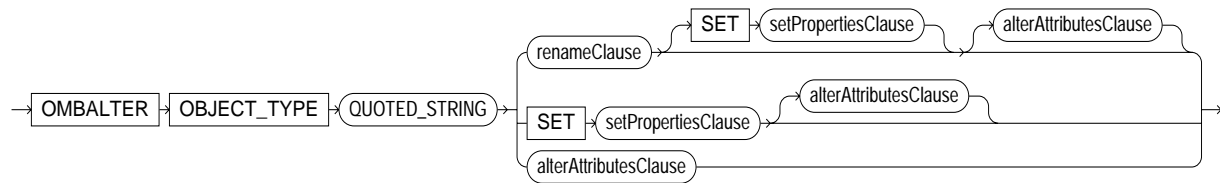
OMBALTER OBJECT\_TYPE - Alter the Object Type by resetting its properties or adding/removing its attributes.

## Prerequisites

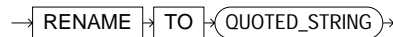
Should be in the context of an Oracle Module.

## Syntax Diagrams

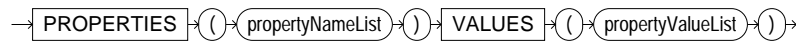
alterObjectTypeCommand



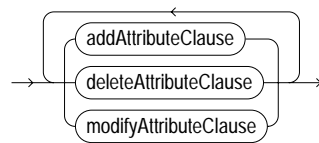
renameClause



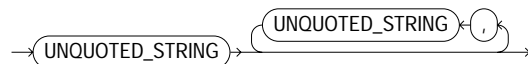
setPropertiesClause



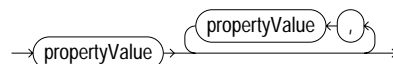
alterAttributesClause



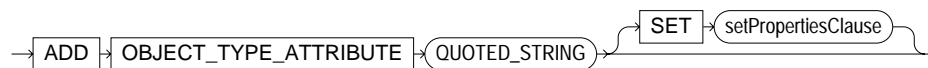
propertyNameList



propertyValueList



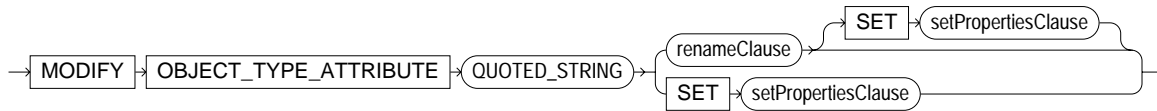
addAttributeClause



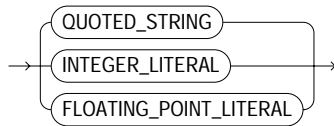
deleteAttributeClause



modifyAttributeClause



propertyValue



## Syntax

```

alterObjectTypeCommand = OMBALTER ( OBJECT_TYPE "QUOTED_STRING" (
"renameClause" [ SET "setPropertiesClause" ] [ "alterAttributesClause" ] | SET
"setPropertiesClause" [ "alterAttributesClause" ] | "alterAttributesClause" ));

renameClause = RENAME TO "QUOTED_STRING";

setPropertiesClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("
"propertyValueList" " )";

alterAttributesClause = ( "addAttributeClause" | "deleteAttributeClause" |
"modifyAttributeClause" )+;

propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };

propertyValueList = "propertyValue" { "," "propertyValue" };

addAttributeClause = ADD OBJECT_TYPE_ATTRIBUTE "QUOTED_STRING" [ SET
"setPropertiesClause" ];

deleteAttributeClause = DELETE OBJECT_TYPE_ATTRIBUTE "QUOTED_STRING";

modifyAttributeClause = MODIFY OBJECT_TYPE_ATTRIBUTE "QUOTED_STRING"
( "renameClause" [ SET "setPropertiesClause" ] | SET "setPropertiesClause" );

propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );
    
```

## Keywords and Parameters

**alterObjectTypeCommand**

Alters an Object Type of the given name by either renaming it, or by setting its properties or by modifying one or more of its Attributes or a combination of these.

**renameClause**

renames a table with a different name.

**setPropertiesClause**



---

Sets properties (core, logical, physical, user-defined) for Object Type or its Attributes.

Basic properties for OBJECT\_TYPE:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the Object Type

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the Object Type

Basic properties for OBJECT\_TYPE\_ATTRIBUTE:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the Attribute

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the Attribute

Name: DATATYPE

Type: STRING(20)

Valid Values: NUMBER, VARCHAR2, VARCHAR, DATE, FLOAT

Default: "

Datatype of the Attribute

Note:

1. N/A means any valid character in supported character set.
2. " represents an empty string

**alterAttributesClause**

Adds, deletes or modifies one or more Attributes of this Object Type.

**propertyNameList**

The list of properties.

**propertyValueList**

The list of property values.

**addAttributeClause**

Adds an Attribute with the given name and properties.

**deleteAttributeClause**

Deletes an Attribute with the given name.

**modifyAttributeClause**

Modifies an Attribute with the given name by either renaming it or changing its properties or both.

**propertyValue**

This clause adds the property values.

**Examples**

```
OMBALTER OBJECT_TYPE 'SOME_OBJECT_TYPE' SET PROPERTIES  
(DESCRIPTION) VALUES
```

```
('This is will be used as Payload type for an AQ.') DELETE OBJECT_TYPE_  
ATTRIBUTE 'ATTR' ADD OBJECT_TYPE_ATTRIBUTE 'ATTR1' SET PROPERTIES  
(DATATYPE) VALUES ('NUMBER') This will set its description to "This is will be  
used as Payload type for an AQ.", remove attribute 'ATTR' and add an attribute  
"ATTR1" of Number type.
```

**See Also**

```
ALTER, OMBRETRIEVE OBJECT_TYPE, OMBCREATE OBJECT_TYPE, OMBDROP  
OBJECT_TYPE
```

# OMBALTER ORACLE\_MODULE

## Purpose

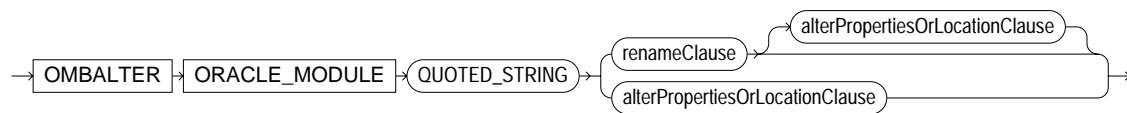
OMBALTER ORACLE\_MODULE - Alter the Oracle module by renaming it, and/or reset its properties.

## Prerequisites

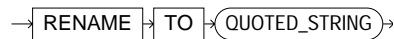
Should be in the context of project.

## Syntax Diagrams

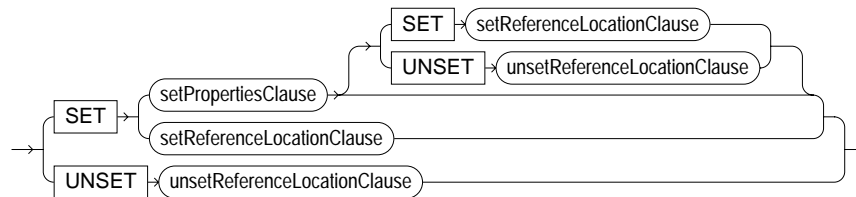
alterOracleModuleCommand



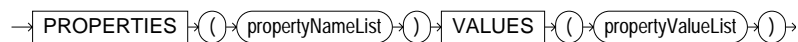
renameClause



alterPropertiesOrLocationClause



setPropertiesClause



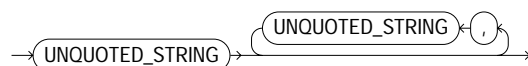
setReferenceLocationClause



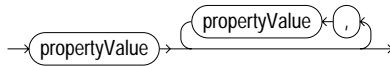
unsetReferenceLocationClause



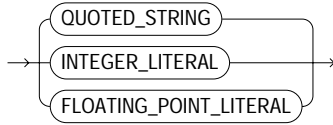
propertyNameList



propertyValueList



propertyValue



## Syntax

```

alterOracleModuleCommand = OMBALTER ( ORACLE_MODULE "QUOTED_
STRING" ("renameClause" [ "alterPropertiesOrLocationClause" ] |
"alterPropertiesOrLocationClause" ) );

renameClause = RENAME TO "QUOTED_STRING";

alterPropertiesOrLocationClause = SET ( "setPropertiesClause" [ SET
"setReferenceLocationClause" | UNSET "unsetReferenceLocationClause" ] |
"setReferenceLocationClause" ) | UNSET "unsetReferenceLocationClause";

setPropertiesClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("
"propertyValueList" ")";

setReferenceLocationClause = ( REFERENCE | REF ) LOCATION "QUOTED_
STRING";

unsetReferenceLocationClause = ( REFERENCE | REF ) LOCATION "QUOTED_
STRING";

propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };

propertyValueList = "propertyValue" { "," "propertyValue" };

propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );

```

## Keywords and Parameters

**alterOracleModuleCommand**

This command modifies an existing Oracle module.

**QUOTED\_STRING**

Name of the existing Oracle module in single quotes.

**renameClause**

Rename an Oracle module.

**alterPropertiesOrLocationClause**

Alter existing Oracle module's properties and/or location.

**setPropertiesClause**

Associate a set of properties with the existing Oracle module.

**Basic properties for ORACLE\_MODULE:**

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: NAME

Business name of an Oracle Module

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of an Oracle Module

Name: UOID

Type: STRING(40)

Valid Values: N/A

Default: N/A

UOID of an Oracle Module

**Properties for ORACLE\_MODULE:**

Name: ABAP\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: abap\

Location where ABAP scripts are stored

Name: ABAP\_EXTENSION

Type: STRING

Valid Values: N/A

Default: .abap

File name extension for ABAP scripts

Name: ABAP\_RUN\_PARAMETER\_FILE

Type: STRING

Valid Values: N/A

Default: \_run.ini

Run Parameter File Suffix for the parameter script in a ABAP job.

Name: ABAP\_SPOOL\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: abap\log\

Location where ABAP scripts are buffered during script generation processing.

Name: APPLICATION\_SHORT\_NAME

Type: STRING

Valid Values: N/A

Default: WB

Application Short Name

Name: ARCHIVE\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: archive\

Archive Directory

Name: CONNECT\_STRING

Type: STRING

Valid Values: N/A

Default: "

A Net\*8 style connection string to the remote database. Alternatively, you can specify machine, port, service name of the remote database.

Name: DDL\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: ddl\

Location where scripts for database objects for the target schema are stored.

Name: DDL\_EXTENSION  
Type: STRING  
Valid Values: N/A  
Default: .ddl  
File name extension for DDL scripts.

Name: DDL\_SPOOL\_DIRECTORY  
Type: STRING  
Valid Values: N/A  
Default: ddl\log\  
Location where DDL scripts are buffered during script generation processing.

Name: DEFAULT\_INDEX\_TABLESPACE  
Type: STRING(30)  
Valid Values: N/A  
Default: "  
Default name of tablespace to install indexes into.

Name: DEFAULT\_OBJECT\_TABLESPACE  
Type: STRING(30)  
Valid Values: N/A  
Default: "  
Default name of tablespace to install objects into.

Name: DEPLOYABLE  
Type: BOOLEAN  
Valid Values: true, false  
Default: true  
Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: END\_OF\_LINE  
Type: STRING  
Valid Values: N/A  
Default: \r\n  
End of Line

Name: INPUT\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: input\

Input Directory

Name: INVALID\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: invalid\

Directory for SQL\*Loader errors and rejected records

Name: LIB\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: lib\

LIB Directory

Name: LIB\_EXTENSION

Type: STRING

Valid Values: N/A

Default: .lib

LIB Extension

Name: LIB\_SPOOL\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: lib\log\

LIB Spool Directory

Name: LOADER\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: ctl\

Location where control files are stored.



Name: LOADER\_EXTENSION

Type: STRING

Valid Values: N/A

Default: .ctl

Suffix for the loader scripts

Name: LOADER\_RUN\_PARAMETER\_FILE

Type: STRING

Valid Values: N/A

Default: \_run.ini

Suffix for the parameter initialization file.

Name: LOG\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: log\

Log Directory for the SQL\*Loader

Name: MAIN\_APPLICATION\_SHORT\_NAME

Type: STRING

Valid Values: N/A

Default: ora

Main Application Short Name

Name: PLSQL\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: pls\

Location where PL/SQL scripts are stored.

Name: PLSQL\_EXTENSION

Type: STRING

Valid Values: N/A

Default: .pls

File name extension for PL/SQL scripts.

Name: PLSQL\_GENERATION\_MODE

Type: STRING

Valid Values: Oracle9i, Oracle8i

Default: Oracle9i

Generation mode controls validation and generation for version specific features.

Name: PLSQL\_RUN\_PARAMETER\_FILE

Type: STRING

Valid Values: N/A

Default: \_run.ini

Suffix for the parameter script in a PL/SQL job.

Name: PLSQL\_SPOOL\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: pls\log\

Location where PL/SQL scripts are buffered during script generation processing.

Name: PORT

Type: STRING

Valid Values: N/A

Default: "

The port number on the machine where the database listens to.

Name: RECEIVE\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: receive\

Receive Directory

Name: REMOTE\_HOST\_NAME

Type: STRING

Valid Values: N/A

Default: "

The machine where the remote database resides on.

Name: SCHEMA\_OWNER

Type: STRING

Valid Values: N/A

Default: OWB

Schema Owner

Name: SERVICE\_NAME

Type: STRING

Valid Values: N/A

Default: "

The service name (global DB name) of the database instance on the remote machine.

Name: SORT\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: sort\

Sort Directory

Name: TCL\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: tcl\

Location for TCL scripts that are generated after registration with Oracle Enterprise Manager

Name: TOP\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: ..\..\codegen\

Top Directory where generated code will get stored

Name: WORK\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: work\

Work Directory

Note:

1. N/A means any valid character in supported character set.

2. " represents an empty string

`setReferenceLocationClause`

Set a location to the existing Oracle module.

`unsetReferenceLocationClause`

Unset a location to the existing Oracle module.

`propertyNameList`

Comma separated list of property names. Property names are unquoted.

`propertyValueList`

Comma separated list of property values.

`propertyValue`

Value of a property.

## Examples

```
OMBALTER ORACLE_MODULE 'src_module' RENAME TO 'tgt_module' SET
PROPERTIES (DESCRIPTION, BUSINESS_NAME) VALUES ('This becomes a target
module.', 'target module') This will rename the Oracle module "src_module" to "tgt_
module", and set its description to "This becomes a target module.", set its business
name to "target module".
```

## See Also

OMBALTER, OMBCREATE ORACLE\_MODULE, OMBDROP ORACLE\_MODULE

## OMBALTER PACKAGE

### Purpose

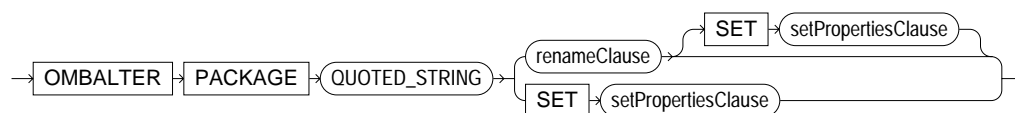
OMBALTER PACKAGE - Alter the Package by renaming it, and/or reset its properties, and/or adding/deleting/modifying the user types.

### Prerequisites

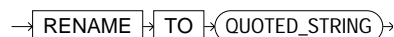
Should be in the context of a Oracle Module or Transformation Module.

### Syntax Diagrams

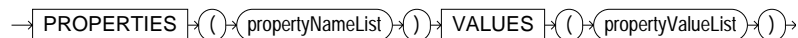
alterPackageCommand



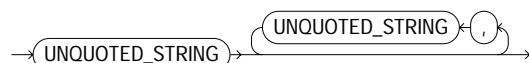
renameClause



setPropertiesClause



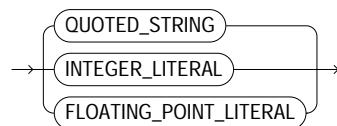
propertyNameList



propertyValueList



propertyValue



### Syntax

```
alterPackageCommand = OMBALTER ( PACKAGE "QUOTED_STRING" (
"renameClause" [ SET "setPropertiesClause" ] | SET "setPropertiesClause" ) );
```

```
renameClause = RENAME TO "QUOTED_STRING";
```

```
setPropertyClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("  
"propertyValueList" ");  
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };  
propertyValueList = "propertyValue" { "," "propertyValue" };  
propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_  
POINT_LITERAL" );
```

## Keywords and Parameters

**alterPackageCommand**

This command modifies an existing Package.

**QUOTED\_STRING**

Name of the existing Package in single quotes.

**renameClause**

Rename a Package.

**setPropertyClause**

Used to set properties (core, user-defined) for packages. Valid properties are shown below:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the package

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the package

Name: PACKAGE\_BODY

Type: STRING

Valid Values: N/A

Default: "

Sets the Package Body for a Imported Package

**Properties for PACKAGE:**

Name: AUTHID

Type: STRING

Valid Values: None, Current\_User, Definer

Default: None

Generate the package with selected AUTHID option. Function will be executed with the permissions defined by the AUTHID clause rather than the function owner's permissions.

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

**Note:**

1. N/A means any valid character in supported character set.
2. " " represents an empty string

propertyNameList

Comma separated list of property names. Property names are unquoted.

propertyValueList

Comma separated list of property values.

propertyValue

Value of a property.

**Examples**

```
OMBALTER PACKAGE 'pkg' RENAME TO 'package_1' SET PROPERTIES  
(DESCRIPTION, BUSINESS_NAME) VALUES ('This becomes a package_1', 'package_  
1') This will rename the Package "pkg" to "package_1", and set its description to "This  
becomes a package_1", set its business name to "package_1"
```

**See Also**

OMBALTER, OMBCREATE PACKAGE, OMBDROP PACKAGE

# OMBALTER PROCEDURE

## Purpose

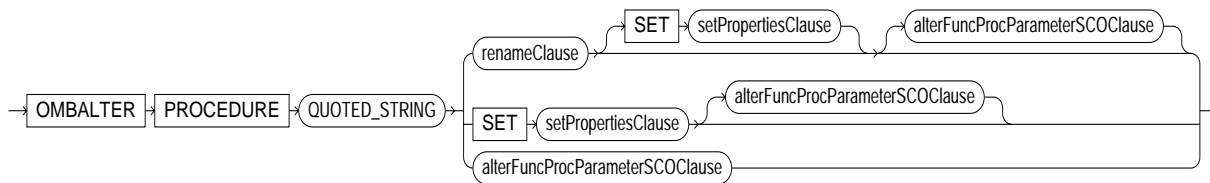
OMBALTER PROCEDURE - Alter the Procedure by renaming it, and/or reset its properties.

## Prerequisites

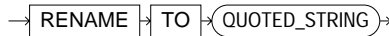
Should be in the context of a Oracle Module or Package or Transformation Module.

## Syntax Diagrams

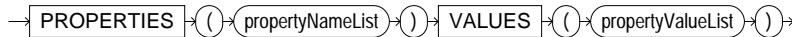
alterProcedureCommand



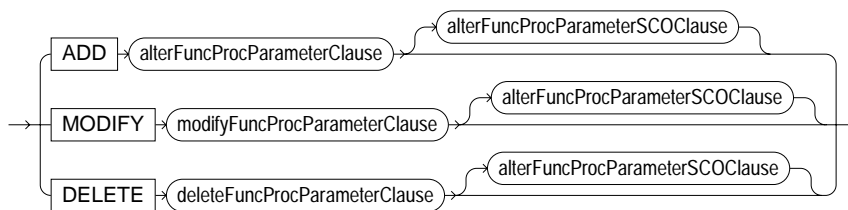
renameClause



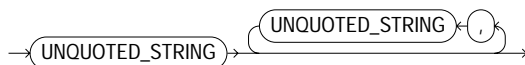
setPropertiesClause



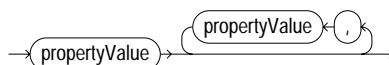
alterFuncProcParameterSCOClause



propertyNameList



propertyValueList

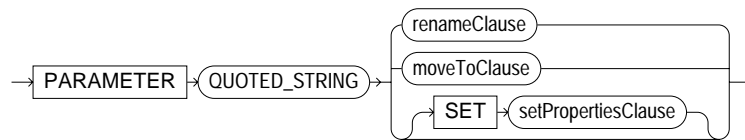


alterFuncProcParameterClause





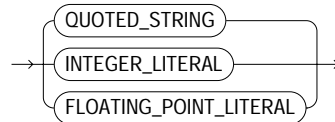
modifyFuncProcParameterClause



deleteFuncProcParameterClause



propertyValue



moveToClause



## Syntax

```
alterProcedureCommand = OMBALTER ( PROCEDURE "QUOTED_STRING" (
"renameClause" [ SET "setPropertiesClause" ] [ "alterFuncProcParameterSCOClaue" ]
| SET "setPropertiesClause" [ "alterFuncProcParameterSCOClaue" ] |
"alterFuncProcParameterSCOClaue" ) );
```

```
renameClause = RENAME TO "QUOTED_STRING";
```

```
setPropertiesClause = PROPERTIES "(" "propertyNameList ")" VALUES "("
"propertyValueList ")";
```

```
alterFuncProcParameterSCOClaue = ( ADD "alterFuncProcParameterClause" [
"alterFuncProcParameterSCOClaue" ] | MODIFY "modifyFuncProcParameterClause"
[ "alterFuncProcParameterSCOClaue" ] | DELETE "deleteFuncProcParameterClause"
[ "alterFuncProcParameterSCOClaue" ] );
```

```
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
```

```
propertyValueList = "propertyValue" { "," "propertyValue" };
```

```
alterFuncProcParameterClause = PARAMETER "QUOTED_STRING" [ AT POSITION
"INTEGER_LITERAL" ] [ SET "setPropertiesClause" ];
```

```
modifyFuncProcParameterClause = ( PARAMETER "QUOTED_STRING" (
"renameClause" | "moveToClause" | [ SET "setPropertiesClause" ] ) );
```

```
deleteFuncProcParameterClause = ( PARAMETER "QUOTED_STRING" );
```

```
propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );
```

```
moveToClause = MOVE TO POSITION "INTEGER_LITERAL";
```

## Keywords and Parameters

alterProcedureCommand

This command modifies an existing Procedure.

**QUOTED\_STRING**

Name of the existing Procedure in single quotes.

**renameClause**

Rename a Procedure.

**setPropertyClause**

Used to set properties (core, user-defined) for procedure. Valid properties are shown below:

**Basic properties for PROCEDURE:**

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the Procedure

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the Procedure

Name: IMPLEMENTATION

Type: STRING

Valid Values: N/A

Default: "

Set the code for Procedure which is included global variable declaration and code between BEGIN and END.

**Basic properties for PARAMETER:**

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the Parameter

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the Parameter

Name: DATATYPE

Type: STRING

Valid Values: PLS\_INTEGER, BINARY\_INTEGER, BOOLEAN, NUMBER, FLOAT, CHAR, VARCHAR, VARCHAR2, DATE

Default: NUMBER

Set the data type for Parameter

Name: IN\_OUT

Type: STRING

Valid Values: IN, OUT, INOUT

Default: 'IN'

Set the parameter mode for Parameter

Name: DEFAULT\_VALUE

Type: STRING

Valid Values: N/A

Default: "

Set the default value for Parameter

Properties for PROCEDURE:

Name: AUTHID

Type: STRING

Valid Values: None, Current\_User, Definer

Default: None

Generate the transformation with selected AUTHID option. Function will be executed with the permissions defined by the AUTHID clause rather than the function owner's permissions.

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Note:

1. N/A means any valid character in supported character set.
2. " " represents an empty string

`alterFuncProcParameterSCOClause`

Second class object clause to modify, delete or add a Parameter for Function/Procedure.

`propertyNameList`

Comma separated list of property names. Property names are unquoted.

`propertyValueList`

Comma separated list of property values.

`alterFuncProcParameterClause`

This clause alters Parameter of a Procedure.

`modifyFuncProcParameterClause`

Modify one or more Parameters to this Function/Procedure.

`deleteFuncProcParameterClause`

Delete one or more Parameters to this Function/Procedure.

`propertyValue`

Value of a property.

`moveToClause`

Move a Parameters of this Function/Procedure.

## Examples

OMBALTER PROCEDURE 'proc' RENAME TO 'proc\_1' SET PROPERTIES (DESCRIPTION, BUSINESS\_NAME) VALUES ('This becomes a proc\_1', 'proc') This will rename the Procedure "proc" to "proc\_1", and set its description to "This becomes a proc\_1", set its business name to "proc". If Packaged Function is overloaded, first find the Signature by using OMBLIST command, and then use OMBALTER command using appropriate signature. Example, if OMBLIST PROCEDURES gives following two signatures, PROC\_1 (NUMBER) PROC\_1 (VARCHAR2, NUMBER) The OMBALTER syntax to modify the first one will be as follows OMBALTER PROCEDURE 'PROC\_1 \ (NUMBER\)' SET PROPERTIES (DESCRIPTION, BUSINESS\_NAME) VALUES ('descri\_PROC\_1', 'PROC\_1')

## See Also

OMBALTER, OMBCREATE PROCEDURE, OMBDROP PROCEDURE

# OMBALTER PROCESS\_FLOW

## Purpose

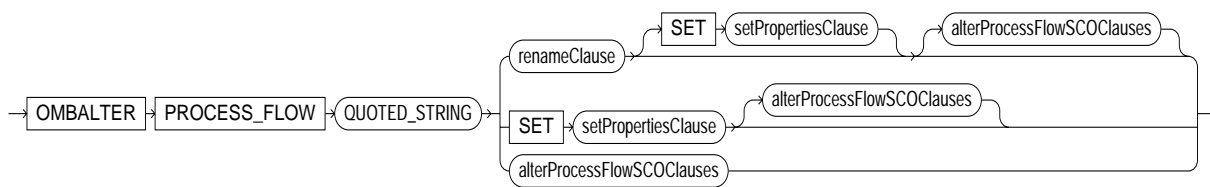
OMBALTER PROCESS\_FLOW - Alter the Process Flow by renaming it, and/or reset its properties.

## Prerequisites

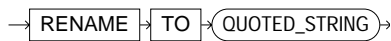
Should be in the context of a Process Flow Package.

## Syntax Diagrams

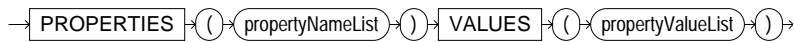
alterProcessFlowCommand



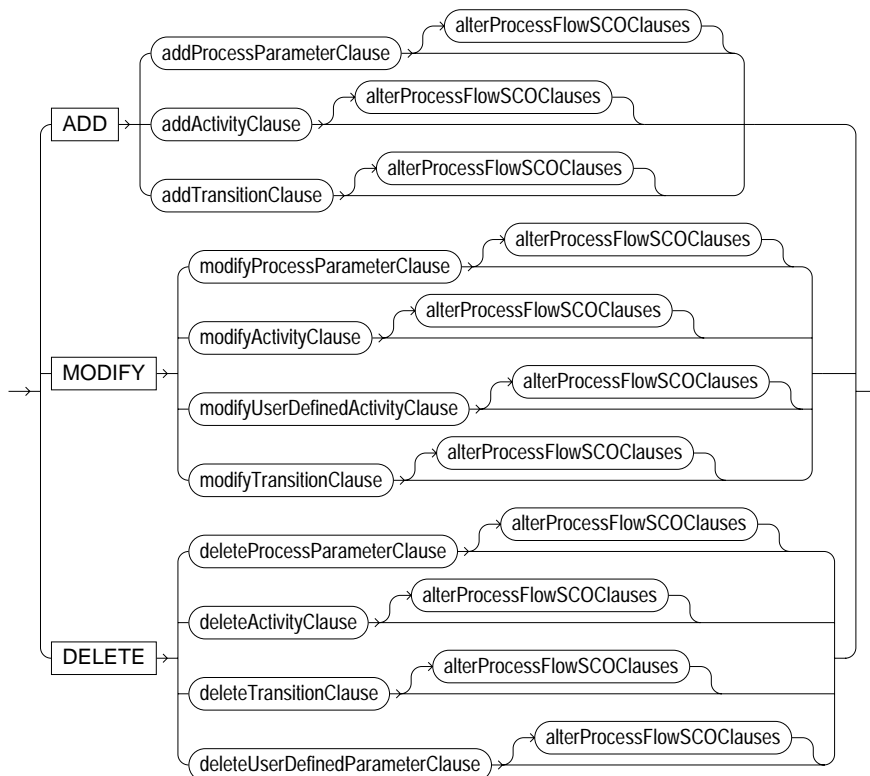
renameClause



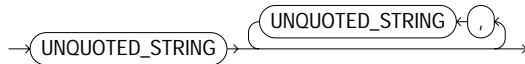
setPropertiesClause



alterProcessFlowSCOClauses



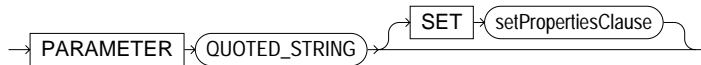
propertyNameList



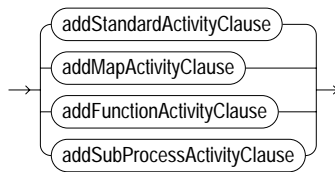
propertyValueList



addProcessParameterClause



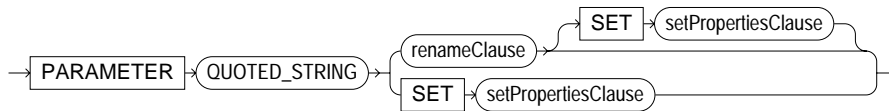
addActivityClause



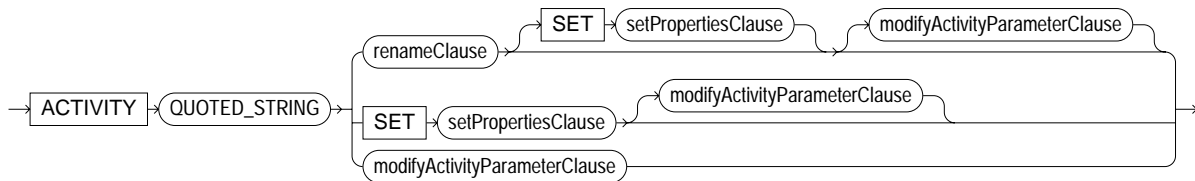
addTransitionClause



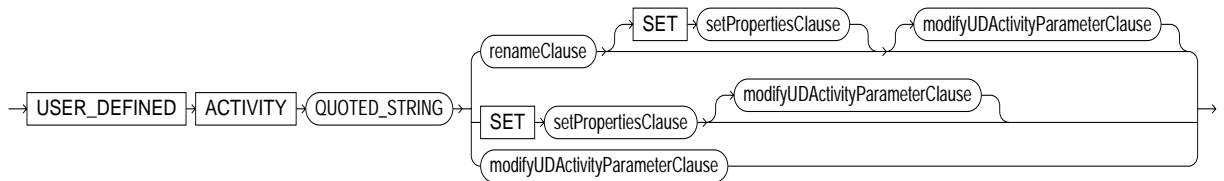
modifyProcessParameterClause



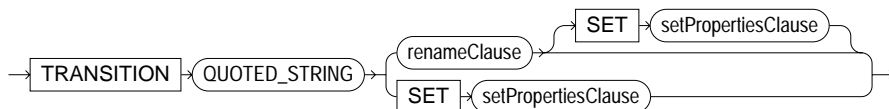
modifyActivityClause



modifyUserDefinedActivityClause



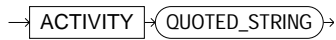
modifyTransitionClause



deleteProcessParameterClause



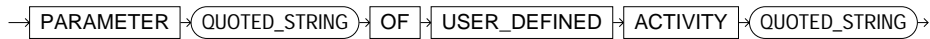
deleteActivityClause



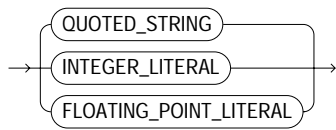
deleteTransitionClause



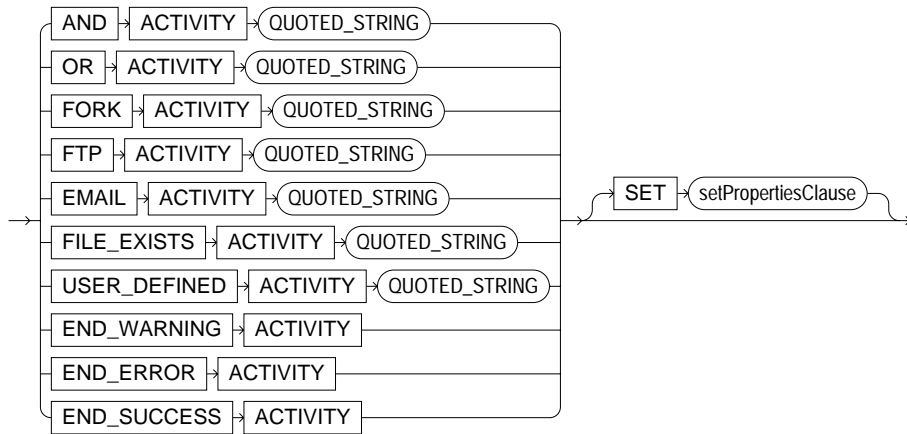
deleteUserDefinedParameterClause



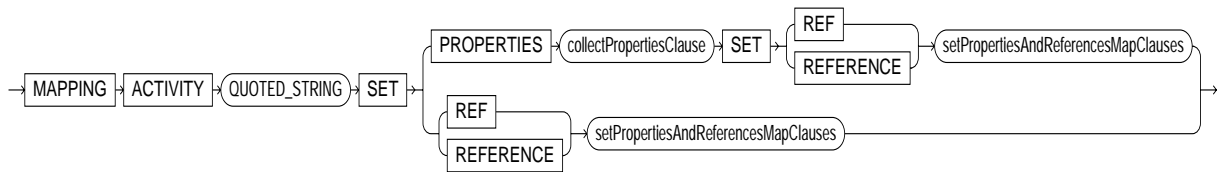
property Value



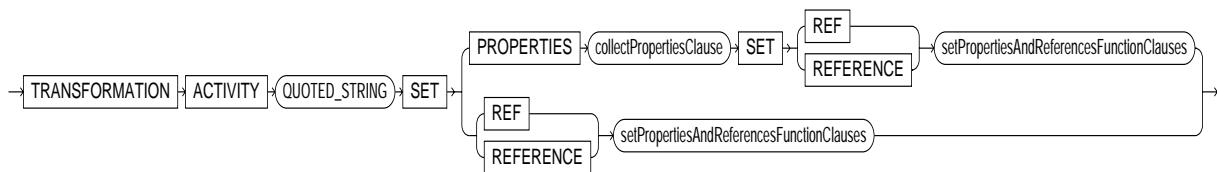
addStandardActivityClause



addMapActivityClause

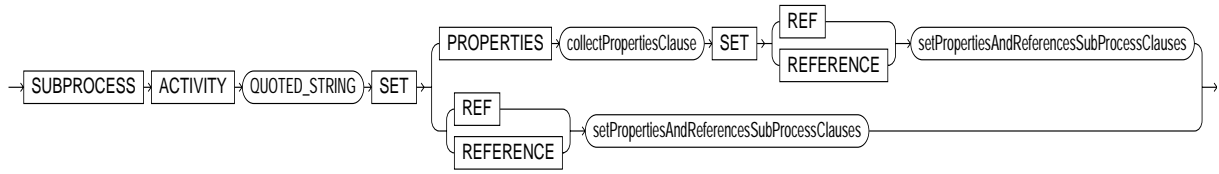


addFunctionActivityClause





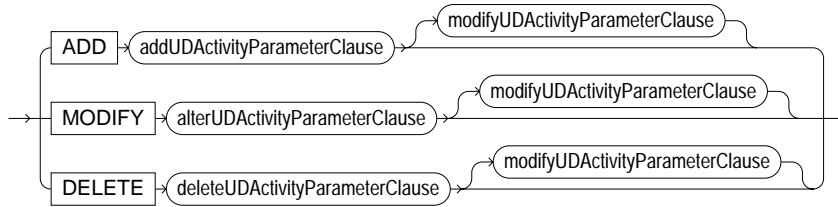
addSubProcessActivityClause



modifyActivityParameterClause



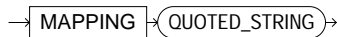
modifyUDActivityParameterClause



collectPropertiesClause



setPropertiesAndReferencesMapClauses



setPropertiesAndReferencesFunctionClauses



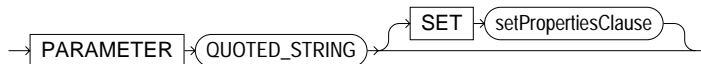
setPropertiesAndReferencesSubProcessClauses



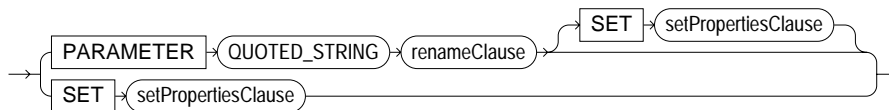
modifyActivityParameter



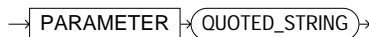
addUDActivityParameterClause



alterUDActivityParameterClause



deleteUDActivityParameterClause



## Syntax

```

alterProcessFlowCommand = OMBALTER PROCESS_FLOW "QUOTED_STRING" (
"renameClause" [ SET "setPropertyClause" ] [ "alterProcessFlowSCOClauses" ] | SET
"setPropertyClause" [ "alterProcessFlowSCOClauses" ] |
"alterProcessFlowSCOClauses");

renameClause = RENAME TO "QUOTED_STRING";

setPropertyClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("
"propertyValueList" ")";

alterProcessFlowSCOClauses = ADD ( "addProcessParameterClause" [
"alterProcessFlowSCOClauses" ] | "addActivityClause" [
"alterProcessFlowSCOClauses" ] | "addTransitionClause" [
"alterProcessFlowSCOClauses" ] ) | MODIFY ( "modifyProcessParameterClause" [
"alterProcessFlowSCOClauses" ] | "modifyActivityClause" [
"alterProcessFlowSCOClauses" ] | "modifyUserDefinedActivityClause" [
"alterProcessFlowSCOClauses" ] | "modifyTransitionClause" [
"alterProcessFlowSCOClauses" ] ) | DELETE ( "deleteProcessParameterClause" [
"alterProcessFlowSCOClauses" ] | "deleteActivityClause" [
"alterProcessFlowSCOClauses" ] | "deleteTransitionClause" [
"alterProcessFlowSCOClauses" ] | "deleteUserDefinedParameterClause" [
"alterProcessFlowSCOClauses" ] );

propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };

propertyValueList = "propertyValue" { "," "propertyValue" };

addProcessParameterClause = ( PARAMETER "QUOTED_STRING" ) [ SET
"setPropertyClause" ];

addActivityClause = ( "addStandardActivityClause" | "addMapActivityClause" |
"addFunctionActivityClause" | "addSubProcessActivityClause" );

addTransitionClause = ( TRANSITION "QUOTED_STRING" ( FROM ACTIVITY
"QUOTED_STRING" ) ( TO "QUOTED_STRING" ) ) [ SET "setPropertyClause" ];

modifyProcessParameterClause = PARAMETER "QUOTED_STRING" (
"renameClause" [ SET "setPropertyClause" ] | SET "setPropertyClause" );

modifyActivityClause = ACTIVITY "QUOTED_STRING" ( "renameClause" [ SET
"setPropertyClause" ] [ "modifyActivityParameterClause" ] | SET
"setPropertyClause" [ "modifyActivityParameterClause" ] |
"modifyActivityParameterClause" );

modifyUserDefinedActivityClause = USER_DEFINED ACTIVITY "QUOTED_
STRING" ( "renameClause" [ SET "setPropertyClause" ] [
"modifyUDActivityParameterClause" ] | SET "setPropertyClause" [
"modifyUDActivityParameterClause" ] | "modifyUDActivityParameterClause" );

modifyTransitionClause = TRANSITION "QUOTED_STRING" ( "renameClause" [ SET
"setPropertyClause" ] | SET "setPropertyClause" );

deleteProcessParameterClause = PARAMETER "QUOTED_STRING";

deleteActivityClause = ACTIVITY "QUOTED_STRING";

deleteTransitionClause = TRANSITION "QUOTED_STRING";

```

```

deleteUserDefinedParameterClause = PARAMETER "QUOTED_STRING" OF USER_
DEFINED ACTIVITY "QUOTED_STRING";

propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );

addStandardActivityClause = ( AND ACTIVITY "QUOTED_STRING" | OR
ACTIVITY "QUOTED_STRING" | FORK ACTIVITY "QUOTED_STRING" | FTP
ACTIVITY "QUOTED_STRING" | EMAIL ACTIVITY "QUOTED_STRING" | FILE_
EXISTS ACTIVITY "QUOTED_STRING" | USER_DEFINED ACTIVITY "QUOTED_
STRING" | END_WARNING ACTIVITY | END_ERROR ACTIVITY | END_
SUCCESS ACTIVITY ) [ SET "setPropertiesClause" ];

addMapActivityClause = ( MAPPING ACTIVITY "QUOTED_STRING" SET (
PROPERTIES "collectPropertiesClause" SET ( REF | REFERENCE )
"setPropertiesAndReferencesMapClauses" | ( REF | REFERENCE )
"setPropertiesAndReferencesMapClauses" ) );

addFunctionActivityClause = ( TRANSFORMATION ACTIVITY "QUOTED_STRING"
SET ( PROPERTIES "collectPropertiesClause" SET ( REF | REFERENCE )
"setPropertiesAndReferencesFunctionClauses" | ( REF | REFERENCE )
"setPropertiesAndReferencesFunctionClauses" ) );

addSubProcessActivityClause = ( SUBPROCESS ACTIVITY "QUOTED_STRING" SET
( PROPERTIES "collectPropertiesClause" SET ( REF | REFERENCE )
"setPropertiesAndReferencesSubProcessClauses" | ( REF | REFERENCE )
"setPropertiesAndReferencesSubProcessClauses" ) );

modifyActivityParameterClause = MODIFY "modifyActivityParameter" [
"modifyActivityParameterClause" ];

modifyUDActivityParameterClause = ADD ( "addUDActivityParameterClause" [
"modifyUDActivityParameterClause" ] ) | MODIFY (
"alterUDActivityParameterClause" [ "modifyUDActivityParameterClause" ] ) |
DELETE ( "deleteUDActivityParameterClause" [ "modifyUDActivityParameterClause"
] );

collectPropertiesClause = "(" "propertyNameList" ")" VALUES "(" "propertyValueList"
)";

setPropertiesAndReferencesMapClauses = ( MAPPING "QUOTED_STRING" );

setPropertiesAndReferencesFunctionClauses = ( TRANSFORMATION "QUOTED_
STRING" );

setPropertiesAndReferencesSubProcessClauses = ( PROCESS_FLOW "QUOTED_
STRING" );

modifyActivityParameter = PARAMETER "QUOTED_STRING" SET
"setPropertiesClause";

addUDActivityParameterClause = PARAMETER "QUOTED_STRING" [ SET
"setPropertiesClause" ];

alterUDActivityParameterClause = PARAMETER "QUOTED_STRING"
"renameClause" [ SET "setPropertiesClause" ] | SET "setPropertiesClause";

deleteUDActivityParameterClause = PARAMETER "QUOTED_STRING";

```

## Keywords and Parameters

alterProcessFlowCommand

Alter process flow.

**renameClause**

Rename process flow/ activity, depending on current context.

**setPropertyClause**

Used to set properties (core, user-defined) for process flow. Note: For MAPPING, TRANSFORMATION and SUBPROCESS activities the `setPropertyAndReferencesMapClauses`, `setPropertyAndReferencesFunctionClauses` and `setPropertyAndReferencesSubProcessClauses` respectively, are mandatory. For MAPPING or TRANSFORMATION activities and the REFERENCE property has to be set to a valid MAP or TRANSFORMATION within the current project. For SUBPROCESS activities the REFERENCE property has to be set to a SUBPROCESS within the same PROCESS\_FLOW\_PACKAGE. Valid properties are shown below:

Base properties for PROCESS\_FLOW:

Basic properties for Process Flow, Activity, Transition and Parameter:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the Process Flow

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the Process Flow Core properties for Transition :

Name: TRANSITION\_CONDITION

Type: STRING

Valid Values: ", SUCCESS, ERROR, WARNING

Default: " i.e. Unconditional

Sets the Transition Condition of a Transition

Description of the Process Flow Core properties for Activity Parameter :

Name: DATATYPE

Type: STRING

Valid Values: INTEGER, FLOAT, DATE, STRING, BOOLEAN

Default: STRING

Sets the datatype of a Activity Parameter

Name: DIRECTION

Type: STRING

Valid Values: IN

Default: IN

Sets the direction of a Activity Parameter

Name: VALUE

Type: STRING

Valid Values: Examples '123', '123.456', 'Jan-08-2003', 'I am String', 'true'

Default: "

For Mapping activities representing PLSQL maps, the allowed value for the parameters: OPERATING\_MODE:'SET\_BASED' 'ROW\_BASED' 'ROW\_BASED\_TARGET\_ONLY' 'SET\_BASED\_FAIL\_OVER\_TO\_ROW\_BASED' 'SET\_BASED\_FAIL\_OVER\_TO\_ROW\_BASED\_TARGET\_ONLY' AUDIT\_LEVEL:'NONE' 'STATISTICS' 'ERROR\_DETAILS' 'COMPLETE' Sets the value of a Activity Parameter

Name: BINDING

Type: STRING

Valid Values: Examples 'PARAM\_1', 'PARAM\_2'

Default: "

Represents the parameter on the process flow that this parameter is bound to. When setting users can specify the name of any PROCESS PARAMETER of same datatype. This feature allows for parameterizing the process flow. If the parameter is bound the VALUE property is ignored when generating the process flow. To unbind a parameter, use an empty quoted string ie. "", and the parameter will be unbound.

Properties for PROCESS\_FLOW:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: "

Physical name of the object that this activity represents.

Name: DEPLOYED\_LOCATION

Type: STRING

Valid Values: N/A

Default: USE\_DEFAULT\_LOCATION

Deployed Location of Transformation Activities only.

Name: EXECUTION\_LOCATION

Type: STRING

Valid Values: N/A

Default: NATIVE\_EXECUTION

The location from which this activity will be executed.

Name: REMOTE\_LOCATION

Type: STRING

Valid Values: N/A

Default: USE\_DEFAULT\_LOCATION

Remote Location for FTP activities only.

Name: USE\_RETURN\_AS\_STATUS

Type: BOOLEAN

Valid Values: true, false

Default: false

Use any return value to select the Transition path.

Name: WORKING\_LOCATION

Type: STRING

Valid Values: N/A

Default: USE\_DEFAULT\_LOCATION

Working Location for FTP, FILE\_EXIST and USER\_DEFINED activities only.

Properties for ACTIVITY:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: "

Physical name of the object that this activity represents.

Name: DEPLOYED\_LOCATION

Type: STRING

Valid Values: N/A

Default: USE\_DEFAULT\_LOCATION

Deployed Location of Transformation Activities only.

Name: EXECUTION\_LOCATION

Type: STRING

Valid Values: N/A

Default: NATIVE\_EXECUTION

The location from which this activity will be executed.

Name: REMOTE\_LOCATION

Type: STRING

Valid Values: N/A

Default: USE\_DEFAULT\_LOCATION

Remote Location for FTP activities only.

Name: USE\_RETURN\_AS\_STATUS

Type: BOOLEAN

Valid Values: true, false

Default: false

Use any return value to select the Transition path.

Name: WORKING\_LOCATION

Type: STRING

Valid Values: N/A

Default: USE\_DEFAULT\_LOCATION

Working Location for FTP, FILE\_EXIST and USER\_DEFINED activities only.

Note:

1. N/A means any valid character in supported character set.
2. " " represents an empty string

alterProcessFlowSCOClauses

This clause is wrapper clause to add, modify or delete Process Flow's second class objects.

`propertyNameList`

A comma delimited set of property names to set.

`propertyValueList`

A comma delimited set of property values to set.

`addProcessParameterClause`

This clause adds the Parameters for a Process Flow.

`addActivityClause`

This clause adds the Activities for a Process Flow.

`addTransitionClause`

This clause adds the Transitions for a Process Flow.

`modifyProcessParameterClause`

This clause modifies the Parameters of a Process Flow.

`modifyActivityClause`

This clause modifies the Activities of a Process Flow.

`modifyUserDefinedActivityClause`

Alter the properties of the specified user defined activity.

`modifyTransitionClause`

This clause modifies the Transitions of a Process Flow.

`deleteProcessParameterClause`

This clause deletes the Parameters of a Process Flow.

`deleteActivityClause`

This clause deletes the Activities of a Process Flow.

`deleteTransitionClause`

This clause deletes the Transitions of a Process Flow.

`deleteUserDefinedParameterClause`



This clause deletes the Parameters of a User Defined Activity of a Process Flow.

propertyValue

Integer value, float value or quoted string literal.

addStandardActivityClause

This clause adds standard activity types AND, FORK, FTP, OR, USER\_DEFINED to a Process Flow.

addMapActivityClause

This clause adds the MAP activity to a Process Flow.

addFunctionActivityClause

This clause adds the Function or Procedure activity to a Process Flow.

addSubProcessActivityClause

This clause adds a Process as an activity to a Process Flow.

modifyActivityParameterClause

Alter the properties of activity parameters.

modifyUDActivityParameterClause

Alter user defined activity by adding new parameters or modifying existing parameter.

collectPropertiesClause

This clause collects core properties of Map, Function/Procedure and Subprocess activity.

setPropertiesAndReferencesMapClauses

This clause sets reference to the existing Map.

setPropertiesAndReferencesFunctionClauses

This clause sets a reference to existing Function or Procedure.

setPropertiesAndReferencesSubProcessClauses

This clause sets a reference to existing Process Flow.

**modifyActivityParameter**

Alter the properties of the parameter.

**addUDActivityParameterClause**

Add more activity parameters to the user defined activity.

**alterUDActivityParameterClause**

In the current user defined activity, alter the properties of the activity parameter eg. rename, set DATATYPE or set VALUE.

**deleteUDActivityParameterClause**

For the current user defined activity, delete the activity parameter.

**Examples**

```
OMBALTER PROCESS_FLOW 'process_flow' RENAME TO 'p_flow' SET
PROPERTIES (DESCRIPTION, BUSINESS_NAME) VALUES ('This becomes a process
flow.', 'process flow') This will rename the Process Flow "process_flow" to "p_flow",
and set its description to "This becomes a process flow", set its business name to
"process flow".
```

**See Also**

OMBALTER, OMBCREATE PROCESS\_FLOW, OMBDROP PROCESS\_FLOW

# OMBALTER PROCESS\_FLOW\_MODULE

## Purpose

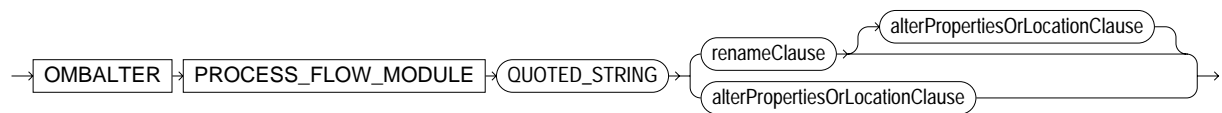
OMBALTER PROCESS\_FLOW\_MODULE - Alter the Process Flow Module by renaming it, and/or reset its properties.

## Prerequisites

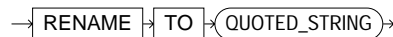
Should be in the context of a project.

## Syntax Diagrams

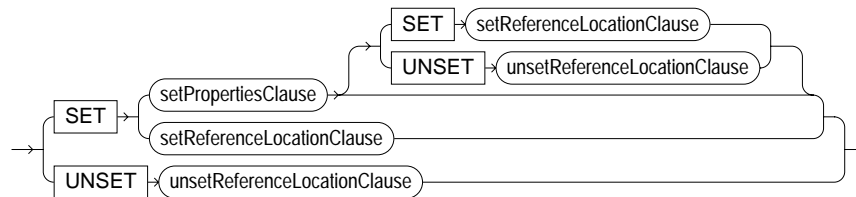
alterProcessFlowModuleCommand



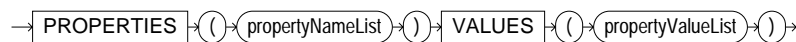
renameClause



alterPropertiesOrLocationClause



setPropertiesClause



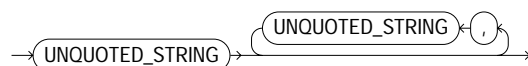
setReferenceLocationClause



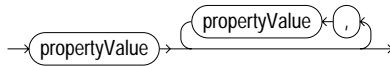
unsetReferenceLocationClause



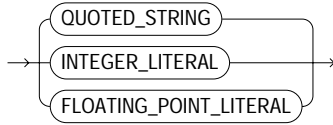
propertyNameList



propertyValueList



propertyValue



## Syntax

```
alterProcessFlowModuleCommand = OMBALTER ( PROCESS_FLOW_MODULE
"QUOTED_STRING" ( "renameClause" [ "alterPropertiesOrLocationClause" ] |
"alterPropertiesOrLocationClause" ) );
```

```
renameClause = RENAME TO "QUOTED_STRING";
```

```
alterPropertiesOrLocationClause = SET ( "setPropertiesClause" [ SET
"setReferenceLocationClause" | UNSET "unsetReferenceLocationClause" ] |
"setReferenceLocationClause" ) | UNSET "unsetReferenceLocationClause";
```

```
setPropertiesClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("
"propertyValueList" ")";
```

```
setReferenceLocationClause = ( REFERENCE | REF ) LOCATION "QUOTED_
STRING";
```

```
unsetReferenceLocationClause = ( REFERENCE | REF ) LOCATION "QUOTED_
STRING";
```

```
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
```

```
propertyValueList = "propertyValue" { "," "propertyValue" };
```

```
propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );
```

## Keywords and Parameters

**alterProcessFlowModuleCommand**

This command modifies an existing process flow module.

**renameClause**

Rename an existing process flow module.

**alterPropertiesOrLocationClause**

Modify the details of this process flow module.

**setPropertiesClause**

Set values of properties of a process flow module.

Base properties for PROCESS\_FLOW\_MODULE:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: NAME

Business name of a Process Flow Module

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of a Process Flow Module

setReferenceLocationClause

Set a location to a supported workflow engine.

unsetReferenceLocationClause

Unset the location of the process flow module.

propertyNameList

Comma-separated list of property names. Property names are not in quotation marks.

propertyValueList

Comma separated list of property values.

propertyValue

Value of a property.

## Examples

```
OMBALTER PROCESS_FLOW_MODULE 'process_module' RENAME TO 'p_module'
SET PROPERTIES (DESCRIPTION, BUSINESS_NAME) VALUES ('This becomes a
process flow module.', 'process module') This will rename the Process Flow Module
"process_module" to "p_module", and set its description to "This becomes a process
flow module", set its business name to "process module".
```

## See Also

OMBALTER, OMBCREATE PROCESS\_FLOW\_MODULE, OMBDROP PROCESS\_FLOW\_MODULE

# OMBALTER PROCESS\_FLOW\_PACKAGE

## Purpose

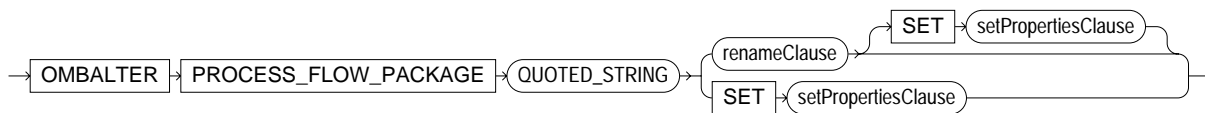
OMBALTER PROCESS\_FLOW\_PACKAGE - Alter the Process Flow Package by renaming it, and/or reset its properties.

## Prerequisites

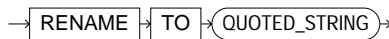
Should be in the context of a Process Flow Module.

## Syntax Diagrams

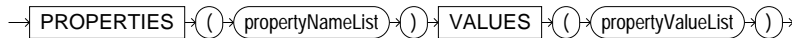
alterProcessFlowPackageCommand



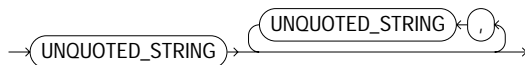
renameClause



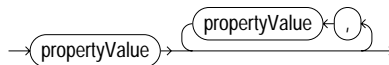
setPropertiesClause



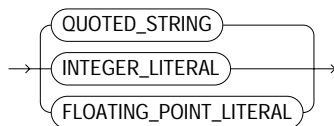
propertyNameList



propertyValueList



propertyValue



## Syntax

```
alterProcessFlowPackageCommand = OMBALTER ( PROCESS_FLOW_PACKAGE
"QUOTED_STRING" ( "renameClause" [ SET "setPropertiesClause" ] | SET
"setPropertiesClause" ) );
```

```
renameClause = RENAME TO "QUOTED_STRING";
```

```

setPropertyClause = PROPERTIES "(" propertyNameList ")" VALUES "("
"propertyValueList" ");
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
propertyValueList = "propertyValue" { "," "propertyValue" };
propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );

```

## Keywords and Parameters

**alterProcessFlowPackageCommand**

Alter process flow package.

**renameClause**

Rename the process flow package.

**setPropertyClause**

Set the properties of the process flow package.

Basic properties for PROCESS\_FLOW\_PACKAGE:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the Process Flow Package

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the Process Flow Package

Note:

1. N/A means any valid character in supported character set.
2. " represents an empty string

**propertyNameList**

A comma delimited set of property names to set.

propertyValueList

A comma delimited set of property values to set.

propertyValue

Integer value, float value or quoted string literal.

## Examples

```
OMBALTER PROCESS_FLOW_PACKAGE 'process_package' RENAME TO 'p_
package' SET PROPERTIES (DESCRIPTION, BUSINESS_NAME) VALUES ('This
becomes a process flow package.', 'process package') This will rename the Process
Flow Package "process_package" to "p_package", and set its description to "This
becomes a process flow package", set its business name to "process package".
```

## See Also

OMBALTER, OMBCREATE PROCESS\_FLOW\_PACKAGE, OMBDROP PROCESS\_
FLOW\_PACKAGE



# OMBALTER PROJECT

## Purpose

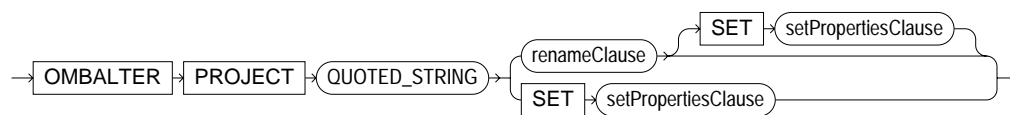
OMBALTER PROJECT - Alter the project by renaming it, and/or reset its properties.

## Prerequisites

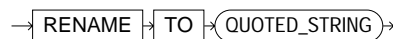
Should be in the top level context.

## Syntax Diagrams

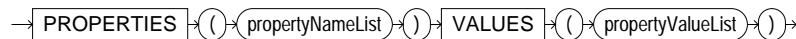
alterProjectCommand



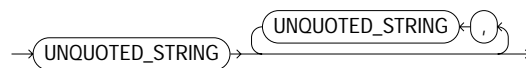
renameClause



setPropertiesClause



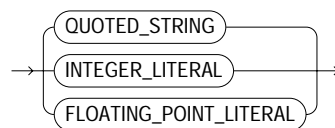
propertyNameList



propertyValueList



propertyValue



## Syntax

```
alterProjectCommand = OMBALTER ( PROJECT "QUOTED_STRING" (
"renameClause" [ SET "setPropertiesClause" ] | SET "setPropertiesClause" ) );
renameClause = RENAME TO "QUOTED_STRING";
```

```
setPropertyClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("  
"propertyValueList" ");  
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };  
propertyValueList = "propertyValue" { "," "propertyValue" };  
propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_  
POINT_LITERAL" );
```

## Keywords and Parameters

**alterProjectCommand**

Modify an existing project.

**QUOTED\_STRING**

Name of the existing project in quotes.

**renameClause**

Rename a project.

**setPropertyClause**

Associate a set of properties with a project.

Basic properties for PROJECT:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: NAME

Business name of a Project

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of a Project

Name: UOID

Type: STRING(40)

Valid Values: N/A

Default: N/A

UUID of a Project

propertyNameList

Comma separated list of property names. Property names are unquoted.

propertyValueList

Comma separated list of property values.

propertyValue

Value of a property.

### Examples

```
OMBALTER PROJECT 'New Project' RENAME TO 'Old Project' SET PROPERTIES  
(DESCRIPTION, BUSINESS_NAME) VALUES ('This becomes an old project.', 'old  
payroll project') This will rename project "New Project" to "Old Project", and set its  
description to "This becomes an old project", set its business name to "old payroll  
project".
```

### See Also

OMBALTER, OMBCREATE PROJECT, OMBDROP PROJECT

# OMBALTER\_RUNTIME\_REPOSITORY\_CONNECTIONS

## Purpose

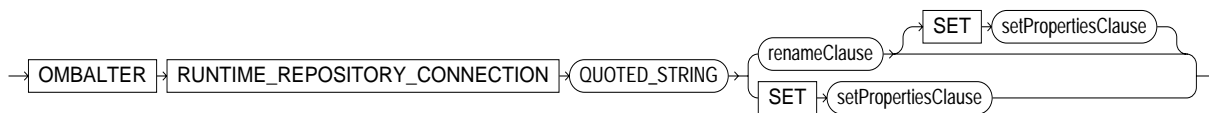
OMBALTER\_RUNTIME\_REPOSITORY\_CONNECTION - Alter the runtime repository connection by renaming it, and/or resetting its properties.

## Prerequisites

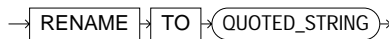
Should be in the context of a project.

## Syntax Diagrams

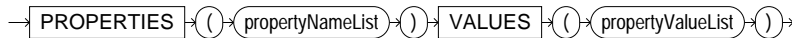
alterRuntimeRepositoryCommand



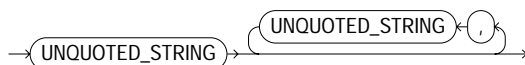
renameClause



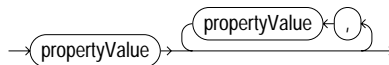
setPropertiesClause



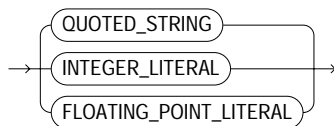
propertyNameList



propertyValueList



propertyValue



## Syntax

```
alterRuntimeRepositoryCommand = OMBALTER ( RUNTIME_REPOSITORY_
CONNECTION "QUOTED_STRING" ( "renameClause" [ SET "setPropertiesClause" ] |
SET "setPropertiesClause" ) );
```

```
renameClause = RENAME TO "QUOTED_STRING";
```

```
setPropertyClause = PROPERTIES "(" propertyNameList ")" VALUES "("  
"propertyValueList" ");  
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };  
propertyValueList = "propertyValue" { "," "propertyValue" };  
propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_  
POINT_LITERAL" );
```

## Keywords and Parameters

**alterRuntimeRepositoryCommand**

Alter a runtime repository connection.

**renameClause**

Rename the runtime repository connection to the value of the following quoted string.

**setPropertyClause**

Set the specified properties of the runtime repository connection.

**propertyNameList**

The names of the properties whose values you want to set.

Properties for RUNTIME\_REPOSITORY\_CONNECTION:

Name: HOST

Type: STRING

Valid Values: N/A

Default: N/A

The host machine the runtime repository is installed on.

Name: PORT

Type: NUMBER

Valid Values: 0 - 65535

Default: N/A

The port number of the database in which the runtime repository is installed.

Name: SERVICE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The service name of the database in which the runtime repository is installed.

Name: CONNECT\_AS\_USER

Type: STRING

Valid Values: N/A

Default: N/A

The name of the database user you wish to connect to the runtime repository as.

Name: RUNTIME\_REPOSITORY\_OWNER

Type: STRING

Valid Values: N/A

Default: N/A

The name of the schema in which the runtime repository is installed.

All of the preceding properties are mandatory for OMBCREATE RUNTIME\_REPOSITORY\_CONNECTION.

Basic properties for RUNTIME\_REPOSITORY\_CONNECTION:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the runtime repository connection.

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the runtime repository connection.

Note:

1. N/A means any valid character in supported character set.
2. " represents an empty string

propertyValueList

The values for the named properties.

propertyValue

A property value.

### Examples

```
OMBALTER RUNTIME_REPOSITORY_CONNECTION 'MY_CONNECTION'  
RENAME TO 'NEW_CONNECTION' SET PROPERTIES (DESCRIPTION, BUSINESS_  
NAME) VALUES ('This becomes a new repository.', 'new repository') This will rename  
the runtime repository connection "MY_CONNECTION" to "NEW_CONNECTION",  
and set its description to "This becomes a new repository", set its business name to  
"new repository".
```

### See Also

```
OMBALTER, OMBCREATE RUNTIME_REPOSITORY_CONNECTION, OMBDROP  
RUNTIME_REPOSITORY_CONNECTION
```

# OMBALTER SEQUENCE

## Purpose

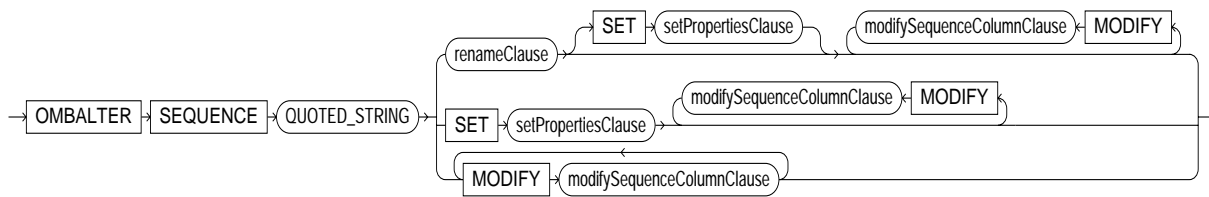
OMBALTER SEQUENCE - To alter properties and definition of a sequence.

## Prerequisites

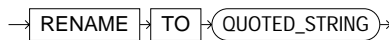
In the context of an Oracle Module.

## Syntax Diagrams

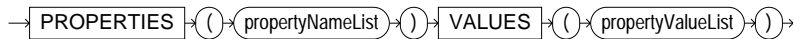
alterSequenceCommand



renameClause



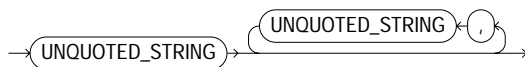
setPropertiesClause



modifySequenceColumnClause



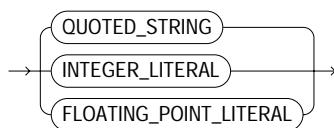
propertyNameList



propertyValueList



propertyValue





## Syntax

```

alterSequenceCommand = OMBALTER ( SEQUENCE "QUOTED_STRING" (
"renameClause" [ SET "setPropertiesClause" ] { MODIFY
"modifySequenceColumnClause" } | SET "setPropertiesClause" { MODIFY
"modifySequenceColumnClause" } | ( MODIFY "modifySequenceColumnClause" )+ )
);
renameClause = RENAME TO "QUOTED_STRING";
setPropertiesClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("
"propertyValueList" ")";
modifySequenceColumnClause = COLUMN "QUOTED_STRING" SET
"setPropertiesClause";
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
propertyValueList = "propertyValue" { "," "propertyValue" };
propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );

```

## Keywords and Parameters

**alterSequenceCommand**

This clause is for sequence alter command.

**renameClause**

renames a table with a different name.

**setPropertiesClause**

set sequences properties.

Basic properties for SEQUENCE:

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the sequence.

Name: CURRVAL

Type: NUMBER

Valid Values: N/A

Default: 1

current increment value.

Name: NEXTVAL

Type: NUMBER

Valid Values: N/A

Default: 1

next increment value. next increment value.

Properties for SEQUENCE:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INCREMENT\_BY

Type: NUMBER

Valid Values: -2147483648 - 2147483647

Default: 1

Sequence Incremented By

Name: START\_WITH

Type: NUMBER

Valid Values: -2147483648 - 2147483647

Default: 1

Sequence Starts With

Note:

1. N/A means any valid character in supported character set.
2. " " represents an empty string

modifySequenceColumnClause

This clause modifies the sequence's column.

propertyNameList

The list of properties.

propertyValueList

The list of property values.

propertyValue

This clause adds the property values.

### Examples

```
OMBALTER SEQUENCE 'NEW_SEQUENCE' SET PROPERTIES (DESCRIPTION)  
VALUES ('this
```

```
is an altered desc of new sequence') This will alter a sequence named "NEW_  
SEQUENCE", its description is "this is an altered desc of new sequence."
```

### See Also

OMBALTER, OMBCREATE SEQUENCE, OMBDROP SEQUENCE

# OMBALTER SNAPSHOT

## Purpose

OMBALTER SNAPSHOT - A snapshot can be altered to remove, add or update components.

## Prerequisites

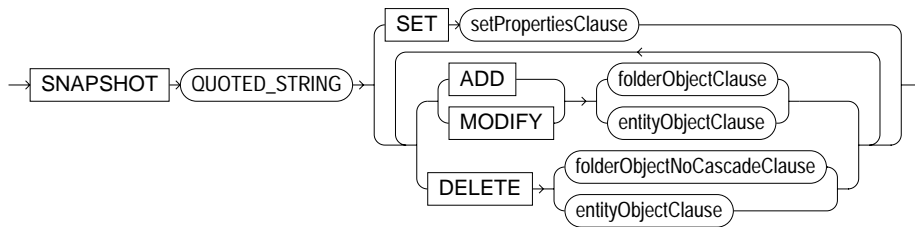
The snapshot to be altered should already exist. This command can be executed for any component regardless of current context.

## Syntax Diagrams

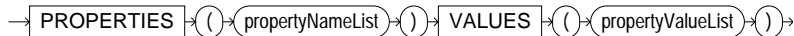
parseAlterCommand



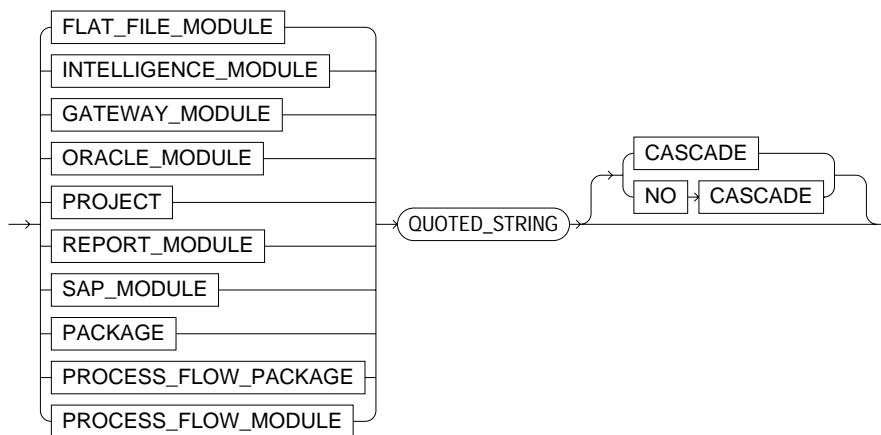
alterSnapshotCommand



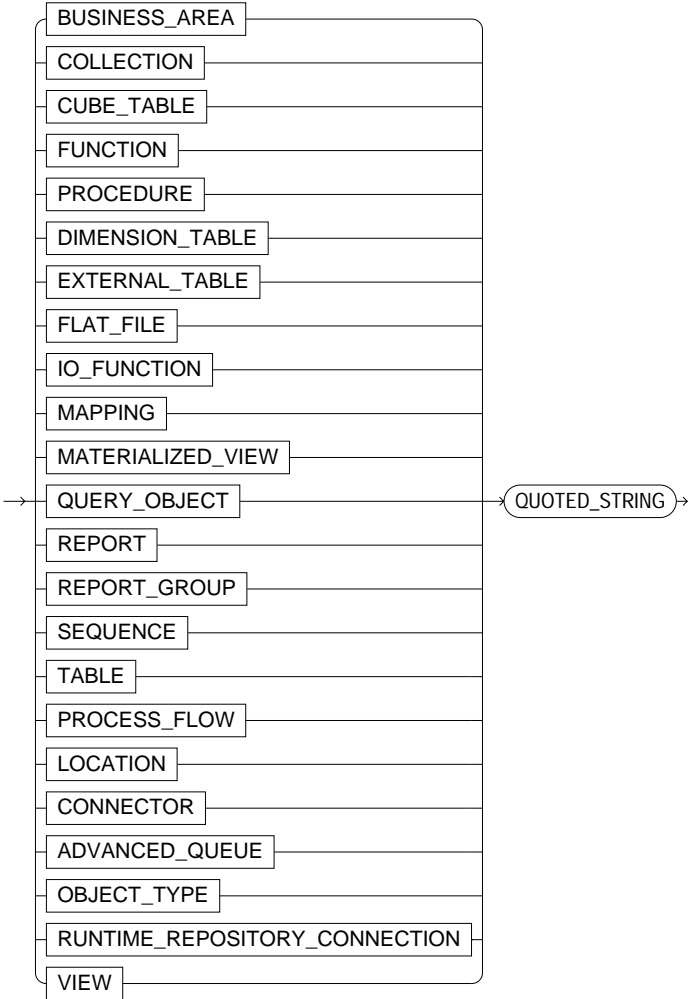
setPropertiesClause



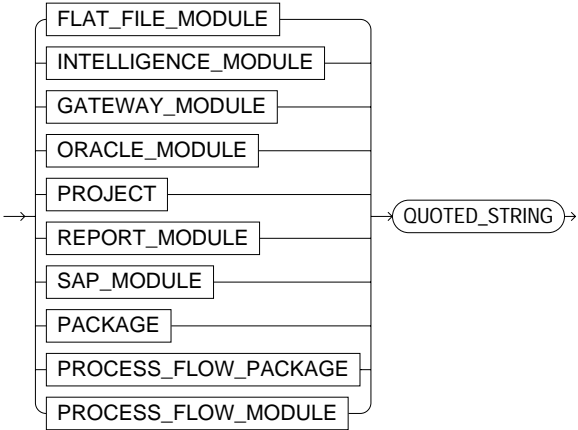
folderObjectClause



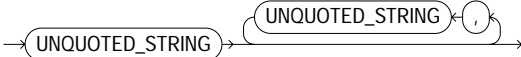
entityObjectClause



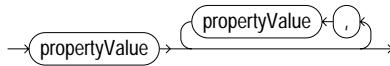
folderObjectNoCascadeClause



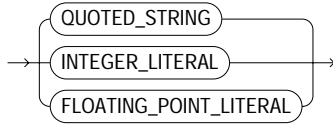
propertyNameList



propertyValueList



propertyValue



## Syntax

```

parseAlterCommand = OMBALTER "alterSnapshotCommand";

alterSnapshotCommand = ( SNAPSHOT "QUOTED_STRING" ( SET
"setPropertyClause" | ( ( ( ADD | MODIFY ) ( "folderObjectClause" |
"entityObjectClause" ) ) | ( DELETE ( "folderObjectNoCascadeClause" |
"entityObjectClause" ) ) )+ ) );

setPropertyClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("
"propertyValueList" ")";

folderObjectClause = ( FLAT_FILE_MODULE | INTELLIGENCE_MODULE |
GATEWAY_MODULE | ORACLE_MODULE | PROJECT | REPORT_MODULE |
SAP_MODULE | PACKAGE | PROCESS_FLOW_PACKAGE | PROCESS_FLOW_
MODULE ) "QUOTED_STRING" [ CASCADE | NO CASCADE ];

entityObjectClause = ( ( BUSINESS_AREA | COLLECTION | CUBE_TABLE |
FUNCTION | PROCEDURE | DIMENSION_TABLE | EXTERNAL_TABLE | FLAT_
FILE | IO_FUNCTION | MAPPING | MATERIALIZED_VIEW | QUERY_OBJECT |
REPORT | REPORT_GROUP | SEQUENCE | TABLE | PROCESS_FLOW |
LOCATION | CONNECTOR | ADVANCED_QUEUE | OBJECT_TYPE | RUNTIME_
REPOSITORY_CONNECTION | VIEW ) "QUOTED_STRING" );

folderObjectNoCascadeClause = ( FLAT_FILE_MODULE | INTELLIGENCE_
MODULE | GATEWAY_MODULE | ORACLE_MODULE | PROJECT | REPORT_
MODULE | SAP_MODULE | PACKAGE | PROCESS_FLOW_PACKAGE |
PROCESS_FLOW_MODULE ) "QUOTED_STRING";

propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };

propertyValueList = "propertyValue" { "," "propertyValue" };

propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );

```

## Keywords and Parameters

parseAlterCommand

Root production for OMBALTER SNAPSHOT.

alterSnapshotCommand

To alter already existing snapshot.

**QUOTED\_STRING**

Name of snapshot to be altered.

**ADD**

Add components to snapshot.

**MODIFY**

Replace an already existing component with the latest definition of the component from repository.

**DELETE**

Remove component from snapshot.

**setPropertiesClause**

Properties of snapshot can be altered through this optional clause. Properties of Snapshot is DESCRIPTION and TYPE. Altering TYPE property is an irreversible operation. Altering snapshot from FULL to SIGNATURE will only keep information required for comparing. SIGNATURE snapshots cannot be exported.

Basic properties for SNAPSHOT:

Name: TYPE

Type: STRING(200)

Valid Values: FULL,SIGNATURE

Default: FULL

This is the type of snapshot

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the snapshot

**folderObjectClause**

Folder component clause, syntax is same as in CREATE SNAPSHOT command. It refers to the folder object and all components contained in it. For the ADD and MODIFY operation, this must be the relative path to the folder within the current repository. For the DELETE operation, this must be the fully-qualified path of the folder within a snapshot. Use OMBRETRIEVE SNAPSHOT to get the fully-qualified path of the folder within the snapshot to be altered.

**entityObjectClause**

Component clause, syntax is same as in CREATE SNAPSHOT. For the ADD and MODIFY operation, this must be the relative path to the component within the current repository. For the DELETE operation, this must be the fully-qualified path of the component within a snapshot. Use OMBRETRIEVE SNAPSHOT to get the fully-qualified path of the component within the snapshot to be altered.

**folderObjectNoCascadeClause**

Folder component with no cascade clause, It only refers to the folder object itself, its properties and none of its contained components. For the ADD and MODIFY operation, this must be the relative path to the folder within the current repository. For the DELETE operation, this must be the fully-qualified path of the folder within a snapshot. Use OMBRETRIEVE SNAPSHOT to get the fully-qualified path of the folder within the snapshot to be altered.

**propertyNameList**

Property names for SNAPSHOT that can be altered.

**propertyValueList**

List of property values for SNAPSHOT.

**propertyValue**

Allowable value types for a snapshot property.

**Examples**

```
OMBALTER SNAPSHOT 'S1' ADD TABLE '/Project1/WH2/T3'
```

This command adds table T3 into snapshot S1.

```
OMBALTER SNAPSHOT 'S1' DELETE TABLE '/Project1/WH2/T1'
```

This command removes T1 table from snapshot S1, if system can find that table in snapshot. OMBALTER SNAPSHOT 'S1' MODIFY TABLE '/Project1/WH2/T1' This command updates definition of component T1 in snapshot from repository.

OMBALTER SNAPSHOT 'S1' SET PROPERTIES (DESCRIPTION) VALUES ('this is new description'); This command sets new description for snapshot. OMBALTER SNAPSHOT 'S1' SET PROPERTIES (TYPE) VALUES('SIGNATURE') This command transforms a FULL snapshot into a SIGNATURE snapshot, which is only useful for the compare service.

**See Also**

OMBCREATE SNAPSHOT, OMBDROP SNAPSHOT, OMBRESTORE SNAPSHOT, OMBCOMPARE SNAPSHOT, OMBLIST SNAPSHOT, OMBRETRIEVE SNAPSHOT



# OMBALTER TABLE

## Purpose

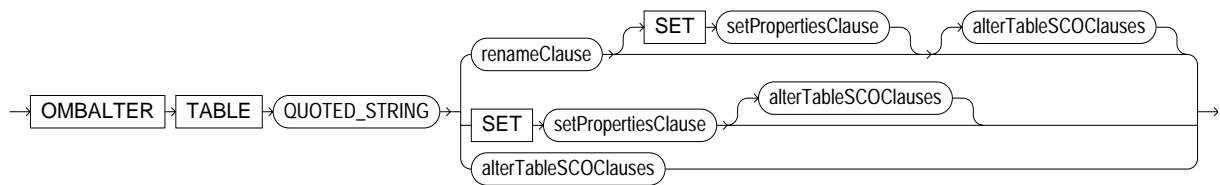
OMBALTER TABLE - To alter properties and definition of a table.

## Prerequisites

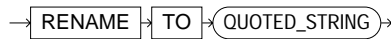
In the context of an Oracle Module.

## Syntax Diagrams

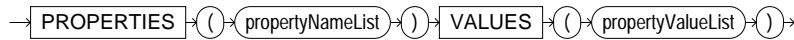
alterTableCommand



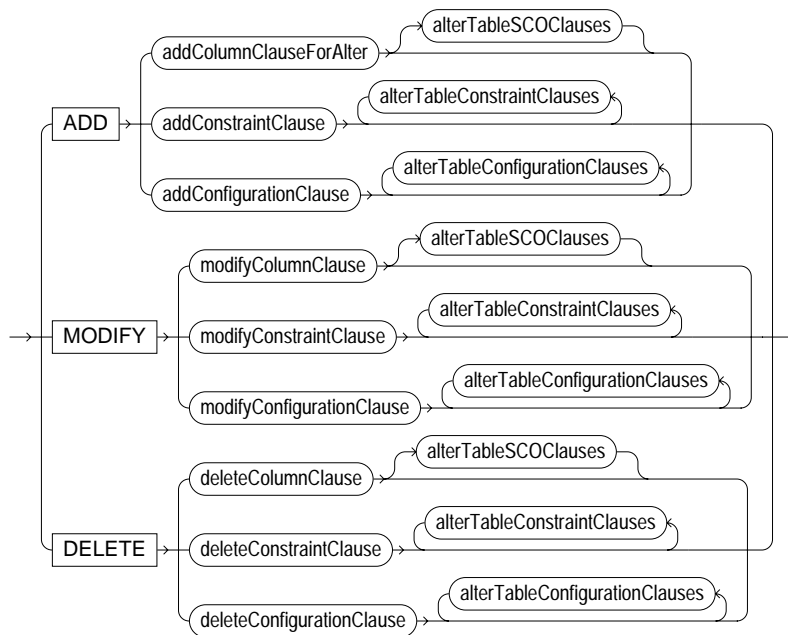
renameClause



setPropertyClause



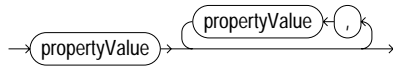
alterTableSCOClauses



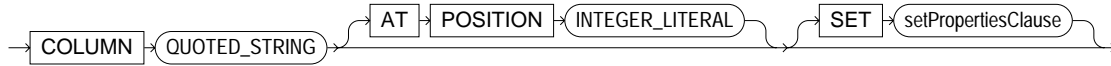
propertyNameList



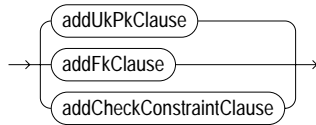
propertyValueList



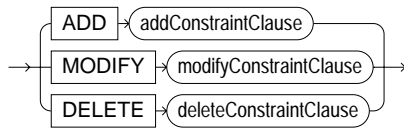
addColumnClauseForAlter



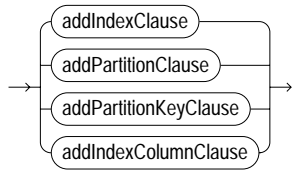
addConstraintClause



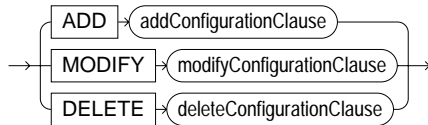
alterTableConstraintClauses



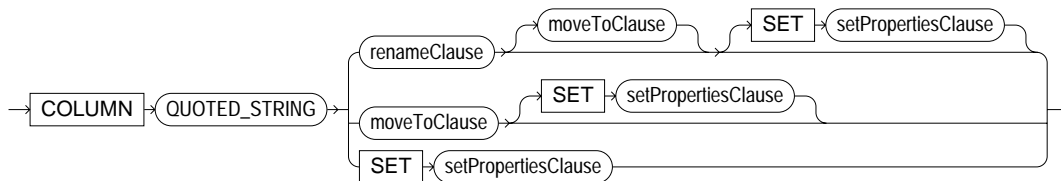
addConfigurationClause



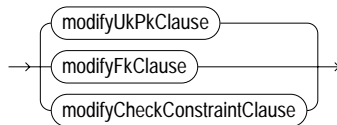
alterTableConfigurationClauses



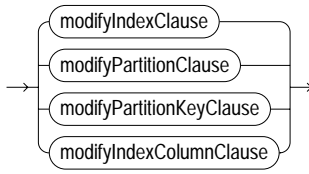
modifyColumnClause



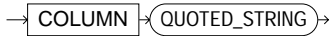
modifyConstraintClause



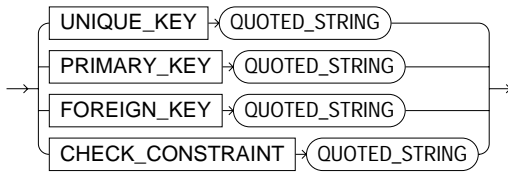
modifyConfigurationClause



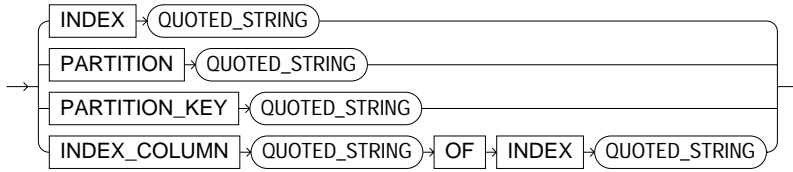
deleteColumnClause



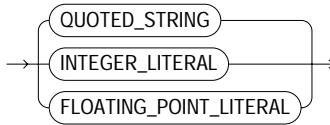
deleteConstraintClause



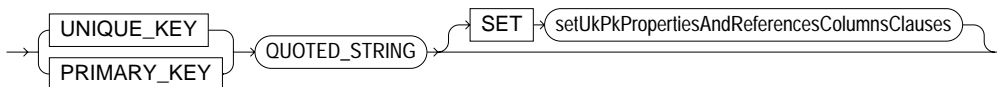
deleteConfigurationClause



propertyValue



addUkPkClause



addFkClause



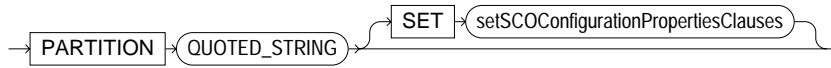
addCheckConstraintClause



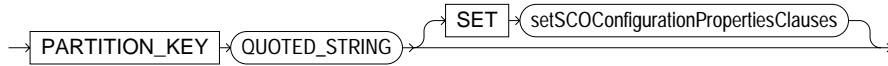
addIndexClause



addPartitionClause



addPartitionKeyClause



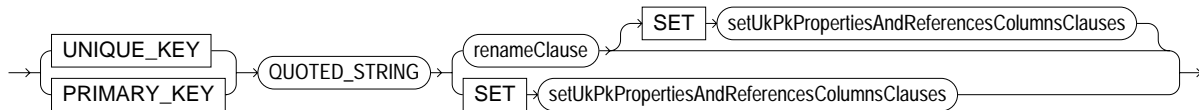
addIndexColumnClause



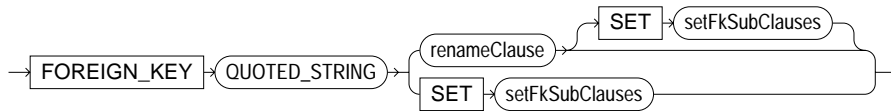
moveToClause



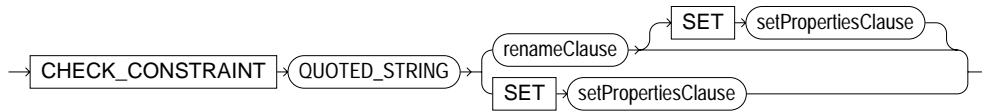
modifyUkPkClause



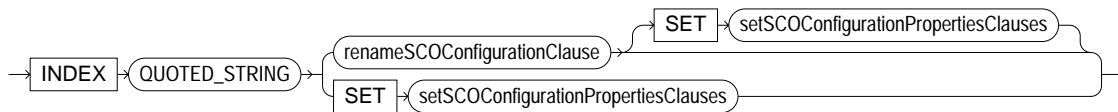
modifyFkClause



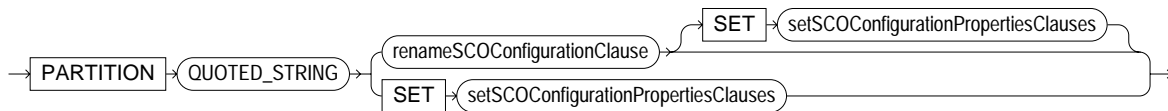
modifyCheckConstraintClause



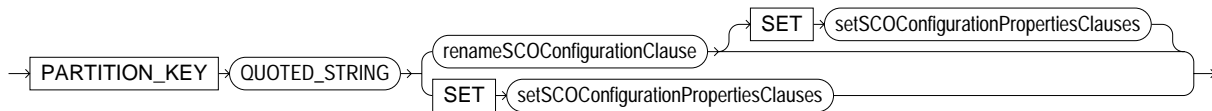
modifyIndexClause



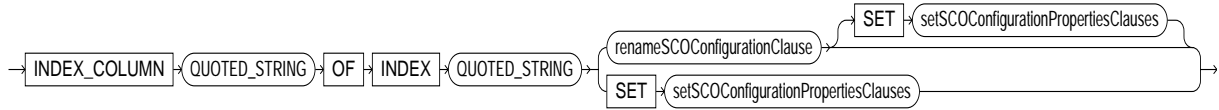
modifyPartitionClause



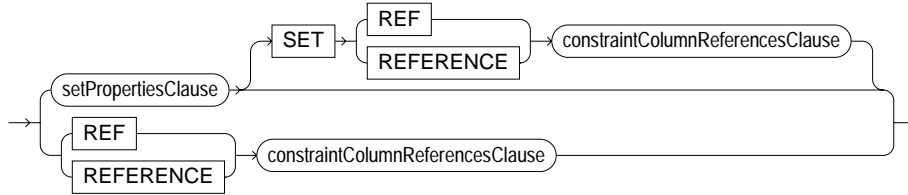
modifyPartitionKeyClause



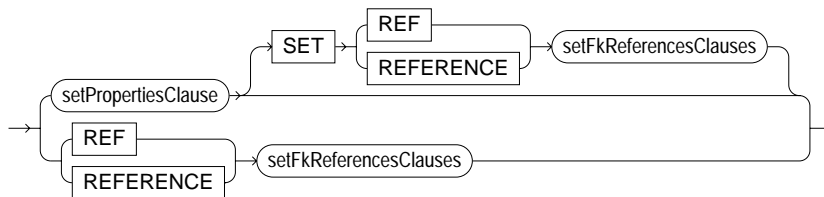
modifyIndexColumnClause



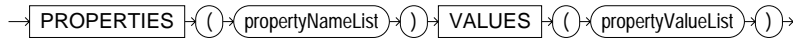
setUkPkPropertiesAndReferencesColumnsClauses



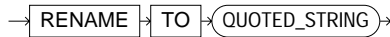
setFkSubClauses



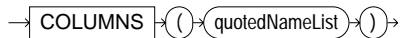
setSCOConfigurationPropertiesClauses



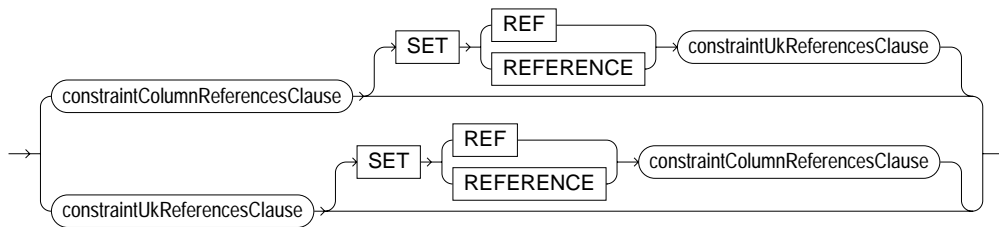
renameSCOConfigurationClause



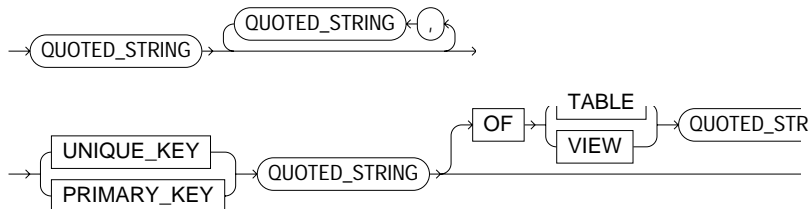
constraintColumnReferencesClause



setFkReferencesClauses



quotedNameList



## Syntax

```

alterTableCommand = OMBALTER ( TABLE "QUOTED_STRING" ( "renameClause" [
SET "setPropertiesClause" ] [ "alterTableSCOClases" ] | SET "setPropertiesClause" [
"alterTableSCOClases" ] | "alterTableSCOClases" ));

renameClause = RENAME TO "QUOTED_STRING";

setPropertiesClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("
"propertyValueList" ")";

alterTableSCOClases = ADD ( "addColumnClauseForAlter" [
"alterTableSCOClases" ] | "addConstraintClause" { "alterTableConstraintClauses" } |
"addConfigurationClause" { "alterTableConfigurationClauses" }) | MODIFY (
"modifyColumnClause" [ "alterTableSCOClases" ] | "modifyConstraintClause" {
"alterTableConstraintClauses" } | "modifyConfigurationClause" {
"alterTableConfigurationClauses" }) | DELETE ( "deleteColumnClause" [
"alterTableSCOClases" ] | "deleteConstraintClause" { "alterTableConstraintClauses" }
| "deleteConfigurationClause" { "alterTableConfigurationClauses" });

propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };

propertyValueList = "propertyValue" { "," "propertyValue" };

addColumnClauseForAlter = COLUMN "QUOTED_STRING" [ AT POSITION
"INTEGER_LITERAL" ] [ SET "setPropertiesClause" ];

addConstraintClause = "addUkPkClause" | "addFkClause" |
"addCheckConstraintClause";

alterTableConstraintClauses = ADD "addConstraintClause" | MODIFY
"modifyConstraintClause" | DELETE "deleteConstraintClause";

addConfigurationClause = "addIndexClause" | "addPartitionClause" |
"addPartitionKeyClause" | "addIndexColumnClause";

alterTableConfigurationClauses = ADD "addConfigurationClause" | MODIFY
"modifyConfigurationClause" | DELETE "deleteConfigurationClause";

modifyColumnClause = COLUMN "QUOTED_STRING" ( "renameClause" [
"moveToClause" ] [ SET "setPropertiesClause" ] | "moveToClause" [ SET
"setPropertiesClause" ] | SET "setPropertiesClause" );

modifyConstraintClause = "modifyUkPkClause" | "modifyFkClause" |
"modifyCheckConstraintClause";

modifyConfigurationClause = "modifyIndexClause" | "modifyPartitionClause" |
"modifyPartitionKeyClause" | "modifyIndexColumnClause";

deleteColumnClause = COLUMN "QUOTED_STRING";

deleteConstraintClause = UNIQUE_KEY "QUOTED_STRING" | PRIMARY_KEY
"QUOTED_STRING" | FOREIGN_KEY "QUOTED_STRING" | CHECK_
CONSTRAINT "QUOTED_STRING";

deleteConfigurationClause = INDEX "QUOTED_STRING" | PARTITION "QUOTED_
STRING" | PARTITION_KEY "QUOTED_STRING" | INDEX_COLUMN "QUOTED_
STRING" OF INDEX "QUOTED_STRING";

propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );

```

```

addUkPkClause = ( UNIQUE_KEY | PRIMARY_KEY ) "QUOTED_STRING" [ SET
"setUkPkPropertiesAndReferencesColumnsClauses" ];
addFkClause = FOREIGN_KEY "QUOTED_STRING" [ SET "setFkSubClauses" ];
addCheckConstraintClause = CHECK_CONSTRAINT "QUOTED_STRING" [ SET
"setPropertiesClause" ]
addIndexClause = INDEX "QUOTED_STRING" [ SET
"setSCOConfigurationPropertiesClauses" ];
addPartitionClause = PARTITION "QUOTED_STRING" [ SET
"setSCOConfigurationPropertiesClauses" ];
addPartitionKeyClause = PARTITION_KEY "QUOTED_STRING" [ SET
"setSCOConfigurationPropertiesClauses" ];
addIndexColumnClause = INDEX_COLUMN "QUOTED_STRING" OF INDEX
"QUOTED_STRING" [ SET "setSCOConfigurationPropertiesClauses" ];
moveToClause = MOVE TO POSITION "INTEGER_LITERAL";
modifyUkPkClause = ( UNIQUE_KEY | PRIMARY_KEY ) "QUOTED_STRING" (
"renameClause" [ SET "setUkPkPropertiesAndReferencesColumnsClauses" ] | SET
"setUkPkPropertiesAndReferencesColumnsClauses" );
modifyFkClause = FOREIGN_KEY "QUOTED_STRING" ( "renameClause" [ SET
"setFkSubClauses" ] | SET "setFkSubClauses" );
modifyCheckConstraintClause = CHECK_CONSTRAINT "QUOTED_STRING" (
"renameClause" [ SET "setPropertiesClause" ] | SET "setPropertiesClause" );
modifyIndexClause = INDEX "QUOTED_STRING" (
"renameSCOConfigurationClause" [ SET "setSCOConfigurationPropertiesClauses" ] |
SET "setSCOConfigurationPropertiesClauses" );
modifyPartitionClause = PARTITION "QUOTED_STRING" (
"renameSCOConfigurationClause" [ SET "setSCOConfigurationPropertiesClauses" ] |
SET "setSCOConfigurationPropertiesClauses" );
modifyPartitionKeyClause = PARTITION_KEY "QUOTED_STRING" (
"renameSCOConfigurationClause" [ SET "setSCOConfigurationPropertiesClauses" ] |
SET "setSCOConfigurationPropertiesClauses" );
modifyIndexColumnClause = INDEX_COLUMN "QUOTED_STRING" OF INDEX
"QUOTED_STRING" ( "renameSCOConfigurationClause" [ SET
"setSCOConfigurationPropertiesClauses" ] | SET
"setSCOConfigurationPropertiesClauses" );
setUkPkPropertiesAndReferencesColumnsClauses = "setPropertiesClause" [ SET (
REF | REFERENCE ) "constraintColumnReferencesClause" ] | ( REF | REFERENCE )
"constraintColumnReferencesClause";
setFkSubClauses = "setPropertiesClause" [ SET ( REF | REFERENCE )
"setFkReferencesClauses" ] | ( REF | REFERENCE ) "setFkReferencesClauses";
setSCOConfigurationPropertiesClauses = PROPERTIES "(" "propertyNameList" ")"
VALUES "(" "propertyValueList" ")";
renameSCOConfigurationClause = RENAME TO "QUOTED_STRING";
constraintColumnReferencesClause = COLUMNS "(" "quotedNameList" ")";

```

```
setFkReferencesClauses = "constraintColumnReferencesClause" [ SET ( REF |  
REFERENCE ) "constraintUkReferencesClause" ] | "constraintUkReferencesClause" [  
SET ( REF | REFERENCE ) "constraintColumnReferencesClause" ] ;  
quotedNameList = "QUOTED_STRING" { "," "QUOTED_STRING" } ;  
constraintUkReferencesClause = ( UNIQUE_KEY | PRIMARY_KEY ) "QUOTED_  
STRING" [ OF ( TABLE | VIEW ) "QUOTED_STRING" ] ;
```

## Keywords and Parameters

**alterTableCommand**

This clause alters a table.

**QUOTED\_STRING**

name of the table.

**renameClause**

renames a table with a different name.

**setPropertyClause**

Used to set properties (core, logical, physical, user-defined) for tables, columns, unique keys, foreign keys, primary keys, and check constraints.

Basic properties for TABLE:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the table

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the table

Name: UOID

Type: STRING

Valid Values: N/A

Default: N/A

Basic properties for COLUMN:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the table



Name: DESCRIPTION  
Type: STRING(4000)  
Valid Values: N/A  
Default: "  
Description of the table

Name: DATATYPE  
Type: STRING  
Valid Values: NUMBER, VARCHAR, VARCHAR2, DATE, FLOAT  
Default: NUMBER  
The datatype of a column

Name: LENGTH  
Type: NUMBER  
Valid Values:  
Default: 1  
The length of a number

Name: PRECISION  
Type: NUMBER  
Valid Values: 0 - 39  
Default: 1  
The precision of a number.

Name: SCALE  
Type: NUMBER  
Valid Values: -85 - 125  
Default: 1  
The scale of a number.

Name: UOID  
Type: STRING  
Valid Values: N/A  
Default: N/A

Basic properties for INDEX, PARTITION, PARTITION\_KEY, INDEX\_COLUMN:

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the Index, Partition, PartitionKey, IndexColumn.

Properties for TABLE:

Name: ANALYZE\_TABLE\_ESTIMATE\_PERCENT

Type: NUMBER

Valid Values: 0 - 100

Default: 99

Value represents the sample size as a percentage of total rows. When set to a nonzero value, Builder generates a DDL script to analyze the table.

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: HASH\_PARTITION\_TABLESPACE\_LIST

Type: STRING

Valid Values: N/A

Default: "

A comma separated list of tablespaces to use for [sub]partition storage.

Name: HASH\_SUBPARTITION\_NUMBER

Type: NUMBER

Valid Values: 2 - 63999

Default: 2

To create Hash partition, specify the number of Hash subpartition.

Name: LOGGING\_MODE

Type: STRING

Valid Values: NOLOGGING, LOGGING

Default: LOGGING

Recovery requirements for a data warehouse : Logging or not logging to Redo Log File.

Name: PARALLEL\_ACCESS\_MODE

Type: STRING

Valid Values: NOPARALLEL, PARALLEL

Default: PARALLEL

Enables or disables parallel processing when the table is created. Also enables or disables parallel processing or access.

Name: TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Use the Tablespace parameter to specify the name of tablespace.

Properties for UNIQUE\_KEY:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: USING\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

Use of Index on Dimension Constraint

## Properties for FOREIGN\_KEY:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: USING\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

Use of Index on Dimension Constraint

## Properties for CHECK\_CONSTRAINT:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: USING\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

Use of Index on Dimension Constraint

Properties for INDEX:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: INDEX\_TYPE

Type: STRING

Valid Values: BITMAP, UNIQUE, NO\_INDEX

Default: UNIQUE

The types of Indexes created on Dimension are BITMAP, UNIQUE or a non-specific index.

Name: LOCAL\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

A local index is constructed so that it reflects the structure of the underlying table. It is equipartitioned with the underlying table, meaning that it is partitioned on the same columns as the underlying table, creates the same number of partitions or subpartitions, and gives them the same partition bounds as corresponding partitions of the underlying table.

Name: LOGGING\_MODE

Type: STRING

Valid Values: NOLOGGING, LOGGING

Default: LOGGING

Recovery requirements for a data warehouse : Logging or not logging to Redo Log File.

Name: PARALLEL\_ACCESS\_MODE

Type: STRING

Valid Values: NOPARALLEL, PARALLEL

Default: PARALLEL

Enables or disables parallel processing when the table is created. Also enables or disables parallel processing or access.

Name: TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Use the Tablespace parameter to specify the name of tablespace.

Name: USING\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

Use of Index on Dimension Constraint

Properties for PARTITION:

Name: DATE\_LESS\_THAN

Type: STRING

Valid Values: N/A

Default: TO\_DATE('01-JAN-1999', 'DD-MON-YYYY')

Value that represents upper bound of partition stored in warehouse key column for the Days Dimension.

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: EMPTY\_STRING

Use the Tablespace parameter to specify the name of tablespace.

Name: VALUES\_LESS\_THAN

Type: STRING

Valid Values: N/A

Default: TO\_DATE('01-JAN-1999', 'DD-MON-YYYY')

Noninclusive upper bound for the current partition. Type a comma separated list in the same order as the columns in range partitioned key.

Name: VALUE\_LESS\_THAN

Type: STRING

Valid Values: N/A

Default: "

Noninclusive upper bound for the current partition. Type a comma separated list in the same order as the columns in range partitioned key.

Properties for PARTITION\_KEY:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: TYPE

Type: STRING

Valid Values: HASH, RANGE

Default: RANGE

Oracle partitions the storage space and stores rows according to a Hash Algorithm or specified ranges.

Note:

1. N/A means any valid character in supported character set.
2. " represents an empty string

**alterTableSCOClauses**

This clause will add, modify, delete, columns, configuration, and keys.

**propertyNameList**

The list of properties.

**propertyValueList**

The list of property values.

**addColumnClauseForAlter**

This clause adds a column at a particular position

**QUOTED\_STRING**

The column name.

**addConstraintClause**

Adds primary and unique key, and add check constraints.

**alterTableConstraintClauses**

This clause alters table constraints through addition, modification, and deletion.

**addConfigurationClause**

This clause will add configuration objects.

**alterTableConfigurationClauses**

This clause has modify and delete configuration clause.

**modifyColumnClause**

This clause renames, set properties, and move columns.

**modifyConstraintClause**

This clause modifies keys and check constraints

**modifyConfigurationClause**



This clause will modify configuration objects.

`deleteColumnClause`

This clause deletes a column.

`deleteConstraintClause`

This clause deletes a key or check constraint.

`deleteConfigurationClause`

This clause deletes a configuration object.

`QUOTED_STRING`

Either index, partition, partition\_key, or index column name.

`propertyValue`

This clause adds the property values.

`addUkPkClause`

This clause adds the adds unique key and primary keys.

`QUOTED_STRING`

name of the unique key or primary key.

`addFkClause`

This clause adds foreign key

`QUOTED_STRING`

Name of the foreign key.

`addCheckConstraintClause`

add a check constraint.

`QUOTED_STRING`

Name of the CheckConstraint.

`addIndexClause`

This clause adds an index.

QUOTED\_STRING

Name of the index.

addPartitionClause

This clause adds a partition.

QUOTED\_STRING

Name of the partition.

addPartitionKeyClause

This clause adds a partition key.

QUOTED\_STRING

Name of the partition key. This should be a column identifier.

addIndexColumnClause

This clause will add indexColumn to a specified index.

QUOTED\_STRING

Index name

moveToClause

This clause will move the column to given position.

modifyUkPkClause

It modifies unique or primary key.

modifyFkClause

This clause modifies the foreign key.

modifyCheckConstraintClause

This clause modifies the check constraint.

modifyIndexClause

This clause modifies the Index

**QUOTED\_STRING**

Name of the index.

**modifyPartitionClause**

This clause modifies a partition

**QUOTED\_STRING**

Name of the partition.

**modifyPartitionKeyClause**

This clause modifies a partition key.

**QUOTED\_STRING**

Name of the partition key.

**modifyIndexColumnClause**

Modifies the Index Column. The first quoted\_string in this clause denotes index column name, and the latter denotes index.

**setUkPkPropertiesAndReferencesColumnsClauses**

This clause adds properties and references to columns

**setFkSubClauses**

This clause set references to a foreign key.

**setSCOConfigurationPropertiesClauses**

Set the configuration properties for the following objects Index: LOGGING\_MODE, PARALLEL\_ACCESS\_MODE, TABLESPACE, INDEX\_TYPE, LOCAL\_INDEX, DEPLOYABLE Partition: DATE\_LESS\_THAN, TABLESPACE, DEPLOYABLE Partition\_key: TYPE, DEPLOYABLE  
RelationalCmdParser\$constraintColumnReferencesClause = This clause provides names of all columns.

**renameSCOConfigurationClause**

This clause renames configuration objects.

**constraintColumnReferencesClause**

RelationalCmdParser\$constraintColumnReferencesClause??

**setFkReferencesClauses**

This clause sets foreign key references.

**quotedNameList**

This clause gives column names.

**constraintUkReferencesClause**

The first QUOTED\_STRING denotes the UniqueKey or Primary key name, and the latter denotes the table's or view's name.

**Examples**

OMBALTER TABLE 'new\_table' SET PROPERTIES (DESCRIPTION, BUSINESS\_NAME) VALUES ('this is an altered desc of new table', 'Altered New Table') This will alter a table named "NEW\_TABLE", its description is "this is an altered desc of new table", and business name is "Altered New Table".

**See Also**

OMBALTER, OMBCREATE TABLE, OMBDROP TABLE

# OMBALTER VIEW

## Purpose

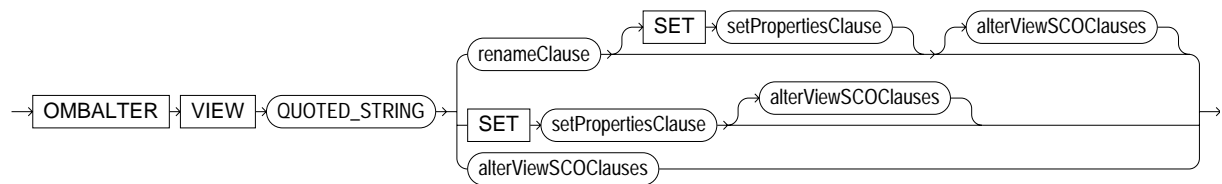
OMBALTER VIEW - To alter properties and definition of a view.

## Prerequisites

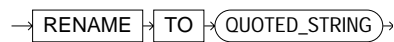
In the context of an Oracle Module.

## Syntax Diagrams

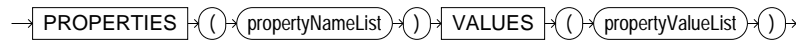
alterViewCommand



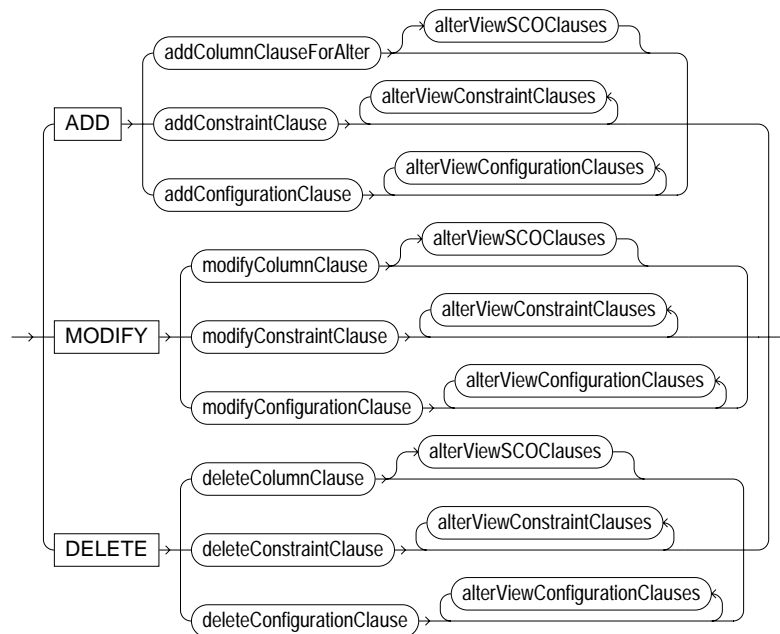
renameClause



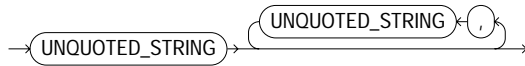
setPropertyClause



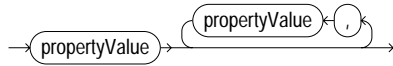
alterViewSCOClauses



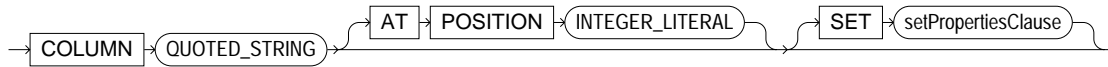
propertyNameList



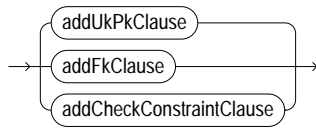
propertyValueList



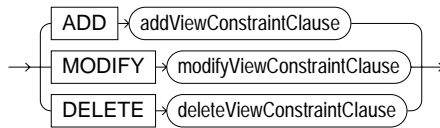
addColumnClauseForAlter



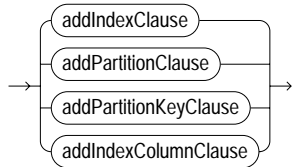
addConstraintClause



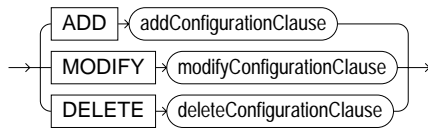
alterViewConstraintClauses



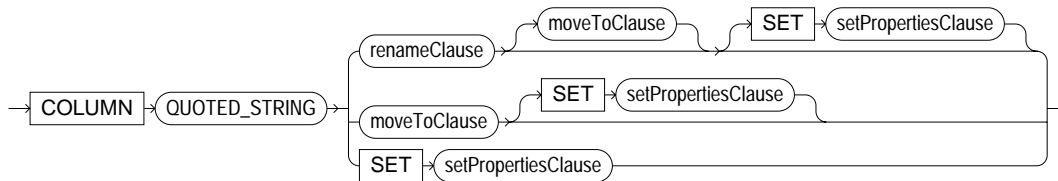
addConfigurationClause



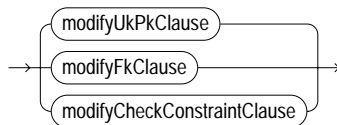
alterViewConfigurationClauses



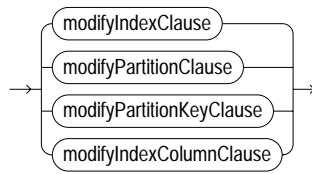
modifyColumnClause



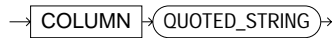
modifyConstraintClause



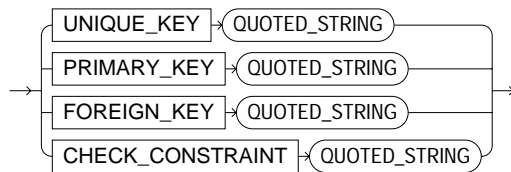
modifyConfigurationClause



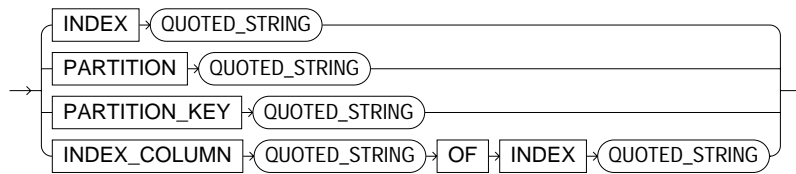
deleteColumnClause



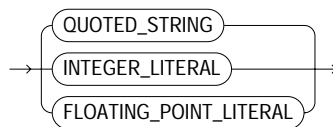
deleteConstraintClause



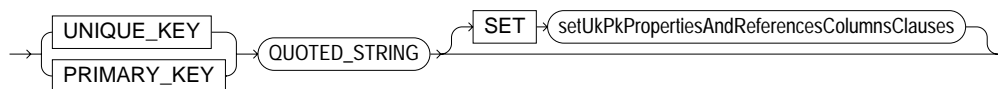
deleteConfigurationClause



property Value



addUkPkClause



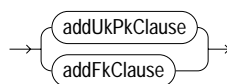
addFkClause



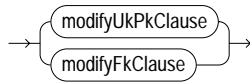
addCheckConstraintClause



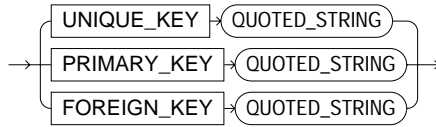
addViewConstraintClause



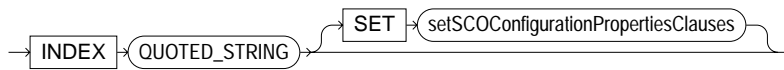
modifyViewConstraintClause



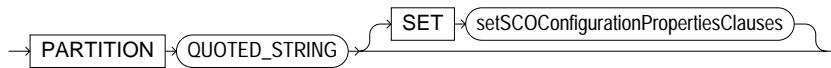
deleteViewConstraintClause



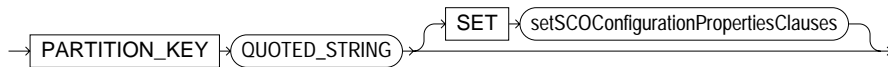
addIndexClause



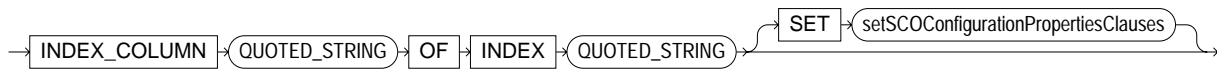
addPartitionClause



addPartitionKeyClause



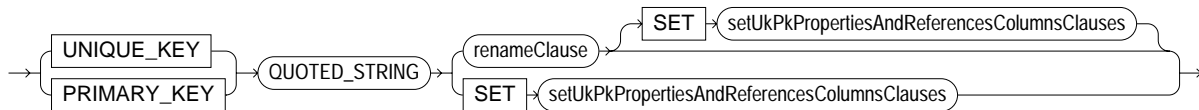
addIndexColumnClause



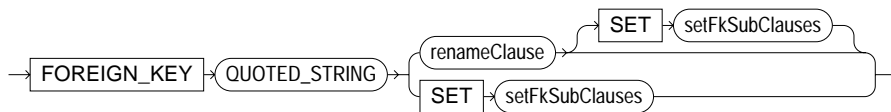
moveToClause



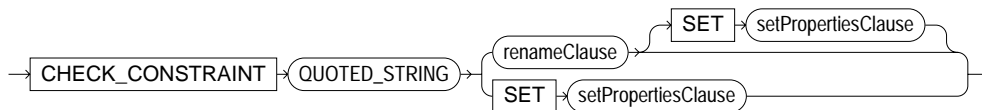
modifyUkPkClause



modifyFkClause

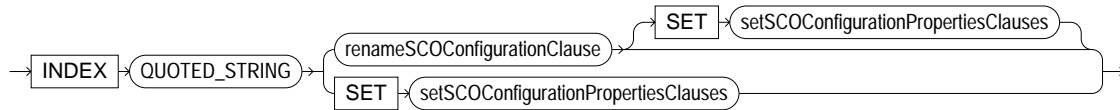


modifyCheckConstraintClause

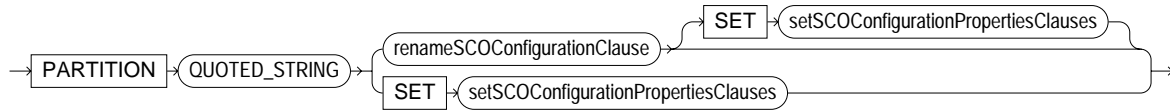




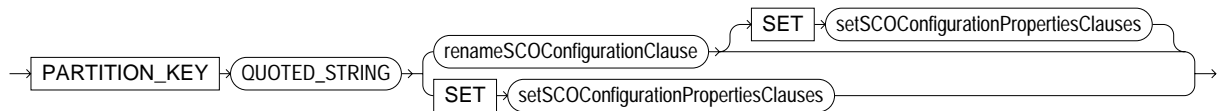
modifyIndexClause



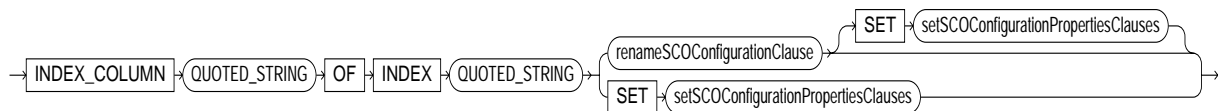
modifyPartitionClause



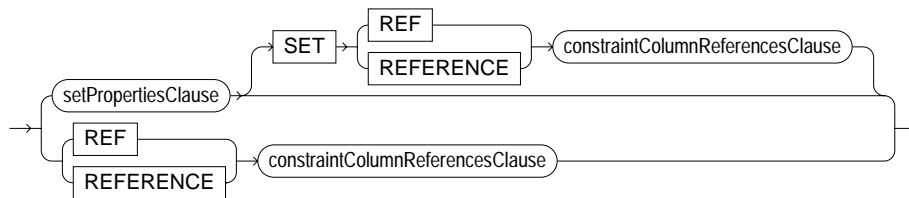
modifyPartitionKeyClause



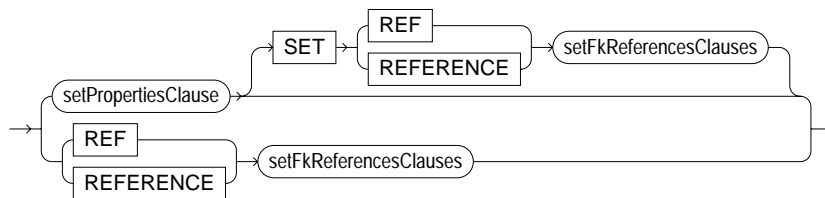
modifyIndexColumnClause



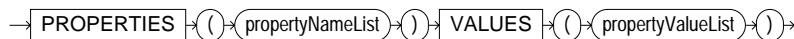
setUkPkPropertiesAndReferencesColumnsClauses



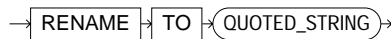
setFkSubClauses



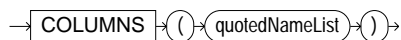
setSCOConfigurationPropertiesClauses



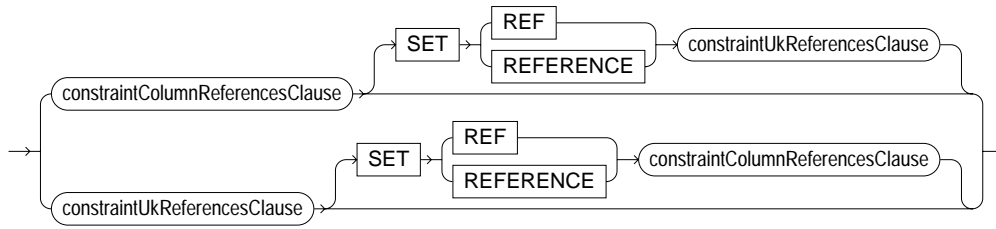
renameSCOConfigurationClause



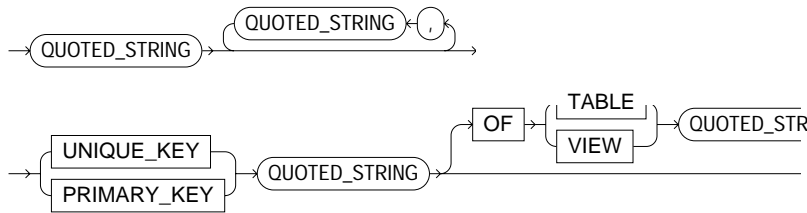
constraintColumnReferencesClause



setFkReferencesClauses



quotedNameList



## Syntax

```
alterViewCommand = OMBALTER ( VIEW "QUOTED_STRING" ( "renameClause" [
SET "setPropertiesClause" ] [ "alterViewSCOClauses" ] | SET "setPropertiesClause" [
"alterViewSCOClauses" ] | "alterViewSCOClauses" ) );

renameClause = RENAME TO "QUOTED_STRING";

setPropertiesClause = PROPERTIES (" "propertyNameList" ") VALUES ("
"propertyValueList" ");

alterViewSCOClauses = ADD ( "addColumnClauseForAlter" [ "alterViewSCOClauses"
] | "addConstraintClause" { "alterViewConstraintClauses" } |
"addConfigurationClause" { "alterViewConfigurationClauses" } ) | MODIFY (
"modifyColumnClause" [ "alterViewSCOClauses" ] | "modifyConstraintClause" {
"alterViewConstraintClauses" } | "modifyConfigurationClause" {
"alterViewConfigurationClauses" } ) | DELETE ( "deleteColumnClause" [
"alterViewSCOClauses" ] | "deleteConstraintClause" { "alterViewConstraintClauses" }
| "deleteConfigurationClause" { "alterViewConfigurationClauses" } );

propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };

propertyValueList = "propertyValue" { "," "propertyValue" };

addColumnClauseForAlter = COLUMN "QUOTED_STRING" [ AT POSITION
"INTEGER_LITERAL" ] | SET "setPropertiesClause" ];

addConstraintClause = "addUkPkClause" | "addFkClause" |
"addCheckConstraintClause";

alterViewConstraintClauses = ADD "addViewConstraintClause" | MODIFY
"modifyViewConstraintClause" | DELETE "deleteViewConstraintClause";

addConfigurationClause = "addIndexClause" | "addPartitionClause" |
"addPartitionKeyClause" | "addIndexColumnClause";

alterViewConfigurationClauses = ADD "addConfigurationClause" | MODIFY
"modifyConfigurationClause" | DELETE "deleteConfigurationClause";
```

```

modifyColumnClause = COLUMN "QUOTED_STRING" ( "renameClause" [
"moveToClause" ] [ SET "setPropertiesClause" ] | "moveToClause" [ SET
"setPropertiesClause" ] | SET "setPropertiesClause" );

modifyConstraintClause = "modifyUkPkClause" | "modifyFkClause" |
"modifyCheckConstraintClause";

modifyConfigurationClause = "modifyIndexClause" | "modifyPartitionClause" |
"modifyPartitionKeyClause" | "modifyIndexColumnClause";

deleteColumnClause = COLUMN "QUOTED_STRING";

deleteConstraintClause = UNIQUE_KEY "QUOTED_STRING" | PRIMARY_KEY
"QUOTED_STRING" | FOREIGN_KEY "QUOTED_STRING" | CHECK_
CONSTRAINT "QUOTED_STRING";

deleteConfigurationClause = INDEX "QUOTED_STRING" | PARTITION "QUOTED_
STRING" | PARTITION_KEY "QUOTED_STRING" | INDEX_COLUMN "QUOTED_
STRING" OF INDEX "QUOTED_STRING";

propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );

addUkPkClause = ( UNIQUE_KEY | PRIMARY_KEY ) "QUOTED_STRING" [ SET
"setUkPkPropertiesAndReferencesColumnsClauses" ];

addFkClause = FOREIGN_KEY "QUOTED_STRING" [ SET "setFkSubClauses" ];

addCheckConstraintClause = CHECK_CONSTRAINT "QUOTED_STRING" [ SET
"setPropertiesClause" ];

addViewConstraintClause = "addUkPkClause" | "addFkClause";

modifyViewConstraintClause = "modifyUkPkClause" | "modifyFkClause";

deleteViewConstraintClause = UNIQUE_KEY "QUOTED_STRING" | PRIMARY_
KEY "QUOTED_STRING" | FOREIGN_KEY "QUOTED_STRING";

addIndexClause = INDEX "QUOTED_STRING" [ SET
"setSCOConfigurationPropertiesClauses" ];

addPartitionClause = PARTITION "QUOTED_STRING" [ SET
"setSCOConfigurationPropertiesClauses" ];

addPartitionKeyClause = PARTITION_KEY "QUOTED_STRING" [ SET
"setSCOConfigurationPropertiesClauses" ];

addIndexColumnClause = INDEX_COLUMN "QUOTED_STRING" OF INDEX
"QUOTED_STRING" [ SET "setSCOConfigurationPropertiesClauses" ];

moveToClause = MOVE TO POSITION "INTEGER_LITERAL";

modifyUkPkClause = ( UNIQUE_KEY | PRIMARY_KEY ) "QUOTED_STRING" (
"renameClause" [ SET "setUkPkPropertiesAndReferencesColumnsClauses" ] | SET
"setUkPkPropertiesAndReferencesColumnsClauses" );

modifyFkClause = FOREIGN_KEY "QUOTED_STRING" ( "renameClause" [ SET
"setFkSubClauses" ] | SET "setFkSubClauses" );

modifyCheckConstraintClause = CHECK_CONSTRAINT "QUOTED_STRING" (
"renameClause" [ SET "setPropertiesClause" ] | SET "setPropertiesClause" );

modifyIndexClause = INDEX "QUOTED_STRING" (
"renameSCOConfigurationClause" [ SET "setSCOConfigurationPropertiesClauses" ] |
SET "setSCOConfigurationPropertiesClauses" );

```

```

modifyPartitionClause = PARTITION "QUOTED_STRING" (
"renameSCOConfigurationClause" [ SET "setSCOConfigurationPropertiesClauses" ] |
SET "setSCOConfigurationPropertiesClauses" );

modifyPartitionKeyClause = PARTITION_KEY "QUOTED_STRING" (
"renameSCOConfigurationClause" [ SET "setSCOConfigurationPropertiesClauses" ] |
SET "setSCOConfigurationPropertiesClauses" );

modifyIndexColumnClause = INDEX_COLUMN "QUOTED_STRING" OF INDEX
"QUOTED_STRING" ( "renameSCOConfigurationClause" [ SET
"setSCOConfigurationPropertiesClauses" ] | SET
"setSCOConfigurationPropertiesClauses" );

setUkPkPropertiesAndReferencesColumnsClauses = "setPropertiesClause" [ SET (
REF | REFERENCE ) "constraintColumnReferencesClause" ] | ( REF | REFERENCE )
"constraintColumnReferencesClause";

setFkSubClauses = "setPropertiesClause" [ SET ( REF | REFERENCE )
"setFkReferencesClauses" ] | ( REF | REFERENCE ) "setFkReferencesClauses";

setSCOConfigurationPropertiesClauses = PROPERTIES "(" "propertyNameList" ")"
VALUES "(" "propertyValueList" ")";

renameSCOConfigurationClause = RENAME TO "QUOTED_STRING";

constraintColumnReferencesClause = COLUMNS "(" "quotedNameList" ")";

setFkReferencesClauses = "constraintColumnReferencesClause" [ SET ( REF |
REFERENCE ) "constraintUkReferencesClause" ] | "constraintUkReferencesClause" [
SET ( REF | REFERENCE ) "constraintColumnReferencesClause" ];

quotedNameList = "QUOTED_STRING" { "," "QUOTED_STRING" };

constraintUkReferencesClause = ( UNIQUE_KEY | PRIMARY_KEY ) "QUOTED_
STRING" [ OF ( TABLE | VIEW ) "QUOTED_STRING" ];

```

## Keywords and Parameters

**alterViewCommand**

This clause alters a view.

**QUOTED\_STRING**

name of the view.

**renameClause**

renames a table with a different name.

**setPropertiesClause**

Used to set properties (core, logical, physical, user-defined) for tables, columns, unique keys, foreign keys, primary keys, and check constraints.

Basic properties for VIEW, MATERIALIZED\_VIEW:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the View, MaterializedView

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the View, MaterializedView

Name: VIEW\_QUERY

Type: STRING(4000)

Valid Values: N/A

Default: "

Sets the query definition in View and MaterializedView.

Name: UOID

Type: STRING

Valid Values: N/A

Default: N/A

Basic properties for INDEX, PARTITION, PARTITION\_KEY, INDEX\_COLUMN:

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the Index, Partition, PartitionKey, IndexColumn in a MaterializedView.

Properties for VIEW:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Properties for UNIQUE\_KEY:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: USING\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

Use of Index on Dimension Constraint

Properties for FOREIGN\_KEY:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: USING\_INDEX  
Type: BOOLEAN  
Valid Values: true, false  
Default: false  
Use of Index on Dimension Constraint

Properties for CHECK\_CONSTRAINT:

Name: DEPLOYABLE  
Type: BOOLEAN  
Valid Values: true, false  
Default: true  
Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE  
Type: STRING(30)  
Valid Values: N/A  
Default: "  
Tablespace to store Index on Table Constraint

Name: USING\_INDEX  
Type: BOOLEAN  
Valid Values: true, false  
Default: false  
Use of Index on Dimension Constraint

Note:

1. N/A means any valid character in supported character set.
2. " represents an empty string

alterViewSCOClauses

This clause alters the view clause.

propertyNameList

The list of properties.

propertyValueList

The list of property values.

addColumnClauseForAlter

This clause adds a column at a particular position

QUOTED\_STRING

The column name.

addConstraintClause

Adds primary and unique key, and add check constraints.

alterViewConstraintClauses

This clause alters the view's constraint clause.

addConfigurationClause

This clause will add configuration objects.

alterViewConfigurationClauses

This clause alters the view's configuration clause.

modifyColumnClause

This clause renames, set properties, and move columns.

modifyConstraintClause

This clause modifies keys and check constraints

modifyConfigurationClause

This clause will modify configuration objects.

deleteColumnClause

This clause deletes a column.

deleteConstraintClause

This clause deletes a key or check constraint.



`deleteConfigurationClause`

This clause deletes a configuration object.

`QUOTED_STRING`

Either index, partition, partition\_key, or index column name.

`propertyValue`

This clause adds the property values.

`addUkPkClause`

This clause adds the adds unique key and primary keys.

`QUOTED_STRING`

name of the unique key or primary key.

`addFkClause`

This clause adds foreign key

`QUOTED_STRING`

Name of the foreign key.

`addCheckConstraintClause`

add a check constraint.

`QUOTED_STRING`

Name of the CheckConstraint.

`addViewConstraintClause`

This clause adds the view's configuration clause.

`modifyViewConstraintClause`

= This clause modifies the view's constraint clause.

`deleteViewConstraintClause`

This clause deletes the view's constraint.

`addIndexClause`

This clause adds an index.

QUOTED\_STRING

Name of the index.

addPartitionClause

This clause adds a partition.

QUOTED\_STRING

Name of the partition.

addPartitionKeyClause

This clause adds a partition key.

QUOTED\_STRING

Name of the partition key. This should be a column identifier.

addIndexColumnClause

This clause will add indexColumn to a specified index.

QUOTED\_STRING

Index name

moveToClause

This clause will move the column to given position.

modifyUkPkClause

It modifies unique or primary key.

modifyFkClause

This clause modifies the foreign key.

modifyCheckConstraintClause

This clause modifies the check constraint.

modifyIndexClause

This clause modifies the Index

QUOTED\_STRING

Name of the index.

modifyPartitionClause

This clause modifies a partition

QUOTED\_STRING

Name of the partition.

modifyPartitionKeyClause

This clause modifies a partition key.

QUOTED\_STRING

Name of the partition key.

modifyIndexColumnClause

Modifies the Index Column. The first quoted\_string in this clause denotes index column name, and the latter denotes index.

setUkPkPropertiesAndReferencesColumnsClauses

This clause adds properties and references to columns

setFkSubClauses

This clause set references to a foreign key.

setSCOConfigurationPropertiesClauses

Set the configuration properties for the following objects Index: LOGGING\_MODE, PARALLEL\_ACCESS\_MODE, TABLESPACE, INDEX\_TYPE, LOCAL\_INDEX, DEPLOYABLE Partition: DATE\_LESS\_THAN, TABLESPACE, DEPLOYABLE Partition\_key: TYPE, DEPLOYABLE  
RelationalCmdParser\$constraintColumnReferencesClause = This clause provides names of all columns.

renameSCOConfigurationClause

This clause renames configuration objects.

constraintColumnReferencesClause

RelationalCmdParser\$constraintColumnReferencesClause??

**setFkReferencesClauses**

This clause sets foreign key references.

**quotedNameList**

This clause gives column names.

**constraintUkReferencesClause**

The first QUOTED\_STRING denotes the UniqueKey or Primary key name, and the latter denotes the table's or view's name.

**Examples**

```
OMBALTER VIEW 'NEW_VIEW' SET PROPERTIES (DESCRIPTION, BUSINESS_
NAME) VALUES
```

('this is an altered desc of new view', 'Altered New View') This will alter a view named "NEW\_VIEW", its description is "this is an altered desc of new view", and business name is "Altered New View".

**See Also**

OMBRETRIEVE VIEW, OMBCREATE VIEW, OMBDROP VIEW

---

---

# OMBCREATE

This chapter contains the following topics:

---

<a href="#">OMBCREATE</a> on page 4-2	<a href="#">OMBCREATE FUNCTION</a> on page 4-47	<a href="#">OMBCREATE PROCESS_FLOW_MODULE</a> on page 4-175
<a href="#">OMBCREATE ADVANCED_QUEUE</a> on page 4-6	<a href="#">OMBCREATE LOCATION</a> on page 4-52	<a href="#">OMBCREATE PROCESS_FLOW_PACKAGE</a> on page 4-178
<a href="#">OMBCREATE COLLECTION</a> on page 4-10	<a href="#">OMBCREATE MAPPING</a> on page 4-55	<a href="#">OMBCREATE PROJECT</a> on page 4-181
<a href="#">OMBCREATE CONNECTOR</a> on page 4-14	<a href="#">OMBCREATE MATERIALIZED_VIEW</a> on page 4-132	<a href="#">OMBCREATE RUNTIME_REPOSITORY_CONNECTION</a> on page 4-184
<a href="#">OMBCREATE CUBE_TABLE</a> on page 4-17	<a href="#">OMBCREATE OBJECT_TYPE</a> on page 4-146	<a href="#">OMBCREATE SEQUENCE</a> on page 4-187
<a href="#">OMBCREATE DEPLOYMENT_ACTION_PLAN</a> on page 4-25	<a href="#">OMBCREATE ORACLE_MODULE</a> on page 4-149	<a href="#">OMBCREATE SNAPSHOT</a> on page 4-190
<a href="#">OMBCREATE EXTERNAL_TABLE</a> on page 4-28	<a href="#">OMBCREATE PACKAGE</a> on page 4-158	<a href="#">OMBCREATE TABLE</a> on page 4-195
<a href="#">OMBCREATE FLAT_FILE</a> on page 4-36	<a href="#">OMBCREATE PROCEDURE</a> on page 4-161	<a href="#">OMBCREATE VIEW</a> on page 4-210
<a href="#">OMBCREATE FLAT_FILE_MODULE</a> on page 4-44	<a href="#">OMBCREATE PROCESS_FLOW</a> on page 4-165	

---

# OMBCREATE

## Purpose

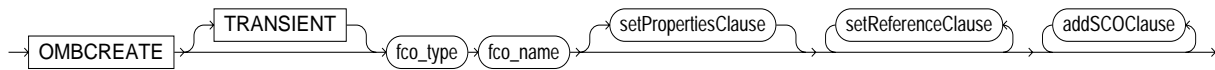
OMBCREATE - Create a new component.

## Prerequisites

Should be in the context that stores the component type.

## Syntax Diagrams

createCommand



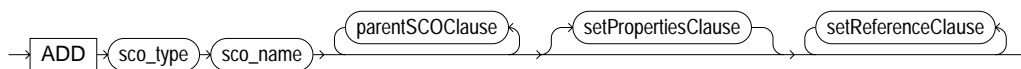
setPropertiesClause



setReferenceClause



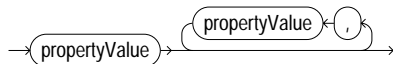
addSCOClauses



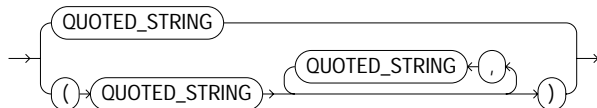
propertyNameList



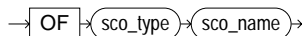
propertyValueList



quotedNameList



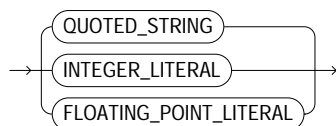
parentSCOClauses



propertyName



propertyValue



## Syntax

```
createCommand = OMBCREATE [ TRANSIENT ] "fco_type" "fco_name" [
  "setPropertyClause" ] { "setReferenceClause" } { "addSCOClause" };
setPropertyClause = SET PROPERTIES "(" "propertyNameList" ")" VALUES "("
  "propertyValueList" ")";
setReferenceClause = SET ( REF | REFERENCE ) [ "qualifier" ] "type"
  "quotedNameList" [ { "parentSCOClause" } OF "fco_type" "fco_name" ];
addSCOClause = ADD "sco_type" "sco_name" { "parentSCOClause" } [
  "setPropertyClause" ] { "setReferenceClause" };
propertyNameList = "propertyName" { "," "propertyName" };
propertyValueList = "propertyValue" { "," "propertyValue" };
quotedNameList = "QUOTED_STRING" | "(" "QUOTED_STRING" { "," "QUOTED_
  STRING" } ")";
parentSCOClause = OF "sco_type" "sco_name";
propertyName = "UNQUOTED_STRING";
propertyValue = "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
  POINT_LITERAL";
```

## Keywords and Parameters

**createCommand**

Specify metadata for a new component.

**TRANSIENT**

Keyword used to specify that the object being created will not be persisted in the repository.

**fco\_type**

The type of the component.

**fco\_name**

The physical name of the component in single quotes.

**setPropertyClause**

Set object properties.

setReferenceClause

Specify reference objects.

qualifier

Specify which reference to set, if there are more than one pointing to the same type.

addSCOClause

Add a new child object to the component.

propertyNameList

A list of property names.

propertyValueList

A list of property values.

quotedNameList

A list of single-quoted physical names.

parentSCOClause

Used to specify the path from a child object to the component

propertyName

An unquoted string representing the name of a property.

propertyValue

The value of a property.

## Examples

This is an example for creating an empty table:

```
OMBCREATE TABLE 'T1' SET PROPERTIES (DESCRIPTION) VALUES ('My First Table')
```

The following statement creates a view with a column:



```
OMBCREATE VIEW 'V1'  
ADD COLUMN 'COL1'  
SET PROPERTIES (DATATYPE) VALUES ('VARCHAR2')
```

**See Also**

OMBALTER, OMBDROP

# OMBCREATE ADVANCED\_QUEUE

## Purpose

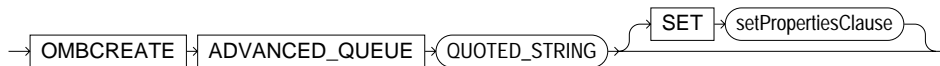
OMBCREATE ADVANCED\_QUEUE - To create an Advanced Queue.

## Prerequisites

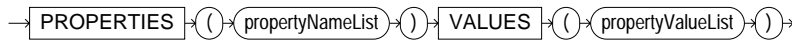
Should be in the context of an Oracle Module. The Object Type set as Payload Type should exist in the same Oracle Module.

## Syntax Diagrams

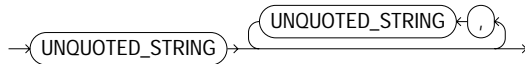
createAQCommand



setPropertiesClause



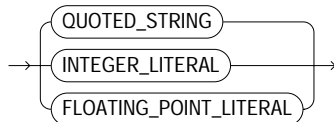
propertyNameList



propertyValueList



propertyValue



## Syntax

```
createAQCommand = OMBCREATE ( ADVANCED_QUEUE "QUOTED_STRING" [
SET "setPropertiesClause" ] );
```

```
setPropertiesClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("
"propertyValueList" " )";
```

```
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
```

```
propertyValueList = "propertyValue" { "," "propertyValue" };
```

```
propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );
```

## Keywords and Parameters

`createAQCommand`

Creates an Advanced Queue with the given name.

`setPropertyClause`

Sets properties (core, logical, physical, user-defined) for Advanced Queue.

Valid properties are shown below:

Basic properties for `ADVANCED_QUEUE`:

Name: `BUSINESS_NAME`

Type: `STRING(200)`

Valid Values: N/A

Default: ""

Business name of the Advanced Queue

Name: `DESCRIPTION`

Type: `STRING(4000)`

Valid Values: N/A

Default: ""

Description of the Advanced Queue

Name: `PAYLOAD_TYPE`

Type: `STRING(4000)`

Valid Values: N/A

Default: ""

PayLoad Type of the Advanced Queue. This has to be the name of an Object Type(`OBJECT_TYPE`) existing in the same Oracle Module.

Properties for `ADVANCED_QUEUE`:

Name: `DEPLOYABLE`

Type: `BOOLEAN`

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those objects marked as Deployable = true.

Name: GENERATE\_ADVANCED\_QUEUE

Type: BOOLEAN

Valid Values: true, false

Default: true

Generate the code to create the Advanced Queue.

Name: GENERATE\_OBJECT\_TYPE

Type: BOOLEAN

Valid Values: true, false

Default: true

Generate a script to create the Object Type.

Name: GENERATE\_QUEUE\_TABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Generate code to create the queue table that will persist the messages of this Advanced Queue.

Name: GENERATE\_TEMPORARY\_TABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Generate the code to create the temporary table.

Name: QUEUE\_TABLE\_NAME

Type: STRING(30)

Valid Values: N/A

Default: "

The name of the queue table that is used to persist the messages in this Advanced Queue.

Name: TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Use the Tablespace parameter to specify the name of tablespace.

Note:

1. N/A means any valid character in supported character set.
2. " " represents an empty string

propertyNameList

The list of properties.

propertyValueList

The list of property values.

propertyValue

This clause adds the property values.

## Examples

```
OMBCREATE ADVANCED_QUEUE 'NEW_ADVANCED_QUEUE' SET PROPERTIES  
(DESCRIPTION, PAYLOAD_TYPE) VALUES ('this is an Advanced Queue', 'SOME_  
OBJECT_TYPE') This will create an Advanced Queue named "NEW_ADVANCED_  
QUEUE", its description is "this is an Advanced Queue" and its Payload type is Object  
Type 'SOME_OBJECT_TYPE'.
```

## See Also

```
OMBCREATE ADVANCED_QUEUE, OMBALTER ADVANCED_QUEUE, OMBDROP  
ADVANCED_QUEUE
```

## OMBCREATE COLLECTION

### Purpose

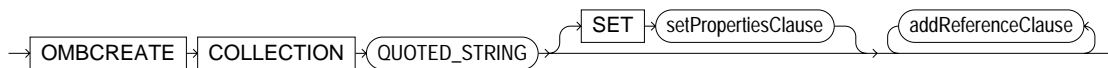
OMBCREATE COLLECTION - This is an arbitrary grouping mechanism in OWB. Any first class object can be added to the collection.

### Prerequisites

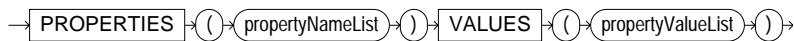
Should be in the context of a project, before creating a collection.

### Syntax Diagrams

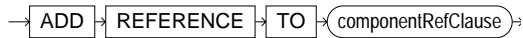
createCollectionCommand



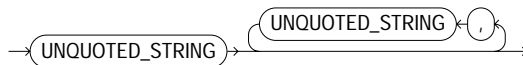
setPropertiesClause



addReferenceClause



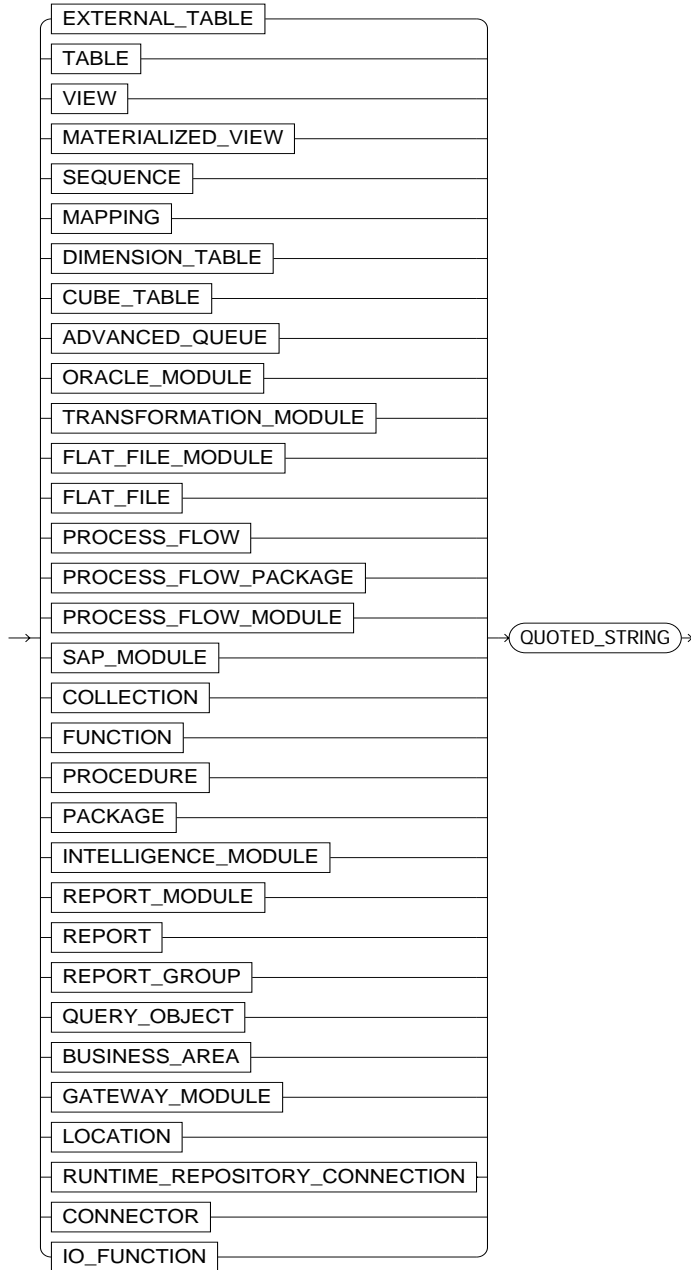
propertyNameList



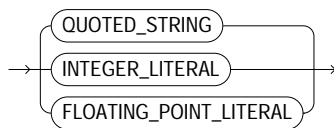
propertyValueList



componentRefClause



propertyValue



## Syntax

```
createCollectionCommand = OMBCREATE ( COLLECTION "QUOTED_STRING" [
SET "setPropertyClause" ] ( { "addReferenceClause" } ) );

setPropertyClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("
"propertyValueList" )";
```

```

addReferenceClause = ADD REFERENCE TO "componentRefClause";
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
propertyValueList = "propertyValue" { "," "propertyValue" };
componentRefClause = ( EXTERNAL_TABLE | TABLE | VIEW | MATERIALIZED_
VIEW | SEQUENCE | MAPPING | DIMENSION_TABLE | CUBE_TABLE |
ADVANCED_QUEUE | ORACLE_MODULE | TRANSFORMATION_MODULE |
FLAT_FILE_MODULE | FLAT_FILE | PROCESS_FLOW | PROCESS_FLOW_
PACKAGE | PROCESS_FLOW_MODULE | SAP_MODULE | COLLECTION |
FUNCTION | PROCEDURE | PACKAGE | INTELLIGENCE_MODULE | REPORT_
MODULE | REPORT | REPORT_GROUP | QUERY_OBJECT | BUSINESS_AREA |
GATEWAY_MODULE | LOCATION | RUNTIME_REPOSITORY_CONNECTION |
CONNECTOR | IO_FUNCTION ) "QUOTED_STRING";
propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );

```

## Keywords and Parameters

**createCollectionCommand**

Create a collection of objects.

**setPropertyClause**

Set values for a number of properties when creating the collection.

Basic properties for COLLECTION:

**Name:** BUSINESS\_NAME

**Type:** STRING(200)

**Valid Values:** N/A

**Default:** "

Business name of the collection

**Name:** DESCRIPTION

**Type:** STRING(4000)

**Valid Values:** N/A

**Default:** "

Description of the collection

**Note:**

1. N/A means any valid character in supported character set.
2. " represents an empty string

**addReferenceClause**



Add a reference to the collection.

**propertyNameList**

Comma separated list of property names to retrieve values. Property names are unquoted.

**propertyValueList**

Comma separated list of property values.

**componentRefClause**

Specify the type of the object to reference.

**propertyValue**

Value of property.

## Examples

```
OMBCREATE COLLECTION 'PURCHASING_WAREHOUSE' ADD REFERENCE TO  
TABLE 'PURCHASING/ORDER' ADD REFERENCE TO TABLE  
'PURCHASING/CUSTOMER' OMBCREATE COLLECTION 'MY_FOLDERS' ADD  
REFERENCE TO ORACLE_MODULE 'PURCHASING' ADD REFERENCE TO  
ORACLE_MODULE 'HUMAN_RESOURCES'
```

## See Also

OMBCREATE, OMBALTER COLLECTION, OMBDROP COLLECTION

# OMBCREATE CONNECTOR

## Purpose

OMBCREATE CONNECTOR - To create a connector.

## Prerequisites

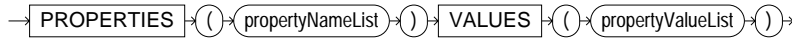
Should be in the context of a location.

## Syntax Diagrams

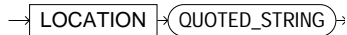
createConnectorCommand



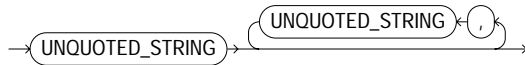
setPropertiesClause



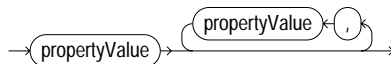
setReferenceToLocationClause



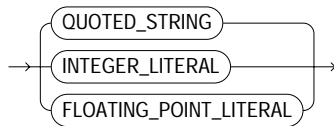
propertyNameList



propertyValueList



propertyValue



## Syntax

```
createConnectorCommand = OMBCREATE ( CONNECTOR "QUOTED_STRING" [
SET "setPropertiesClause" ] [ SET ( REFERENCE | REF )
"setReferenceToLocationClause" ] );
```

```
setPropertiesClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("
"propertyValueList" " )";
```

```
setReferenceToLocationClause = LOCATION "QUOTED_STRING";
```

```
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
propertyValueList = "propertyValue" { "," "propertyValue" };
propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );
```

## Keywords and Parameters

`createConnectorCommand`

Create a new connector.

`setPropertyClause`

Set specified properties of the connector.

`setReferenceToLocationClause`

Set the name of the location which the connector references.

`propertyNameList`

The names of the properties whose values you want to set.

Properties for CONNECTOR:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those objects marked as Deployable = true

Note:

1. N/A means any valid character in supported character set.
2. " represents an empty string

`propertyValueList`

The values for the named properties.

`propertyValue`

A property value.

## Examples

OMBCREATE CONNECTOR 'NEW\_CONNECTOR' SET PROPERTIES  
(DESCRIPTION, BUSINESS\_NAME) VALUES ('this is a connector', 'connector') This  
will create a connector named "NEW\_CONNECTOR", its description is "this is a  
connector", and business name is "connector".

OMBCREATE CONNECTOR 'NEW\_CONNECTOR\_2' SET REFERENCE LOCATION  
'MY\_LOCATION'

## See Also

OMBCREATE, OMBALTER CONNECTOR, OMBDROP CONNECTOR

# OMBCREATE CUBE\_TABLE

## Purpose

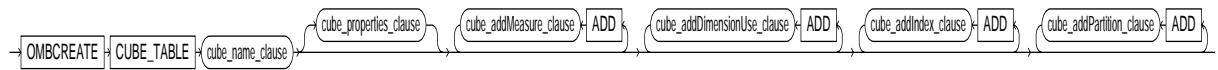
OMBCREATE CUBE\_TABLE - This command creates a cube.

## Prerequisites

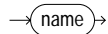
Should be in Oracle Module context.

## Syntax Diagrams

OMBCreateCube



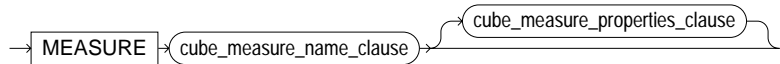
cube\_name\_clause



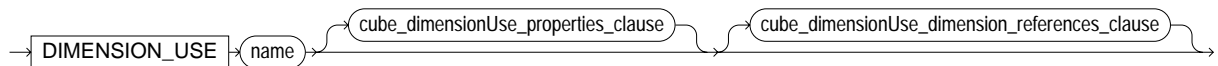
cube\_properties\_clause



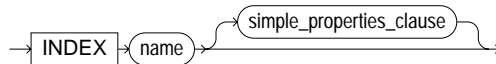
cube\_addMeasure\_clause



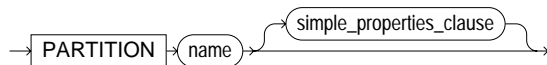
cube\_addDimensionUse\_clause



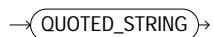
cube\_addIndex\_clause



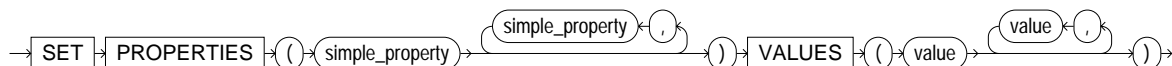
cube\_addPartition\_clause



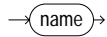
name



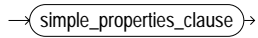
simple\_properties\_clause



cube\_measure\_name\_clause



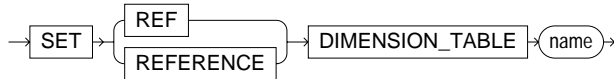
cube\_measure\_properties\_clause



cube\_dimensionUse\_properties\_clause



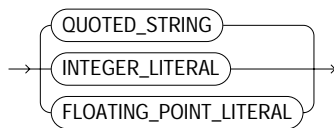
cube\_dimensionUse\_dimension\_references\_clause



simple\_property



value



## Syntax

```

OMBCreateCube = OMBCREATE CUBE_TABLE "cube_name_clause" [ "cube_
properties_clause" ] { ADD "cube_addMeasure_clause" } { ADD "cube_
addDimensionUse_clause" } { ADD "cube_addIndex_clause" } { ADD "cube_
addPartition_clause" };

cube_name_clause = "name";

cube_properties_clause = "simple_properties_clause";

cube_addMeasure_clause = MEASURE "cube_measure_name_clause" [ "cube_
measure_properties_clause" ];

cube_addDimensionUse_clause = DIMENSION_USE ( "name" [ "cube_
dimensionUse_properties_clause" ] [ "cube_dimensionUse_dimension_referenc
es_clause" ] );

cube_addIndex_clause = INDEX "name" [ "simple_properties_clause" ];

cube_addPartition_clause = PARTITION "name" [ "simple_properties_clause" ];

name = ( "QUOTED_STRING" );

simple_properties_clause = SET PROPERTIES "(" "simple_property" { "," "simple_
property" } ")" VALUES "(" "value" { "," "value" } ")";

cube_measure_name_clause = "name";

cube_measure_properties_clause = "simple_properties_clause";

cube_dimensionUse_properties_clause = "simple_properties_clause";

cube_dimensionUse_dimension_references_clause = SET ( REF | REFERENCE ) (
DIMENSION_TABLE "name" );
    
```

```
simple_property = "UNQUOTED_STRING";  
value = ("QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_POINT_  
LITERAL");
```

## Keywords and Parameters

OMBCreateCube

This command creates cube\_table.

cube\_name\_clause

The name of the cube\_table to be created.

cube\_properties\_clause

This clause uses the simple properties.

cube\_addMeasure\_clause

This clause adds the measure name and its properties.

cube\_addDimensionUse\_clause

This clause adds the dimensionUse and its properties.

cube\_addIndex\_clause

This clause adds Index name and its properties.

cube\_addPartition\_clause

This clause adds partition and its properties.

name

The name has to be a quoted string or an integer, or a decimal number.

simple\_properties\_clause

Sets properties and their values.

Basic properties for TABLE:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the table

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the table

Basic properties for INDEX, PARTITION, PARTITION\_KEY, INDEX\_COLUMN:

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the Index, Partition, PartitionKey, IndexColumn

Properties for CUBE\_TABLE:

Name: ANALYZE\_TABLE\_ESTIMATE\_PERCENT

Type: NUMBER

Valid Values: 0 - 100

Default: 99

Value represents the sample size as a percentage of total rows. When set to a nonzero value, Builder generates a DDL script to analyze the table.

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: HASH\_PARTITION\_TABLESPACE\_LIST

Type: STRING

Valid Values: N/A

Default: "

A comma separated list of tablespaces to use for [sub]partition storage.



Name: HASH\_SUBPARTITION\_NUMBER

Type: NUMBER

Valid Values: 2 - 63999

Default: 2

To create Hash partition, specify the number of Hash subpartition.

Name: LOGGING\_MODE

Type: STRING

Valid Values: NOLOGGING, LOGGING

Default: LOGGING

Recovery requirements for a data warehouse : Logging or not logging to Redo Log File.

Name: PARALLEL\_ACCESS\_MODE

Type: STRING

Valid Values: NOPARALLEL, PARALLEL

Default: PARALLEL

Enables or disables parallel processing when the table is created. Also enables or disables parallel processing or access.

Name: TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Use the Tablespace parameter to specify the name of tablespace.

Properties for INDEX:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: INDEX\_TYPE

Type: STRING

Valid Values: BITMAP, UNIQUE, NO\_INDEX

Default: UNIQUE

The types of Indexes created on Dimension are BITMAP, UNIQUE or a non-specific index.

Name: LOCAL\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

A local index is constructed so that it reflects the structure of the underlying table. It is equipartitioned with the underlying table, meaning that it is partitioned on the same columns as the underlying table, creates the same number of partitions or subpartitions, and gives them the same partition bounds as corresponding partitions of the underlying table.

Name: LOGGING\_MODE

Type: STRING

Valid Values: NOLOGGING, LOGGING

Default: LOGGING

Recovery requirements for a data warehouse : Logging or not logging to Redo Log File.

Name: PARALLEL\_ACCESS\_MODE

Type: STRING

Valid Values: NOPARALLEL, PARALLEL

Default: PARALLEL

Enables or disables parallel processing when the table is created. Also enables or disables parallel processing or access.

Name: TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Use the Tablespace parameter to specify the name of tablespace.

Name: USING\_INDEX  
Type: BOOLEAN  
Valid Values: true, false  
Default: false  
Use of Index on Dimension Constraint

Properties for PARTITION:

Name: DATE\_LESS\_THAN  
Type: STRING  
Valid Values: N/A  
Default: TO\_DATE('01-JAN-1999', 'DD-MON-YYYY')  
Value that represents upper bound of partition stored in warehouse key column for the Days Dimension.

Name: DEPLOYABLE  
Type: BOOLEAN  
Valid Values: true, false  
Default: true  
Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: TABLESPACE  
Type: STRING(30)  
Valid Values: N/A  
Default: EMPTY\_STRING  
Use the Tablespace parameter to specify the name of tablespace.

Name: VALUES\_LESS\_THAN  
Type: STRING  
Valid Values: N/A  
Default: TO\_DATE('01-JAN-1999', 'DD-MON-YYYY')  
Noninclusive upper bound for the current partition. Type a comma separated list in the same order as the columns in range partitioned key.

Name: VALUE\_LESS\_THAN  
Type: STRING

Valid Values: N/A

Default: "

Noninclusive upper bound for the current partition. Type a comma separated list in the same order as the columns in range partitioned key.

Note:

1. N/A means any valid character in supported character set.
2. " represents an empty string

cube\_measure\_name\_clause

name of the measure.

cube\_measure\_properties\_clause

alters simple properties of a measure.

cube\_dimensionUse\_properties\_clause

alters simple properties of a dimensionUse.

cube\_dimensionUse\_dimension\_references\_clause

The dimensionUse references to the first level of the dimension.

simple\_property

gets the simple property.

UNQUOTED\_STRING

Name of the simple property.

value

The quoted string name

## Examples

```
OMBCREATE CUBE_TABLE 'CUBE1' ADD MEASURE 'MEASURE_1' ADD
DIMENSION_USE 'CUBE1_DIM1' SET REF DIMENSION_TABLE 'DIM1'
```

## See Also

OMBALTER CUBE\_TABLE, OMBDROP CUBE\_TABLE, OMBRETRIEVE CUBE\_TABLE

# OMBCREATE DEPLOYMENT\_ACTION\_PLAN

## Purpose

OMBCREATE DEPLOYMENT\_ACTION\_PLAN - Create a deployment action plan.

## Prerequisites

There must be a current working project.

## Syntax Diagrams

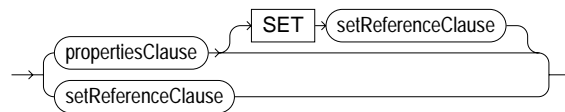
CreateActionPlanCommand



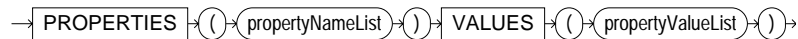
addActionClause



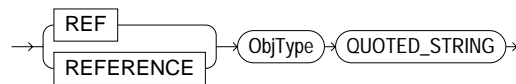
setClause



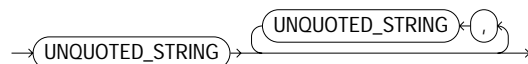
propertiesClause



setReferenceClause



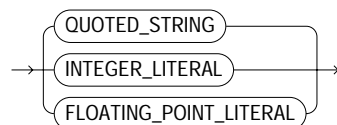
propertyNameList



propertyValueList



propertyValue



## Syntax

```

CreateActionPlanCommand = ( OMBCREATE TRANSIENT ( DEPLOYMENT_
ACTION_PLAN ) "QUOTED_STRING" { "addActionClause" } );
addActionClause = ADD ACTION "QUOTED_STRING" [ SET "setClause" ];
setClause = ( "propertiesClause" [ SET "setReferenceClause" ] ) | "setReferenceClause";
propertiesClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("
"propertyValueList" ")";
setReferenceClause = ( REF | REFERENCE ) "ObjType" "QUOTED_STRING";
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
propertyValueList = "propertyValue" { "," "propertyValue" };
propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );

```

## Keywords and Parameters

**CreateActionPlanCommand**

Create a deployment action plan.

**QUOTED\_STRING**

Name of the action plan in a single-quoted string. It is case-insensitive.

Must be unique within a project.

**addActionClause**

Add an action to an action plan.

**QUOTED\_STRING**

Name of the action in a single-quoted string. It is case-insensitive. Must be unique within an action plan.

**setClause**

Set the properties of an action and/or associate an object with an action.

**propertiesClause**

Associate a set of properties with an action.

**PROPERTIES**

The only valid property is OPERATION, which specifies the type of action to be taken.

**setReferenceClause**

Associate an object with an action.

ObjType

Object type. Valid values are ADVANCED\_QUEUE, CUBE\_TABLE, DIMENSION\_TABLE,

EXTERNAL\_TABLE, CONNECTOR, FUNCTION, MAPPING, MATERIALIZED\_VIEW, PROCEDURE,

PROCESS\_FLOW\_PACKAGE, SEQUENCE, TABLE, and VIEW.

QUOTED\_STRING

Absolute or relative path name of an object (e.g. '/MY\_PROJECT/MODULE\_X/TABLE\_Y').

propertyNameList

Comma separated list of property names. Property names are unquoted.

propertyValueList

Comma separated list of property values.

propertyValue

Value of a property. Valid values for OPERATION are DROP and CREATE.

## Examples

```
OMBCREATE TRANSIENT DEPLOYMENT_ACTION_PLAN 'MY_PLAN'  
OMBCREATE TRANSIENT DEPLOYMENT_ACTION_PLAN 'MY_PLAN' ADD  
ACTION 'DUMMY'  
OMBCREATE TRANSIENT DEPLOYMENT_ACTION_PLAN 'MY_PLAN'  
ADD ACTION 'MY_TABLE_DEPLOY' SET PROPERTIES (OPERATION) VALUES  
( 'DROP' )  
SET REFERENCE TABLE '/MY_PROJECT/MY_MODULE/MY_TABLE'  
ADD ACTION 'MY_VIEW_CREATE' SET PROPERTIES (OPERATION) VALUES  
( 'CREATE' )  
SET REFERENCE VIEW 'MY_MODULE/MY_VIEW'
```

## See Also

OMBDEPLOY

# OMBCREATE EXTERNAL\_TABLE

## Purpose

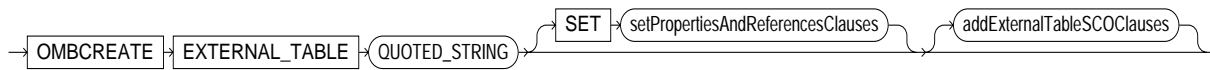
OMBCREATE EXTERNAL\_TABLE - To create an external table.

## Prerequisites

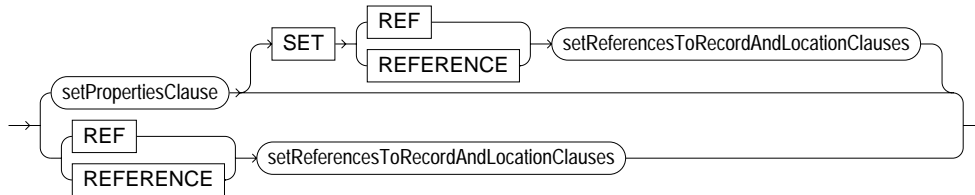
Should be in the context of an Oracle Module.

## Syntax Diagrams

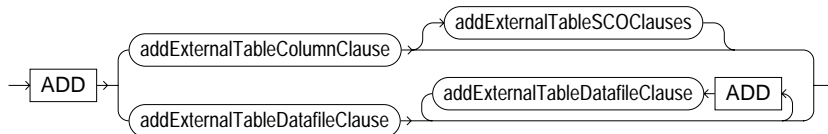
createExternalTableCommand



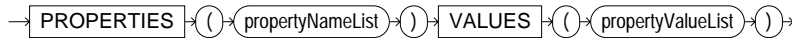
setPropertiesAndReferencesClauses



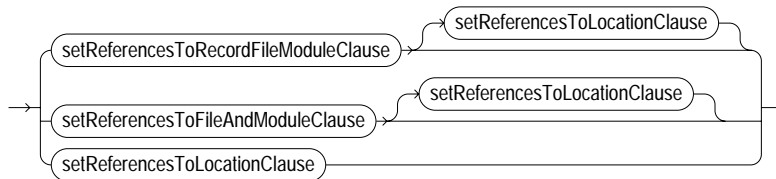
addExternalTableSCOClauses



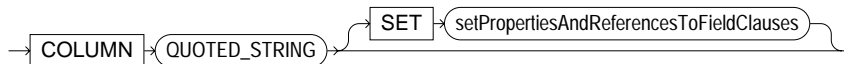
setPropertiesClause



setReferencesToRecordAndLocationClauses



addExternalTableColumnClause

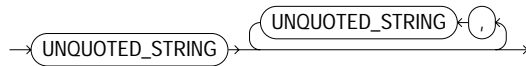




addExternalTableDatafileClause



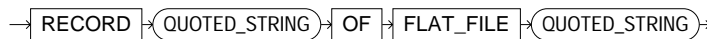
propertyNameList



propertyValueList



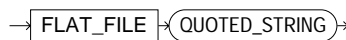
setReferencesToRecordFileModuleClause



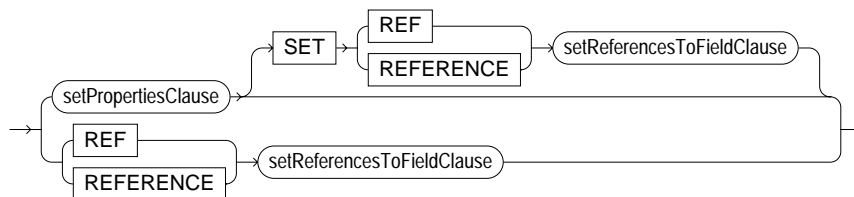
setReferencesToLocationClause



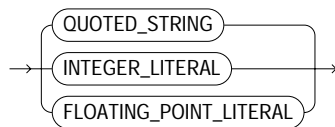
setReferencesToFileAndModuleClause



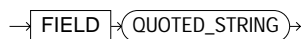
setPropertiesAndReferencesToFieldClauses



propertyValue



setReferencesToFieldClause



## Syntax

```

createExternalTableCommand = OMBCREATE ( EXTERNAL_TABLE "QUOTED_
STRING" [ SET "setPropertiesAndReferencesClauses" ] [
"addExternalTableSCOClauses" ] );
  
```

```

setPropertiesAndReferencesClauses = "setPropertiesClause" [ SET ( REF |
REFERENCE ) "setReferencesToRecordAndLocationClauses" ] | ( REF | REFERENCE
) "setReferencesToRecordAndLocationClauses";
  
```

```

addExternalTableSCOClauses = ADD ( "addExternalTableColumnClause" [
"addExternalTableSCOClauses" ] | "addExternalTableDatafileClause" { ADD
"addExternalTableDatafileClause" } );
  
```

```

setPropertyClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("
"propertyValueList" ");
setReferencesToRecordAndLocationClauses =
"setReferencesToRecordFileModuleClause" [ "setReferencesToLocationClause" ] |
"setReferencesToFileAndModuleClause" [ "setReferencesToLocationClause" ] |
"setReferencesToLocationClause";
addExternalTableColumnClause = COLUMN "QUOTED_STRING" [ SET
"setPropertiesAndReferencesToFieldClauses" ];
addExternalTableDatafileClause = DATA_FILE "QUOTED_STRING" [ SET
"setPropertyClause" ];
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
propertyValueList = "propertyValue" { "," "propertyValue" };
setReferencesToRecordFileModuleClause = RECORD "QUOTED_STRING" OF FLAT_
FILE "QUOTED_STRING";
setReferencesToLocationClause = DEFAULT_LOCATION "QUOTED_STRING";
setReferencesToFileAndModuleClause = FLAT_FILE "QUOTED_STRING";
setPropertiesAndReferencesToFieldClauses = "setPropertyClause" [ SET ( REF |
REFERENCE ) "setReferencesToFieldClause" ] | ( REF | REFERENCE )
"setReferencesToFieldClause";
propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );
setReferencesToFieldClause = FIELD "QUOTED_STRING";

```

## Keywords and Parameters

**createExternalTableCommand**

Create a new external table.

**QUOTED\_STRING**

The name of the new external table.

**setPropertyAndReferencesClauses**

Set the properties and/or flat file reference of the external table.

**addExternalTableSCOClauses**

Add columns and/or data files to the external table.

**setPropertyClause**

Set specified properties of the external table.

**setReferencesToRecordAndLocationClauses**

Set the referenced record and/or default location.

**addExternalTableColumnClause**

Add an external table column.

**addExternalTableDatafileClause**

Add a data file to the external table.

**propertyNameList**

The names of the properties whose values you want to set.

**Properties for EXTERNAL\_TABLE:**

Name: BAD\_FILE\_LOCATION

Type: STRING

Valid Values: N/A

Default: USE\_DEFAULT\_LOCATION

Location to use when creating the bad file. When the Access Parameters property is specified for the External Table, this configuration parameter is ignored.

Name: BAD\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: "

Name to use when creating the bad file. When the Access Parameters property is specified for the External Table, this configuration parameter is ignored.

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

When deployable is set to true, a script to create an External Table is generated.

Name: DISCARD\_FILE\_LOCATION

Type: STRING

Valid Values: N/A

Default: USE\_DEFAULT\_LOCATION

Location to use when creating the discard file. When the Access Parameters property is specified for the External Table, this configuration parameter is ignored.

Name: DISCARD\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: "

Name to use when creating the discard file. When the Access Parameters property is specified for the External Table, this configuration parameter is ignored.

Name: ENDIAN

Type: STRING

Valid Values: BIG, LITTLE, PLATFORM

Default: PLATFORM

Data endian should be platform default, little or big. When the Access Parameters property is specified for the External Table, this configuration parameter is ignored.

Name: LOAD\_NULLS\_WHEN\_MISSING\_VALUES

Type: BOOLEAN

Valid Values: true, false

Default: false

If TRUE, then NULLs are loaded for any missing values in the record. If FALSE, then records with missing values are rejected. When the Access Parameters property is specified for the External Table, this configuration parameter is ignored.

Name: LOG\_FILE\_LOCATION

Type: STRING

Valid Values: N/A

Default: USE\_DEFAULT\_LOCATION

Location to use when creating the log file. When the Access Parameters property is specified for the External Table, this configuration parameter is ignored.

Name: LOG\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: "

Name to use when creating the log file. When the Access Parameters property is specified for the External Table, this configuration parameter is ignored.

Name: NUMBER\_OF\_REJECTS\_ALLOWED

Type: NUMBER

Valid Values: 0 - 2147483647

Default: 0

The number of rejects allowed before processing is terminated.

Name: PARALLEL\_ACCESS\_DRIVERS

Type: NUMBER

Valid Values: 1 - 63999

Default: 1

The number of parallel access drivers to enable.

Name: PARALLEL\_ACCESS\_MODE

Type: BOOLEAN

Valid Values: true, false

Default: false

Enable or disable parallel processing.

Name: REJECTS\_ARE\_UNLIMITED

Type: BOOLEAN

Valid Values: true, false

Default: true

Enable or disable limiting the number of rejected records.

Name: STRING\_SIZES\_IN

Type: STRING

Valid Values: CHARACTERS, BYTES

Default: BYTES

String sizes are in bytes or characters. When the Access Parameters property is specified for the External Table, this configuration parameter is ignored.

Name: TRIM

Type: STRING

Valid Values: LEFT, NONE, SQL\*LOADER, BOTH, RIGHT

Default: NONE

Specification from trim option on input fields. When the Access Parameters property is specified for the External Table, this configuration parameter is ignored.

Properties for DATA\_FILE:

Name: DATA\_FILE\_LOCATION

Type: STRING

Valid Values: N/A

Default: USE\_DEFAULT\_LOCATION

The location of this data file for the external table.

Name: DATA\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: "

Name of this data file.

Note:

1. N/A means any valid character in supported character set.
2. " represents an empty string

propertyValueList

The values for the named properties.

setReferencesToRecordFileModuleClause

Specify the record and full path to the flat file for the external table to reference.

setReferencesToLocationClause

The name of the external table's default location.

setReferencesToFileAndModuleClause

Specify the full path to the flat file for the external table to reference.

setPropertyAndReferencesToFieldClauses

Set the properties and/or field reference of the external table column.

propertyValue

A property value.

setReferencesToFieldClause

Set the name of the field which the external table column references.

**Examples**

OMBCREATE EXTERNAL\_TABLE 'SRC\_TABLE' SET REFERENCE RECORD 'REC\_1'  
OF FLAT\_FILE '../SRC\_FILES/FILE\_1' This will create an external table named "SRC\_  
TABLE" based upon the record "REC\_1" of flat file "FILE\_1" of flat file module "SRC\_  
FILES". ombcreate\_external\_table\$createExternalTableCommand =

**See Also**

OMBCREATE, OMBALTER EXTERNAL\_TABLE, OMBDROP EXTERNAL\_TABLE

# OMBCREATE FLAT\_FILE

## Purpose

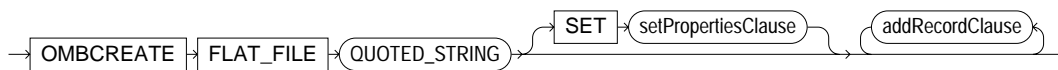
OMBCREATE FLAT\_FILE - To create a flat file.

## Prerequisites

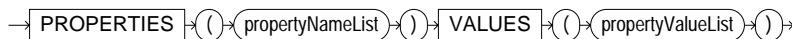
Should be in the context of a flat file module.

## Syntax Diagrams

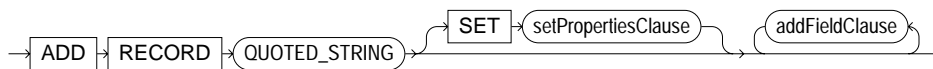
createFlatFileCommand



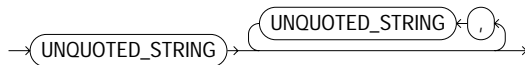
setPropertiesClause



addRecordClause



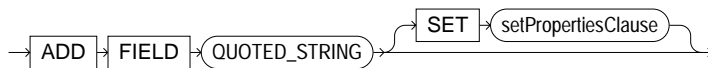
propertyNameList



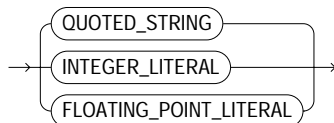
propertyValueList



addFieldClause



propertyValue



## Syntax

```

createFlatFileCommand = OMBCREATE ( FLAT_FILE "QUOTED_STRING" [ SET
"setPropertiesClause" ] { "addRecordClause" } );
  
```



```

setPropertyClause = PROPERTIES "(" propertyNameList ")" VALUES "("
"propertyValueList ")";
addRecordClause = ADD ( RECORD "QUOTED_STRING" [ SET
"setPropertyClause" ] { "addFieldClause" } );
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
propertyValueList = "propertyValue" { "," "propertyValue" };
addFieldClause = ADD FIELD "QUOTED_STRING" [ SET "setPropertyClause" ];
propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );

```

## Keywords and Parameters

**createFlatFileCommand**

Create a new flat file object.

**QUOTED\_STRING**

The name of the new flat file.

**setPropertyClause**

Set the properties of the flat file, record, or field.

**addRecordClause**

Add a record to the flat file.

**QUOTED\_STRING**

The name of the new record.

**propertyNameList**

The names of the properties whose values you want to set.

Properties for FLAT\_FILE:

Name: DATA\_FILE\_NAME

Type: STRING

Valid Values: N/A Default " The name of the "sampled" file. Also the default data file value used in SQL\*Loader maps and External Tables.

Name: IS\_DELIMITED

Type: BOOLEAN

Valid Values: true, false, 1, 0

Default: true

True indicates that this flat file is delimited. False indicates that its fields are defined by fixed lengths

Name: CHARACTERSET

Type: STRING

Valid Values:

AL24UTFFSS,AR8ARABICMAC,AR8ARABICMACS,AR8ISO8859P6,AR8MSAWIN,A  
R8MSWIN1256,BLT8CP921,BLT8EBCDIC1112,BLT8MSWIN1257,BLT8PC775,CDN8PC  
863,CL8EBCDIC1025,CL8EBCDIC1025X,CL8ISO8859P5,CL8KOI8R,CL8MACCYRILLI  
C,CL8MACCYRILLICS,CL8MSWIN1251,D8EBCDIC273,DK8EBCDIC277,EE8EBCDIC  
870,EE8ISO8859P2,EE8MACCE,EE8MACCES,EE8MACCROATIAN,EE8MACCROATI  
ANS,EE8MSWIN1250,EE8PC852,EL8EBCDIC875,EL8ISO8859P7,EL8MACGREEK,EL8  
MACGREEKS,EL8MSWIN1253,EL8PC437S,EL8PC737,EL8PC869,F8EBCDIC297,I8EBC  
DIC280,IS8MACICELANDIC,IS8MACICELANDICS,IS8PC861,IW8EBCDIC424,IW8IS  
O8859P8,IW8MACHEBREW,IW8MACHEBREWS,IW8MSWIN1255,JA16EBCDIC930,J  
A16EUC,JA16EUCYEN,JA16MACSJIS,JA16SJIS,JA16SJISYEN,JA16VMS,KO16KSC560  
1,LT8MSWIN921,N8PC865,NEE8ISO8859P4,RU8PC855,RU8PC866,S8EBCDIC278,SE8I  
SO8859P3,TH8MACTHAI,TH8MACTHAIS,TH8TISASCII,TR8EBCDIC1026,TR8MAC  
TURKISH,TR8MACTURKISHS,TR8MSWIN1254,TR8PC857,US7ASCII,US8PC437,UTF  
8,WE8EBCDIC284,WE8EBCDIC285,WE8EBCDIC37,WE8EBCDIC37C,WE8EBCDIC500  
,WE8EBCDIC500C,WE8EBCDIC871,WE8ISO8859P1,WE8ISO8859P9,WE8MACROMA  
N8,WE8MACROMAN8S,WE8MSWIN1252,WE8PC850,WE8PC860,ZHS16CGB231280,  
ZHS16GBK,ZHS16MACCGB231280,ZHT16BIG5,ZHT16MSWIN950,ZHT32EUC

Default: WE8MSWIN1252

The character set of the data file.

Name: RECORD\_DELIMITER

Type: STRING

Valid Values: N/A

Default: "

The character(s) which denote the end of a physical record in a data file. (Please note that this is not the FIELD\_DELIMITER.)

Name: RECORD\_LENGTH

Type: NUMBER

Valid Values: 0+

Default: 0 (Records are delimited by default)

The length (in characters) of the records in the data file.

Name: RECORD\_TYPE\_COLUMN\_NUMBER

Type: NUMBER

Valid Values: 0+

Default: 1

The column which contains the record type values for a delimited, multi-record type file.

Name: RECORD\_TYPE\_START\_POSITION

Type: NUMBER

Valid Values: 0+

Default: 0

The starting position of the field (relative to 1) which contains the record type values for a fixed-length, multi-record type file.

Name: RECORD\_TYPE\_END\_POSITION

Type: NUMBER

Valid Values: 0+

Default: 0

The ending position of the field (relative to 1) which contains the record type values for a fixed-length, multi-record type file.

Name: NUMBER\_OF\_RECORDS\_TO\_SKIP

Type: NUMBER

Valid Values: 0 - 1000

Default: 0

The default number of records to skip when loading this file.

Name: FIELD\_DELIMITER

Type: STRING

Valid Values: Any single character

Default: ',' (Comma)

The character to divide the fields in a delimited file.

Name: FIELD\_LEFT\_ENCLOSURE

Type: STRING

Valid Values: Any single character

Default: None

A character to enclose fields which may contain the field delimiter.

Name: FIELD\_RIGHT\_ENCLOSURE

Type: STRING

Valid Values: Any single character

Default: None

A character to enclose fields which may contain the field delimiter.

Name: NUMBER\_OF\_PHYSICAL\_RECORDS\_PER\_LOGICAL

Type: Number

Valid Values: 0+

Default: 0

Set this value if you wish to concatenate a fixed number of physical records to form a single logical record.

Name: CONTINUE\_IF\_ENDS\_WITH

Type: STRING

Valid Values: Any single character

Default: None

Set this value if you wish to concatenate a variable number of physical records to form a single logical record, determined by records ending with this character.

Name: CONTINUE\_IF\_STARTS\_WITH

Type: STRING

Valid Values: N/A

Default: None

Set this value if you wish to concatenate a variable number of physical records to form a single logical record, determined by records beginning with this character.

Properties for RECORD:

Name: RECORD\_TYPE\_VALUE

Type: STRING

Valid Values: N/A

Default: None

This is a mandatory property for each record of a multi-record type file. It is the string which will identify this record type in the data file.

Properties for FIELD:

Name: DATATYPE

Type: STRING

---

Valid Values: CHAR, DATE, DECIMAL EXTERNAL, FLOAT EXTERNAL, INTEGER EXTERNAL, ZONED EXTERNAL, ZONED

Default: CHAR

This is the SQL\*Loader data type for the field.

Name: LENGTH

Type: NUMBER

Valid Values: 1+

Default: 1

This is the length of the field in a fixed length file. This is the max length of the field in a delimited file.

Name: PRECISION

Type: NUMBER

Valid Values: Depends on data type

Default: 0

Precision of the field.

Name SCALE

Type: NUMBER

Valid Values: Depends on data type

Default: 0

Scale of the field

Name: START\_POSITION

Type: NUMBER

Valid Values: 0+

Default: 0

The starting position of a field for a fixed length file.

Name: SQL\_DATATYPE

Type: STRING

Valid Values: CHAR, DATE, FLOAT, NUMBER, VARCHAR, VARCHAR2, DEFAULT

Default: DEFAULT. This will derive the SQL\_DATATYPE from the value of

DATATYPE. The data type which the field will be treated as in mapping and for External Tables.

Name: SQL\_LENGTH

Type: NUMBER

Valid Values: for 'CHAR' : 1 - 2000 for 'VARCHAR' and 'VARCHAR2' : 1 - 4000

Default: Depends on Sql data type.

Name: SQL\_PRECISION

Type: NUMBER

Valid Values: 1 - 38

Default: 1

Name: SQL\_SCALE

Type: NUMBER

Valid Values: -84 - 127

Default: 0

Name: MASK

Type: STRING

Valid Values: N/A

Default: None

This is the mask used to define the format of DATE fields in the data file.

Name: NULL\_IF

Type: STRING

Valid Values: Either = or != followed by either 'BLANKS', a single quoted string, or a hexadecimal string

Default: None

If this condition is true for a field, the value loaded will be NULL.

Name: DEFAULT\_IF

Type: STRING

Valid Values: Either = or != followed by either 'BLANKS', a single quoted string, or a hexadecimal string

Default: None

If this condition is true for a field, the value loaded will be either NULL or 0, dependent on data type.

propertyValueList

The values for the named properties.

addFieldClause

Add a field to the record.

QUOTED\_STRING

The name of the new field.

propertyValue

A property value.

## Examples

```
OMBCREATE FLAT_FILE 'SRC_FILE' \
  SET PROPERTIES (DATA_FILE_NAME, FIELD_DELIMITER, FIELD_LEFT_
ENCLOSURE, FIELD_RIGHT_ENCLOSURE, RECORD_TYPE_COLUMN_NUMBER)
\ VALUES ('src_data.dat', '|', '{', '}', 1) \ ADD RECORD 'REC1' \ SET PROPERTIES
(RECORD_TYPE_VALUE) VALUES ('E') \ ADD FIELD 'F1' \ SET PROPERTIES
(DATATYPE, MASK) VALUES ('DATE', 'dd-mon-yyyy') \ ADD FIELD 'F2' \ ADD
RECORD 'REC2' \ SET PROPERTIES (RECORD_TYPE_VALUE) VALUES ('P') \
ADD FIELD 'F1' \ ADD FIELD 'F2' \ ADD FIELD 'F3' This will create a flat file
named "SRC_FILE" with multiple record types.
```

## See Also

OMBCREATE, OMBALTER FLAT\_FILE, OMBDROP FLAT\_FILE

# OMBCREATE FLAT\_FILE\_MODULE

## Purpose

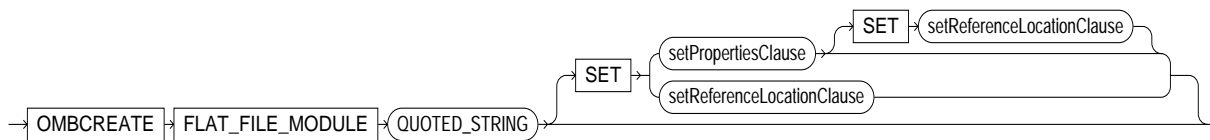
OMBCREATE FLAT\_FILE\_MODULE - To create a flat file module.

## Prerequisites

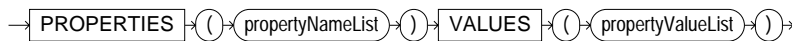
Should be in the context of project.

## Syntax Diagrams

createFlatFileModuleCommand



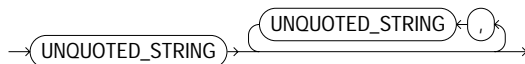
setPropertiesClause



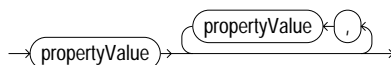
setReferenceLocationClause



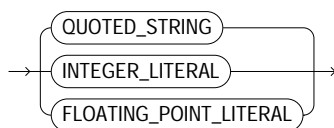
propertyNameList



propertyValueList



propertyValue



## Syntax

```
createFlatFileModuleCommand = OMBCREATE ( FLAT_FILE_MODULE "QUOTED_
STRING" [ SET ( "setPropertiesClause" [ SET "setReferenceLocationClause" ] |
"setReferenceLocationClause" ) ] );
```

```
setPropertiesClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("
"propertyValueList" ")";
```



```
setReferenceLocationClause = ( REFERENCE | REF ) LOCATION "QUOTED_
STRING";
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
propertyValueList = "propertyValue" { "," "propertyValue" };
propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );
```

## Keywords and Parameters

createFlatFileModuleCommand

Create a new flat file module.

QUOTED\_STRING

The name for the new flat file module.

setPropertyClause

Set specified properties of the new flat file module.

setReferenceLocationClause

Set the location for the new flat file module to the location specified by the quoted string.

propertyNameList

The names of the properties whose values you want to set.

Basic properties for FLAT\_FILE\_MODULE:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: NAME

Business name of the flat file module.

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the flat file module.

Name: UOID

Type: STRING(40)

Valid Values: N/A

Default: N/A

UUID of the flat file module.

propertyValueList

The values for the named properties.

propertyValue

A property value.

## Examples

OMBCREATE FLAT\_FILE\_MODULE 'src\_module' SET PROPERTIES (DESCRIPTION, BUSINESS\_NAME) VALUES ('this is a flat file module', 'source module') This will create a flat file module named "src\_module", its description is "this is a flat file module", and business name is "source module".

## See Also

OMBCREATE, OMBALTER FLAT\_FILE\_MODULE, OMBDROP FLAT\_FILE\_MODULE

# OMBCREATE FUNCTION

## Purpose

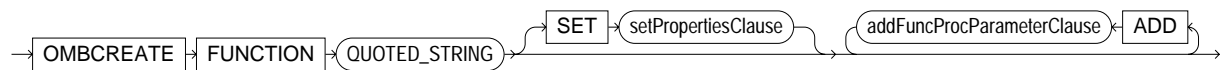
OMBCREATE FUNCTION - To create a Function.

## Prerequisites

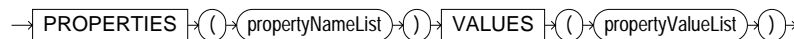
Should be in the context of a Oracle Module or Package or Transformation Module.

## Syntax Diagrams

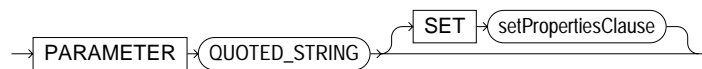
createFunctionCommand



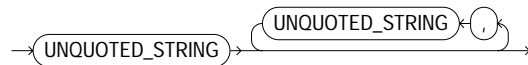
setPropertiesClause



addFuncProcParameterClause



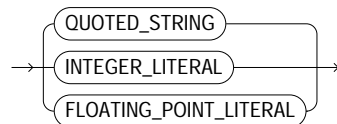
propertyNameList



propertyValueList



propertyValue



## Syntax

```
createFunctionCommand = OMBCREATE ( FUNCTION "QUOTED_STRING" [ SET
"setPropertiesClause" ] { ADD "addFuncProcParameterClause" } );
```

```
setPropertiesClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("
"propertyValueList" )";
```

```
addFuncProcParameterClause = PARAMETER "QUOTED_STRING" [ SET
"setPropertiesClause" ];
```

```
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
```

```
propertyValueList = "propertyValue" { "," "propertyValue" };
propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );
```

## Keywords and Parameters

**createCommand**

This command creates a Function

**QUOTED\_STRING**

Name of the Function to be created.

**setPropertyClause**

Used to set properties (core, user-defined) for function. Valid properties are shown below:

Basic properties for FUNCTION:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the Function

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the Function

Name: RETURN\_TYPE

Type: STRING

Valid Values: PLS\_INTEGER, BINARY\_INTEGER, BOOLEAN, NUMBER, FLOAT, CHAR, VARCHAR, VARCHAR2, DATE

Default: NUMBER

Set the Return Type for Function

Name: IMPLEMENTATION

Type: STRING

Valid Values: N/A

Default: "

Set the code for Function which is included global variable declaration and code between BEGIN and END.

Name: IS\_DETERMINISTIC

Type: BOOLEAN

Valid Values: true, false

Default: false

This setting helps the optimizer avoid redundant function calls.

Name: IS\_PARALLEL\_ENABLE

Type: BOOLEAN

Valid Values: true, false

Default: false

This option sets flag to a stored function can be used safely in the slave sessions of parallel DML evaluations.

Basic properties for PARAMETER:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the Parameter

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the Parameter

Name: DATATYPE

Type: STRING

Valid Values: PLS\_INTEGER, BINARY\_INTEGER, BOOLEAN, NUMBER, FLOAT, CHAR, VARCHAR, VARCHAR2, DATE

Default: NUMBER

Set the data type for Parameter

Name: IN\_OUT

Type: STRING

Valid Values: IN, OUT, INOUT

Default: 'IN'

Set the parameter mode for Parameter

Name: DEFAULT\_VALUE

Type: STRING

Valid Values: N/A

Default: "

Set the default value for Parameter

Properties for FUNCTION:

Name: AUTHID

Type: STRING

Valid Values: None, Current\_User, Definer

Default: None

Generate the transformation with selected AUTHID option. Function will be executed with the permissions defined by the AUTHID clause rather than the function owner's permissions.

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Note:

1. N/A means any valid character in supported character set.
2. " represents an empty string

addFuncProcParameterClause

Adds one or more Parameters to this Function.

propertyNameList

Comma separated list of property names. Property names are unquoted.

propertyValueList

Comma separated list of property values.

propertyValue

Value of a property.

## Examples

```
OMBCREATE FUNCTION 'func' SET PROPERTIES (DESCRIPTION, BUSINESS_
NAME, RETURN_TYPE, IMPLEMENTATION, IS_DETERMINISTIC, IS_PARALLEL_
ENABLE) VALUES ('this is a Function', 'function', 'NUMBER', 'BEGIN RETURN 1
END func \;', 'true', 'true') ADD PARAMETER 'PARAM_1' SET PROPERTIES
(DESCRIPTION, BUSINESS_NAME, IN_OUT, DATATYPE, DEFAULT_VALUE)
VALUES ('param_1', 'this is a param_1', 'IN', 'VARCHAR2', 'this is a Varchar2') ADD
PARAMETER 'PARAM_2' SET PROPERTIES (DESCRIPTION, BUSINESS_NAME, IN_
OUT, DATATYPE, DEFAULT_VALUE) VALUES ('param_2', 'this is a param_
2', 'INOUT', 'DATE', 'this is a Date') This will create a Function named "func", its
description is "this is a Function", and business name is "function", return datatype
NUMBER, and body of function as 'BEGIN RETURN 1 END func;'. It creates two
parameters 'PARAM_1' and 'PARAM_2'
```

## See Also

OMBCREATE, OMBALTER FUNCTION, OMBDROP FUNCTION

# OMBCREATE LOCATION

## Purpose

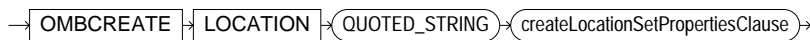
OMBCREATE LOCATION - To create a location.

## Prerequisites

Should be in the context of a project.

## Syntax Diagrams

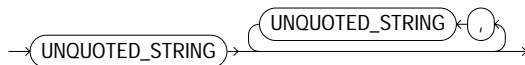
createLocationCommand



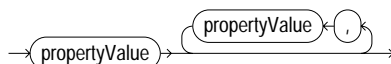
createLocationSetPropertiesClause



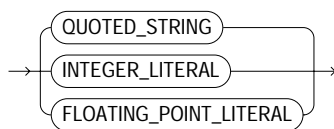
propertyNameList



propertyValueList



propertyValue



## Syntax

```

createLocationCommand = OMBCREATE ( LOCATION "QUOTED_STRING"
"createLocationSetPropertiesClause" );

createLocationSetPropertiesClause = SET PROPERTIES "(" "propertyNameList" ")"
VALUES "(" "propertyValueList" ")";

propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };

propertyValueList = "propertyValue" { "," "propertyValue" };

propertyValue = ("QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );
  
```

## Keywords and Parameters

createLocationCommand



---

Create a new location.

createLocationSetPropertiesClause

Set specified properties of the new location.

propertyNameList

The names of the properties whose values you want to set.

Properties for LOCATION:

Name: TYPE

Type: STRING

Valid Values: 'Oracle Gateway', 'File System', 'Oracle Database', 'OEM Agent', 'Oracle Workflow', 'SAP'

Default: N/A

The type of system the location represents.

Name: VERSION

Type: STRING

Valid Values:

for 'Oracle Gateway' : '0'

for 'File System' : '0'

for 'Oracle Database' : '9.2','9.0','8.1','8.0','7.3.4'

for 'OEM Agent' : '9.2','9.0'

for 'Oracle Workflow' : '2.6'

for 'SAP' : '4.x','3.x'

Default: N/A

The version of the system(s) the location represents.

Basic properties for LOCATION:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the location.

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the location.

Note:

1. N/A means any valid character in supported character set.

2. " represents an empty string

propertyValueList

The values for the named properties.

propertyValue

A property value.

## Examples

OMBCREATE LOCATION 'MY\_LOCATION' SET PROPERTIES (TYPE, VERSION, DESCRIPTION, BUSINESS\_NAME) VALUES ('Oracle Database', '9.2', 'this is a location', 'location') This will create a location named "MY\_LOCATION". Its type is "Oracle Database", version is "9.2", description is "this is a location", and business name is "location".

## See Also

OMBCREATE, OMBALTER LOCATION, OMBDROP LOCATION

# OMBCREATE MAPPING

## Purpose

OMBCREATE MAPPING - Create a mapping in an Oracle Module.

## Prerequisites

1. The current context of scripting must be an Oracle Module.
2. No concurrent user should be locking the Oracle Module or any of its ancestors exclusively at the moment the map is being created.
3. The map name must not conflict with existing map names and the maps names that concurrent user tries to use.

## Syntax Diagrams

createMappingCommand



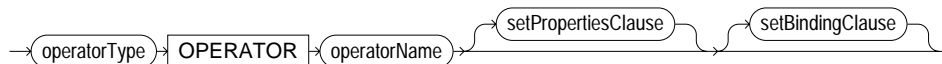
mappingName



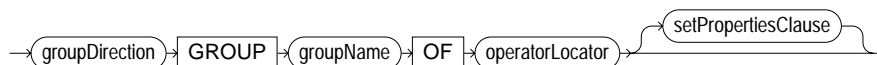
setPropertiesClause



addOperatorClause



addGroupClause



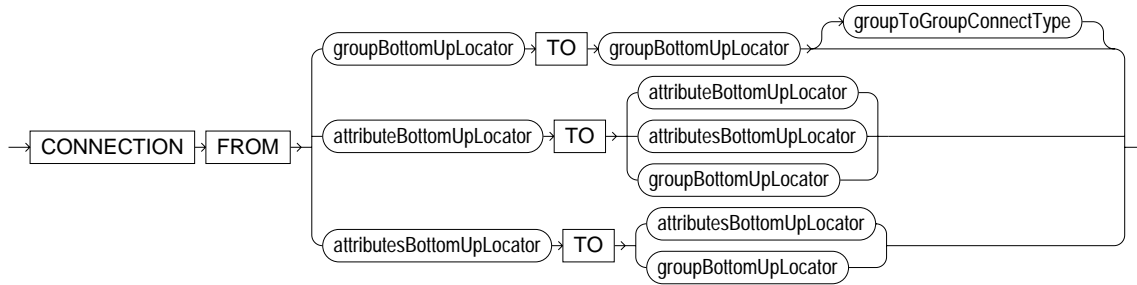
addAttributeClause



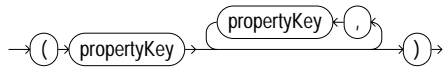
addChildClause



addConnectionClause



propertyKeyList



propertyValueList



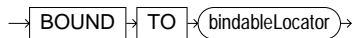
operatorType



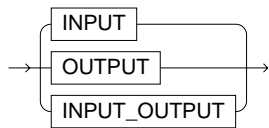
operatorName



setBindingClause



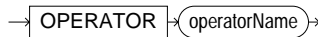
groupDirection



groupName



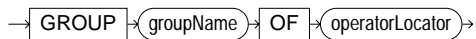
operatorLocator



attributeName



groupBottomUpLocator



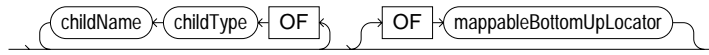
childType



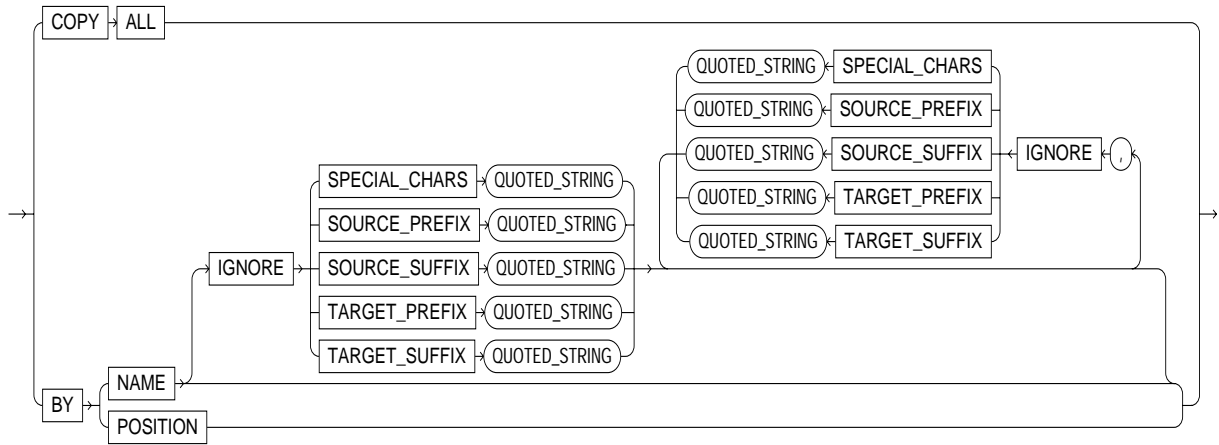
childName



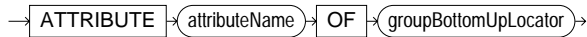
childOwnerBottomUpLocator



groupToGroupConnectType



attributeBottomUpLocator



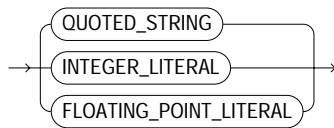
attributesBottomUpLocator



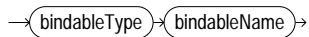
propertyKey



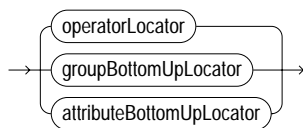
propertyValue



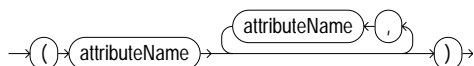
bindableLocator



mappableBottomUpLocator



attributeNameList



bindableType

→ UNQUOTED\_STRING →

bindableName

→ QUOTED\_STRING →

## Syntax

```
createMappingCommand = OMBCREATE MAPPING "mappingName" [
"setPropertiesClause" ] { ADD ( "addOperatorClause" | "addGroupClause" |
"addAttributeClause" | "addChildClause" | "addConnectionClause" ) };
mappingName = "QUOTED_STRING";
setPropertiesClause = SET PROPERTIES "propertyKeyList" VALUES
"propertyValueList";
addOperatorClause = "operatorType" OPERATOR "operatorName" [
"setPropertiesClause" ] [ "setBindingClause" ];
addGroupClause = "groupDirection" GROUP "groupName" OF "operatorLocator" [
"setPropertiesClause" ];
addAttributeClause = ATTRIBUTE "attributeName" OF "groupBottomUpLocator" [
"setPropertiesClause" ];
addChildClause = "childType" "childName" "childOwnerBottomUpLocator" [
"setPropertiesClause" ];
addConnectionClause = CONNECTION FROM ( "groupBottomUpLocator" TO
"groupBottomUpLocator" [ "groupToGroupConnectType" ] |
"attributeBottomUpLocator" TO ( "attributeBottomUpLocator" |
"attributesBottomUpLocator" | "groupBottomUpLocator" ) |
"attributesBottomUpLocator" TO ( "attributesBottomUpLocator" |
"groupBottomUpLocator" ) );
propertyKeyList = "(" "propertyKey" { "," "propertyKey" } ")";
propertyValueList = "(" "propertyValue" { "," "propertyValue" } ")";
operatorType = "UNQUOTED_STRING";
operatorName = "QUOTED_STRING";
setBindingClause = BOUND TO "bindableLocator";
groupDirection = INPUT | OUTPUT | INPUT_OUTPUT;
groupName = "QUOTED_STRING";
operatorLocator = OPERATOR "operatorName";
attributeName = "QUOTED_STRING";
groupBottomUpLocator = GROUP "groupName" OF "operatorLocator";
childType = "UNQUOTED_STRING";
childName = "QUOTED_STRING";
childOwnerBottomUpLocator = { OF "childType" "childName" } [ OF
"mappableBottomUpLocator" ];
groupToGroupConnectType = COPY ALL | BY ( NAME [ IGNORE ( SPECIAL_
CHARS "QUOTED_STRING" | SOURCE_PREFIX "QUOTED_STRING" | SOURCE_
```

```

SUFFIX "QUOTED_STRING" | TARGET_PREFIX "QUOTED_STRING" | TARGET_
SUFFIX "QUOTED_STRING" ) { "," IGNORE ( SPECIAL_CHARS "QUOTED_STRING"
| SOURCE_PREFIX "QUOTED_STRING" | SOURCE_SUFFIX "QUOTED_STRING" |
TARGET_PREFIX "QUOTED_STRING" | TARGET_SUFFIX "QUOTED_STRING" ) } ]
| POSITION );

attributeBottomUpLocator = ATTRIBUTE "attributeName" OF
"groupBottomUpLocator";

attributesBottomUpLocator = ATTRIBUTES "attributeNameList" OF
"groupBottomUpLocator";

propertyKey = "UNQUOTED_STRING";

propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );

bindableLocator = "bindableType" "bindableName";

mappableBottomUpLocator = "operatorLocator" | "groupBottomUpLocator" |
"attributeBottomUpLocator";

attributeNameList = "(" "attributeName" { "," "attributeName" } ")";

bindableType = "UNQUOTED_STRING";

bindableName = "QUOTED_STRING";

```

## Keywords and Parameters

**createMappingCommand**

Create a mapping in an Oracle Module.

**mappingName**

Name of the mapping.

**setPropertyClause**

Describe the keys of properties for the map or objects in the map.

**addOperatorClause**

Adds a mapping operator to a map. When you add an operator, Warehouse Builder creates default groups and parameters for the operator. Please see the appendix section of the Scripting Reference.

The following is an example for creating a child object under a mapping (which is not an operator) `OMBALTER MAPPING 'M1' ADD SOURCE_DATA_FILE 'FILE1'`

The following is an example for creating an operator:

```
OMBALTER MAPPING 'M1' ADD TABLE OPERATOR 'T1'
```

In the second example, when user forgets to type "OPERATOR" "GROUP" "ATTRIBUTE" key word, instead of complaining the keywords are missing, OMBPlus

will complain about error getting child objects. Here is an example: OMB+>  
OMBALTER MAPPING 'M1' ADD TABLE 'T1' OMB02932: Error getting child objects  
of type TABLE in M1

TO A USER: it looks like OMBPlus should complain they forget to type a keyword.

TO OMBPLUS: the syntax is actually for creating a non-operator child object  
under the mapping. Therefore, it goes and tries to find type definition for  
non-operator child object "TABLE" and cannot find it. Therefore the exception is  
thrown.

#### addGroupClause

Add a mapping group to a mapping operator.

#### addAttributeClause

Add a mapping attribute to a mapping group.

#### addChildClause

Add a child to a mapping, mapping operator, mapping group or mapping attribute.

#### addConnectionClause

Add connections between mapping groups or mapping attributes.

#### propertyKeyList

The list of property keys

#### propertyValueList

A list of property values.

#### operatorType

Type of a mapping operator. The following operator types are available: ADVANCED\_  
QUEUE, AGGREGATOR, CONSTANT, CUBE, DATA\_GENERATOR,  
DEDUPLICATOR, DIMENSION, EXPRESSION, EXTERNAL\_PROCESS, EXTERNAL\_  
TABLE, FILTER, FLAT\_FILE, INPUT\_PARAMETER, JOINER, KEY\_LOOKUP,  
MATCHMERGE, MATERIALIZED\_VIEW, NAME\_AND\_ADDRESS, OUTPUT\_  
PARAMETER, PIVOT, POSTMAPPING\_PROCESS, PREMAPPING\_PROCESS,  
SEQUENCE, SET\_OPERATION, SORTER, SPLITTER, TABLE, TRANSFORMATION,  
UNPIVOT, VIEW.

#### operatorName



Name of a mapping operator.

setBindingClause

Set the binding during the creation of a mapping operator or mapping attribute.

groupDirection

Direction of a mapping group.

groupName

Name of a mapping group.

operatorLocator

Location of a mapping operator.

attributeName

Name of a mapping attribute.

groupBottomUpLocator

Location of a mapping group.

childType

Type of a child that belongs to map, mapping operator, mapping group or mapping attribute.

childName

Name of a child that belongs to map, mapping operator, mapping group or mapping attribute.

childOwnerBottomUpLocator

Location of a child owner. A child owner can be a map, mapping operator, mapping group, mapping attribute or a child.

groupToGroupConnectType

Connecting from a mapping group in one mapping operator to a mapping group in another mapping operator.

attributeBottomUpLocator

Location of a mapping attribute.

`attributesBottomUpLocator`

Location of a list of mapping attributes.

`propertyKey`

A property key for an object.

Basic properties for MAPPING:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the mapping

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the mapping

Basic properties for OPERATOR:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the operator

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the operator

Basic properties for GROUP:

Name: BUSINESS\_NAME  
Type: STRING(200)  
Valid Values: N/A  
Default: "  
Business name of the group

Name: DESCRIPTION  
Type: STRING(4000)  
Valid Values: N/A  
Default: "  
Description of the group

Basic properties for ATTRIBUTE:

Name: BUSINESS\_NAME  
Type: STRING(200)  
Valid Values: N/A  
Default: "  
Business name of the attribute

Name: DESCRIPTION  
Type: STRING(4000)  
Valid Values: N/A  
Default: "  
Description of the attribute

Name: DATATYPE  
Type: STRING(20)  
Valid Values: NUMBER, VARCHAR2, VARCHAR, DATE, LONG  
Default: "  
Datatype of the Attribute

Name: LENGTH  
Type: NUMBER  
Valid Values: N/A  
Default: 0  
Length of the attribute.

Name: PRECISION  
Type: NUMBER  
Valid Values: N/A  
Default: 0  
Precision of the attribute.

Name: SCALE  
Type: NUMBER  
Valid Values: N/A  
Default: 0  
Scale of the attribute.

Properties for MAPPING:

Name: DEPLOYABLE  
Type: BOOLEAN  
Valid Values: true, false  
Default: true  
Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: STEP\_TYPE  
Type: STRING  
Valid Values: ABAP, SQLLOADER, PLSQL  
Default: UNKNOWN  
The step type used to generate this mapping

Properties for MAPPING ABAP STEP:

Name: CONTROL\_FILE\_NAME  
Type: STRING  
Valid Values: N/A  
Default: owb.dat  
Control File Name

Name: DATA\_FILE\_NAME

Type: STRING  
Valid Values: N/A  
Default: owb.dat  
Data File Name

Name: FILE\_DELIMITER\_FOR\_STAGING\_FILE  
Type: STRING  
Valid Values: N/A  
Default: ~  
File Delimiter for Staging File

Name: LOG\_FILE\_NAME  
Type: STRING  
Valid Values: N/A  
Default: owb.log  
Log File Name

Name: NESTED\_LOOP  
Type: STRING  
Valid Values: TRUE\_DEFAULT, TRUE, FALSE  
Default: TRUE\_DEFAULT  
Nested Loop

Name: PRIMARY\_FOREIGN\_KEY\_FOR\_JOIN  
Type: STRING  
Valid Values: N/A  
Default: DEFAULT  
Primary Foreign Key for Join

Name: SAP\_SYSTEM\_VERSION  
Type: STRING  
Valid Values: SAP\_R3\_4X, SAP\_R3\_3X  
Default: SAP\_R3\_4X  
SAP System Version

Name: SQL\_JOIN\_COLLAPSING  
Type: STRING

Valid Values: TRUE\_DEFAULT, TRUE, FALSE

Default: TRUE\_DEFAULT

Sql Join Collapsing

Name: STAGING\_FILE\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: C:\temp\

Staging File Directory

Name: USE\_SELECT\_SINGLE

Type: STRING

Valid Values: TRUE\_DEFAULT, TRUE, FALSE

Default: TRUE\_DEFAULT

Use Select Single

Properties for MAPPING PLSQL STEP:

Name: ANALYZE\_TABLE\_SAMPLE\_PERCENTAGE

Type: NUMBER

Valid Values: N/A

Default: 90

The default percentage of rows to be sampled when the target tables are analyzed for statistics to improve performance during insertion.

Name: ANALYZE\_TABLE\_STATEMENTS

Type: BOOLEAN

Valid Values: true, false

Default: true

Generate statistics collection statement if this is true.

Name: BULK\_PROCESSING\_CODE

Type: BOOLEAN

Valid Values: true, false

Default: true

Generate bulk processing code if this is true.

Name: BULK\_SIZE

Type: NUMBER

Valid Values: N/A

Default: 50

The default number of rows to be fetched in batch during cursor processing.

Name: COMMIT\_FREQUENCY

Type: NUMBER

Valid Values: N/A

Default: 1000

The default number of rows processed before a commit is issued.

Name: CORRELATED\_COMMIT

Type: BOOLEAN

Valid Values: true, false

Default: false

The mapping commits or rolls back correlated rows together.

Name: DEFAULT\_AUDIT\_LEVEL

Type: STRING

Valid Values: NONE, STATISTICS, ERROR\_DETAILS, COMPLETE

Default: ERROR\_DETAILS

The default audit level when the step is executed.

Name: DEFAULT\_OPERATING\_MODE

Type: STRING

Valid Values: SET\_BASED, ROW\_BASED, ROW\_BASED\_TARGET\_ONLY, SET\_BASED\_FAIL\_OVER\_TO\_ROW\_BASED, SET\_BASED\_FAIL\_OVER\_TO\_ROW\_BASED\_TARGET\_ONLY

Default: SET\_BASED\_FAIL\_OVER\_TO\_ROW\_BASED

The default operating mode.

Name: DEFAULT\_PURGE\_GROUP

Type: STRING

Valid Values: N/A

Default: WB

The default purge group to be used when the step is executed.

Name: MAXIMUM\_NUMBER\_OF\_ERRORS

Type: NUMBER

Valid Values: N/A

Default: 50

The default maximum number of errors encountered before aborting the step execution.

Name: OPTIMIZED\_CODE

Type: BOOLEAN

Valid Values: true, false

Default: true

Attempt to generate optimized code if this is true.

Name: PARALLEL\_ROW\_CODE

Type: BOOLEAN

Valid Values: true, false

Default: false

Generate parallel row code if this is true.

Properties for MAPPING SQLLOADER STEP:

Name: AUDIT

Type: BOOLEAN

Valid Values: true, false

Default: true

Perform audit when the step is executed.

Name: BIND\_SIZE

Type: NUMBER

Valid Values: N/A

Default: 50000

Bind Size

Name: CONTINUE\_LOAD

Type: BOOLEAN

Valid Values: true, false

Default: false



---

## Continue Load

Name: CONTROL\_FILE\_LOCATION

Type: STRING

Valid Values: N/A

Default: N/A

The Control File Location

Name: CONTROL\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The control file name used in TCL generation

Name: DATABASE\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Database file name from which to allocate extents

Name: DEFAULT\_PURGE\_GROUP

Type: STRING

Valid Values: N/A

Default: WB

The default purge group to be used when the step is executed.

Name: DELIMITED\_FILE\_RECORD\_TERMINATION

Type: STRING

Valid Values: N/A

Default: N/A

This property has been deprecated. Please set the record delimiter in the Flat File Sample Wizard or Property Sheet.

Name: DIRECT\_MODE

Type: BOOLEAN

Valid Values: true, false

Default: true

Direct Mode

Name: ERRORS\_ALLOWED

Type: NUMBER

Valid Values: N/A

Default: 50

Number of errors to allow

Name: LOG\_FILE\_LOCATION

Type: STRING

Valid Values: N/A

Default: N/A

The log file location.

Name: LOG\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The log file name.

Name: NLS\_CHARACTERSET

Type: STRING

Valid Values: AL24UTFFSS, AR8ARABICMAC, AR8ARABICMACS, AR8ISO8859P6, AR8MSAWIN, AR8MSWIN1256, BLT8CP921, BLT8EBCDIC1112, BLT8MSWIN1257, BLT8PC775, CDN8PC863, CL8EBCDIC1025, CL8EBCDIC1025X, CL8ISO8859P5, CL8KOI8R, CL8MACCYRILLIC, CL8MACCYRILLICS, CL8MSWIN1251, D8EBCDIC273, DK8EBCDIC277, EE8EBCDIC870, EE8ISO8859P2, EE8MACCE, EE8MACCES, EE8MACCROATIAN, EE8MACCROATIANS, EE8MSWIN1250, EE8PC852, EL8EBCDIC875, EL8ISO8859P7, EL8MACGREEK, EL8MACGREEKS, EL8MSWIN1253, EL8PC437S, EL8PC737, EL8PC869, F8EBCDIC297, I8EBCDIC280, IS8MACICELANDIC, IS8MACICELANDICS, IS8PC861, IW8EBCDIC424, IW8ISO8859P8, IW8MACHEBREW, IW8MACHEBREWS, IW8MSWIN1255, JA16EBCDIC930, JA16EUC, JA16EUCYEN, JA16MACSJIS, JA16SJIS, JA16SJISYEN, JA16VMS, KO16KSC5601, LT8MSWIN921, N8PC865, NEE8ISO8859P4, RU8PC855, RU8PC866, S8EBCDIC278, SE8ISO8859P3, TH8MACTHAI, TH8MACTHAIS, TH8TISASCII, TR8EBCDIC1026, TR8MACTURKISH, TR8MACTURKISHS, TR8MSWIN1254, TR8PC857, US7ASCII, US8PC437, UTF8, WE8EBCDIC284, WE8EBCDIC285, WE8EBCDIC37, WE8EBCDIC37C, WE8EBCDIC500, WE8EBCDIC500C, WE8EBCDIC871, WE8ISO8859P1, WE8ISO8859P9, WE8MACROMAN8, WE8MACROM AN8S, WE8MSWIN1252, WE8PC850, WE8PC860, ZHS16CGB231280, ZHS16GBK, ZHS16MACCGB231280, ZHT16BIG5, ZHT16MSWIN950, ZHT32EUC

Default: WE8MSWIN1252

**Nls Characterset****Name: OPERATION\_RECOVERABLE****Type: BOOLEAN****Valid Values: true, false****Default: true****Operation Recoverable****Name: PERFORM\_PARALLEL\_LOAD****Type: BOOLEAN****Valid Values: true, false****Default: false****Perform Parallel Load****Name: PRESERVE\_BLANKS****Type: BOOLEAN****Valid Values: true, false****Default: false****Preserve Blanks****Name: READ\_BUFFERS****Type: NUMBER****Valid Values: N/A****Default: 4****The Number of Buffers****Name: READ\_SIZE****Type: NUMBER****Valid Values: N/A****Default: 65536****The size of the read buffer****Name: RECORDS\_TO\_LOAD****Type: NUMBER****Valid Values: N/A****Default: N/A****Number of logical records to load. The default value of 0 indicates to load all records.**

Name: RECORDS\_TO\_SKIP

Type: NUMBER

Valid Values: N/A

Default: N/A

Number of logical records to skip

Name: ROWS\_PER\_COMMIT

Type: NUMBER

Valid Values: N/A

Default: 200

Rows per Commit

Properties for MAPPING SQLLOADER STEP SOURCE\_DATA\_FILE CHILD:

Name: BAD\_FILE\_LOCATION

Type: STRING

Valid Values: N/A

Default: N/A

Bad file location.

Name: BAD\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Bad file name. If the Bad File Location is set then this must be a relative file name. Otherwise this should contain a fully qualified path.

Name: DATA\_FILE\_LOCATION

Type: STRING

Valid Values: N/A

Default: N/A

The location for this component.

Name: DATA\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The data file name for this component. If the Data File Location is set then this must be a relative file name. Otherwise this should contain a fully qualified path.

Name: DISCARD\_FILE\_LOCATION

Type: STRING

Valid Values: N/A

Default: N/A

Discard file location

Name: DISCARD\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Discard file name. If the Discard File Location is set then this must be a relative file name. Otherwise this should contain a fully qualified path.

Name: DISCARD\_MAX

Type: NUMBER

Valid Values: N/A

Default: 0

Discard Max

Properties for ADVANCED\_QUEUE OPERATOR:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Properties for AGGREGATOR OPERATOR:

Name: GROUP\_BY\_CLAUSE

Type: STRING

Valid Values: N/A

Default: N/A

The Group By clause for the aggregation

Name: HAVING\_CLAUSE

Type: STRING

Valid Values: N/A

Default: N/A

The Having clause for the aggregation

Properties for CUBE OPERATOR:

Name: ADVANCED\_MATCH\_BY\_CONSTRAINT

Type: STRING

Valid Values: N/A

Default: No constraints

This property is used by public API and scripting to influence how columns are used for UPDATE or DELETE DMLs. If this property is set with the name of a primary or unique key, all the columns in the key will be used for matching during UPDATE or DELETE; and all the columns not in the key are used for loading. The property can also be assigned the value "All constraints" or "No constraints". If the DML type is INSERT, TRUNCATE/INSERT, or CHECK/INSERT, setting this property causes no effect.

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DATABASE\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Database file name to allocate extents from

Name: DATABASE\_LINK

Type: STRING

Valid Values: N/A

Default: N/A

The database link used to access this entity during mapping.

Name: DATA\_COLLECTION\_FREQUENCY

Type: STRING

Valid Values: UNKNOWN, YEAR, QUARTER, MONTH, DAY, HOUR, MINUTE

Default: UNKNOWN

New Data Granularity

Name: DIRECT

Type: BOOLEAN

Valid Values: true, false

Default: false

Directly swap source into target as a partition without first creating a staging table.

Name: ENABLE\_CONSTRAINTS

Type: BOOLEAN

Valid Values: true, false

Default: true

Enable Constraints

Name: EXCEPTIONS\_TABLE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Exceptions Table Name

Name: EXTRACTION\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when extracting from this table using SQL

Name: LOADING\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when loading into this table using SQL

Name: LOADING\_TYPE

Type: STRING

Valid Values: INSERT, UPDATE, INSERT\_UPDATE, UPDATE\_INSERT, DELETE, NONE, TRUNCATE\_INSERT, DELETE\_INSERT, CHECK\_INSERT

Default: INSERT

The loading operation to be performed when this is a target.

Name: MATCH\_BY\_CONSTRAINT

Type: STRING

Valid Values: N/A

Default: YES

A property to indicate whether unique or primary key information on this target will override the matching criteria obtained from Match by constraint property on the attributes of this target.

Name: PARTITION\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Partition Name

Name: PEL\_ENABLED

Type: BOOLEAN

Valid Values: true, false

Default: false

PEL Enabled

Name: PRIMARY\_SOURCE

Type: STRING

Valid Values: YES, NO

Default: NO

A boolean value to indicate whether this is a primary source or not (only used in EDW).

Name: RECORDS\_TO\_SKIP

Type: NUMBER

Valid Values: N/A



Default: N/A

Number of records to skip

Name: REPLACE\_DATA

Type: BOOLEAN

Valid Values: true, false

Default: false

Replace existing data in target partition if there is any.

Name: SCHEMA

Type: STRING

Valid Values: N/A

Default: N/A

Schema

Name: SINGLEROW

Type: BOOLEAN

Valid Values: true, false

Default: false

Singlerow

Name: SORTED\_INDEXES\_CLAUSE

Type: STRING

Valid Values: N/A

Default: N/A

Sorted Indexes Clause

Name: SUBPARTITION\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Subpartition Name

Name: TARGET\_FILTER\_FOR\_DELETE

Type: STRING

Valid Values: N/A

Default: N/A

A condition on the rows in the target and if evaluated to true, that row will participate in the delete loading operation.

Name: TARGET\_FILTER\_FOR\_UPDATE

Type: STRING

Valid Values: N/A

Default: N/A

A condition on the rows in the target and if evaluated to true, that row will participate in the update loading operation.

Name: TRAILING\_NULLCOLS

Type: BOOLEAN

Valid Values: true, false

Default: false

Trailing Nullcols

Properties for DIMENSION OPERATOR:

Name: ADVANCED\_MATCH\_BY\_CONSTRAINT

Type: STRING

Valid Values: N/A

Default: No constraints

This property is used by public API and scripting to influence how columns are used for UPDATE or DELETE DMLs. If this property is set with the name of a primary or unique key, all the columns in the key will be used for matching during UPDATE or DELETE; and all the columns not in the key are used for loading. The property can also be assigned the value "All constraints" or "No constraints". If the DML type is INSERT, TRUNCATE/INSERT, or CHECK/INSERT, setting this property causes no effect.

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DATABASE\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Database file name to allocate extents from

Name: DATABASE\_LINK

Type: STRING

Valid Values: N/A

Default: N/A

The database link used to access this entity during mapping.

Name: DATA\_COLLECTION\_FREQUENCY

Type: STRING

Valid Values: UNKNOWN, YEAR, QUARTER, MONTH, DAY, HOUR, MINUTE

Default: UNKNOWN

New Data Granularity

Name: DIRECT

Type: BOOLEAN

Valid Values: true, false

Default: false

Directly swap source into target as a partition without first creating a staging table.

Name: ENABLE\_CONSTRAINTS

Type: BOOLEAN

Valid Values: true, false

Default: true

Enable Constraints

Name: EXCEPTIONS\_TABLE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Exceptions Table Name

Name: EXTRACTION\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when extracting from this table using SQL

Name: LOADING\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when loading into this table using SQL

Name: LOADING\_TYPE

Type: STRING

Valid Values: INSERT, UPDATE, INSERT\_UPDATE, UPDATE\_INSERT, DELETE, NONE, TRUNCATE\_INSERT, DELETE\_INSERT, CHECK\_INSERT

Default: INSERT

The loading operation to be performed when this is a target.

Name: MATCH\_BY\_CONSTRAINT

Type: STRING

Valid Values: N/A

Default: YES

A property to indicate whether unique or primary key information on this target will override the matching criteria obtained from Match by constraint property on the attributes of this target.

Name: PARTITION\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Partition Name

Name: PEL\_ENABLED

Type: BOOLEAN

Valid Values: true, false

Default: false

PEL Enabled

Name: PRIMARY\_SOURCE

Type: STRING

Valid Values: YES, NO

Default: NO

A boolean value to indicate whether this is a primary source or not (only used in EDW).

Name: RECORDS\_TO\_SKIP

Type: NUMBER

Valid Values: N/A

Default: N/A

Number of records to skip

Name: REPLACE\_DATA

Type: BOOLEAN

Valid Values: true, false

Default: false

Replace existing data in target partition if there is any.

Name: SCHEMA

Type: STRING

Valid Values: N/A

Default: N/A

Schema

Name: SINGLEROW

Type: BOOLEAN

Valid Values: true, false

Default: false

Singlerow

Name: SORTED\_INDEXES\_CLAUSE

Type: STRING

Valid Values: N/A

Default: N/A

Sorted Indexes Clause

Name: SUBPARTITION\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Subpartition Name

Name: TARGET\_FILTER\_FOR\_DELETE

Type: STRING

Valid Values: N/A

Default: N/A

A condition on the rows in the target and if evaluated to true, that row will participate in the delete loading operation.

Name: TARGET\_FILTER\_FOR\_UPDATE

Type: STRING

Valid Values: N/A

Default: N/A

A condition on the rows in the target and if evaluated to true, that row will participate in the update loading operation.

Name: TRAILING\_NULLCOLS

Type: BOOLEAN

Valid Values: true, false

Default: false

Trailing Nullcols

Properties for EXTERNAL\_TABLE OPERATOR:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DATABASE\_LINK

Type: STRING

Valid Values: N/A

Default: N/A

The database link used to access this entity during mapping.

Name: EXTRACTION\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when extracting from this table using SQL

Name: LOADING\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when loading into this table using SQL

Name: PRIMARY\_SOURCE

Type: STRING

Valid Values: YES, NO

Default: NO

A boolean value to indicate whether this is a primary source or not (only used in EDW).

Name: SCHEMA

Type: STRING

Valid Values: N/A

Default: N/A

Schema

Properties for FILTER OPERATOR:

Name: FILTER\_CONDITION

Type: STRING

Valid Values: N/A

Default: N/A

The boolean filtering condition that identifies what data is to be processed. Any row with a false condition will be ignored.

Properties for FLAT\_FILE OPERATOR:

Name: CONCATENATE\_RECORDS

Type: NUMBER

Valid Values: N/A

Default: 0

Number of Physical Records per Logical Record.

Name: CONTINUATION\_CHARACTER

Type: STRING

Valid Values: N/A

Default: N/A

Character that indicates the record is continued on the next line.

Name: CONTINUATION\_CHARACTER\_ON\_NEXT\_LINE

Type: BOOLEAN

Valid Values: true, false

Default: FALSE

If there is a continuation character, is it at the start of the line.

Name: FIELD\_ENCLOSURE\_CHARACTERS

Type: STRING

Valid Values: N/A

Default: N/A

Characters that wrap fields. Example ' or ".

Name: FIELD\_NAMES\_IN\_THE\_FIRST\_ROW

Type: BOOLEAN

Valid Values: true, false

Default: FALSE

Indicates whether to write the field names in the first row of the output file.

Name: FIELD\_TERMINATION\_CHARACTER

Type: STRING

Valid Values: N/A

Default: ,

Character that separates the fields of a delimited file.

Name: FILE\_FORMAT

Type: STRING

Valid Values: N/A



Default: DELIMITED  
File Format (Fixed or Delimited).

Name: LOADING\_TYPE  
Type: STRING  
Valid Values: INSERT, UPDATE, NONE  
Default: INSERT  
The loading operation to be performed

Name: NLS\_CHARACTERSET  
Type: STRING  
Valid Values: N/A  
Default: WE8MSWIN1252  
NLS Characterset

Name: RECORD\_DELIMITER  
Type: STRING  
Valid Values: N/A  
Default: N/A  
Character that indicates the end of the record.

Name: RECORD\_SIZE  
Type: NUMBER  
Valid Values: N/A  
Default: 0  
Size of a fixed length record.

Name: RECORD\_TYPE\_LENGTH  
Type: NUMBER  
Valid Values: N/A  
Default: 0  
If this is a multi record file, this will indicate the length of the data that identifies the type of record. It is used with the Record Type Position.

Name: RECORD\_TYPE\_POSITION  
Type: NUMBER  
Valid Values: N/A

Default: 0

If this is a multi record file, this will indicate the position of the field that identifies the type of record.

Name: SAMPLED\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name of the file that was sampled to get the metadata for this file.

Name: TARGET\_DATA\_FILE\_LOCATION

Type: STRING

Valid Values: N/A

Default: N/A

The name of the location in which to open the target data file. Make sure the root path of this location, as registered in the Runtime Platform, is exactly specified in the initialization file (INIT.ORA) of your runtime database using the UTL\_FILE\_DIR parameter.

Name: TARGET\_DATA\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Name of the target data file, including extension (file type). This name should not include the file path. To specify where the target data file will be created/appended, set the Target Data File Location.

Properties for INPUT\_PARAMETER OPERATOR:

Properties for JOINER OPERATOR:

Name: JOIN\_CONDITION

Type: STRING

Valid Values: N/A

Default: N/A

The Join Condition for the join operator

Properties for KEY\_LOOKUP OPERATOR:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DATABASE\_LINK

Type: STRING

Valid Values: N/A

Default: N/A

The database link used to access this entity during mapping.

Name: LOOKUP\_CONDITION

Type: STRING

Valid Values: N/A

Default: N/A

Key lookup condition based on the source inputs. This condition is used to lookup a value in the bound table. If the condition is not met, the default value expression will be returned. If a default expression is not defined, null is used.

Name: SCHEMA

Type: STRING

Valid Values: N/A

Default: N/A

Schema

Properties for MATCHMERGE OPERATOR:

Name: MATCH\_KEYS

Type: STRING

Valid Values: N/A

Default: N/A

Ordered list of attributes that control the set of records to be matched at any at any particular time.

Name: MERGED\_PREFIX

Type: STRING

Valid Values: N/A

Default: N/A

Set the prefix used for the merged attributes in the cross-reference group.

Properties for MATERIALIZED\_VIEW OPERATOR:

Name: ADVANCED\_MATCH\_BY\_CONSTRAINT

Type: STRING

Valid Values: N/A

Default: No constraints

This property is used by public API and scripting to influence how columns are used for UPDATE or DELETE DMLs. If this property is set with the name of a primary or unique key, all the columns in the key will be used for matching during UPDATE or DELETE; and all the columns not in the key are used for loading. The property can also be assigned the value "All constraints" or "No constraints". If the DML type is INSERT, TRUNCATE/INSERT, or CHECK/INSERT, setting this property causes no effect.

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DATABASE\_LINK

Type: STRING

Valid Values: N/A

Default: N/A

The database link used to access this entity during mapping.

Name: EXTRACTION\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when extracting from this table using SQL

Name: LOADING\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when loading into this table using SQL

Name: LOADING\_TYPE

Type: STRING

Valid Values: INSERT, UPDATE, INSERT\_UPDATE, UPDATE\_INSERT, DELETE, NONE, TRUNCATE\_INSERT, DELETE\_INSERT, CHECK\_INSERT

Default: INSERT

The loading operation to be performed when this is a target.

Name: MATCH\_BY\_CONSTRAINT

Type: STRING

Valid Values: N/A

Default: YES

A property to indicate whether unique or primary key information on this target will override the matching criteria obtained from Match by constraint property on the attributes of this target.

Name: PRIMARY\_SOURCE

Type: STRING

Valid Values: YES, NO

Default: NO

A boolean value to indicate whether this is a primary source or not (only used in EDW).

Name: SCHEMA

Type: STRING

Valid Values: N/A

Default: N/A

Schema

Name: TARGET\_FILTER\_FOR\_DELETE

Type: STRING

Valid Values: N/A

Default: N/A

A condition on the rows in the target and if evaluated to true, that row will participate in the delete loading operation.

Name: TARGET\_FILTER\_FOR\_UPDATE

Type: STRING

Valid Values: N/A

Default: N/A

A condition on the rows in the target and if evaluated to true, that row will participate in the update loading operation.

Properties for NAME\_AND\_ADDRESS OPERATOR:

Name: ADDRESS\_LINE\_1

Type: STRING

Valid Values: N/A

Default: N/A

Address line 1

Name: ADDRESS\_LINE\_2

Type: STRING

Valid Values: N/A

Default: N/A

Address line 2

Name: ADDRESS\_LINE\_3

Type: STRING

Valid Values: N/A

Default: N/A

Address line 3

Name: ADDRESS\_LINE\_4

Type: STRING

Valid Values: N/A

Default: N/A

Address line 4

Name: DUAL\_ADDRESS\_ASSIGNMENT

Type: STRING

Valid Values: NA\_DUALADDR\_STREET, NA\_DUALADDR\_POBOX, NA\_DUALADDR\_CLOSESTTOLASTLINE

Default: NA\_DUALADDR\_STREET

---

A dual address refers to two address lines for the same destination. For example, a record contains both a street address and a P.O. Box; this is common with business data. Select which of the two address lines should be assigned in these cases.

Name: GENERATE\_CASS\_REPORT

Type: STRING

Valid Values: NA\_YES, NA\_NO

Default: NA\_NO

Select "Yes" to generate CASS (Coding Accuracy Support System) report. CASS report is a text file specified by the United States Postal Service. The report is written to the [nas/bin/admin/reports] folder under the home folder of the name/address server.

Name: LIST\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The list name is optional and provides a reference for tracking multiple CASS reports.

Name: PARSING\_TYPE

Type: STRING

Valid Values: NA\_NAMEONLY, NA\_ADDRESSONLY, NA\_NAMEANDADDRESS

Default: NA\_NAMEANDADDRESS

Select a name-address parsing type to be performed on the input data

Name: PRIMARY\_COUNTRY

Type: STRING

Valid Values: NA\_ARG, NA\_AUS, NA\_BEL, NA\_BRA, NA\_CAN, NA\_CHL, NA\_COL, NA\_DNK, NA\_FRA, NA\_DEU, NA\_HKG, NA\_IND, NA\_IRL, NA\_ITA, NA\_MEX, NA\_MYS, NA\_NLD, NA\_NZL, NA\_PER, NA\_PHL, NA\_PRT, NA\_SGP, NA\_ZAF, NA\_ESP, NA\_SWE, NA\_CHE, NA\_ARE, NA\_GBR, NA\_USA, NA\_VEN

Default: NA\_USA

Select the primary parsing country which best represents the input data. Input addresses having the same country as the primary parsing country will only need to be parsed once. Input addresses having a different country than the primary parsing country may be reparsed by a different parser. For performance reasons, it is best to minimize the percentage of 2-pass parses by selecting the optimal parser.

Name: PROCESSOR\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The processor name is the name of the organization submitting the CASS report.

Properties for OUTPUT\_PARAMETER OPERATOR:

Properties for PIVOT OPERATOR:

Name: PIVOT\_GROUP\_SIZE

Type: NUMBER

Valid Values: 1 - 1000

Default: 2

A number specifying the pivot group size for the pivot operation. Pivot group size determines the number of output rows that are produced from each input row.

Properties for POSTMAPPING\_PROCESS OPERATOR:

Name: FUNCTION\_CALL

Type: STRING

Valid Values: N/A

Default: N/A

Expression template for procedure call

Name: FUNCTION\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Name of the transformation to be called.

Name: POSTMAPPING\_PROCESS\_RUN\_CONDITION

Type: STRING

Valid Values: ALWAYS, ON\_SUCCESS, ON\_ERROR, ON\_WARNING

Default: ON\_SUCCESS

Indicates under what condition of the mapping the post-mapping process will be run.

Name: ROWBASED\_ONLY

Type: BOOLEAN

Valid Values: true, false

Default: N/A



Indicates if this transformation must be used only Row Based mode. Some transformations can be used in SQL mode as well as Row Based mode.

Properties for PREMAPPING\_PROCESS OPERATOR:

Name: FUNCTION\_CALL

Type: STRING

Valid Values: N/A

Default: N/A

Expression template for procedure call

Name: FUNCTION\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Name of the transformation to be called.

Name: MAPPING\_RUN\_CONDITION

Type: STRING

Valid Values: ALWAYS, ON\_SUCCESS, ON\_ERROR

Default: ON\_SUCCESS

Indicates under what condition of the pre-mapping process the mapping will be run.

Name: ROWBASED\_ONLY

Type: BOOLEAN

Valid Values: true, false

Default: N/A

Indicates if this transformation must be used only Row Based mode. Some transformations can be used in SQL mode as well as Row Based mode.

Properties for SEQUENCE OPERATOR:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DATABASE\_LINK

Type: STRING

Valid Values: N/A

Default: N/A

The database link used to access this entity during mapping.

Name: SCHEMA

Type: STRING

Valid Values: N/A

Default: N/A

Schema

Properties for SET\_OPERATION OPERATOR:

Name: SET\_OPERATION

Type: STRING

Valid Values: UNION, UNIONALL, INTERSECT, MINUS

Default: UNION

Specifies the set operation that is to be performed by this operator.

Properties for SORTER OPERATOR:

Name: ORDER\_BY\_CLAUSE

Type: STRING

Valid Values: N/A

Default: N/A

The Order By Clause

Properties for TABLE OPERATOR:

Name: ADVANCED\_MATCH\_BY\_CONSTRAINT

Type: STRING

Valid Values: N/A

Default: No constraints

This property is used by public API and scripting to influence how columns are used for UPDATE or DELETE DMLs. If this property is set with the name of a primary or unique key, all the columns in the key will be used for matching during UPDATE or

DELETE; and all the columns not in the key are used for loading. The property can also be assigned the value "All constraints" or "No constraints". If the DML type is INSERT, TRUNCATE/INSERT, or CHECK/INSERT, setting this property causes no effect.

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DATABASE\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Database file name to allocate extents from

Name: DATABASE\_LINK

Type: STRING

Valid Values: N/A

Default: N/A

The database link used to access this entity during mapping.

Name: DATA\_COLLECTION\_FREQUENCY

Type: STRING

Valid Values: UNKNOWN, YEAR, QUARTER, MONTH, DAY, HOUR, MINUTE

Default: UNKNOWN

New Data Granularity

Name: DIRECT

Type: BOOLEAN

Valid Values: true, false

Default: false

Directly swap source into target as a partition without first creating a staging table.

Name: ENABLE\_CONSTRAINTS

Type: BOOLEAN

Valid Values: true, false

Default: true

Enable Constraints

Name: EXCEPTIONS\_TABLE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Exceptions Table Name

Name: EXTRACTION\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when extracting from this table using SQL

Name: LOADING\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when loading into this table using SQL

Name: LOADING\_TYPE

Type: STRING

Valid Values: INSERT, UPDATE, INSERT\_UPDATE, UPDATE\_INSERT, DELETE, NONE, TRUNCATE\_INSERT, DELETE\_INSERT, CHECK\_INSERT

Default: INSERT

The loading operation to be performed when this is a target.

Name: MATCH\_BY\_CONSTRAINT

Type: STRING

Valid Values: N/A

Default: YES

A property to indicate whether unique or primary key information on this target will override the matching criteria obtained from Match by constraint property on the attributes of this target.

Name: PARTITION\_NAME

Type: STRING  
Valid Values: N/A  
Default: N/A  
Partition Name

Name: PEL\_ENABLED  
Type: BOOLEAN  
Valid Values: true, false  
Default: false  
PEL Enabled

Name: PRIMARY\_SOURCE  
Type: STRING  
Valid Values: YES, NO  
Default: NO

A boolean value to indicate whether this is a primary source or not (only used in EDW).

Name: RECORDS\_TO\_SKIP  
Type: NUMBER  
Valid Values: N/A  
Default: N/A  
Number of records to skip

Name: REPLACE\_DATA  
Type: BOOLEAN  
Valid Values: true, false  
Default: false  
Replace existing data in target partition if there is any.

Name: SCHEMA  
Type: STRING  
Valid Values: N/A  
Default: N/A  
Schema

Name: SINGLEROW

Type: BOOLEAN  
Valid Values: true, false  
Default: false  
Singlerow

Name: SORTED\_INDEXES\_CLAUSE  
Type: STRING  
Valid Values: N/A  
Default: N/A  
Sorted Indexes Clause

Name: SUBPARTITION\_NAME  
Type: STRING  
Valid Values: N/A  
Default: N/A  
Subpartition Name

Name: TARGET\_FILTER\_FOR\_DELETE  
Type: STRING  
Valid Values: N/A  
Default: N/A  
A condition on the rows in the target and if evaluated to true, that row will participate in the delete loading operation.

Name: TARGET\_FILTER\_FOR\_UPDATE  
Type: STRING  
Valid Values: N/A  
Default: N/A  
A condition on the rows in the target and if evaluated to true, that row will participate in the update loading operation.

Name: TRAILING\_NULLCOLS  
Type: BOOLEAN  
Valid Values: true, false  
Default: false  
Trailing Nullcols

Properties for TABLE\_FUNCTION OPERATOR:

Name: TABLE\_FUNCTION\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name of the Table Function

Properties for TRANSFORMATION OPERATOR:

Name: FUNCTION\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Name of the transformation to be called.

Name: ROWBASED\_ONLY

Type: BOOLEAN

Valid Values: true, false

Default: N/A

Indicates if this transformation must be used only Row Based mode. Some transformations can be used in SQL mode as well as Row Based mode.

Properties for VIEW OPERATOR:

Name: ADVANCED\_MATCH\_BY\_CONSTRAINT

Type: STRING

Valid Values: N/A

Default: No constraints

This property is used by public API and scripting to influence how columns are used for UPDATE or DELETE DMLs. If this property is set with the name of a primary or unique key, all the columns in the key will be used for matching during UPDATE or DELETE; and all the columns not in the key are used for loading. The property can also be assigned the value "All constraints" or "No constraints". If the DML type is INSERT, TRUNCATE/INSERT, or CHECK/INSERT, setting this property causes no effect.

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DATABASE\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Database file name to allocate extents from

Name: DATABASE\_LINK

Type: STRING

Valid Values: N/A

Default: N/A

The database link used to access this entity during mapping.

Name: EXTRACTION\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when extracting from this table using SQL

Name: LOADING\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when loading into this table using SQL

Name: LOADING\_TYPE

Type: STRING

Valid Values: INSERT, UPDATE, INSERT\_UPDATE, UPDATE\_INSERT, DELETE, NONE, TRUNCATE\_INSERT, DELETE\_INSERT, CHECK\_INSERT

Default: INSERT

The loading operation to be performed when this is a target.

Name: MATCH\_BY\_CONSTRAINT

Type: STRING

Valid Values: N/A



Default: YES

A property to indicate whether unique or primary key information on this target will override the matching criteria obtained from Match by constraint property on the attributes of this target.

Name: PARTITION\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Partition Name

Name: PRIMARY\_SOURCE

Type: STRING

Valid Values: YES, NO

Default: NO

A boolean value to indicate whether this is a primary source or not (only used in EDW).

Name: RECORDS\_TO\_SKIP

Type: NUMBER

Valid Values: N/A

Default: N/A

Number of records to skip

Name: SCHEMA

Type: STRING

Valid Values: N/A

Default: N/A

Schema

Name: SINGLEROW

Type: BOOLEAN

Valid Values: true, false

Default: false

Singlerow

Name: SORTED\_INDEXES\_CLAUSE

Type: STRING

Valid Values: N/A  
 Default: N/A  
 Sorted Indexes Clause

Name: SUBPARTITION\_NAME  
 Type: STRING  
 Valid Values: N/A  
 Default: N/A  
 Subpartition Name

Name: TARGET\_FILTER\_FOR\_DELETE  
 Type: STRING  
 Valid Values: N/A  
 Default: N/A

A condition on the rows in the target and if evaluated to true, that row will participate in the delete loading operation.

Name: TARGET\_FILTER\_FOR\_UPDATE  
 Type: STRING  
 Valid Values: N/A  
 Default: N/A

A condition on the rows in the target and if evaluated to true, that row will participate in the update loading operation.

Name: TRAILING\_NULLCOLS  
 Type: BOOLEAN  
 Valid Values: true, false  
 Default: false

Trailing Nullcols

Properties for CUBE, DIMENSION, KEY\_LOOKUP, MATERIALIZED\_VIEW, TABLE, VIEW OPERATOR KEYS\_READONLY CHILD:

Name: KEY\_COLUMNS  
 Type: STRING  
 Valid Values: N/A  
 Default: N/A

Local columns that define this key (Comma separated if more than one).

Name: KEY\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Name of the primary, foreign or unique key (primary, foreign, or unique).

Name: KEY\_TYPE

Type: STRING

Valid Values: N/A

Default: UNIQUE

Type of key - primary, foreign or unique.

Name: REFERENCED\_KEYS

Type: STRING

Valid Values: N/A

Default: N/A

If the key is a foreign key, this will contain the key or keys used of the referenced object.

Properties for MATCHMERGE OPERATOR MATCH\_RULES CHILD:

Name: ADDRESS\_ADDRESS\_LINE\_SIMILARITY\_SCORE

Type: NUMBER

Valid Values: N/A

Default: N/A

Similarity score for address line in Address Match Rule.

Name: ADDRESS\_ALLOW\_DIFFERING\_SECONDARY\_ADDRESSES

Type: BOOLEAN

Valid Values: true, false

Default: false

Allow differing secondary addresses to match in Address Match Rule.

Name: ADDRESS\_LAST\_LINE\_SIMILARITY\_SCORE

Type: NUMBER

Valid Values: N/A

Default: N/A

Similarity score for last line in Address Match Rule.

Name: ADDRESS\_MATCH\_ON\_ADDRESS\_LINE\_SIMILARITY

Type: BOOLEAN

Valid Values: true, false

Default: false

Check address line similarity in Address Match Rule.

Name: ADDRESS\_MATCH\_ON\_BLANK\_SECONDARY\_ADDRESSES

Type: BOOLEAN

Valid Values: true, false

Default: false

Match on blank secondary address in Address Match Rule.

Name: ADDRESS\_MATCH\_ON\_LAST\_LINE\_SIMILARITY

Type: BOOLEAN

Valid Values: true, false

Default: false

Check last line similarity in Address Match Rule.

Name: ADDRESS\_MATCH\_ON\_STREET\_OR\_PO\_BOX

Type: BOOLEAN

Valid Values: true, false

Default: false

Match on Street or Post Office (PO) Box in Address Match Rule.

Name: CUSTOM\_RULE

Type: STRING

Valid Values: N/A

Default: N/A

Custom Merge Rule

Name: DESCRIPTION

Type: STRING

Valid Values: N/A

Default: N/A

Description of match rule.

Name: FIRM\_CROSS\_MATCH\_FIRM1\_AND\_FIRM2

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a firm match rule, should Warehouse Builder cross match firm 1 and firm 2?

Name: FIRM\_MATCH\_ON\_ABBREVIATIONS

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a firm match rule, should Warehouse Builder match on abbreviations?

Name: FIRM\_MATCH\_ON\_ACRONYMS

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a firm match rule, should Warehouse Builder match on acronyms?

Name: FIRM\_MATCH\_ON\_PARTIAL\_NAMES

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a firm match rule, should Warehouse Builder match on partial names?

Name: FIRM\_MATCH\_ON\_SIMILARITY

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a firm match rule, should Warehouse Builder check firm similarity?

Name: FIRM\_SIMILARITY\_SCORE

Type: NUMBER

Valid Values: N/A

Default: N/A

Similarity score for firm in Firm Match Rule.

Name: FIRM\_STRIP\_NOISE\_WORDS

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a firm match rule, should Warehouse Builder strip noise words?

Name: PERSON\_DETECT\_SWITCHED\_NAME\_ORDER

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person rule, should Warehouse Builder detect the switched name order.

Name: PERSON\_FN\_DETECT\_COMPOUND\_NAME

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder detect compound names?

Name: PERSON\_FN\_MATCH\_ON\_INITIALS

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder match on initials.

Name: PERSON\_FN\_MATCH\_ON\_SIMILARITY

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder match on first name similarity?

Name: PERSON\_FN\_MATCH\_ON\_SOUNDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder match on first name soundex?

Name: PERSON\_FN\_MATCH\_ON\_SUBSTRINGS

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder match on first name substrings?

Name: PERSON\_FN\_MRS\_MATCH

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder detect MRS? (For example, should Mrs John Smith match Mrs Smith)

Name: PERSON\_FN\_SIMILARITY\_SCORE

Type: NUMBER

Valid Values: N/A

Default: N/A

If matching in first name similarity, what is the similarity score?

Name: PERSON\_LN\_DETECT\_MISSING\_HYPHEN

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder detect missing hyphens?

Name: PERSON\_LN\_MATCH\_HYPHENATED\_NAMES

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder match on hyphenated last name?

Name: PERSON\_LN\_MATCH\_ON\_SIMILARITY

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder match on last name similarity?

Name: PERSON\_LN\_MATCH\_ON\_SOUNDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder match on last name soundex?

Name: PERSON\_LN\_SIMILARITY\_SCORE

Type: NUMBER

Valid Values: N/A

Default: N/A

If Warehouse Builder is matching on last name similarity, what is the similarity score?

Name: PERSON\_MN\_MATCH\_ON\_INITIALS

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder match on middle name initials?

Name: PERSON\_MN\_MATCH\_ON\_SIMILARITY

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder match on middle name similarity?

Name: PERSON\_MN\_MATCH\_ON\_SOUNDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder match on middle name soundex?

Name: PERSON\_MN\_MATCH\_ON\_SUBSTRINGS

Type: BOOLEAN



Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder match on middle name substrings?

Name: PERSON\_MN\_SIMILARITY\_SCORE

Type: NUMBER

Valid Values: N/A

Default: N/A

If Warehouse Builder is matching on middle name similarity, what is the similarity score?

Name: TOTAL\_WEIGHT\_SCORE

Type: NUMBER

Valid Values: N/A

Default: N/A

If this is a weight rule, the sum of all weights must be equal to or greater than this score for the records to pass the weight rule.

Name: TYPE

Type: STRING

Valid Values: MM\_TRUE, MM\_FALSE, MM\_WEIGHT, MM\_CONDITIONAL, MM\_CUSTOM, MM\_PERSON, MM\_FIRM, MM\_ADDRESS

Default: MM\_TRUE

What type of match rule is this? Possible types are: MM\_TRUE, MM\_FALSE, MM\_WEIGHT, MM\_CONDITIONAL, MM\_CUSTOM, MM\_PERSON, MM\_FIRM, MM\_ADDRESS

Name: USAGE

Type: STRING

Valid Values: MM\_ACTIVE, MM\_PASSIVE

Default: MM\_ACTIVE

Is this an active or passive rule? Possible usages are: MM\_ACTIVE, MM\_PASSIVE.

Properties for MATCHMERGE OPERATOR MERGE\_RULES CHILD:

Name: ATTRIBUTE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Merge Attribute

Name: COPY\_ATTRIBUTE

Type: STRING

Valid Values: N/A

Default: N/A

Copy this merged attribute to merge rule target attribute.

Name: CUSTOM\_TEXT

Type: STRING

Valid Values: N/A

Default: N/A

Implementation text for custom merge rule. Include "BEGIN and END statements.

Name: DESCRIPTION

Type: STRING

Valid Values: N/A

Default: N/A

Description

Name: MATCH\_ID\_SEQUENCE\_MODULE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Physical sequence module name for match id merge rule. This sequence will be used to generate the match id.

Name: MATCH\_ID\_SEQUENCE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Physical sequence name for match id merge rule. This sequence will be used to generate the match id.

Name: MIN\_MAX\_ATTRIBUTE

Type: STRING

Valid Values: N/A

Default: N/A

Selecting attribute for min/max merge rule

Name: MIN\_MAX\_TYPE

Type: STRING

Valid Values: MM\_MIN, MM\_MAX, MM\_SHORTEST, MM\_LONGEST

Default: MM\_MAX

Select record where attribute is min,max, shortest, longest. Possible values are MM\_MIN, MM\_MAX, MM\_SHORTEST, MM\_LONGEST.

Name: SEQUENCE\_MODULE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Physical sequence module name for sequence merge rule.

Name: SEQUENCE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Physical sequence name for sequence merge rule.

Name: TYPE

Type: STRING

Valid Values: MM\_ANY, MM\_MATCH\_ID, MM\_RANK, MM\_SEQUENCE, MM\_MIN\_MAX, MM\_COPY, MM\_CUSTOM, MM\_RECORD\_ANY, MM\_RECORD\_RANK, MM\_RECORD\_MIN\_MAX, MM\_RECORD\_CUSTOM

Default: MM\_ANY

Merge Rule Type. Possible values are:MM\_ANY,MM\_MATCH\_ID,MM\_RANK,MM\_SEQUENCE,MM\_MIN\_MAX,MM\_COPY,MM\_CUSTOM,MM\_RECORD\_ANY,MM\_RECORD\_RANK,MM\_RECORD\_MIN\_MAX,MM\_RECORD\_CUSTOM.

Properties for MATCHMERGE OPERATOR MATCH\_RULES CHILD ADDRESS\_ROLES CHILD:

Name: ATTRIBUTE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

What is the attribute for this role?

Name: TYPE

Type: STRING

Valid Values: MM\_PRIMARY\_ADDR, MM\_UNIT\_NUM, MM\_PO\_BOX, MM\_DUAL\_PRIMARY\_ADDR, MM\_DUAL\_UNIT\_NUM, MM\_DUAL\_PO\_BOX, MM\_CITY, MM\_STATE, MM\_POSTAL\_CODE, MM\_IS\_FOUND

Default: MM\_PRIMARY\_ADDR

What role is this attribute? Possible values are: MM\_PRIMARY\_ADDR, MM\_UNIT\_NUM, MM\_PO\_BOX, MM\_DUAL\_PRIMARY\_ADDR, MM\_DUAL\_UNIT\_NUM, MM\_DUAL\_PO\_BOX, MM\_CITY, MM\_STATE, MM\_POSTAL\_CODE, MM\_IS\_FOUND.

Properties for MATCHMERGE OPERATOR MATCH\_RULES CHILD CONDITIONS CHILD:

Name: ALGORITHM

Type: STRING

Valid Values: MM\_EXACT, MM\_STD\_EXACT, MM\_SOUNDEX, MM\_SIMILARITY, MM\_STD\_SIMILARITY, MM\_PARTIAL\_NAME, MM\_ABBREVIATION, MM\_ACRONYM

Default: MM\_EXACT

Algorithm of this condition. Possible values are: MM\_EXACT, MM\_STD\_EXACT, MM\_SOUNDEX, MM\_SIMILARITY, MM\_STD\_SIMILARITY, MM\_PARTIAL\_NAME, MM\_ABBREVIATION, MM\_ACRONYM.

Name: ATTRIBUTE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

To which attribute does this condition apply?

Name: BLANK\_MATCHING

Type: STRING

Valid Values: MM\_MATCH\_BOTH\_BLANK, MM\_MATCH\_EITHER\_BLANK, MM\_NO\_MATCH\_IF\_BLANK

Default: MM\_MATCH\_BOTH\_BLANK

How do you want blanks to be handled? Possible values are: MM\_MATCH\_BOTH\_BLANK, MM\_MATCH\_EITHER\_BLANK, MM\_NO\_MATCH\_IF\_BLANK.

Name: SIMILARITY\_SCORE

Type: NUMBER

Valid Values: N/A

Default: N/A

If this is a similarity condition, what score must the similarity equal or exceed for records to meet the condition?

Properties for MATCHMERGE OPERATOR MATCH\_RULES CHILD FIRM\_ROLES CHILD:

Name: ATTRIBUTE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

What is the attribute for this role?

Name: TYPE

Type: STRING

Valid Values: MM\_FIRM1, MM\_FIRM2

Default: MM\_FIRM1

What role is this attribute? Possible values are: MM\_FIRM1,MM\_FIRM2.

Properties for MATCHMERGE OPERATOR MATCH\_RULES CHILD PERSON\_ROLES CHILD:

Name: ATTRIBUTE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

What is the attribute for this role.

Name: TYPE

Type: STRING

Valid Values: MM\_PRENAME, MM\_FIRST\_NAME\_STD, MM\_MIDDLE\_NAME\_STD, MM\_MIDDLE\_NAME\_2\_STD, MM\_MIDDLE\_NAME\_3\_STD, MM\_LAST\_NAME, MM\_MATURITY\_POST\_NAME

Default: MM\_PRENAME

What role is this attribute? Possible values are: MM\_PRENAME, MM\_FIRST\_NAME\_STD, MM\_MIDDLE\_NAME\_STD, MM\_MIDDLE\_NAME\_2\_STD, MM\_MIDDLE\_NAME\_3\_STD, MM\_LAST\_NAME, MM\_MATURITY\_POST\_NAME.

Properties for MATCHMERGE OPERATOR MATCH\_RULES CHILD WEIGHTS  
CHILD:

Name: BLANK\_SCORE  
Type: NUMBER  
Valid Values: N/A  
Default: N/A  
Score if either attribute is blank

Name: WEIGHT\_ATTRIBUTE  
Type: STRING  
Valid Values: N/A  
Default: N/A  
Attribute

Name: WEIGHT\_SCORE  
Type: NUMBER  
Valid Values: N/A  
Default: N/A  
Score used for this weight if the attributes in the two match records are identical.  
Similarity will be used to generate this score.

Properties for MATCHMERGE OPERATOR MERGE\_RULES CHILD ATTRIBUTES  
CHILD:

Name: ATTRIBUTE\_NAME  
Type: STRING  
Valid Values: N/A  
Default: N/A  
Record merge rule attribute

Properties for MATCHMERGE OPERATOR MERGE\_RULES CHILD RANK\_RULES  
CHILD:

Name: EXPRESSION  
Type: STRING  
Valid Values: N/A

Default: N/A

Rank expression.

Properties for FLAT\_FILE OPERATOR INOUT GROUP:

Name: RECORD\_TYPE\_VALUES

Type: STRING

Valid Values: N/A

Default: N/A

Record Type Values.

Properties for JOINER OPERATOR INPUT GROUP:

Properties for PIVOT OPERATOR OUTPUT GROUP:

Name: ROW\_LOCATOR

Type: STRING

Valid Values: N/A

Default: N/A

An expression indicating which attribute within the output group is the row locator.

Properties for SPLITTER OPERATOR OUTPUT GROUP:

Name: SPLIT\_CONDITION

Type: STRING

Valid Values: N/A

Default: N/A

Condition that defines when to perform the attribute maps for the attributes in this group.

Properties for TABLE\_FUNCTION OPERATOR INPUT GROUP:

Name: INPUT\_PARAMETER\_TYPE

Type: STRING

Valid Values: REF\_CURSOR, SCALAR

Default: REF\_CURSOR

This property specifies whether the input parameter is a scalar or a ref cursor type

Name: PARAMETER\_POSITION

Type: NUMBER

Valid Values: N/A

Default: N/A

The position of the argument in the table function signature corresponding to this parameter group

Properties for TABLE\_FUNCTION OPERATOR OUTPUT GROUP:

Name: RETURN\_TABLE\_OF\_SCALAR

Type: BOOLEAN

Valid Values: true, false

Default: false

This property specifies whether the return of the table function is a TABLE of SCALAR or not.

Properties for UNPIVOT OPERATOR INPUT GROUP:

Name: ROW\_LOCATOR

Type: STRING

Valid Values: N/A

Default: N/A

An expression indicating which attribute within the input group is the row locator.

Name: ROW\_LOCATOR\_VALUES

Type: STRING

Valid Values: N/A

Default: NULL, NULL

A comma-separated expressions that gives the possible values of the row locator within a unpivot group.

Properties for ADVANCED\_QUEUE OPERATOR INOUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A



---

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Properties for CUBE OPERATOR INOUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: LOAD\_COLUMN\_WHEN\_INSERTING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will participate in the insert load operation.

Name: LOAD\_COLUMN\_WHEN\_UPDATING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will participate in the update load operation.

Name: MATCH\_COLUMN\_WHEN\_DELETING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will be used to construct the matching criteria between the incoming data and the existing data on the target during the delete load operation.

Name: MATCH\_COLUMN\_WHEN\_UPDATING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will be used to construct the matching criteria between the incoming data and the existing data on the target during the update load operation.

Name: UPDATE\_OPERATION

Type: STRING

Valid Values: =, +=, -=, =-, =| |, || =

Default: =

The computation to be performed on this attribute between the incoming data and the existing data on the target during the update load operation.

Properties for DIMENSION OPERATOR INOUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: LOAD\_COLUMN\_WHEN\_INSERTING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will participate in the insert load operation.

Name: LOAD\_COLUMN\_WHEN\_UPDATING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will participate in the update load operation.

Name: MATCH\_COLUMN\_WHEN\_DELETING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

---

A boolean value to indicate whether this attribute will be used to construct the matching criteria between the incoming data and the existing data on the target during the delete load operation.

Name: MATCH\_COLUMN\_WHEN\_UPDATING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will be used to construct the matching criteria between the incoming data and the existing data on the target during the update load operation.

Name: UPDATE\_OPERATION

Type: STRING

Valid Values: =, +=, -=, =-, =| |, | | =

Default: =

The computation to be performed on this attribute between the incoming data and the existing data on the target during the update load operation.

Properties for EXTERNAL\_TABLE OPERATOR INOUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Properties for INPUT\_PARAMETER OPERATOR OUTPUT GROUP ATTRIBUTE:

Name: DEFAULT\_VALUE

Type: STRING

Valid Values: N/A

Default: N/A

The default value applies when the map is not given a value for this attribute. The value can be specified only by the calling program.

Properties for KEY\_LOOKUP OPERATOR OUTPUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DEFAULT\_VALUE

Type: STRING

Valid Values: N/A

Default: N/A

The default value used for this attribute if no key lookup table row exists for a given input row.

Properties for MATCHMERGE OPERATOR OUTPUT GROUP ATTRIBUTE:

Name: MERGE\_ATTR

Type: STRING

Valid Values: N/A

Default: N/A

Related merge attribute

Properties for MATERIALIZED\_VIEW OPERATOR INOUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: LOAD\_COLUMN\_WHEN\_INSERTING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will participate in the insert load operation.

Name: LOAD\_COLUMN\_WHEN\_UPDATING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will participate in the update load operation.

Name: MATCH\_COLUMN\_WHEN\_DELETING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will be used to construct the matching criteria between the incoming data and the existing data on the target during the delete load operation.

Name: MATCH\_COLUMN\_WHEN\_UPDATING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will be used to construct the matching criteria between the incoming data and the existing data on the target during the update load operation.

Name: UPDATE\_OPERATION

Type: STRING

Valid Values: =, +=, -=, =-, =| |, || =

Default: =

The computation to be performed on this attribute between the incoming data and the existing data on the target during the update load operation.

Properties for NAME\_AND\_ADDRESS OPERATOR INPUT GROUP ATTRIBUTE:

Name: INPUT\_ROLE

Type: STRING

Valid Values: NA\_NONE, NA\_FIRSTNAME, NA\_MIDDLENAME, NA\_MIDDLENAME2, NA\_MIDDLENAME3, NA\_LASTNAME, NA\_FIRSTPARTNAME, NA\_LASTPARTNAME, NA\_PRENAME, NA\_POSTNAME, NA\_PERSON, NA\_PERSON2, NA\_PERSON3, NA\_FIRMNAME, NA\_PRIMARYADDRESS, NA\_SECONDARYADDRESS, NA\_ADDRESS, NA\_ADDRESS2, NA\_NEIGHBORHOOD, NA\_LASTLINE, NA\_CITY, NA\_STATE, NA\_POSTALCODE, NA\_COUNTRYNAME,

NA\_COUNTRYCODE, NA\_LINE1, NA\_LINE2, NA\_LINE3, NA\_LINE4, NA\_LINE5,  
NA\_LINE6, NA\_LINE7, NA\_LINE8, NA\_LINE9, NA\_LINE10

Default: NA\_NONE

Assigns a name-address input role to the selected input attribute

Properties for NAME\_AND\_ADDRESS OPERATOR OUTPUT GROUP ATTRIBUTE:

Name: ADDRESS\_TYPE

Type: STRING

Valid Values: NA\_ADDRTYPE\_NORMAL, NA\_ADDRTYPE\_DUAL

Default: NA\_ADDRTYPE\_NORMAL

You can designate an address type as Normal or Dual. For example, a dual address occurs when a record contains both a street address and a P.O. Box; this is common with business data. A normal address contains only one type of address.

Name: INSTANCE

Type: STRING

Valid Values: NA\_INSTANCE\_FIRST, NA\_INSTANCE\_SECOND, NA\_INSTANCE\_THIRD,  
NA\_INSTANCE\_FOURTH, NA\_INSTANCE\_FIFTH, NA\_INSTANCE\_SIXTH

Default: NA\_INSTANCE\_FIRST

The instance option is used when an address contains multiple names, you can specify which name in the group should be used. In addition, you can use this option to assign an address type to a miscellaneous address component.

Name: OUTPUT\_COMPONENT

Type: STRING

Valid Values: NA\_NONE, NA\_PRENAME, NA\_FIRSTNAMESTD, NA\_MIDDLENAMESTD,  
NA\_MIDDLENAME2STD, NA\_MIDDLENAME3STD, NA\_POSTNAME, NA\_OTHERPOSTNAME,  
NA\_NAMEDESIGNATOR, NA\_RELATIONSHIP, NA\_PERSON, NA\_FIRSTNAME, NA\_MIDDLENAME,  
NA\_MIDDLENAME2, NA\_MIDDLENAME3, NA\_LASTNAME, NA\_GENDER, NA\_PERSONCOUNT,  
NA\_FIRMNAME, NA\_FIRMCOUNT, NA\_ADDRESS, NA\_PRIMARYADDRESS, NA\_STREETNUMBER,  
NA\_PREDIRECTIONAL, NA\_STREETNAME, NA\_STREETTYPE, NA\_POSTDIRECTIONAL,  
NA\_SECONDARYADDRESS, NA\_UNITDESIGNATOR, NA\_UNITNUMBER, NA\_BOXNAME,  
NA\_BOXNUMBER, NA\_ROUTENAME, NA\_ROUTENUMBER, NA\_BUILDINGNAME, NA\_COMPLEX,  
NA\_MISCADDRESS, NA\_LASTLINE, NA\_NEIGHBORHOOD, NA\_CITY, NA\_STATE, NA\_POSTALCODE,  
NA\_POSTALCODEFORMATTED, NA\_DELIVERYPOINT, NA\_COUNTRYCODE, NA\_COUNTRYCODE3,  
NA\_COUNTRYNAM, NA\_ISGOODGROUP, NA\_ISPARSED, NA\_PARSESTATUS, NA\_PARSESTATUSDESC,  
NA\_ISGOODNAME, NA\_NAMEWARNING, NA\_ISGOODADDRESS, NA\_ISFOUND, NA\_CITYMATCH,  
NA\_STREETNAMEMATCH, NA\_STREETNUMBERMATCH, NA\_STREETCOMPMATCH, NA\_NONAMBIGUOUSMATCH,  
NA\_CITYWARNING, NA\_STREETWARNING, NA\_ISADDRESSVERIFIABLE, NA\_ADDRESSCORRECTED,

NA\_POSTALCODECORRECTED, NA\_CITYCORRECTED, NA\_STREETCORRECTED, NA\_STREETCOMPCORRECTED, NA\_ADDRESSTYPE, NA\_PARSINGCOUNTRY, NA\_INSTALLATIONTYPE, NA\_INSTALLATIONNAME, NA\_DELIVERYOFFICECODE, NA\_DELIVERYBEATCODE, NA\_ADDRESS2, NA\_LOCALITYCODE, NA\_LOCALITYNAME, NA\_COUNTYNAME, NA\_ZIP5, NA\_ZIP4, NA\_URBANIZATIONNAME, NA\_LACS, NA\_CART, NA\_CHECKDIGIT, NA\_MSA, NA\_MCD, NA\_LATITUDE, NA\_LONGITUDE, NA\_FIPSCOUNTY, NA\_FIPS, NA\_CENSUSID

Default: NA\_NONE

Assigns a Name and Address output component to the selected output attribute.

Properties for PIVOT OPERATOR INPUT GROUP ATTRIBUTE:

Name: GROUP\_KEY

Type: BOOLEAN

Valid Values: true, false

Default: false

A boolean value to indicate whether this input attribute is a part of the pivot group key.

Properties for PIVOT OPERATOR OUTPUT GROUP ATTRIBUTE:

Name: GROUP\_KEY

Type: BOOLEAN

Valid Values: true, false

Default: false

A boolean value to indicate whether this output attribute is a part of the pivot group key, which obtains its value from its corresponding input attribute.

Name: PIVOT\_EXPRESSION

Type: STRING

Valid Values: N/A

Default: N/A

A comma-separated expression that gives the input attribute to be used for each output row in the pivot group.

Properties for POSTMAPPING\_PROCESS OPERATOR INPUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DEFAULT\_VALUE

Type: STRING

Valid Values: N/A

Default: N/A

The Default Value for the function input parameter

Name: IS\_OPTIONAL

Type: BOOLEAN

Valid Values: true, false

Default: N/A

If true, the input is not required to be connected

Properties for POSTMAPPING\_PROCESS OPERATOR OUTPUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: FUNCTION\_RETURN

Type: BOOLEAN

Valid Values: true, false

Default: N/A

Specifies whether this output is the return value of this function

Properties for PREMAPPING\_PROCESS OPERATOR INPUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A



Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DEFAULT\_VALUE

Type: STRING

Valid Values: N/A

Default: N/A

The Default Value for the function input parameter

Name: IS\_OPTIONAL

Type: BOOLEAN

Valid Values: true, false

Default: N/A

If true, the input is not required to be connected

Properties for PREMAPPING\_PROCESS OPERATOR OUTPUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: FUNCTION\_RETURN

Type: BOOLEAN

Valid Values: true, false

Default: N/A

Specifies whether this output is the return value of this function

Properties for TABLE OPERATOR INOUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: LOAD\_COLUMN\_WHEN\_INSERTING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will participate in the insert load operation.

Name: LOAD\_COLUMN\_WHEN\_UPDATING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will participate in the update load operation.

Name: MATCH\_COLUMN\_WHEN\_DELETING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will be used to construct the matching criteria between the incoming data and the existing data on the target during the delete load operation.

Name: MATCH\_COLUMN\_WHEN\_UPDATING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will be used to construct the matching criteria between the incoming data and the existing data on the target during the update load operation.

Name: UPDATE\_OPERATION

Type: STRING

Valid Values: =, +=, -=, =-, =|, |=

Default: =

The computation to be performed on this attribute between the incoming data and the existing data on the target during the update load operation.

---

Properties for TABLE\_FUNCTION OPERATOR INPUT GROUP ATTRIBUTE:

Name: PARAMETER\_POSITION

Type: NUMBER

Valid Values: N/A

Default: N/A

The position of the argument in the table function signature corresponding to this parameter

Properties for TABLE\_FUNCTION OPERATOR OUTPUT GROUP ATTRIBUTE:

Name: TYPE\_ATTRIBUTE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name of the field of the PLS Record or attribute of the Object Type or column of the ROWTYPE that corresponds to this attribute. This property is not applicable if the return type is TABLE or SCALAR.

Properties for TRANSFORMATION OPERATOR INPUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DEFAULT\_VALUE

Type: STRING

Valid Values: N/A

Default: N/A

The Default Value for the function input parameter

Name: IS\_OPTIONAL

Type: BOOLEAN

Valid Values: true, false

Default: N/A

If true, the input is not required to be connected

Properties for TRANSFORMATION OPERATOR OUTPUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: FUNCTION\_RETURN

Type: BOOLEAN

Valid Values: true, false

Default: N/A

Specifies whether this output is the return value of this function

Properties for TRANSFORMATION OPERATOR INOUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Properties for UNPIVOT OPERATOR INPUT GROUP ATTRIBUTE:

Name: GROUP\_KEY

Type: BOOLEAN

Valid Values: true, false

Default: false

A boolean value to indicate whether this input attribute is a part of the unpivot group key.

Properties for UNPIVOT OPERATOR OUTPUT GROUP ATTRIBUTE:

Name: GROUP\_KEY

Type: BOOLEAN

Valid Values: true, false

Default: false

A boolean value to indicate whether this output attribute is a part of the unpivot group key, which obtains its value from its corresponding input attribute.

Name: MATCHING\_ROW

Type: NUMBER

Valid Values: 1 - 1000

Default: 1

An positive integer to indicate from which row within the unpivot group this output attribute obtains its data.

Name: UNPIVOT\_EXPRESSION

Type: STRING

Valid Values: N/A

Default: NULL

An expression that gives the input attribute to be used as the output of this attribute.

Properties for VIEW OPERATOR INOUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: LOAD\_COLUMN\_WHEN\_INSERTING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will participate in the insert load operation.

Name: LOAD\_COLUMN\_WHEN\_UPDATING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will participate in the update load operation.

Name: MATCH\_COLUMN\_WHEN\_DELETING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will be used to construct the matching criteria between the incoming data and the existing data on the target during the delete load operation.

Name: MATCH\_COLUMN\_WHEN\_UPDATING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will be used to construct the matching criteria between the incoming data and the existing data on the target during the update load operation.

Name: UPDATE\_OPERATION

Type: STRING

Valid Values: =, +=, -=, =|, ||, ||=

Default: =

The computation to be performed on this attribute between the incoming data and the existing data on the target during the update load operation.

Note:

1. N/A means any valid character in supported character set.
2. " " represents an empty string

propertyValue

A single property value. It can be a number, float, boolean or single-quoted string.

bindableLocator

Location of the object to be bound to a mapping operator or mapping attribute.

mappableBottomUpLocator

Location of the object to be bound to a mapping mapping operator or mapping

attribute.

attributeNameList

A list of attribute names.

bindableType

Type of object bound to a mapping operator or mapping attribute.

bindableName

Name of the object bound to a mapping operator or mapping attribute.

## Examples

```
OMBCREATE MAPPING 'MAP1'
```

```
OMBCREATE MAPPING 'MAP1'
```

```
SET PROPERTIES (business_name, description)
```

```
VALUES ('My map', 'Map to load customer look up table')
```

```
ADD TABLE OPERATOR 'CUST_SRC'
```

```
BOUND TO TABLE '../SRC_MODULE/CUST_SRC'
```

```
ADD TABLE OPERATOR 'CUST_LOOK_UP'
```

```
ADD CONNECTION FROM GROUP 'INOUTGRP1' OF OPERATOR 'CUST_SRC'
```

```
TO GROUP 'INOUTGRP1' OF OPERATOR 'CUST_LOOK_UP'
```

## See Also

OMBCREATE, OMBALTER MAPPING, OMBRETRIEVE MAPPING, OMBDROP MAPPING

# OMBCREATE MATERIALIZED\_VIEW

## Purpose

OMBCREATE MATERIALIZED\_VIEW - To create a materialized view.

## Prerequisites

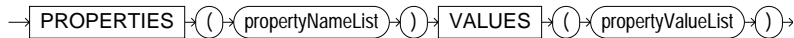
In the context of an Oracle Module.

## Syntax Diagrams

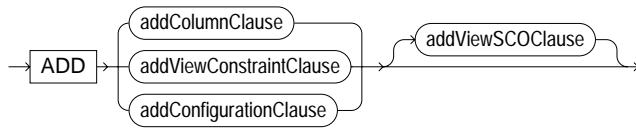
createMaterializedViewCommand



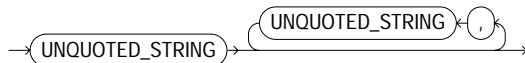
setPropertyClause



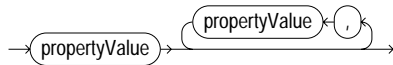
addViewSCOClauses



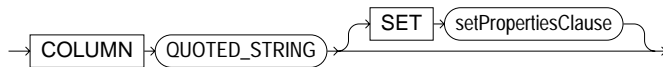
propertyNameList



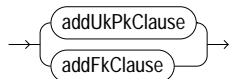
propertyValueList



addColumnClause

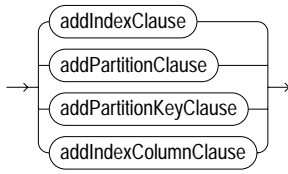


addViewConstraintClause

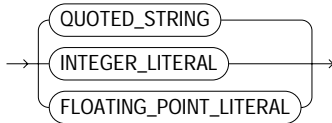




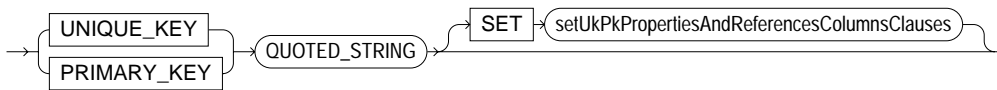
addConfigurationClause



property Value



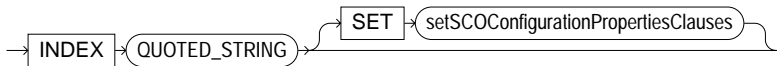
addUkPkClause



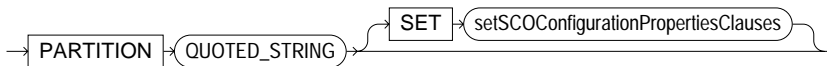
addFkClause



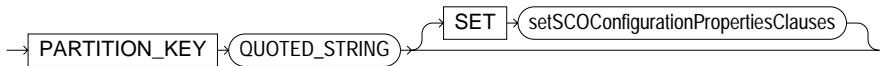
addIndexClause



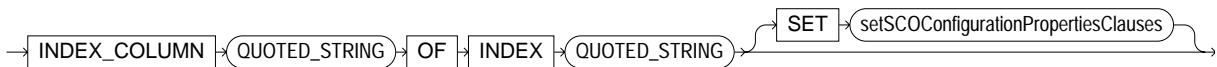
addPartitionClause



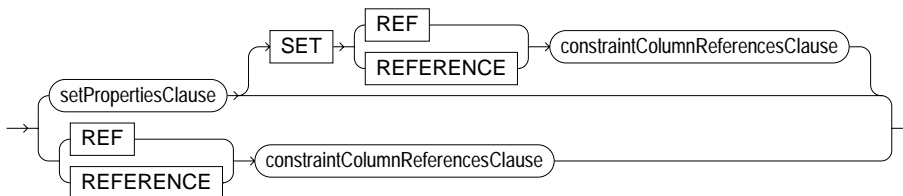
addPartitionKeyClause



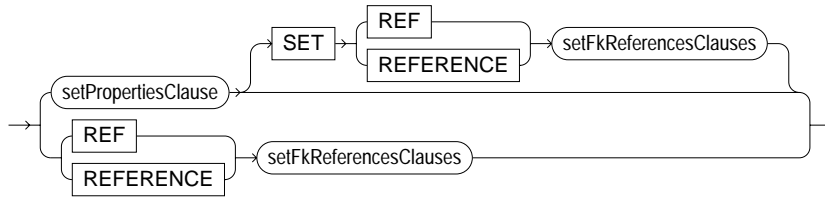
addIndexColumnClause



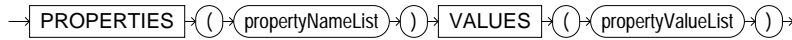
setUkPkPropertiesAndReferencesColumnsClauses



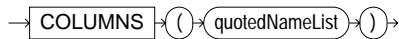
setFkSubClauses



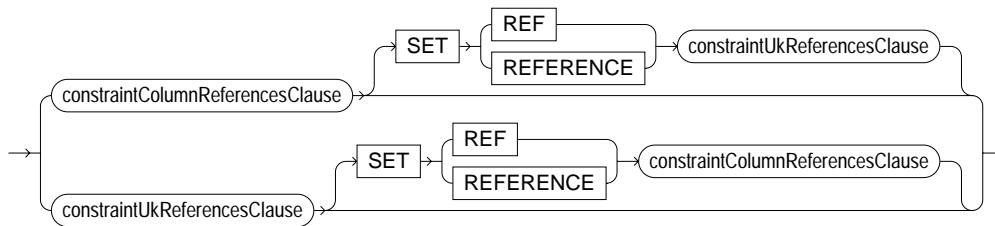
setSCOConfigurationPropertiesClauses



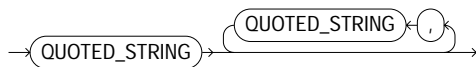
constraintColumnReferencesClause



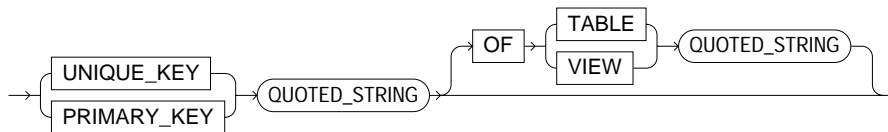
setFkReferencesClauses



quotedNameList



constraintUkReferencesClause



## Syntax

```
createMaterializedViewCommand = OMBCREATE ( MATERIALIZED_VIEW
"QUOTED_STRING" [ SET "setPropertiesClause" ] [ "addViewSCOClause" ] );
setPropertiesClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("
"propertyValueList" ")";
addViewSCOClause = ADD ( "addColumnClause" | "addViewConstraintClause" |
"addConfigurationClause" ) [ "addViewSCOClause" ];
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
propertyValueList = "propertyValue" { "," "propertyValue" };
addColumnClause = COLUMN "QUOTED_STRING" [ SET "setPropertiesClause" ];
addViewConstraintClause = "addUkPkClause" | "addFkClause";
addConfigurationClause = "addIndexClause" | "addPartitionClause" |
"addPartitionKeyClause" | "addIndexColumnClause";
```

```

propertyValue = ("QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL");
addUkPkClause = ( UNIQUE_KEY | PRIMARY_KEY ) "QUOTED_STRING" [ SET
"setUkPkPropertiesAndReferencesColumnsClauses" ];
addFkClause = FOREIGN_KEY "QUOTED_STRING" [ SET "setFkSubClauses" ];
addIndexClause = INDEX "QUOTED_STRING" [ SET
"setSCOConfigurationPropertiesClauses" ];
addPartitionClause = PARTITION "QUOTED_STRING" [ SET
"setSCOConfigurationPropertiesClauses" ];
addPartitionKeyClause = PARTITION_KEY "QUOTED_STRING" [ SET
"setSCOConfigurationPropertiesClauses" ];
addIndexColumnClause = INDEX_COLUMN "QUOTED_STRING" OF INDEX
"QUOTED_STRING" [ SET "setSCOConfigurationPropertiesClauses" ];
setUkPkPropertiesAndReferencesColumnsClauses = "setPropertiesClause" [ SET (
REF | REFERENCE ) "constraintColumnReferencesClause" ] | ( REF | REFERENCE )
"constraintColumnReferencesClause";
setFkSubClauses = "setPropertiesClause" [ SET ( REF | REFERENCE )
"setFkReferencesClauses" ] | ( REF | REFERENCE ) "setFkReferencesClauses";
setSCOConfigurationPropertiesClauses = PROPERTIES "(" "propertyNameList" ")"
VALUES "(" "propertyValueList" ")";
constraintColumnReferencesClause = COLUMNS "(" "quotedNameList" ")";
setFkReferencesClauses = "constraintColumnReferencesClause" [ SET ( REF |
REFERENCE ) "constraintUkReferencesClause" ] | "constraintUkReferencesClause" [
SET ( REF | REFERENCE ) "constraintColumnReferencesClause" ];
quotedNameList = "QUOTED_STRING" { "," "QUOTED_STRING" };
constraintUkReferencesClause = ( UNIQUE_KEY | PRIMARY_KEY ) "QUOTED_
STRING" [ OF ( TABLE | VIEW ) "QUOTED_STRING" ];

```

## Keywords and Parameters

createMaterializedViewCommand

This command creates a materialized view.

QUOTED\_STRING

Specify the name of the materialized view to be created.

setPropertiesClause

Used to set properties (core, logical, physical, user-defined) for tables, columns, unique keys, foreign keys, primary keys, and check constraints. Valid properties are shown below:

Basic properties for VIEW, MATERIALIZED\_VIEW:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the View, MaterializedView

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the View, MaterializedView

Name: VIEW\_QUERY

Type: STRING(4000)

Valid Values: N/A

Default: "

Sets the query definition in View and MaterializedView.

Name: UOID

Type: STRING

Valid Values: N/A

Default: N/A

Basic properties for INDEX, PARTITION, PARTITION\_KEY, INDEX\_COLUMN:

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the Index, Partition, PartitionKey, IndexColumn in a MaterializedView.

Properties for MATERIALIZED\_VIEW:

Name: BASE\_TABLES

Type: STRING

Valid Values: N/A

Default: "

---

Comma separated list of base tables.

Name: BUILD

Type: STRING

Valid Values: DEFERRED, IMMEDIATE

Default: IMMEDIATE

Immediate : populates the view when it is created. Deferred : delays population until the next refresh operation.

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: HASH\_PARTITION\_TABLESPACE\_LIST

Type: STRING

Valid Values: N/A

Default: "

A comma separated list of tablespaces to use for [sub]partition storage.

Name: HASH\_SUBPARTITION\_NUMBER

Type: NUMBER

Valid Values: 2 - 63999

Default: 2

Hash SubPartition Number

Name: LOGGING\_MODE

Type: STRING

Valid Values: NOLOGGING, LOGGING

Default: LOGGING

Recovery requirements for a data warehouse : Logging or not logging to Redo Log File.

Name: PARALLEL\_ACCESS\_MODE

Type: STRING

Valid Values: NOPARALLEL, PARALLEL

Default: PARALLEL

Enables or disables parallel processing when the table is created. Also enables or disables parallel processing or access.

Name: QUERY\_REWRITE

Type: STRING

Valid Values: DISABLE, ENABLE

Default: ENABLE

Enable marks the View eligible for query rewrite and disable marks the View ineligible for query rewrite

Name: REFRESH

Type: STRING

Valid Values: COMPLETE, FAST, ON\_COMMIT, ON\_DEMAND, FORCE

Default: COMPLETE

Complete : specifies the complete refresh method implemented by executing the query of the view. Fast : specifies the incremental refresh method which refreshes the view according to changes that have occurred to the master tables. Force : specifies that when a refresh occurs, Oracle performs a fast refresh if possible or a complete refresh otherwise.

Name: TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Use the Tablespace parameter to specify the name of tablespace.

Properties for UNIQUE\_KEY:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: USING\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

Use of Index on Dimension Constraint

Properties for FOREIGN\_KEY:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: USING\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

Use of Index on Dimension Constraint

Properties for CHECK\_CONSTRAINT:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: USING\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

Use of Index on Dimension Constraint

Properties for INDEX:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: INDEX\_TYPE

Type: STRING

Valid Values: BITMAP, UNIQUE, NO\_INDEX

Default: UNIQUE

The types of Indexes created on Dimension are BITMAP, UNIQUE or a non-specific index.

Name: LOCAL\_INDEX

Type: BOOLEAN

Valid Values: true, false



Default: false

A local index is constructed so that it reflects the structure of the underlying table. It is equipartitioned with the underlying table, meaning that it is partitioned on the same columns as the underlying table, creates the same number of partitions or subpartitions, and gives them the same partition bounds as corresponding partitions of the underlying table.

Name: LOGGING\_MODE

Type: STRING

Valid Values: NOLOGGING, LOGGING

Default: LOGGING

Recovery requirements for a data warehouse : Logging or not logging to Redo Log File.

Name: PARALLEL\_ACCESS\_MODE

Type: STRING

Valid Values: NOPARALLEL, PARALLEL

Default: PARALLEL

Enables or disables parallel processing when the table is created. Also enables or disables parallel processing or access.

Name: TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Use the Tablespace parameter to specify the name of tablespace.

Name: USING\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

Use of Index on Dimension Constraint

Properties for PARTITION:

Name: DATE\_LESS\_THAN

Type: STRING

Valid Values: N/A

Default: TO\_DATE('01-JAN-1999', 'DD-MON-YYYY')

Value that represents upper bound of partition stored in warehouse key column for the Days Dimension.

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: EMPTY\_STRING

Use the Tablespace parameter to specify the name of tablespace.

Name: VALUES\_LESS\_THAN

Type: STRING

Valid Values: N/A

Default: TO\_DATE('01-JAN-1999', 'DD-MON-YYYY')

Noninclusive upper bound for the current partition. Type a comma separated list in the same order as the columns in range partitioned key.

Name: VALUE\_LESS\_THAN

Type: STRING

Valid Values: N/A

Default: "

Noninclusive upper bound for the current partition. Type a comma separated list in the same order as the columns in range partitioned key.

Properties for PARTITION\_KEY:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: TYPE

Type: STRING

Valid Values: HASH, RANGE

Default: RANGE

Oracle partitions the storage space and stores rows according to a Hash Algorithm or specified ranges.

Note:

1. N/A means any valid character in supported character set.
2. " " represents an empty string

addViewSCOClause

This clause adds components like column etc.

propertyNameList

The list of properties.

propertyValueList

The list of property values.

addColumnClause

This column adds a column.

addViewConstraintClause

This clause adds the view's configuration clause.

addConfigurationClause

This clause will add configuration objects.

propertyValue

This clause adds the property values.

addUkPkClause

This clause adds the adds unique key and primary keys.

QUOTED\_STRING

name of the unique key or primary key.

**addFkClause**

This clause adds foreign key

**QUOTED\_STRING**

Name of the foreign key.

**addIndexClause**

This clause adds an index.

**QUOTED\_STRING**

Name of the index.

**addPartitionClause**

This clause adds a partition.

**QUOTED\_STRING**

Name of the partition.

**addPartitionKeyClause**

This clause adds a partition key.

**QUOTED\_STRING**

Name of the partition key. This should be a column identifier.

**addIndexColumnClause**

This clause will add indexColumn to a specified index.

**QUOTED\_STRING**

Index name

**setUkPkPropertiesAndReferencesColumnsClauses**

This clause adds properties and references to columns

**setFkSubClauses**

This clause set references to a foreign key.

**setSCOConfigurationPropertiesClauses**

Set the configuration properties for the following objects Index: LOGGING\_MODE, PARALLEL\_ACCESS\_MODE, TABLESPACE, INDEX\_TYPE, LOCAL\_INDEX, DEPLOYABLE Partition: DATE\_LESS\_THAN, TABLESPACE, DEPLOYABLE Partition\_key: TYPE, DEPLOYABLE  
 RelationalCmdParser\$constraintColumnReferencesClause = This clause provides names of all columns.

**constraintColumnReferencesClause**

RelationalCmdParser\$constraintColumnReferencesClause??

**setFkReferencesClauses**

This clause sets foreign key references.

**quotedNameList**

This clause gives column names.

**constraintUkReferencesClause**

The first QUOTED\_STRING denotes the UniqueKey or Primary key name, and the latter denotes the table's or view's name.

**Examples**

```
OMBCREATE MATERIALIZED_VIEW 'NEW_MATERIALIZED_VIEW' SET
PROPERTIES (DESCRIPTION, BUSINESS_NAME) VALUES ('this is a new
materialized view', 'New MaterializedView') This will create a materialized view
named "NEW_VIEW", its description is "this is a new materialized view", and
business name is "New MaterializedView".
```

**See Also**

OMBCREATE, OMBALTER MATERIALIZED\_VIEW, OMBDROP MATERIALIZED\_VIEW

# OMBCREATE OBJECT\_TYPE

## Purpose

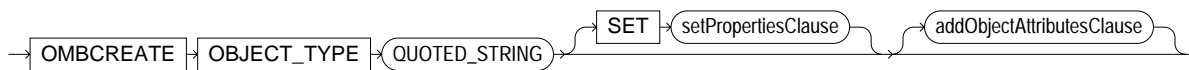
OMBCREATE OBJECT\_TYPE - To create an Object Type.

## Prerequisites

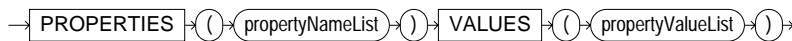
Should be in the context of an Oracle Module.

## Syntax Diagrams

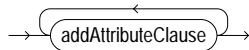
createObjectTypeCommand



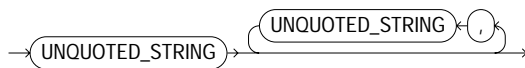
setPropertiesClause



addObjectAttributesClause



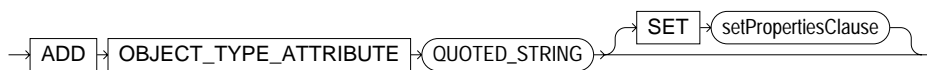
propertyNameList



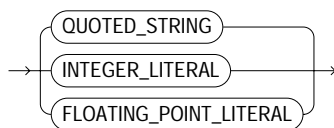
propertyValueList



addAttributeClause



propertyValue



## Syntax

```
createObjectTypeCommand = OMBCREATE ( OBJECT_TYPE "QUOTED_STRING" [
SET "setPropertiesClause" ] [ "addObjectAttributesClause" ] );
```

```
setPropertiesClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("
"propertyValueList" ")";
```

```

addObjectAttributesClause = "addAttributeClause"+;
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
propertyValueList = "propertyValue" { "," "propertyValue" };
addAttributeClause = ADD OBJECT_TYPE_ATTRIBUTE "QUOTED_STRING" [ SET
"setPropertyClause" ];
propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );

```

## Keywords and Parameters

**createObjectTypeCommand**

Creates an Object Type with the given name.

**setPropertyClause**

Sets properties (core, logical, physical, user-defined) for Object Type or its Attributes. Valid properties are shown below:

Basic properties for OBJECT\_TYPE:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the Object Type

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the Object Type

Basic properties for OBJECT\_TYPE\_ATTRIBUTE:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the Attribute

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the Attribute

Name: DATATYPE

Type: STRING(20)

Valid Values: NUMBER, VARCHAR2, VARCHAR, DATE, FLOAT

Default: "

Datatype of the Attribute

Note:

1. N/A means any valid character in supported character set.

2. " represents an empty string

`addObjectAttributesClause`

Adds one or more Attributes in this Object Type.

`propertyNameList`

The list of properties.

`propertyValueList`

The list of property values.

`addAttributeClause`

Adds an Attribute with the given name and properties.

`propertyValue`

This clause adds the property values.

## Examples

```
OMBCREATE OBJECT_TYPE 'NEW_OBJECT_TYPE' SET PROPERTIES
(DESCRIPTION) VALUES
```

```
('this is an object type') ADD OBJECT_TYPE_ATTRIBUTE 'ATTR' SET PROPERTIES
(DATATYPE) VALUES ('VARCHAR2') This will create an Object Type named "NEW_
OBJECT_TYPE", its description is "this is an object type" and an Attribute 'ATTR' of
Varchar2 type.
```

## See Also

OMBCREATE, OMBALTER OBJECT\_TYPE, OMBDROP OBJECT\_TYPE



# OMBCREATE ORACLE\_MODULE

## Purpose

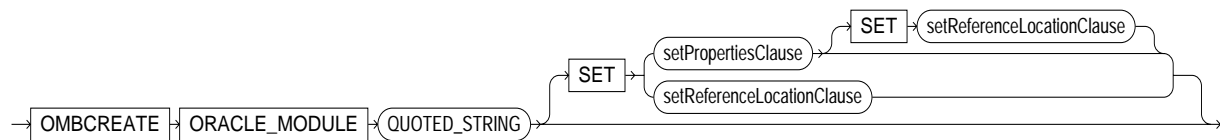
OMBCREATE ORACLE\_MODULE - To create an Oracle module.

## Prerequisites

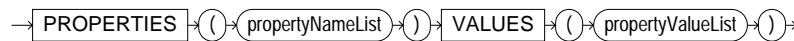
Should be in the context of project.

## Syntax Diagrams

createOracleModuleCommand



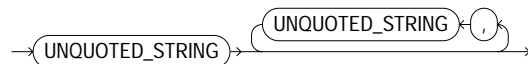
setPropertiesClause



setReferenceLocationClause



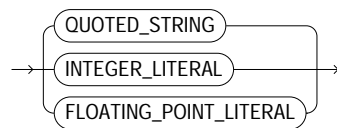
propertyNameList



propertyValueList



propertyValue



## Syntax

```
createOracleModuleCommand = OMBCREATE ( ORACLE_MODULE "QUOTED_
STRING" [ SET ( "setPropertiesClause" [ SET "setReferenceLocationClause" ] |
"setReferenceLocationClause" ) ] );
```

```
setPropertiesClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("
"propertyValueList" ")";
```

```

setReferenceLocationClause = ( REFERENCE | REF ) LOCATION "QUOTED_
STRING";
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
propertyValueList = "propertyValue" { "," "propertyValue" };
propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );

```

## Keywords and Parameters

createOracleModuleCommand

This command creates an Oracle module

QUOTED\_STRING

Name of the Oracle module to be created.

setPropertyClause

Associate a set of properties with an Oracle module.

Basic properties for ORACLE\_MODULE:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: NAME

Business name of an Oracle Module

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of an Oracle Module

Name: UOID

Type: STRING(40)

Valid Values: N/A

Default: N/A

UOID of an Oracle Module

Properties for ORACLE\_MODULE:

Name: ABAP\_DIRECTORY  
Type: STRING  
Valid Values: N/A  
Default: abap\  
Location where ABAP scripts are stored

Name: ABAP\_EXTENSION  
Type: STRING  
Valid Values: N/A  
Default: .abap  
File name extension for ABAP scripts

Name: ABAP\_RUN\_PARAMETER\_FILE  
Type: STRING  
Valid Values: N/A  
Default: \_run.ini  
Run Parameter File Suffix for the parameter script in a ABAP job.

Name: ABAP\_SPOOL\_DIRECTORY  
Type: STRING  
Valid Values: N/A  
Default: abap\log\  
Location where ABAP scripts are buffered during script generation processing.

Name: APPLICATION\_SHORT\_NAME  
Type: STRING  
Valid Values: N/A  
Default: WB  
Application Short Name

Name: ARCHIVE\_DIRECTORY  
Type: STRING  
Valid Values: N/A  
Default: archive\  
Archive Directory

Name: CONNECT\_STRING

Type: STRING

Valid Values: N/A

Default: "

A Net\*8 style connection string to the remote database. Alternatively, you can specify machine, port, service name of the remote database.

Name: DDL\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: ddl\

Location where scripts for database objects for the target schema are stored.

Name: DDL\_EXTENSION

Type: STRING

Valid Values: N/A

Default: .ddl

File name extension for DDL scripts.

Name: DDL\_SPOOL\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: ddl\log\

Location where DDL scripts are buffered during script generation processing.

Name: DEFAULT\_INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Default name of tablespace to install indexes into.

Name: DEFAULT\_OBJECT\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Default name of tablespace to install objects into.

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: END\_OF\_LINE

Type: STRING

Valid Values: N/A

Default: \r\n

End of Line

Name: INPUT\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: input\

Input Directory

Name: INVALID\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: invalid\

Directory for SQL\*Loader errors and rejected records

Name: LIB\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: lib\

LIB Directory

Name: LIB\_EXTENSION

Type: STRING

Valid Values: N/A

Default: .lib

LIB Extension

Name: LIB\_SPOOL\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: lib\log\

LIB Spool Directory

Name: LOADER\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: ctl\

Location where control files are stored.

Name: LOADER\_EXTENSION

Type: STRING

Valid Values: N/A

Default: .ctl

Suffix for the loader scripts

Name: LOADER\_RUN\_PARAMETER\_FILE

Type: STRING

Valid Values: N/A

Default: \_run.ini

Suffix for the parameter initialization file.

Name: LOG\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: log\

Log Directory for the SQL\*Loader

Name: MAIN\_APPLICATION\_SHORT\_NAME

Type: STRING

Valid Values: N/A

Default: ora

Main Application Short Name

Name: PLSQL\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: pls\

Location where PL/SQL scripts are stored.

Name: PLSQL\_EXTENSION

Type: STRING

Valid Values: N/A

Default: .pls

File name extension for PL/SQL scripts.

Name: PLSQL\_GENERATION\_MODE

Type: STRING

Valid Values: Oracle9i, Oracle8i

Default: Oracle9i

Generation mode controls validation and generation for version specific features.

Name: PLSQL\_RUN\_PARAMETER\_FILE

Type: STRING

Valid Values: N/A

Default: \_run.ini

Suffix for the parameter script in a PL/SQL job.

Name: PLSQL\_SPOOL\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: pls\log\

Location where PL/SQL scripts are buffered during script generation processing.

Name: PORT

Type: STRING

Valid Values: N/A

Default: "

The port number on the machine where the database listens to.

Name: RECEIVE\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: receive\

Receive Directory

Name: REMOTE\_HOST\_NAME

Type: STRING

Valid Values: N/A

Default: "

The machine where the remote database resides on.

Name: SCHEMA\_OWNER

Type: STRING

Valid Values: N/A

Default: OWB

Schema Owner

Name: SERVICE\_NAME

Type: STRING

Valid Values: N/A

Default: "

The service name (global DB name) of the database instance on the remote machine.

Name: SORT\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: sort\

Sort Directory

Name: TCL\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: tcl\

Location for TCL scripts that are generated after registration with Oracle Enterprise Manager

Name: TOP\_DIRECTORY

Type: STRING



Valid Values: N/A

Default: ..\..\codegen\

Top Directory where generated code will get stored

Name: WORK\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: work\

Work Directory

Note:

1. N/A means any valid character in supported character set.
2. " " represents an empty string

setReferenceLocationClause

Set a location to an Oracle module.

propertyNameList

Comma separated list of property names. Property names are unquoted.

propertyValueList

Comma separated list of property values.

propertyValue

Value of a property.

## Examples

```
OMBCREATE ORACLE_MODULE 'src_module' SET PROPERTIES (DESCRIPTION,
BUSINESS_NAME) VALUES ('this is an Oracle module', 'source module') This will
create an Oracle module named "src_module", its description is "this is an Oracle
module", and business name is "source module".
```

## See Also

OMBCREATE, OMBALTER ORACLE\_MODULE, OMBDROP ORACLE\_MODULE

# OMBCREATE PACKAGE

## Purpose

OMBCREATE PACKAGE - To create a Package.

## Prerequisites

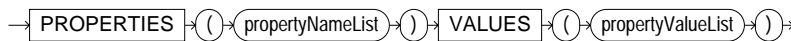
Should be in the context of a Oracle Module or Transformation Module.

## Syntax Diagrams

createPackageCommand



setPropertiesClause



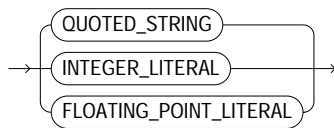
propertyNameList



propertyValueList



propertyValue



## Syntax

```
createPackageCommand = OMBCREATE ( PACKAGE "QUOTED_STRING" [ SET "setPropertiesClause" ] );
```

```
setPropertiesClause = PROPERTIES "(" "propertyNameList" ")" VALUES "(" "propertyValueList" ")";
```

```
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
```

```
propertyValueList = "propertyValue" { "," "propertyValue" };
```

```
propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_POINT_LITERAL" );
```

## Keywords and Parameters

createPackageCommand

This command creates a Package

QUOTED\_STRING

Name of the Package to be created.

setPropertiesClause

Used to set properties (core, user-defined) for packages. Valid properties are shown below:

Basic properties for PACKAGE:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the Package

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the Package

Properties for PACKAGE:

Name: AUTHID

Type: STRING

Valid Values: None, Current\_User, Definer

Default: None

Generate the package with selected AUTHID option. Function will be executed with the permissions defined by the AUTHID clause rather than the function owner's permissions.

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Note:

1. N/A means any valid character in supported character set.
2. " represents an empty string

propertyNameList

Comma separated list of property names. Property names are unquoted.

propertyValueList

Comma separated list of property values.

propertyValue

Value of a property.

## Examples

OMBCREATE PACKAGE 'pkg' SET PROPERTIES (DESCRIPTION, BUSINESS\_NAME) VALUES ('this is a Package', 'package') This will create a Package named "pkg", its description is "this is a Package", and business name is "package".

## See Also

OMBCREATE, OMBALTER PACKAGE, OMBDROP PACKAGE

# OMBCREATE PROCEDURE

## Purpose

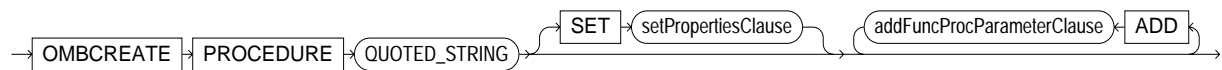
OMBCREATE PROCEDURE - To create a Procedure.

## Prerequisites

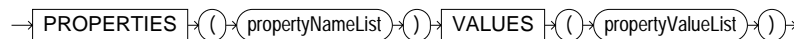
Should be in the context of a Oracle Module or Package or Transformation Module.

## Syntax Diagrams

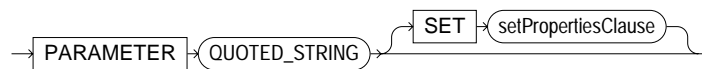
createProcedureCommand



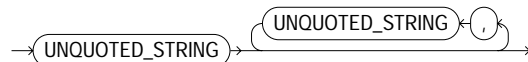
setPropertiesClause



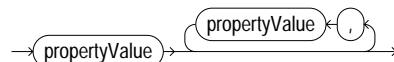
addFuncProcParameterClause



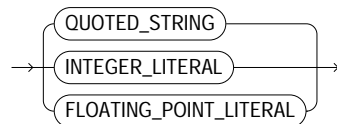
propertyNameList



propertyValueList



propertyValue



## Syntax

```
createProcedureCommand = OMBCREATE ( PROCEDURE "QUOTED_STRING" [
SET "setPropertiesClause" ] { ADD "addFuncProcParameterClause" } );
```

```
setPropertiesClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("
"propertyValueList" )";
```

```
addFuncProcParameterClause = PARAMETER "QUOTED_STRING" [ SET
"setPropertiesClause" ];
```

```
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
```

```
propertyValueList = "propertyValue" { "," "propertyValue" };  
propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_  
POINT_LITERAL" );
```

## Keywords and Parameters

**createProcedureCommand**

This command creates a Procedure

**QUOTED\_STRING**

Name of the Procedure to be created.

**setPropertyClause**

Used to set properties (core, user-defined) for procedurefunction. Valid properties are shown below:

Basic properties for PROCEDURE:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the Procedure

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the Procedure

Name: IMPLEMENTATION

Type: STRING

Valid Values: N/A

Default: "

Set the code for Procedure which is included global variable declaration and code between BEGIN and END.

Basic properties for PARAMETER:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the Parameter

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the Parameter

Name: DATATYPE

Type: STRING

Valid Values: PLS\_INTEGER, BINARY\_INTEGER, BOOLEAN, NUMBER, FLOAT, CHAR, VARCHAR, VARCHAR2, DATE

Default: NUMBER

Set the data type for Parameter

Name: IN\_OUT

Type: STRING

Valid Values: IN, OUT, INOUT

Default: 'IN'

Set the parameter mode for Parameter

Name: DEFAULT\_VALUE

Type: STRING

Valid Values: N/A

Default: "

Set the default value for Parameter

Properties for PROCEDURE:

Name: AUTHID

Type: STRING

Valid Values: None, Current\_User, Definer

Default: None

Generate the transformation with selected AUTHID option. Function will be executed with the permissions defined by the AUTHID clause rather than the function owner's permissions.

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Note:

1. N/A means any valid character in supported character set.

2. " " represents an empty string

`addFuncProcParameterClause`

Adds one or more Parameters to this Procedure.

`propertyNameList`

Comma separated list of property names. Property names are unquoted.

`propertyValueList`

Comma separated list of property values.

`propertyValue`

Value of a property.

## Examples

```
OMBCREATE PROCEDURE 'proc' SET PROPERTIES (DESCRIPTION, BUSINESS_
NAME, IMPLEMENTATION) VALUES ('this is a Procedure', 'proc', 'BEGIN END proc
\;') ADD PARAMETER 'PARAM_1' SET PROPERTIES (DEXCRIPTION, BUSINESS_
NAME, IN_OUT, DATATYPE, DEFAULT_VALUE) VALUES ('param_1', 'this is a
param_1','IN', 'VARCHAR2', 'this is a Varchar2') ADD PARAMETER 'PARAM_2' SET
PROPERTIES (DEXCRIPTION, BUSINESS_NAME, IN_OUT, DATATYPE, DEFAULT_
VALUE) VALUES ('param_2', 'this is a param_2','INOUT', 'DATE', 'this is a Date') This
will create a Procedure named "proc", its description is "this is a Procedure", and
business name is "proc", return datatype NUMBER, and body of function as 'BEGIN
END proc;'. It creates two parameters 'PARAM_1' and 'PARAM_2'
```

## See Also

OMBCREATE, OMBALTER PROCEDURE, OMBDROP PROCEDURE



# OMBCREATE PROCESS\_FLOW

## Purpose

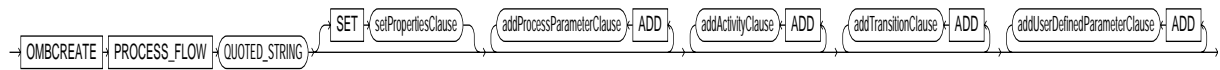
OMBCREATE PROCESS\_FLOW - To create a Process Flow.

## Prerequisites

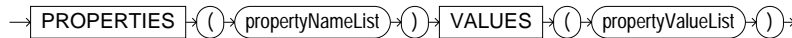
Should be in the context of a Process Flow Package.

## Syntax Diagrams

createProcessFlowCommand



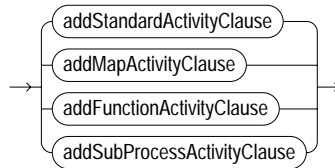
setPropertiesClause



addProcessParameterClause



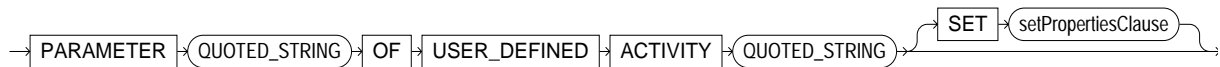
addActivityClause



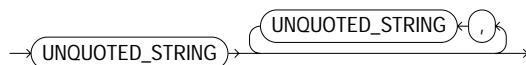
addTransitionClause



addUserDefinedParameterClause



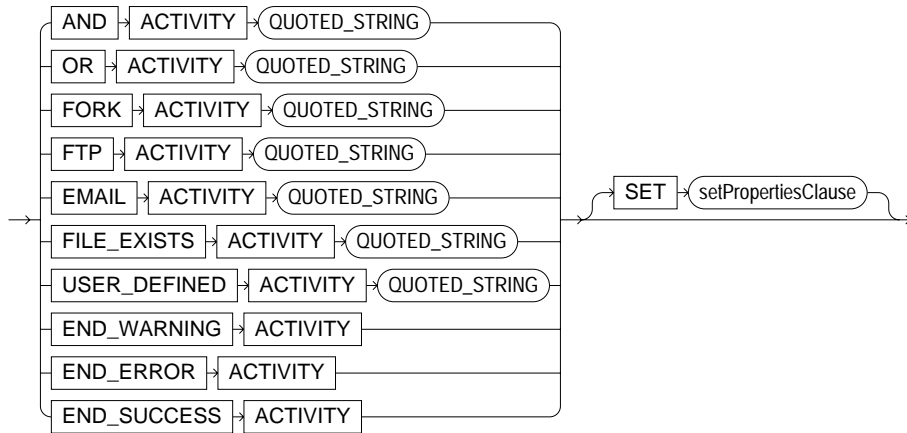
propertyNameList



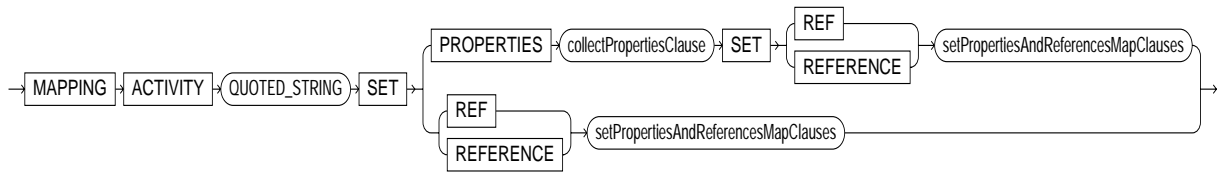
propertyValueList



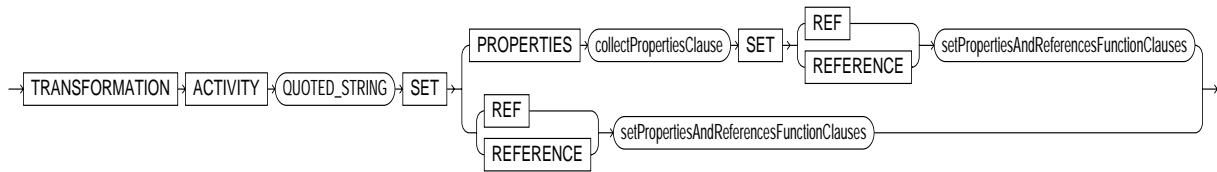
addStandardActivityClause



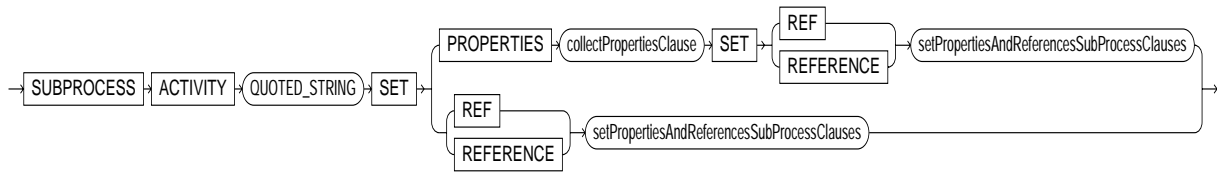
addMapActivityClause



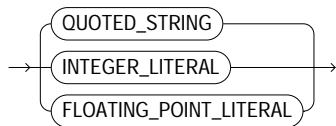
addFunctionActivityClause



addSubProcessActivityClause



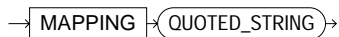
propertyValue



collectPropertiesClause



setPropertiesAndReferencesMapClauses



setPropertiesAndReferencesFunctionClauses



setPropertiesAndReferencesSubProcessClauses



## Syntax

```
createProcessFlowCommand = OMBCREATE ( PROCESS_FLOW "QUOTED_
STRING" [ SET "setPropertiesClause" ] ( { ADD "addProcessParameterClause" } { ADD
"addActivityClause" } { ADD "addTransitionClause" } { ADD
"addUserDefinedParameterClause" } ) );
```

```
setPropertiesClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("
"propertyValueList" )";
```

```
addProcessParameterClause = ( PARAMETER "QUOTED_STRING" ) [ SET
"setPropertiesClause" ];
```

```
addActivityClause = ( "addStandardActivityClause" | "addMapActivityClause" |
"addFunctionActivityClause" | "addSubProcessActivityClause" );
```

```
addTransitionClause = ( TRANSITION "QUOTED_STRING" ( FROM ACTIVITY
"QUOTED_STRING" ) ( TO "QUOTED_STRING" ) ) [ SET "setPropertiesClause" ];
```

```
addUserDefinedParameterClause = ( PARAMETER "QUOTED_STRING" OF USER_
DEFINED ACTIVITY "QUOTED_STRING" ) [ SET "setPropertiesClause" ];
```

```
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
```

```
propertyValueList = "propertyValue" { "," "propertyValue" };
```

```
addStandardActivityClause = ( AND ACTIVITY "QUOTED_STRING" | OR
ACTIVITY "QUOTED_STRING" | FORK ACTIVITY "QUOTED_STRING" | FTP
ACTIVITY "QUOTED_STRING" | EMAIL ACTIVITY "QUOTED_STRING" | FILE_
EXISTS ACTIVITY "QUOTED_STRING" | USER_DEFINED ACTIVITY "QUOTED_
STRING" | END_WARNING ACTIVITY | END_ERROR ACTIVITY | END_
SUCCESS ACTIVITY ) [ SET "setPropertiesClause" ];
```

```
addMapActivityClause = ( MAPPING ACTIVITY "QUOTED_STRING" SET (
PROPERTIES "collectPropertiesClause" SET ( REF | REFERENCE )
"setPropertiesAndReferencesMapClauses" | ( REF | REFERENCE )
"setPropertiesAndReferencesMapClauses" ) );
```

```
addFunctionActivityClause = ( TRANSFORMATION ACTIVITY "QUOTED_STRING"
SET ( PROPERTIES "collectPropertiesClause" SET ( REF | REFERENCE )
"setPropertiesAndReferencesFunctionClauses" | ( REF | REFERENCE )
"setPropertiesAndReferencesFunctionClauses" ) );
```

```
addSubProcessActivityClause = ( SUBPROCESS ACTIVITY "QUOTED_STRING" SET
( PROPERTIES "collectPropertiesClause" SET ( REF | REFERENCE )
"setPropertiesAndReferencesSubProcessClauses" | ( REF | REFERENCE )
"setPropertiesAndReferencesSubProcessClauses" ) );
```

```
propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );
```

```
collectPropertiesClause = "(" "propertyNameList" ")" VALUES "(" "propertyValueList"
)";
```

```
setPropertiesAndReferencesMapClauses = ( MAPPING "QUOTED_STRING" );
```

```

setPropertiesAndReferencesFunctionClauses = ( TRANSFORMATION "QUOTED_
STRING" );
setPropertiesAndReferencesSubProcessClauses = ( PROCESS_FLOW "QUOTED_
STRING" );

```

## Keywords and Parameters

**createProcessFlowCommand**

Create a new process flow using quoted name.

**setPropertiesClause**

Used to set properties (core, user-defined) for process flow. Note: For MAPPING, TRANSFORMATION and SUBPROCESS activities the setPropertiesAndReferencesMapClauses, setPropertiesAndReferencesFunctionClauses and setPropertiesAndReferencesSubProcessClauses respectively, are mandatory. For MAPPING or TRANSFORMATION activities and the REFERENCE property has to be set to a valid MAP or TRANSFORMATION within the current project. For SUBPROCESS activities the REFERENCE property has to be set to a SUBPROCESS within the same PROCESS\_FLOW\_PACKAGE. Valid properties are shown below:

Base properties for PROCESS\_FLOW:

Basic properties for Process Flow, Activity, Transition and Parameter:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the Process Flow

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the Process Flow Core properties for Transition :

Name: TRANSITION\_CONDITION

Type: STRING

Valid Values: ", SUCCESS, ERROR, WARNING

Default: " i.e. Unconditional

Sets the Transition Condition of a Transition

Description of the Process Flow Core properties for Activity Parameter :

Name: DATATYPE

Type: STRING

Valid Values: INTEGER, FLOAT, DATE, STRING, BOOLEAN

Default: STRING

Sets the datatype of a Activity Parameter

Name: DIRECTION

Type: STRING

Valid Values: IN

Default: IN

Sets the direction of a Activity Parameter

Name: VALUE

Type: STRING

Valid Values: Examples '123', '123.456', 'Jan-08-2003', 'I am String', 'true'

Default: "

For Mapping activities representing PLSQL maps, the allowed value for the parameters: OPERATING\_MODE:'SET\_BASED' 'ROW\_BASED' 'ROW\_BASED\_TARGET\_ONLY' 'SET\_BASED\_FAIL\_OVER\_TO\_ROW\_BASED' 'SET\_BASED\_FAIL\_OVER\_TO\_ROW\_BASED\_TARGET\_ONLY' AUDIT\_LEVEL:'NONE' 'STATISTICS' 'ERROR\_DETAILS' 'COMPLETE' Sets the value of a Activity Parameter

Name: BINDING

Type: STRING

Valid Values: Examples 'PARAM\_1', 'PARAM\_2'

Default: "

Represents the parameter on the process flow that this parameter is bound to. When setting users can specify the name of any PROCESS PARAMETER of same datatype. This feature allows for parameterizing the process flow. If the parameter is bound the VALUE property is ignored when generating the process flow. To unbind a parameter, use an empty quoted string ie. "", and the parameter will be unbound.

Properties for PROCESS\_FLOW:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: "

Physical name of the object that this activity represents.

Name: DEPLOYED\_LOCATION

Type: STRING

Valid Values: N/A

Default: USE\_DEFAULT\_LOCATION

Deployed Location of Transformation Activities only.

Name: EXECUTION\_LOCATION

Type: STRING

Valid Values: N/A

Default: NATIVE\_EXECUTION

The location from which this activity will be executed.

Name: REMOTE\_LOCATION

Type: STRING

Valid Values: N/A

Default: USE\_DEFAULT\_LOCATION

Remote Location for FTP activities only.

Name: USE\_RETURN\_AS\_STATUS

Type: BOOLEAN

Valid Values: true, false

Default: false

Use any return value to select the Transition path.

Name: WORKING\_LOCATION

Type: STRING

Valid Values: N/A

Default: USE\_DEFAULT\_LOCATION

Working Location for FTP, FILE\_EXIST and USER\_DEFINED activities only.

Properties for ACTIVITY:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: "

Physical name of the object that this activity represents.

Name: DEPLOYED\_LOCATION

Type: STRING

Valid Values: N/A

Default: USE\_DEFAULT\_LOCATION

Deployed Location of Transformation Activities only.

Name: EXECUTION\_LOCATION

Type: STRING

Valid Values: N/A

Default: NATIVE\_EXECUTION

The location from which this activity will be executed.

Name: REMOTE\_LOCATION

Type: STRING

Valid Values: N/A

Default: USE\_DEFAULT\_LOCATION

Remote Location for FTP activities only.

Name: USE\_RETURN\_AS\_STATUS

Type: BOOLEAN

Valid Values: true, false

Default: false

Use any return value to select the Transition path.

Name: WORKING\_LOCATION

Type: STRING

Valid Values: N/A

Default: USE\_DEFAULT\_LOCATION

Working Location for FTP, FILE\_EXIST and USER\_DEFINED activities only.

Note:

1. N/A means any valid character in supported character set.
2. " " represents an empty string

addProcessParameterClause

This clause adds the Parameters for a Process Flow.

`addActivityClause`

This clause adds the Activities for a Process Flow.

`addTransitionClause`

This clause adds the Transitions for a Process Flow.

`addUserDefinedParameterClause`

This clause adds the Parameters for a User Defined Activity of a Process Flow.

`propertyNameList`

A comma delimited set of property names to set.

`propertyValueList`

A comma delimited set of property values to set.

`addStandardActivityClause`

This clause adds standard activity types AND, FORK, FTP, OR, USER\_DEFINED to a Process Flow.

`addMapActivityClause`

This clause adds the MAP activity to a Process Flow.

`addFunctionActivityClause`

This clause adds the Function or Procedure activity to a Process Flow.

`addSubProcessActivityClause`

This clause adds a Process as an activity to a Process Flow.

`propertyValue`

Integer value, float value or quoted string literal.

`collectPropertiesClause`

This clause collects core properties of Map, Function/Procedure and Subprocess activity.

`setPropertyAndReferencesMapClauses`



This clause sets reference to the existing Map.

setPropertiesAndReferencesFunctionClauses

This clause sets a reference to existing Function or Procedure.

setPropertiesAndReferencesSubProcessClauses

This clause sets a reference to existing Process Flow.

## Examples

```
OMBCREATE PROCESS_FLOW 'TEST_PROCESS_FLOW' SET PROPERTIES
(DESCRIPTION, BUSINESS_NAME) VALUES ('this is a Process Flow', 'process flow')
ADD PARAMETER 'PARAM_1' ADD PARAMETER 'PARAM_2' ADD FORK
ACTIVITY 'FORK_ACTIVITY' ADD USER_DEFINED ACTIVITY 'UD_ACTIVITY'
ADD FTP ACTIVITY 'FTP_ACTIVITY' ADD OR ACTIVITY 'OR_ACTIVITY' ADD
AND ACTIVITY 'AND_ACTIVITY' ADD MAPPING ACTIVITY 'ACTIVITY_MAP_1'
SET REFERENCE MAPPING '/PROCESS_FLOW_PROJECT/WAREHOUSE_P/MAP_
1' ADD SUBPROCESS ACTIVITY 'SUBPROCESS_ACTIVITY' SET REFERENCE
PROCESS_FLOW 'REPORT_PROCESS_FLOW' ADD MAPPING ACTIVITY
'ACTIVITY_MAP_2' SET REFERENCE MAPPING '/PROCESS_FLOW_
PROJECT/WAREHOUSE_P/MAP_2' ADD TRANSITION 'T1' FROM ACTIVITY
'START' TO 'FORK_ACTIVITY' SET PROPERTIES (TRANSITION_CONDITION)
VALUES ('SUCCESS') ADD TRANSITION 'T2' FROM ACTIVITY 'FORK_ACTIVITY'
TO 'FTP_ACTIVITY' SET PROPERTIES (TRANSITION_CONDITION) VALUES
('SUCCESS') ADD TRANSITION 'T3' FROM ACTIVITY 'FORK_ACTIVITY' TO
'ACTIVITY_MAP_1' SET PROPERTIES (TRANSITION_CONDITION) VALUES
('SUCCESS') ADD TRANSITION 'T4' FROM ACTIVITY 'FTP_ACTIVITY' TO 'OR_
ACTIVITY' SET PROPERTIES (TRANSITION_CONDITION) VALUES ('SUCCESS')
ADD TRANSITION 'T5' FROM ACTIVITY 'FTP_ACTIVITY' TO 'UD_ACTIVITY' SET
PROPERTIES (TRANSITION_CONDITION) VALUES ('ERROR') ADD TRANSITION
'T6' FROM ACTIVITY 'UD_ACTIVITY' TO 'OR_ACTIVITY' SET PROPERTIES
(TRANSITION_CONDITION) VALUES ('SUCCESS') ADD TRANSITION 'T7' FROM
ACTIVITY 'OR_ACTIVITY' TO 'AND_ACTIVITY' ADD TRANSITION 'T8' FROM
ACTIVITY 'ACTIVITY_MAP_1' TO 'AND_ACTIVITY' ADD TRANSITION 'T9' FROM
ACTIVITY 'ACTIVITY_MAP_1' TO 'SUBPROCESS_ACTIVITY' SET PROPERTIES
(TRANSITION_CONDITION) VALUES ('WARNING') ADD TRANSITION 'T10'
FROM ACTIVITY 'AND_ACTIVITY' TO 'ACTIVITY_MAP_2' ADD TRANSITION
'T11' FROM ACTIVITY 'ACTIVITY_MAP_2' TO 'SUBPROCESS_ACTIVITY' SET
PROPERTIES (TRANSITION_CONDITION) VALUES ('WARNING') ADD
TRANSITION 'T12' FROM ACTIVITY 'ACTIVITY_MAP_2' TO 'END' ADD
TRANSITION 'T13' FROM ACTIVITY 'SUBPROCESS_ACTIVITY' TO 'END' ADD
PARAMETER 'PARAM_1' OF USER_DEFINED ACTIVITY 'UD_ACTIVITY'
```

Prerequisite for this examples are the existence of a MAP\_1, MAP\_2 and a Process Flow 'REPORT\_PROCESS\_FLOW'. The example here will create a process flow and its parameters, it creates activity of types FTP, AND, OR, FORK, MAPPING, USER\_DEFINED, and SUBPROCESS. It creates various types of transitions among these activities. At the end it creates a parameter for a user\_defined activity type.

**See Also**

OMBCREATE, OMBALTER PROCESS\_FLOW, OMBDROP PROCESS\_FLOW

# OMBCREATE PROCESS\_FLOW\_MODULE

## Purpose

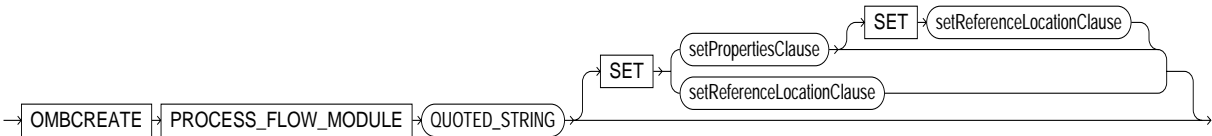
OMBCREATE PROCESS\_FLOW\_MODULE - To create a Process Flow Module.

## Prerequisites

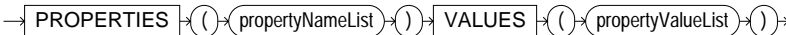
Should be in the context of a project.

## Syntax Diagrams

createProcessFlowModuleCommand



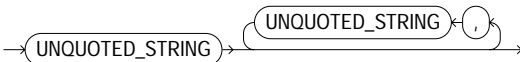
setPropertiesClause



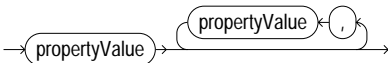
setReferenceLocationClause



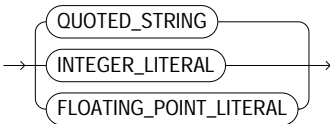
propertyNameList



propertyValueList



propertyValue



## Syntax

```
createProcessFlowModuleCommand = OMBCREATE ( PROCESS_FLOW_MODULE
"QUOTED_STRING" [ SET ( "setPropertiesClause" [ SET "setReferenceLocationClause"
] | "setReferenceLocationClause" ) ] );

setPropertiesClause = PROPERTIES "(" "propertyNameList ")" VALUES "("
"propertyValueList" )";
```

```

setReferenceLocationClause = ( REFERENCE | REF ) LOCATION "QUOTED_
STRING";
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
propertyValueList = "propertyValue" { "," "propertyValue" };
propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );

```

## Keywords and Parameters

`createProcessFlowModuleCommand`

Create a new process flow module.

`setPropertyClause`

Used to set properties (core, user-defined) for process flow module.

Base properties for PROCESS\_FLOW\_MODULE:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: NAME

Business name of a Process Flow Module

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of a Process Flow Module

`setReferenceLocationClause`

Set a location to a supported workflow engine.

`propertyNameList`

Comma-separated list of property names. Property names are not in quotation marks.

`propertyValueList`

Comma-separated list of property values.

`propertyValue`

Value of a property.

### Examples

```
OMBCREATE PROCESS_FLOW_MODULE 'process_Module' SET PROPERTIES  
(DESCRIPTION,  
BUSINESS_NAME) VALUES ('this is a Process Flow Module', 'process flow module')  
This will create a Process Flow Module named "process_Module", its description is  
"this is a Process Flow Module", and business name is "process flow module".
```

### See Also

OMBCREATE, OMBALTER PROCESS\_FLOW\_MODULE, OMBDROP PROCESS\_  
FLOW\_MODULE

# OMBCREATE PROCESS\_FLOW\_PACKAGE

## Purpose

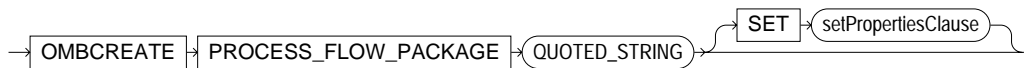
OMBCREATE PROCESS\_FLOW\_PACKAGE - To create a Process Flow Package.

## Prerequisites

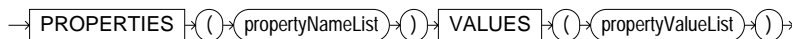
Should be in the context of a Process Flow Module.

## Syntax Diagrams

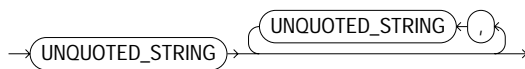
createProcessFlowPackageCommand



setPropertiesClause



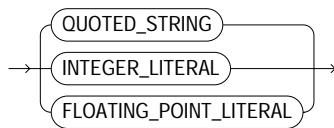
propertyNameList



propertyValueList



propertyValue



## Syntax

```
createProcessFlowPackageCommand = OMBCREATE ( PROCESS_FLOW_PACKAGE
"QUOTED_STRING" [ SET "setPropertiesClause" ] );
```

```
setPropertiesClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("
"propertyValueList" ");
```

```
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
```

```
propertyValueList = "propertyValue" { "," "propertyValue" };
```

```
propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );
```

## Keywords and Parameters

`createProcessFlowPackageCommand`

Create a new process flow package.

`setPropertyClause`

Used to set properties (core, user-defined) for process flow packages. Valid properties are shown below:

Basic properties for `PROCESS_FLOW_PACKAGE`:

Name: `BUSINESS_NAME`

Type: `STRING(200)`

Valid Values: N/A

Default: ""

Business name of the Process Flow Package

Name: `DESCRIPTION`

Type: `STRING(4000)`

Valid Values: N/A

Default: ""

Description of the Process Flow Package

Note:

1. N/A means any valid character in supported character set.
2. "" represents an empty string

`propertyNameList`

A comma delimited set of property names to set.

`propertyValueList`

A comma delimited set of property values to set.

`propertyValue`

Integer value, float value or quoted string literal.

## Examples

OMBCREATE PROCESS\_FLOW\_PACKAGE 'process\_Package' SET PROPERTIES (DESCRIPTION, BUSINESS\_NAME) VALUES ('this is a Process Flow Package', 'process flow package') This will create a Process Flow Package named "process\_Package", its description is "this is a Process Flow Package", and business name is "process flow package".

## See Also

OMBCREATE, OMBALTER PROCESS\_FLOW\_PACKAGE, OMBDROP PROCESS\_FLOW\_PACKAGE



# OMBCREATE PROJECT

## Purpose

OMBCREATE PROJECT - To create a project.

## Prerequisites

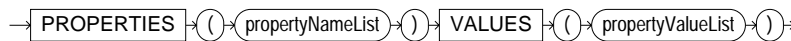
Should be in the top level context.

## Syntax Diagrams

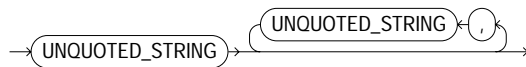
createProjectCommand



setPropertiesClause



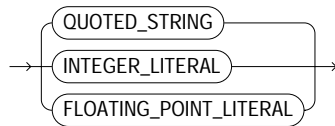
propertyNameList



propertyValueList



propertyValue



## Syntax

```
createProjectCommand = OMBCREATE ( PROJECT "QUOTED_STRING" [ SET
"setPropertiesClause" ] );
```

```
setPropertiesClause = PROPERTIES "(" "propertyNameList ")" VALUES "("
"propertyValueList" );
```

```
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
```

```
propertyValueList = "propertyValue" { "," "propertyValue" };
```

```
propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );
```

## Keywords and Parameters

`createProjectCommand`

Create a project.

`setPropertyClause`

Associate a set of properties with a project.

Basic properties for PROJECT:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: NAME

Business name of a Project

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of a Project

Name: UOID

Type: STRING(40)

Valid Values: N/A

Default: N/A

UOID of a Project

`propertyNameList`

Comma separated list of property names. Property names are unquoted.

`propertyValueList`

Comma separated list of property values.

`propertyValue`

Value of a property.

**Examples**

OMBCREATE PROJECT 'New Project' SET PROPERTIES (DESCRIPTION, BUSINESS\_NAME)

VALUES ('this is a project', 'payroll project') This will create a project named "New Project", its description is "this is a project", and business name is "payroll project".

**See Also**

OMBCREATE, OMBALTER PROJECT, OMBDROP PROJECT

# OMBCREATE RUNTIME\_REPOSITORY\_CONNECTION

## Purpose

OMBCREATE RUNTIME\_REPOSITORY\_CONNECTION - To create a runtime repository connection.

## Prerequisites

Should be in the context of a project.

## Syntax Diagrams

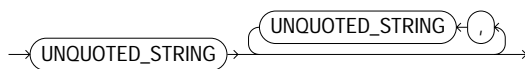
createRuntimeRepositoryCommand



createRunReposSetPropertiesClause



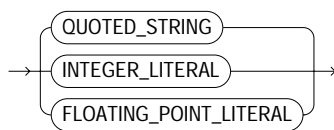
propertyNameList



propertyValueList



propertyValue



## Syntax

```

createRuntimeRepositoryCommand = OMBCREATE ( RUNTIME_REPOSITORY_
CONNECTION "QUOTED_STRING" "createRunReposSetPropertiesClause" );

createRunReposSetPropertiesClause = SET PROPERTIES "(" "propertyNameList" ")"
VALUES "(" "propertyValueList" ")";

propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };

propertyValueList = "propertyValue" { "," "propertyValue" };

propertyValue = ("QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );
  
```

## Keywords and Parameters

`createRuntimeRepositoryCommand`

Create a new runtime repository connection.

`createRunReposSetPropertiesClause`

Set the specified properties of the runtime repository connection.

`propertyNameList`

The names of the properties whose values you want to set.

Properties for `RUNTIME_REPOSITORY_CONNECTION`:

Name: `HOST`

Type: `STRING`

Valid Values: `N/A`

Default: `N/A`

The host machine the runtime repository is installed on.

Name: `PORT`

Type: `NUMBER`

Valid Values: `0 - 65535`

Default: `N/A`

The port number of the database in which the runtime repository is installed.

Name: `SERVICE_NAME`

Type: `STRING`

Valid Values: `N/A`

Default: `N/A`

The service name of the database in which the runtime repository is installed.

Name: `CONNECT_AS_USER`

Type: `STRING`

Valid Values: `N/A`

Default: `N/A`

The name of the database user you wish to connect to the runtime repository as.

Name: `RUNTIME_REPOSITORY_OWNER`

Type: STRING

Valid Values: N/A

Default: N/A

The name of the schema in which the runtime repository is installed.

All of the preceding properties are mandatory for OMBCREATE RUNTIME\_REPOSITORY\_CONNECTION.

Basic properties for RUNTIME\_REPOSITORY\_CONNECTION:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the runtime repository connection.

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the runtime repository connection.

Note:

1. N/A means any valid character in supported character set.
2. " represents an empty string

propertyValueList

The values for the named properties.

propertyValue

A property value.

## Examples

```
OMBCREATE RUNTIME_REPOSITORY_CONNECTION 'NEW_RR_CONNECTION'
SET PROPERTIES (DESCRIPTION, BUSINESS_NAME, HOST, PORT, SERVICE_
NAME, CONNECT_AS_USER, RUNTIME_REPOSITORY_OWNER) VALUES ('this is
a runtime repository', 'runtime repository', 'localhost', 1521, 'orcl9i', 'scott', 'runtime_
repos') This will create a runtime repository connection named "new_repository", its
description is "this is a runtime repository", and business name is "runtime repository".
```

## See Also

OMBCREATE, OMBALTER RUNTIME\_REPOSITORY\_CONNECTION, OMBDROP RUNTIME\_REPOSITORY\_CONNECTION

# OMBCREATE SEQUENCE

## Purpose

OMBCREATE SEQUENCE - To create a sequence.

## Prerequisites

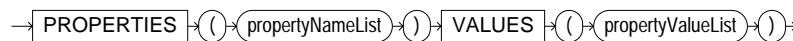
In the context of an Oracle Module.

## Syntax Diagrams

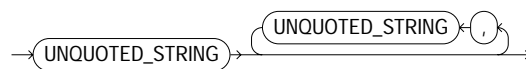
createSequenceCommand



setPropertiesClause



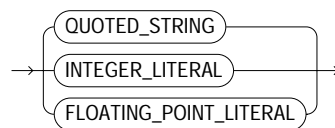
propertyNameList



propertyValueList



propertyValue



## Syntax

```
createSequenceCommand = OMBCREATE ( SEQUENCE "QUOTED_STRING" [ SET "setPropertiesClause" ] );
```

```
setPropertiesClause = PROPERTIES "(" "propertyNameList" ")" VALUES "(" "propertyValueList" )";
```

```
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
```

```
propertyValueList = "propertyValue" { "," "propertyValue" };
```

```
propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_POINT_LITERAL" );
```

## Keywords and Parameters

createSequenceCommand

This command creates a sequence.

### setPropertiesClause

Used to set properties (core, logical, physical, user-defined) for tables, columns, unique keys, foreign keys, primary keys, and check constraints. Valid properties are shown below:

#### Basic properties for SEQUENCE:

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the sequence.

Name: CURRVAL

Type: NUMBER

Valid Values: N/A

Default: 1

current increment value.

Name: NEXTVAL

Type: NUMBER

Valid Values: N/A

Default: 1

next increment value. next increment value.

#### Properties for SEQUENCE:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INCREMENT\_BY

Type: NUMBER

Valid Values: -2147483648 - 2147483647



Default: 1

Sequence Incremented By

Name: START\_WITH

Type: NUMBER

Valid Values: -2147483648 - 2147483647

Default: 1

Sequence Starts With

Note:

1. N/A means any valid character in supported character set.
2. " represents an empty string

propertyNameList

The list of properties.

propertyValueList

The list of property values.

propertyValue

This clause adds the property values.

## Examples

OMBCREATE SEQUENCE 'new\_sequence' SET PROPERTIES (DESCRIPTION, BUSINESS\_NAME) VALUES ('this is a new sequence', 'New Sequence') This will create a sequence named "NEW\_SEQUENCE", its description is "this is a new sequence", and business name is "New Sequence".

## See Also

OMBCREATE, OMBALTER SEQUENCE, OMBDROP SEQUENCE

# OMBCREATE SNAPSHOT

## Purpose

OMBCREATE SNAPSHOT - To create a snapshot of a component.

## Prerequisites

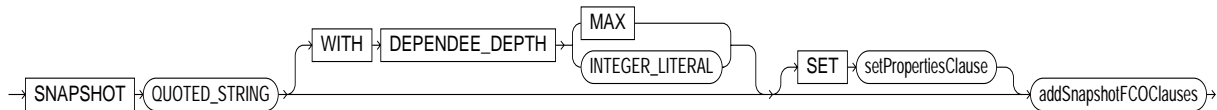
Component on which snapshot is to be created should already exist. This command can be executed for any component regardless of current context.

## Syntax Diagrams

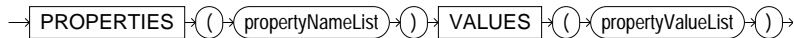
parseCreateCommand



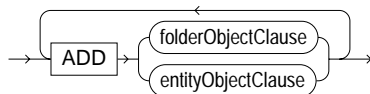
createSnapshotCommand



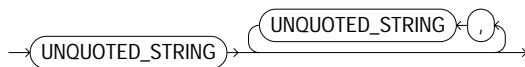
setPropertiesClause



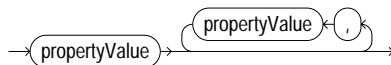
addSnapshotFCOClases



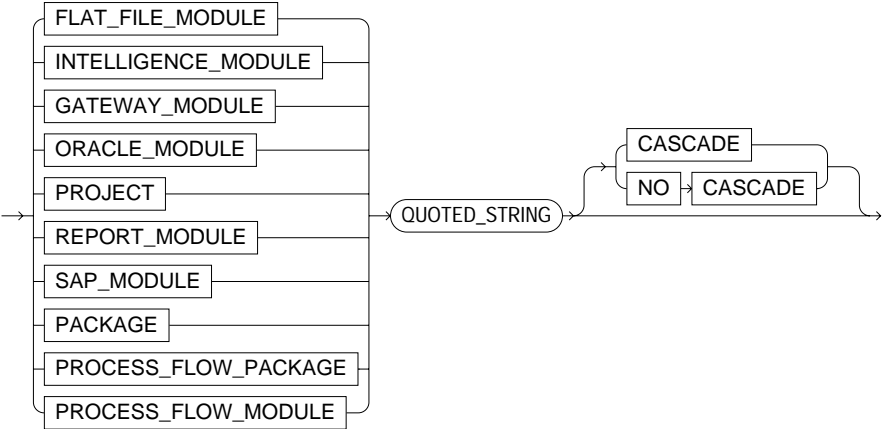
propertyNameList



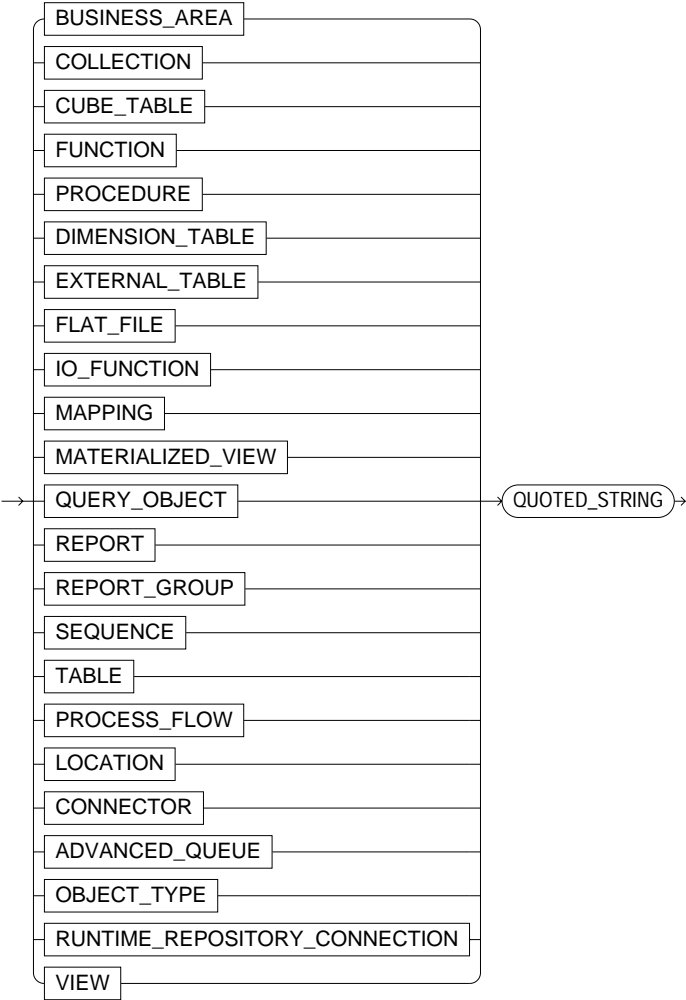
propertyValueList



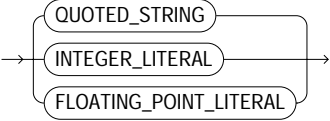
folderObjectClause



entityObjectClause



propertyValue



## Syntax

```

parseCreateCommand = OMBCREATE "createSnapshotCommand";
createSnapshotCommand = ( SNAPSHOT "QUOTED_STRING" [ WITH DEPENDEE_
DEPTH ( MAX | "INTEGER_LITERAL" ) ] [ SET "setPropertyClause" ]
"addSnapshotFCOClauses" );
setPropertyClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("
"propertyValueList" ");
addSnapshotFCOClauses = ( ADD ( "folderObjectClause" | "entityObjectClause" ) );
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
propertyValueList = "propertyValue" { "," "propertyValue" };
folderObjectClause = ( FLAT_FILE_MODULE | INTELLIGENCE_MODULE |
GATEWAY_MODULE | ORACLE_MODULE | PROJECT | REPORT_MODULE |
SAP_MODULE | PACKAGE | PROCESS_FLOW_PACKAGE | PROCESS_FLOW_
MODULE ) "QUOTED_STRING" [ CASCADE | NO CASCADE ];
entityObjectClause = ( ( BUSINESS_AREA | COLLECTION | CUBE_TABLE |
FUNCTION | PROCEDURE | DIMENSION_TABLE | EXTERNAL_TABLE | FLAT_
FILE | IO_FUNCTION | MAPPING | MATERIALIZED_VIEW | QUERY_OBJECT |
REPORT | REPORT_GROUP | SEQUENCE | TABLE | PROCESS_FLOW |
LOCATION | CONNECTOR | ADVANCED_QUEUE | OBJECT_TYPE | RUNTIME_
REPOSITORY_CONNECTION | VIEW ) "QUOTED_STRING" );
propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );

```

## Keywords and Parameters

**parseCreateCommand**

Root production for OMBCREATE SNAPSHOT.

**createSnapshotCommand**

Clause for creating snapshots.

**QUOTED\_STRING**

Name of the snapshot to be created.

**DEPENDEE\_DEPTH**

Optional clause to indicate whether the component's depedee components are to be included in the snapshot.

**setPropertyClause**

Optional clause to set the properties of a snapshot.

Basic properties for SNAPSHOT:

Name: TYPE  
Type: STRING(200)  
Valid Values: FULL,SIGNATURE  
Default: FULL  
This is the type of snapshot

Name: DESCRIPTION  
Type: STRING(4000)  
Valid Values: N/A  
Default: "  
Description of the snapshot

#### PROPERTIES

Valid set of properties are DESCRIPTION and TYPE

#### VALUES

Values for the corresponding properties. Default values of properties for DESCRIPTION is null and valid values for TYPE are FULL or SIGNATURE

#### addSnapshotFCOClauses

Components to be added to the snapshot

#### propertyNameList

Property names for SNAPSHOT.

#### propertyValueList

List of property values for SNAPSHOT.

#### folderObjectClause

Folder components to be added to the snapshot

#### QUOTED\_STRING

Absolute or relative path name of the component to be added to the snapshot

#### CASCADE

Cascade option for folder components. By default, any folder is snapshotted along with its contents. The NO CASCADE option allows the user to snapshot only the folder object itself along with its associated properties.

entityObjectClause

Any non-folder component can be added to a snapshot.

propertyValue

Allowable value types for a snapshot property.

## Examples

```
OMBCREATE SNAPSHOT 'S1' SET PROPERTIES (DESCRIPTION, TYPE) VALUES  
(this is  
snapshot', 'FULL') ADD TABLE '/Project1/WH1/T1'
```

This will create a snapshot named S1 with the T1 table component in it. `OMBCREATE SNAPSHOT 'S1' WITH DEPENDEE_DEPTH 1 SET PROPERTIES (DESCRIPTION, TYPE) VALUES('this is snapshot with dependees', 'FULL') ADD MAPPING '/Project1/WH1/MAP1'` This will create snapshot named S1 with the MAP1 mapping component. This command will find all the dependee components, which in this case would be all the components which the map references. For e.g. if MAP1 contains T1,T2 and T3, then this `WITH DEPENDEE_DEPTH 1` option will take a snapshot of the map MAP1 and tables T1,T2, and T3.

## See Also

OMBALTER SNAPSHOT, OMBDROP SNAPSHOT, OMBRESTORE SNAPSHOT,  
OMBCOMPARE SNAPSHOT, OMBLIST SNAPSHOT, OMBRETRIEVE SNAPSHOT

# OMBCREATE TABLE

## Purpose

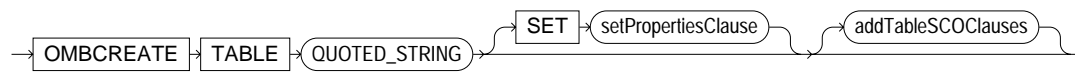
OMBCREATE TABLE - To create a table.

## Prerequisites

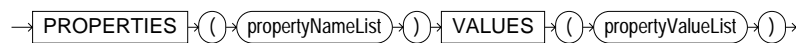
In the context of an Oracle Module.

## Syntax Diagrams

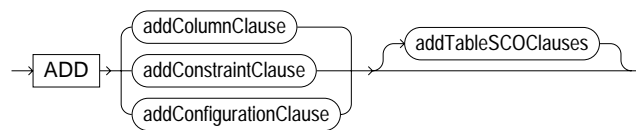
createTableCommand



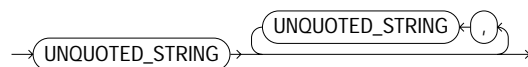
setPropertyClause



addTableSCOClauses



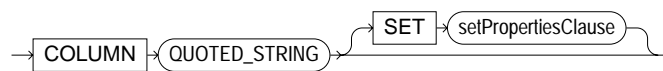
propertyNameList



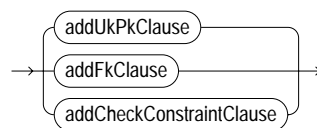
propertyValueList



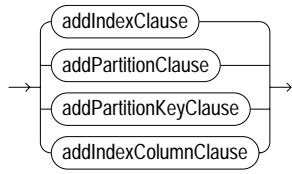
addColumnClause



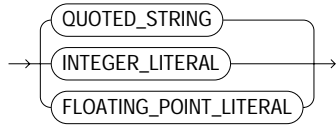
addConstraintClause



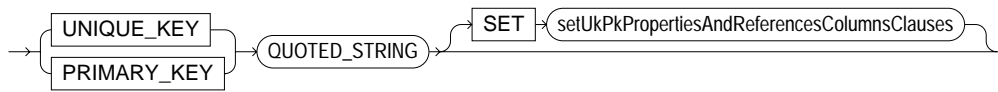
addConfigurationClause



property Value



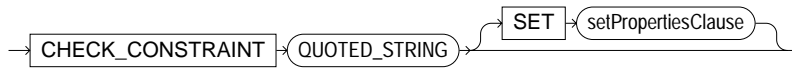
addUkPkClause



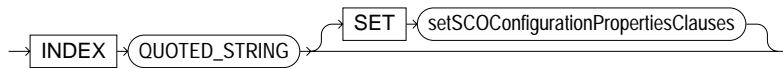
addFkClause



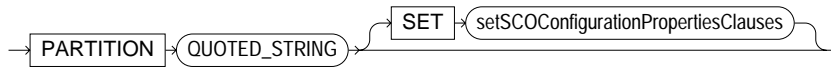
addCheckConstraintClause



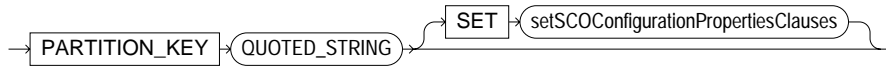
addIndexClause



addPartitionClause



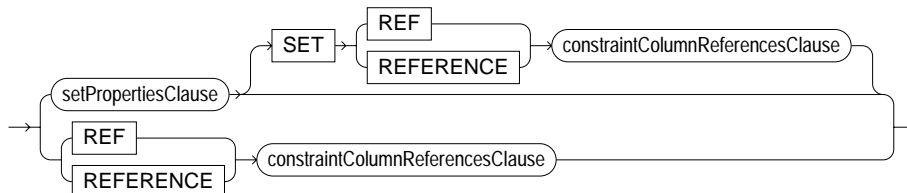
addPartitionKeyClause



addIndexColumnClause

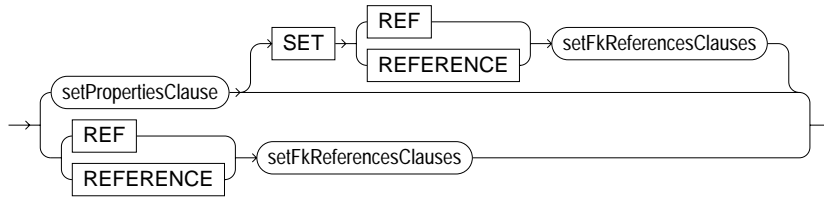


setUkPkPropertiesAndReferencesColumnsClauses

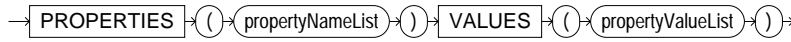




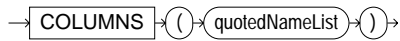
setFkSubClauses



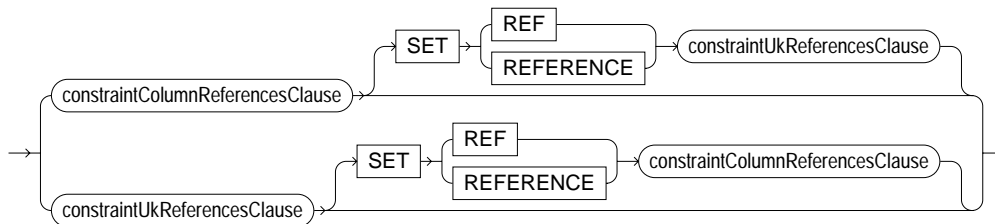
setSCOConfigurationPropertiesClauses



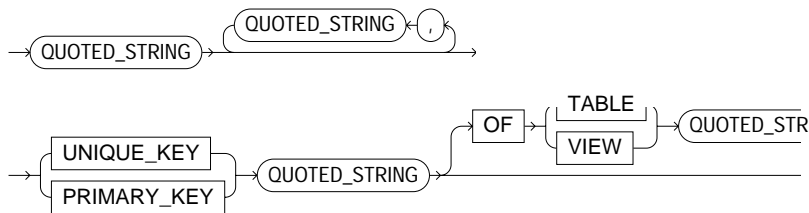
constraintColumnReferencesClause



setFkReferencesClauses



quotedNameList



## Syntax

```
createTableCommand = OMBCREATE ( TABLE "QUOTED_STRING" [ SET
"setPropertiesClause" ] [ "addTableSCOclauses" ] );
```

```
setPropertiesClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("
"propertyValueList" ")";
```

```
addTableSCOclauses = ADD ( "addColumnClause" | "addConstraintClause" |
"addConfigurationClause" ) [ "addTableSCOclauses" ];
```

```
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
```

```
propertyValueList = "propertyValue" { "," "propertyValue" };
```

```
addColumnClause = COLUMN "QUOTED_STRING" [ SET "setPropertiesClause" ];
```

```
addConstraintClause = "addUkPkClause" | "addFkClause" |
"addCheckConstraintClause";
```

```
addConfigurationClause = "addIndexClause" | "addPartitionClause" |
"addPartitionKeyClause" | "addIndexColumnClause";
```

```
propertyValue = ( "QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL" );
```

```

addUkPkClause = ( UNIQUE_KEY | PRIMARY_KEY ) "QUOTED_STRING" [ SET
"setUkPkPropertiesAndReferencesColumnsClauses" ];
addFkClause = FOREIGN_KEY "QUOTED_STRING" [ SET "setFkSubClauses" ];
addCheckConstraintClause = CHECK_CONSTRAINT "QUOTED_STRING" [ SET
"setPropertiesClause" ];
addIndexClause = INDEX "QUOTED_STRING" [ SET
"setSCOConfigurationPropertiesClauses" ];
addPartitionClause = PARTITION "QUOTED_STRING" [ SET
"setSCOConfigurationPropertiesClauses" ];
addPartitionKeyClause = PARTITION_KEY "QUOTED_STRING" [ SET
"setSCOConfigurationPropertiesClauses" ];
addIndexColumnClause = INDEX_COLUMN "QUOTED_STRING" OF INDEX
"QUOTED_STRING" [ SET "setSCOConfigurationPropertiesClauses" ];
setUkPkPropertiesAndReferencesColumnsClauses = "setPropertiesClause" [ SET (
REF | REFERENCE ) "constraintColumnReferencesClause" ] | ( REF | REFERENCE )
"constraintColumnReferencesClause";
setFkSubClauses = "setPropertiesClause" [ SET ( REF | REFERENCE )
"setFkReferencesClauses" ] | ( REF | REFERENCE ) "setFkReferencesClauses";
setSCOConfigurationPropertiesClauses = PROPERTIES "(" "propertyNameList" ")"
VALUES "(" "propertyValueList" ")";
constraintColumnReferencesClause = COLUMNS "(" "quotedNameList" ")";
setFkReferencesClauses = "constraintColumnReferencesClause" [ SET ( REF |
REFERENCE ) "constraintUkReferencesClause" ] | "constraintUkReferencesClause" [
SET ( REF | REFERENCE ) "constraintColumnReferencesClause" ];
quotedNameList = "QUOTED_STRING" { "," "QUOTED_STRING" };
constraintUkReferencesClause = ( UNIQUE_KEY | PRIMARY_KEY ) "QUOTED_
STRING" [ OF ( TABLE | VIEW ) "QUOTED_STRING" ];

```

## Keywords and Parameters

**createTableCommand**

This command creates a table.

**QUOTED\_STRING**

Specify the name of the table to be created.

**setPropertiesClause**

Used to set properties (core, logical, physical, user-defined) for tables, columns, unique keys, foreign keys, primary keys, and check constraints. Valid properties are shown below:

Basic properties for TABLE:

Name: BUSINESS\_NAME

Type: STRING(200)  
Valid Values: N/A  
Default: "  
Business name of the table

Name: DESCRIPTION  
Type: STRING(4000)  
Valid Values: N/A  
Default: "  
Description of the table

Name: UOID  
Type: STRING  
Valid Values: N/A  
Default: N/A

Basic properties for COLUMN:

Name: BUSINESS\_NAME  
Type: STRING(200)  
Valid Values: N/A  
Default: "  
Business name of the table

Name: DESCRIPTION  
Type: STRING(4000)  
Valid Values: N/A  
Default: "  
Description of the table

Name: DATATYPE  
Type: STRING  
Valid Values: NUMBER, VARCHAR, VARCHAR2, DATE, FLOAT  
Default: NUMBER  
The datatype of a column

Name: LENGTH  
Type: NUMBER  
Valid Values:  
Default: 1  
The length of a number

Name: PRECISION  
Type: NUMBER  
Valid Values: 0 - 39  
Default: 1  
The precision of a number.

Name: SCALE  
Type: NUMBER  
Valid Values: -85 - 125  
Default: 1  
The scale of a number.

Name: UOID  
Type: STRING  
Valid Values: N/A  
Default: N/A

Basic properties for INDEX, PARTITION, PARTITION\_KEY, INDEX\_COLUMN:

Name: DESCRIPTION  
Type: STRING(4000)  
Valid Values: N/A  
Default: "  
Description of the Index, Partition, PartitionKey, IndexColumn.

Properties for TABLE:

Name: ANALYZE\_TABLE\_ESTIMATE\_PERCENT  
Type: NUMBER  
Valid Values: 0 - 100

Default: 99

Value represents the sample size as a percentage of total rows. When set to a nonzero value, Builder generates a DDL script to analyze the table.

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: HASH\_PARTITION\_TABLESPACE\_LIST

Type: STRING

Valid Values: N/A

Default: "

A comma separated list of tablespaces to use for [sub]partition storage.

Name: HASH\_SUBPARTITION\_NUMBER

Type: NUMBER

Valid Values: 2 - 63999

Default: 2

To create Hash partition, specify the number of Hash subpartition.

Name: LOGGING\_MODE

Type: STRING

Valid Values: NOLOGGING, LOGGING

Default: LOGGING

Recovery requirements for a data warehouse : Logging or not logging to Redo Log File.

Name: PARALLEL\_ACCESS\_MODE

Type: STRING

Valid Values: NOPARALLEL, PARALLEL

Default: PARALLEL

Enables or disables parallel processing when the table is created. Also enables or disables parallel processing or access.

Name: TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Use the Tablespace parameter to specify the name of tablespace.

Properties for UNIQUE\_KEY:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: USING\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

Use of Index on Dimension Constraint

Properties for FOREIGN\_KEY:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: USING\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

Use of Index on Dimension Constraint

Properties for CHECK\_CONSTRAINT:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: USING\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

Use of Index on Dimension Constraint

Properties for INDEX:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: INDEX\_TYPE

Type: STRING

Valid Values: BITMAP, UNIQUE, NO\_INDEX

Default: UNIQUE

The types of Indexes created on Dimension are BITMAP, UNIQUE or a non-specific index.

Name: LOCAL\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

A local index is constructed so that it reflects the structure of the underlying table. It is equipartitioned with the underlying table, meaning that it is partitioned on the same columns as the underlying table, creates the same number of partitions or subpartitions, and gives them the same partition bounds as corresponding partitions of the underlying table.

Name: LOGGING\_MODE

Type: STRING

Valid Values: NOLOGGING, LOGGING

Default: LOGGING

Recovery requirements for a data warehouse : Logging or not logging to Redo Log File.

Name: PARALLEL\_ACCESS\_MODE

Type: STRING

Valid Values: NOPARALLEL, PARALLEL

Default: PARALLEL

Enables or disables parallel processing when the table is created. Also enables or disables parallel processing or access.

Name: TABLESPACE



Type: STRING(30)

Valid Values: N/A

Default: "

Use the Tablespace parameter to specify the name of tablespace.

Name: USING\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

Use of Index on Dimension Constraint

Properties for PARTITION:

Name: DATE\_LESS\_THAN

Type: STRING

Valid Values: N/A

Default: TO\_DATE('01-JAN-1999', 'DD-MON-YYYY')

Value that represents upper bound of partition stored in warehouse key column for the Days Dimension.

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: EMPTY\_STRING

Use the Tablespace parameter to specify the name of tablespace.

Name: VALUES\_LESS\_THAN

Type: STRING

Valid Values: N/A

Default: TO\_DATE('01-JAN-1999', 'DD-MON-YYYY')

Noninclusive upper bound for the current partition. Type a comma separated list in the same order as the columns in range partitioned key.

Name: VALUE\_LESS\_THAN

Type: STRING

Valid Values: N/A

Default: "

Noninclusive upper bound for the current partition. Type a comma separated list in the same order as the columns in range partitioned key.

Properties for PARTITION\_KEY:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: TYPE

Type: STRING

Valid Values: HASH, RANGE

Default: RANGE

Oracle partitions the storage space and stores rows according to a Hash Algorithm or specified ranges.

Note:

1. N/A means any valid character in supported character set.
2. " represents an empty string

addTableSCOClauses

This clause adds the secon class objects.

propertyNameList

The list of properties.

propertyValueList

The list of property values.

addColumnClause

This column adds a column.

addConstraintClause

Adds primary and unique key, and add check constraints.

addConfigurationClause

This clause will add configuration objects.

propertyValue

This clause adds the property values.

addUkPkClause

This clause adds the adds unique key and primary keys.

QUOTED\_STRING

name of the unique key or primary key.

addFkClause

This clause adds foreign key

QUOTED\_STRING

Name of the foreign key.

addCheckConstraintClause

add a check constraint.

QUOTED\_STRING

Name of the CheckConstraint.

addIndexClause

This clause adds an index.

QUOTED\_STRING

Name of the index.

`addPartitionClause`

This clause adds a partition.

`QUOTED_STRING`

Name of the partition.

`addPartitionKeyClause`

This clause adds a partition key.

`QUOTED_STRING`

Name of the partition key. This should be a column identifier.

`addIndexColumnClause`

This clause will add `indexColumn` to a specified index.

`QUOTED_STRING`

Index name

`setUkPkPropertiesAndReferencesColumnsClauses`

This clause adds properties and references to columns

`setFkSubClauses`

This clause set references to a foreign key.

`setSCOConfigurationPropertiesClauses`

Set the configuration properties for the following objects Index: `LOGGING_MODE`, `PARALLEL_ACCESS_MODE`, `TABLESPACE`, `INDEX_TYPE`, `LOCAL_INDEX`, `DEPLOYABLE` Partition: `DATE_LESS_THAN`, `TABLESPACE`, `DEPLOYABLE` Partition\_key: `TYPE`, `DEPLOYABLE`

`RelationalCmdParser$constraintColumnReferencesClause` = This clause provides names of all columns.

`constraintColumnReferencesClause`

`RelationalCmdParser$constraintColumnReferencesClause??`

`setFkReferencesClauses`

This clause sets foreign key references.

`quotedNameList`

This clause gives column names.

`constraintUkReferencesClause`

The first QUOTED\_STRING denotes the UniqueKey or Primary key name, and the latter denotes the table's or view's name.

### Examples

OMBCREATE TABLE 'new\_table' SET PROPERTIES (DESCRIPTION, BUSINESS\_NAME) VALUES ('this is a new table', 'New Table') This will create a table named "NEW\_TABLE", its description is "this is a new table", and business name is "New Table".

### See Also

OMBCREATE, OMBALTER TABLE, OMBDROP TABLE

# OMBCREATE VIEW

## Purpose

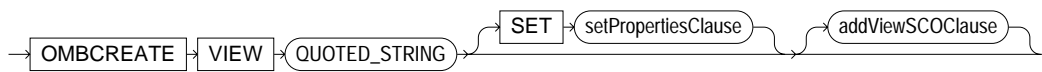
OMBCREATE VIEW - To create a view.

## Prerequisites

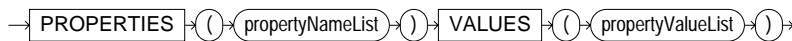
In the context of an Oracle Module.

## Syntax Diagrams

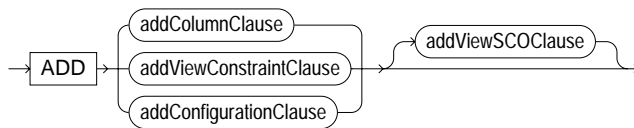
createViewCommand



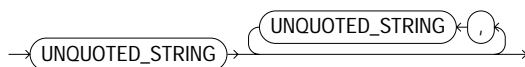
setPropertiesClause



addViewSCOClause



propertyNameList



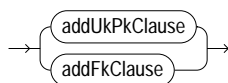
propertyValueList



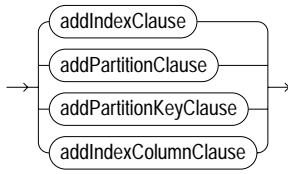
addColumnClause



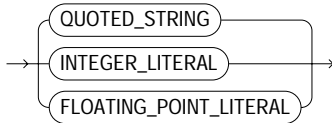
addViewConstraintClause



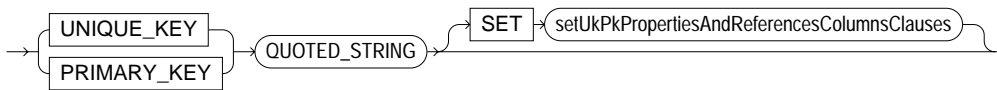
addConfigurationClause



property Value



addUkPkClause



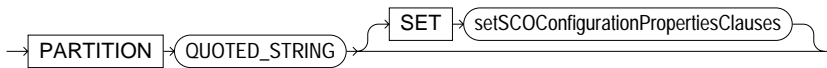
addFkClause



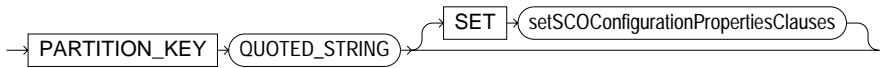
addIndexClause



addPartitionClause



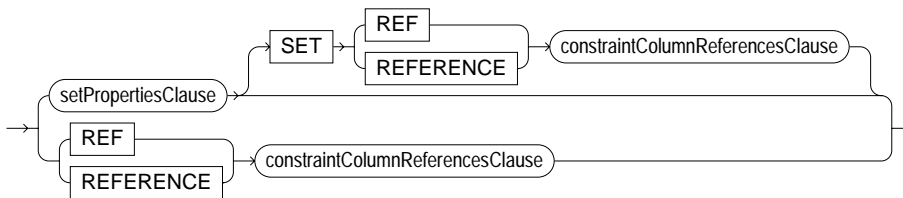
addPartitionKeyClause



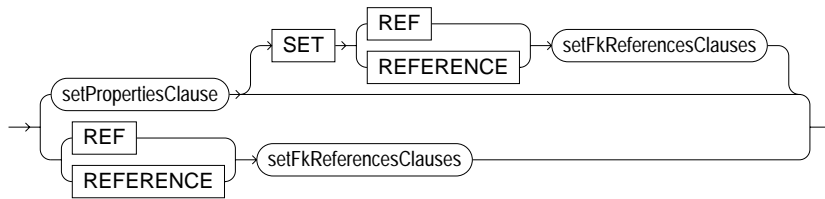
addIndexColumnClause



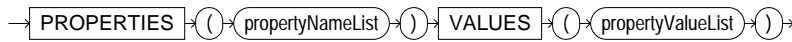
setUkPkPropertiesAndReferencesColumnsClauses



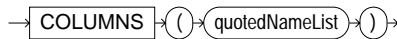
setFkSubClauses



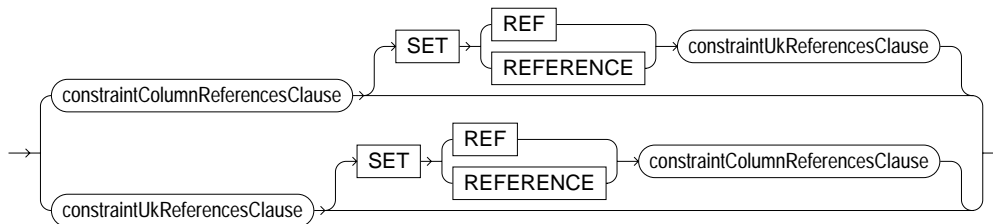
setSCOConfigurationPropertiesClauses



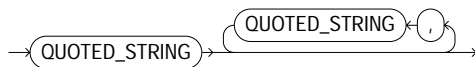
constraintColumnReferencesClause



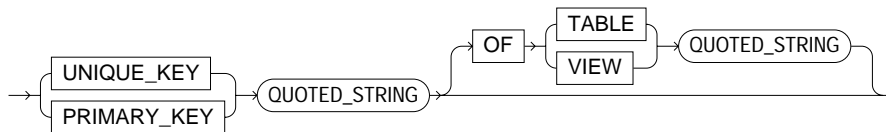
setFkReferencesClauses



quotedNameList



constraintUkReferencesClause



## Syntax

```

createViewCommand = OMBCREATE ( VIEW "QUOTED_STRING" [ SET
"setPropertiesClause" ] [ "addViewSCOClause" ] );

setPropertiesClause = PROPERTIES "(" "propertyNameList" ")" VALUES "("
"propertyValueList" ")";

addViewSCOClause = ADD ( "addColumnClause" | "addViewConstraintClause" |
"addConfigurationClause" ) [ "addViewSCOClause" ];

propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };

propertyValueList = "propertyValue" { "," "propertyValue" };

addColumnClause = COLUMN "QUOTED_STRING" [ SET "setPropertiesClause" ];

addViewConstraintClause = "addUkPkClause" | "addFkClause";

addConfigurationClause = "addIndexClause" | "addPartitionClause" |
"addPartitionKeyClause" | "addIndexColumnClause";
  
```



```

propertyValue = ("QUOTED_STRING" | "INTEGER_LITERAL" | "FLOATING_
POINT_LITERAL");
addUkPkClause = ( UNIQUE_KEY | PRIMARY_KEY ) "QUOTED_STRING" [ SET
"setUkPkPropertiesAndReferencesColumnsClauses" ];
addFkClause = FOREIGN_KEY "QUOTED_STRING" [ SET "setFkSubClauses" ];
addIndexClause = INDEX "QUOTED_STRING" [ SET
"setSCOConfigurationPropertiesClauses" ];
addPartitionClause = PARTITION "QUOTED_STRING" [ SET
"setSCOConfigurationPropertiesClauses" ];
addPartitionKeyClause = PARTITION_KEY "QUOTED_STRING" [ SET
"setSCOConfigurationPropertiesClauses" ];
addIndexColumnClause = INDEX_COLUMN "QUOTED_STRING" OF INDEX
"QUOTED_STRING" [ SET "setSCOConfigurationPropertiesClauses" ];
setUkPkPropertiesAndReferencesColumnsClauses = "setPropertiesClause" [ SET (
REF | REFERENCE ) "constraintColumnReferencesClause" ] | ( REF | REFERENCE )
"constraintColumnReferencesClause";
setFkSubClauses = "setPropertiesClause" [ SET ( REF | REFERENCE )
"setFkReferencesClauses" ] | ( REF | REFERENCE ) "setFkReferencesClauses";
setSCOConfigurationPropertiesClauses = PROPERTIES "(" "propertyNameList" ")"
VALUES "(" "propertyValueList" ")";
constraintColumnReferencesClause = COLUMNS "(" "quotedNameList" ")";
setFkReferencesClauses = "constraintColumnReferencesClause" [ SET ( REF |
REFERENCE ) "constraintUkReferencesClause" ] | "constraintUkReferencesClause" [
SET ( REF | REFERENCE ) "constraintColumnReferencesClause" ];
quotedNameList = "QUOTED_STRING" { "," "QUOTED_STRING" };
constraintUkReferencesClause = ( UNIQUE_KEY | PRIMARY_KEY ) "QUOTED_
STRING" [ OF ( TABLE | VIEW ) "QUOTED_STRING" ];

```

## Keywords and Parameters

**createViewCommand**

This command creates a view.

**QUOTED\_STRING**

Specify the name of the view to be created.

**setPropertiesClause**

Used to set properties (core, logical, physical, user-defined) for tables, columns, unique keys, foreign keys, primary keys, and check constraints.

Basic properties for VIEW, MATERIALIZED\_VIEW:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the View, MaterializedView

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the View, MaterializedView

Name: VIEW\_QUERY

Type: STRING(4000)

Valid Values: N/A

Default: "

Sets the query definition in View and MaterializedView.

Name: UOID

Type: STRING

Valid Values: N/A

Default: N/A

Basic properties for INDEX, PARTITION, PARTITION\_KEY, INDEX\_COLUMN:

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the Index, Partition, PartitionKey, IndexColumn in a MaterializedView.

Properties for VIEW:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

---

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Properties for UNIQUE\_KEY:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: USING\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

Use of Index on Dimension Constraint

Properties for FOREIGN\_KEY:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: USING\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

Use of Index on Dimension Constraint

Properties for CHECK\_CONSTRAINT:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: USING\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

Use of Index on Dimension Constraint

Note:

1. N/A means any valid character in supported character set.

2. " represents an empty string

addViewSCOClause

This clause adds components like column etc.

propertyNameList

The list of properties.

propertyValueList

The list of property values.

addColumnClause

This column adds a column.

addViewConstraintClause

This clause adds the view's configuration clause.

addConfigurationClause

This clause will add configuration objects.

propertyValue

This clause adds the property values.

addUkPkClause

This clause adds the adds unique key and primary keys.

QUOTED\_STRING

name of the unique key or primary key.

addFkClause

This clause adds foreign key

QUOTED\_STRING

Name of the foreign key.

addIndexClause

This clause adds an index.

QUOTED\_STRING

Name of the index.

addPartitionClause

This clause adds a partition.

QUOTED\_STRING

Name of the partition.

addPartitionKeyClause

This clause adds a partition key.

QUOTED\_STRING

Name of the partition key. This should be a column identifier.

addIndexColumnClause

This clause will add indexColumn to a specified index.

QUOTED\_STRING

Index name

setUkPkPropertiesAndReferencesColumnsClauses

This clause adds properties and references to columns

setFkSubClauses

This clause set references to a foreign key.

setSCOConfigurationPropertiesClauses

Set the configuration properties for the following objects Index: LOGGING\_MODE, PARALLEL\_ACCESS\_MODE, TABLESPACE, INDEX\_TYPE, LOCAL\_INDEX, DEPLOYABLE Partition: DATE\_LESS\_THAN, TABLESPACE, DEPLOYABLE Partition\_key: TYPE, DEPLOYABLE RelationalCmdParser\$constraintColumnReferencesClause = This clause provides names of all columns.

constraintColumnReferencesClause

RelationalCmdParser\$constraintColumnReferencesClause??

setFkReferencesClauses

This clause sets foreign key references.

quotedNameList

This clause gives column names.

constraintUkReferencesClause

The first QUOTED\_STRING denotes the UniqueKey or Primary key name, and the latter denotes the table's or view's name.

### Examples

OMBCREATE VIEW 'NEW\_VIEW' SET PROPERTIES (DESCRIPTION, BUSINESS\_NAME) VALUES ('this is a new view', 'New View') This will create a view named "NEW\_VIEW", its description is "this is a new view", and business name is "New View".

### See Also

OMBCREATE, OMBALTER VIEW, OMBDROP VIEW





This chapter contains the following topics:

---

<a href="#">OMBDROP</a> on page 5-2	<a href="#">OMBDROP GATEWAY_ MODULE</a> on page 5-16	<a href="#">OMBDROP PROCESS_FLOW_ PACKAGE</a> on page 5-29
<a href="#">OMBDROP ADVANCED_ QUEUE</a> on page 5-4	<a href="#">OMBDROP LOCATION</a> on page 5-18	<a href="#">OMBDROP PROJECT</a> on page 5-30
<a href="#">OMBDROP COLLECTION</a> on page 5-5	<a href="#">OMBDROP MAPPING</a> on page 5-19	<a href="#">OMBDROP RUNTIME_ REPOSITORY_CONNECTION</a> on page 5-32
<a href="#">OMBDROP CONNECTOR</a> on page 5-6	<a href="#">OMBDROP MATERIALIZED_ VIEW</a> on page 5-21	<a href="#">OMBDROP SAP_MODULE</a> on page 5-34
<a href="#">OMBDROP CUBE_TABLE</a> on page 5-7	<a href="#">OMBDROP OBJECT_TYPE</a> on page 5-22	<a href="#">OMBDROP SEQUENCE</a> on page 5-35
<a href="#">OMBDROP DEPLOYMENT_ ACTION_PLAN</a> on page 5-9	<a href="#">OMBDROP ORACLE_MODULE</a> on page 5-23	<a href="#">OMBDROP SNAPSHOT</a> on page 5-36
<a href="#">OMBDROP EXTERNAL_TABLE</a> on page 5-10	<a href="#">OMBDROP PACKAGE</a> on page 5-25	<a href="#">OMBDROP TABLE</a> on page 5-38
<a href="#">OMBDROP FLAT_FILE</a> on page 5-12	<a href="#">OMBDROP PROCEDURE</a> on page 5-26	<a href="#">OMBDROP VIEW</a> on page 5-39
<a href="#">OMBDROP FLAT_FILE_ MODULE</a> on page 5-14	<a href="#">OMBDROP PROCESS_FLOW</a> on page 5-27	
<a href="#">OMBDROP FUNCTION</a> on page 5-15	<a href="#">OMBDROP PROCESS_FLOW_ MODULE</a> on page 5-28	

---

## OMBDROP

### Purpose

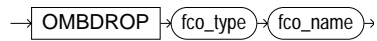
OMBDROP - Drop a component.

### Prerequisites

Should be in the parent context of the component to drop.

### Syntax Diagrams

dropCommand



### Syntax

```
dropCommand = OMBDROP "fco_type" "fco_name";
```

### Keywords and Parameters

dropCommand

Specify the component to drop.

fco\_type

The type of the component.

fco\_name

The physical name of the component in single quotes.

### Examples

This is an example of dropping a table:

```
OMBDROP TABLE 'T1'
```

### See Also

OMBCREATE, OMBALTER



## OMBDROP ADVANCED\_QUEUE

### Purpose

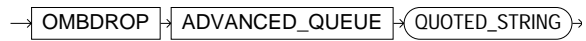
OMBDROP ADVANCED\_QUEUE - Delete the Advanced Queue.

### Prerequisites

Should be in the context of an Oracle Module.

### Syntax Diagrams

dropAQCommand



### Syntax

```
dropAQCommand = OMBDROP ADVANCED_QUEUE "QUOTED_STRING";
```

### Keywords and Parameters

dropAQCommand

Drops the Advanced Queue with the given name.

### Examples

OMBDROP ADVANCED\_QUEUE 'SOME\_ADVANCED\_QUEUE' This will delete the "SOME\_ADVANCED\_QUEUE" Advanced Queue.

### See Also

OMBDROP, OMBCREATE ADVANCED\_QUEUE, OMBALTER ADVANCED\_QUEUE, OMBRETRIEVE ADVANCED\_QUEUE

## OMBDROP COLLECTION

### Purpose

OMBDROP COLLECTION - Drop the collection object from this project.

### Prerequisites

Should be in the context of a project, before dropping a collection.

### Syntax Diagrams

dropCollectionCommand



### Syntax

```
dropCollectionCommand = OMBDROP ( COLLECTION "QUOTED_STRING" );
```

### Keywords and Parameters

dropCollectionCommand

Drop a collection of objects.

### Examples

```
OMBDROP COLLECTION 'PURCHASING_WAREHOUSE'
```

### See Also

OMBDROP, OMBALTER COLLECTION, OMBCREATE COLLECTION

## OMBDROP CONNECTOR

### Purpose

OMBDROP CONNECTOR - Delete a connector.

### Prerequisites

Should be in the context of the connector's owning location.

### Syntax Diagrams

dropConnectorCommand



### Syntax

```
dropConnectorCommand = OMBDROP ( CONNECTOR "QUOTED_STRING" );
```

### Keywords and Parameters

dropConnectorCommand

Drops the named connector from the repository.

### Examples

OMBDROP CONNECTOR 'A\_CONNECTOR' This will delete the "A\_CONNECTOR" connector.

### See Also

OMBDROP, OMBCREATE CONNECTOR, OMBALTER CONNECTOR

---

## OMBDROP CUBE\_TABLE

### Purpose

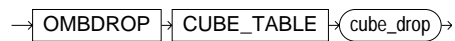
OMBDROP CUBE\_TABLE - This command drops a cube.

### Prerequisites

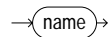
Should be in Oracle Module context.

### Syntax Diagrams

OMBDropCube



cube\_drop



name



### Syntax

```

OMBDropCube = OMBDROP CUBE_TABLE "cube_drop";
cube_drop = "name";
name = ("QUOTED_STRING");
  
```

### Keywords and Parameters

OMBDropCube

OMBDROP CUBE\_TABLE "cube\_drop"

cube\_drop

The name of the cube\_table to be dropped.

name

The name has to be a quoted string or an integer, or a decimal number.

### Examples

```
OMBDROP CUBE_TABLE 'CUBE1'
```

### See Also

OMBCREATE CUBE\_TABLE, OMBALTER CUBE\_TABLE, OMBRETRIEVE CUBE\_TABLE

## OMBDROP DEPLOYMENT\_ACTION\_PLAN

### Purpose

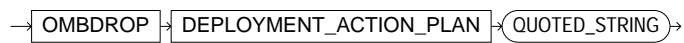
OMBDROP DEPLOYMENT\_ACTION\_PLAN - Remove an existing deployment action plan.

### Prerequisites

There must be a current working project.

### Syntax Diagrams

DropActionPlanCommand



### Syntax

```
DropActionPlanCommand = ( OMBDROP ( DEPLOYMENT_ACTION_PLAN )
"QUOTED_STRING" );
```

### Keywords and Parameters

DropActionPlanCommand

Remove an existing deployment action plan.

### Examples

```
OMBDROP DEPLOYMENT_ACTION_PLAN 'MY_PLAN'
```

### See Also

OMBCREATE DEPLOYMENT\_ACTION\_PLAN, OMBDEPLOY



---

## OMBDROP EXTERNAL\_TABLE

### Purpose

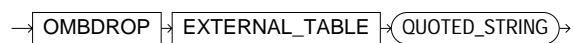
OMBDROP EXTERNAL\_TABLE - Delete the external table.

### Prerequisites

Should be in the context of an Oracle module.

### Syntax Diagrams

dropExternalTableCommand



### Syntax

```
dropExternalTableCommand = OMBDROP EXTERNAL_TABLE "QUOTED_
STRING";
```

### Keywords and Parameters

dropExternalTableCommand

Drop an external table from the repository.

QUOTED\_STRING

The name of the external table to drop.

### Examples

OMBDROP EXTERNAL\_TABLE 'SRC\_TABLE' This will delete the external table "SRC\_TABLE".

### See Also

OMBDROP, OMBCREATE EXTERNAL\_TABLE, OMBALTER EXTERNAL\_TABLE

## OMBDROP FLAT\_FILE

### Purpose

OMBDROP FLAT\_FILE - Delete a flat file.

### Prerequisites

Should be in the context of a flat file module.

### Syntax Diagrams

dropFlatFileCommand



### Syntax

dropFlatFileCommand = OMBDROP FLAT\_FILE "QUOTED\_STRING";

### Keywords and Parameters

dropFlatFileCommand

Drop a flat file.

QUOTED\_STRING

The name of the flat file to drop.

### Examples

OMBDROP FLAT\_FILE 'SRC\_FILE' This will delete the flat file "SRC\_FILE".

### See Also

OMBDROP, OMBCREATE FLAT\_FILE, OMBALTER FLAT\_FILE

---

## OMBDROP FLAT\_FILE\_MODULE

### Purpose

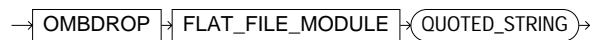
OMBDROP FLAT\_FILE\_MODULE - Delete a flat file module.

### Prerequisites

Should be in the context of a project.

### Syntax Diagrams

dropFlatFileModuleCommand



### Syntax

```
dropFlatFileModuleCommand = OMBDROP ( FLAT_FILE_MODULE "QUOTED_
STRING" );
```

### Keywords and Parameters

dropFlatFileModuleCommand

Drop a flat file module.

QUOTED\_STRING

The name of the flat file module to drop.

### Examples

OMBDROP FLAT\_FILE\_MODULE 'src\_module' This will delete the "src\_module" flat file module.

### See Also

OMBDROP, OMBCREATE FLAT\_FILE\_MODULE, OMBALTER FLAT\_FILE\_MODULE

## OMBDROP FUNCTION

### Purpose

OMBDROP FUNCTION - Delete the Function.

### Prerequisites

Should be in the context of a Oracle Module or Package or Transformation Module.

### Syntax Diagrams

dropFunctionCommand



### Syntax

```
dropFunctionCommand = OMBDROP ( FUNCTION "QUOTED_STRING" );
```

### Keywords and Parameters

dropFunctionCommand

Remove an existing Function.

QUOTED\_STRING

Name of the existing Function in quotes.

### Examples

**OMBDROP FUNCTION 'func'** This will delete the "func" Function. If Packaged Function is over-loaded, first find the Signature by using OMBLIST command, and then use OMBALTER command using appropriate signature. Example, if OMBLIST FUNCTIONS gives following two signatures, FUNC\_1 (NUMBER) RETURN NUMBER FUNC\_1 (VARCHAR2, NUMBER) RETURN NUMBER The OMBDROP Syntax to drop the first one will be as follows OMBDROP FUNCTION 'FUNC\_1 \ (NUMBER) RETURN NUMBER'

### See Also

OMBDROP, OMBCREATE FUNCTION, OMBALTER FUNCTION

---

## OMBDROP GATEWAY\_MODULE

### Purpose

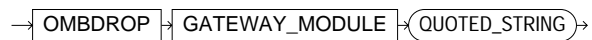
OMBDROP GATEWAY\_MODULE - Drop a gateway module.

### Prerequisites

Should be in the context of project.

### Syntax Diagrams

dropGatewayModuleCommand



### Syntax

```
dropGatewayModuleCommand = OMBDROP ( GATEWAY_MODULE "QUOTED_
STRING" );
```

### Keywords and Parameters

dropGatewayModuleCommand

Specify the gateway module to be dropped.

### Examples

The following example drops a gateway module named IFMX:

```
OMBDROP GATEWAY_MODULE 'IFMX'
```

### See Also

OMBDROP, OMBCREATE GATEWAY\_MODULE, OMBALTER GATEWAY\_MODULE

## OMBDROP LOCATION

### Purpose

OMBDROP LOCATION - Delete the location.

### Prerequisites

Should be in the context of a project.

### Syntax Diagrams

dropLocationCommand



### Syntax

```
dropLocationCommand = OMBDROP ( LOCATION "QUOTED_STRING" );
```

### Keywords and Parameters

dropLocationCommand

Drop a location from the design repository.

### Examples

OMBDROP LOCATION 'OLD\_LOCATION' This will delete the location "OLD\_LOCATION".

### See Also

OMBDROP, OMBCREATE LOCATION, OMBALTER LOCATION

---

## OMBDROP MAPPING

### Purpose

OMBDROP MAPPING - Drop an existing mapping.

### Prerequisites

The current context of scripting must be an Oracle

### Syntax Diagrams

dropMappingCommand



mappingName



### Syntax

```
dropMappingCommand = OMBDROP MAPPING "mappingName";
mappingName = "QUOTED_STRING";
```

### Keywords and Parameters

dropMappingCommand

Drop an existing mapping.

mappingName

Name of the mapping.

### Examples

```
OMBDROP MAPPING 'MAP1'
```

### See Also

OWBDROP, OMBCREATE MAPPING, OMBALTER MAPPING, OMBRETRIEVE MAPPING

## OMBDROP MATERIALIZED\_VIEW

### Purpose

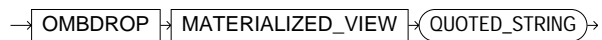
OMBDROP MATERIALIZED\_VIEW - To drop a materialized view.

### Prerequisites

In the context of an Oracle Module.

### Syntax Diagrams

dropMaterializedViewCommand



### Syntax

```
dropMaterializedViewCommand = OMBDROP MATERIALIZED_VIEW "QUOTED_
STRING";
```

### Keywords and Parameters

dropMaterializedViewCommand

This clause drops a MaterializedView.

### Examples

```
OMBDROP MATERIALIZED_VIEW 'NEW_MATERIALIZED_VIEW'.
```

### See Also

OMBRETRIEVE MATERIALIZED\_VIEW, OMBCREATE MATERIALIZED\_VIEW,  
OMBALTER MATERIALIZED\_VIEW



---

## OMBDROP OBJECT\_TYPE

### Purpose

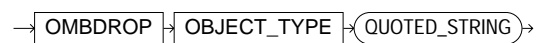
OMBDROP OBJECT\_TYPE - Delete the Object Type.

### Prerequisites

Should be in the context of an Oracle Module.

### Syntax Diagrams

dropObjectTypeCommand



### Syntax

```
dropObjectTypeCommand = OMBDROP OBJECT_TYPE "QUOTED_STRING";
```

### Keywords and Parameters

dropObjectTypeCommand

Drops the Object Type with given name.

### Examples

OMBDROP OBJECT\_TYPE 'SOME\_OBJECT\_TYPE' This will delete the "SOME\_OBJECT\_TYPE" Object Type.

### See Also

OMBDROP, OMBCREATE OBJECT\_TYPE, OMBALTER OBJECT\_TYPE

## OMBDROP ORACLE\_MODULE

### Purpose

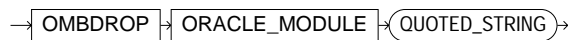
OMBDROP ORACLE\_MODULE - Delete the Oracle module.

### Prerequisites

Should be in the context of project.

### Syntax Diagrams

dropOracleModuleCommand



### Syntax

```
dropOracleModuleCommand = OMBDROP ( ORACLE_MODULE "QUOTED_
STRING" );
```

### Keywords and Parameters

dropOracleModuleCommand

Remove an existing Oracle module.

QUOTED\_STRING

Name of the existing Oracle module in quotes.

### Examples

OMBDROP ORACLE\_MODULE 'src\_module' This will delete the "src\_module"  
Oracle module.

### See Also

OMBDROP, OMBCREATE ORACLE\_MODULE, OMBALTER ORACLE\_MODULE

---

## OMBDROP PACKAGE

### Purpose

OMBDROP PACKAGE - Delete the Package.

### Prerequisites

Should be in the context of a Oracle Module or Transformation Module.

### Syntax Diagrams

dropPackageCommand



### Syntax

```
dropPackageCommand = OMBDROP ( PACKAGE "QUOTED_STRING" );
```

### Keywords and Parameters

dropPackageCommand

Remove an existing Package.

QUOTED\_STRING

Name of the existing Package in quotes.

### Examples

OMBDROP PACKAGE 'package\_1' This will delete the "package\_1" Package.

### See Also

OMBDROP, OMBCREATE PACKAGE, OMBALTER PACKAGE

## OMBDROP PROCEDURE

### Purpose

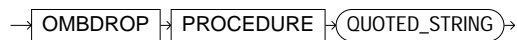
OMBDROP PROCEDURE - Delete the Procedure.

### Prerequisites

Should be in the context of a Oracle Module or Package or Transformation Module.

### Syntax Diagrams

dropProcedureCommand



### Syntax

dropProcedureCommand = OMBDROP ( PROCEDURE "QUOTED\_STRING" );

### Keywords and Parameters

dropProcedureCommand

Remove an existing Procedure.

QUOTED\_STRING

Name of the existing Procedure in quotes.

### Examples

**OMBDROP PROCEDURE 'proc'** This will delete the "proc" Procedure. If Packaged Function is over-loaded, first find the Signature by using OMBLIST command, and then use OMBALTER command using appropriate signature. Example, if OMBLIST PROCEDURES gives following two signatures, PROC\_1 (NUMBER) PROC\_1 (VARCHAR2, NUMBER) The OMBDROP Syntax to drop the first one will be as follows OMBDROP PROCEDURE 'PROC\_1 \ (NUMBER\)'

### See Also

OMBDROP, OMBCREATE PROCEDURE, OMBALTER PROCEDURE

---

## OMBDROP PROCESS\_FLOW

### Purpose

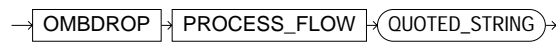
OMBDROP PROCESS\_FLOW - Delete the Process Flow.

### Prerequisites

Should be in the context of a Process Flow Package.

### Syntax Diagrams

dropProcessFlowCommand



### Syntax

```
dropProcessFlowCommand = OMBDROP ( PROCESS_FLOW "QUOTED_STRING" );
```

### Keywords and Parameters

dropProcessFlowCommand

Delete a process flow.

### Examples

OMBDROP PROCESS\_FLOW 'process\_flow' This will delete the "process\_flow" Process Flow.

### See Also

OMBDROP, OMBCREATE PROCESS\_FLOW, OMBALTER PROCESS\_FLOW

## OMBDROP PROCESS\_FLOW\_MODULE

### Purpose

OMBDROP PROCESS\_FLOW\_MODULE - Delete the Process Flow Module.

### Prerequisites

Should be in the context of a project.

### Syntax Diagrams

dropProcessFlowModuleCommand



### Syntax

```
dropProcessFlowModuleCommand = OMBDROP ( PROCESS_FLOW_MODULE
"QUOTED_STRING" );
```

### Keywords and Parameters

dropProcessFlowModuleCommand

Drop an existing process flow module.

### Examples

OMBDROP PROCESS\_FLOW\_MODULE 'process\_module' This will delete the "process\_module" Process Flow Module.

### See Also

OMBDROP, OMBCREATE PROCESS\_FLOW\_MODULE, OMBALTER PROCESS\_FLOW\_MODULE

## OMBDROP PROCESS\_FLOW\_PACKAGE

### Purpose

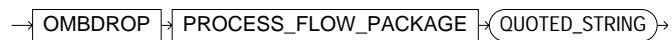
OMBDROP PROCESS\_FLOW\_PACKAGE - Delete the Process Flow Package.

### Prerequisites

Should be in the context of a Process Flow Module.

### Syntax Diagrams

dropProcessFlowPackageCommand



### Syntax

```
dropProcessFlowPackageCommand = OMBDROP ( PROCESS_FLOW_PACKAGE  
"QUOTED_STRING" );
```

### Keywords and Parameters

dropProcessFlowPackageCommand

Delete a process flow package.

### Examples

OMBDROP PROCESS\_FLOW\_PACKAGE 'process\_package' This will delete the "process\_package" Process Flow Package.

### See Also

OMBDROP, OMBCREATE PROCESS\_FLOW\_PACKAGE, OMBALTER PROCESS\_FLOW\_PACKAGE

## OMBDROP PROJECT

### Purpose

OMBDROP PROJECT - Delete the project.

### Prerequisites

Should be in the top level context.

### Syntax Diagrams

dropProjectCommand



### Syntax

```
dropProjectCommand = OMBDROP ( PROJECT "QUOTED_STRING" );
```

### Keywords and Parameters

dropProjectCommand

Remove an existing project.

QUOTED\_STRING

Name of the existing project in quotes.

### Examples

OMBDROP PROJECT 'New Project' This will delete the "New Project" project.

### See Also

OMBDROP, OMBCREATE PROJECT, OMBALTER PROJECT



---

## OMBDROP RUNTIME\_REPOSITORY\_CONNECTION

### Purpose

OMBDROP RUNTIME\_REPOSITORY\_CONNECTION - Delete the runtime repository connection.

### Prerequisites

Should be in the context of a project.

### Syntax Diagrams

dropRuntimeRepositoryCommand



### Syntax

```
dropRuntimeRepositoryCommand = OMBDROP ( RUNTIME_REPOSITORY_
CONNECTION "QUOTED_STRING" );
```

### Keywords and Parameters

dropRuntimeRepositoryCommand

Drop the runtime repository connection from the repository.

### Examples

OMBDROP RUNTIME\_REPOSITORY\_CONNECTION 'MY\_CONNECTION' This will delete the "MY\_CONNECTION" runtime repository connection.

### See Also

OMBDROP, OMBCREATE RUNTIME\_REPOSITORY\_CONNECTION, OMBALTER RUNTIME\_REPOSITORY\_CONNECTION

## OMBDROP SAP\_MODULE

### Purpose

OMBDROP SAP\_MODULE - Remove an existing SAP module.

### Prerequisites

You must open a project to drop a SAP module.

### Syntax Diagrams

dropSAPModuleCommand



### Syntax

```
dropSAPModuleCommand = OMBDROP ( SAP_MODULE "QUOTED_STRING" );
```

### Keywords and Parameters

dropSAPModuleCommand

Remove an existing SAP module.

### Examples

OMBDROP SAP\_MODULE 'src\_module' This will delete the "src\_module" SAP module.

### See Also

OMBDROP

---

## OMBDROP SEQUENCE

### Purpose

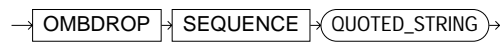
OMBDROP SEQUENCE - To drop a sequence.

### Prerequisites

In the context of an Oracle Module.

### Syntax Diagrams

dropSequenceCommand



### Syntax

```
dropSequenceCommand = OMBDROP SEQUENCE "QUOTED_STRING";
```

### Keywords and Parameters

dropSequenceCommand

This clause drops a Sequence.

### Examples

```
OMBDROP SEQUENCE 'new_sequence';
```

### See Also

OMBRETRIEVE SEQUENCE, OMBCREATE SEQUENCE, OMBALTER SEQUENCE

## OMBDROP SNAPSHOT

### Purpose

OMBDROP SNAPSHOT - A snapshot can be dropped.

### Prerequisites

The snapshot to be dropped should already exist. This command can be executed for any snapshot regardless of current context.

### Syntax Diagrams

parseDropCommand



### Syntax

```
parseDropCommand = OMBDROP ( SNAPSHOT "QUOTED_STRING" );
```

### Keywords and Parameters

parseDropCommand

To drop a snapshot.

QUOTED\_STRING

Name of snapshot to be dropped.

### Examples

```
OMBDROP SNAPSHOT 'S1'
```

### See Also

OMBCREATE SNAPSHOT, OMBALTER SNAPSHOT, OMBRESTORE SNAPSHOT,  
 OMBCOMPARE SNAPSHOT, OMBLIST SNAPSHOT, OMBRETRIEVE SNAPSHOT



## OMBDROP VIEW

### Purpose

OMBDROP VIEW - To drop a view.

### Prerequisites

In the context of an Oracle Module.

### Syntax Diagrams

dropViewCommand



### Syntax

dropViewCommand = OMBDROP VIEW "QUOTED\_STRING";

### Keywords and Parameters

dropViewCommand

This clause drops a View.

### Examples

OMBDROP VIEW 'NEW\_VIEW'.

### See Also

OMBRETRIEVE VIEW, OMBCREATE VIEW, OMBALTER VIEW

---



---

# OMBRETRIEVE

This chapter contains the following topics:

---

<a href="#">OMBRETRIEVE</a> on page 6-2	<a href="#">OMBRETRIEVE GATEWAY_</a> <a href="#">MODULE</a> on page 6-54	<a href="#">OMBRETRIEVE PROCESS_</a> <a href="#">FLOW_PACKAGE</a> on page 6-181
<a href="#">OMBRETRIEVE ADVANCED_</a> <a href="#">QUEUE</a> on page 6-5	<a href="#">OMBRETRIEVE LOCATION</a> on page 6-57	<a href="#">OMBRETRIEVE PROJECT</a> on page 6-184
<a href="#">OMBRETRIEVE COLLECTION</a> on page 6-9	<a href="#">OMBRETRIEVE MAPPING</a> on page 6-60	<a href="#">OMBRETRIEVE RUNTIME_</a> <a href="#">REPOSITORY_CONNECTION</a> on page 6-186
<a href="#">OMBRETRIEVE CONNECTOR</a> on page 6-14	<a href="#">OMBRETRIEVE</a> <a href="#">MATERIALIZED_VIEW</a> on page 6-137	<a href="#">OMBRETRIVE SAP_MODULE</a> on page 6-190
<a href="#">OMBRETRIEVE CUBE_TABLE</a> on page 6-17	<a href="#">OMBRETRIEVE OBJECT_TYPE</a> on page 6-150	<a href="#">OMBRETRIEVE SEQUENCE</a> on page 6-193
<a href="#">OMBRETRIEVE</a> <a href="#">DEPLOYMENT_ACTION_</a> <a href="#">PLAN</a> on page 6-27	<a href="#">OMBRETRIEVE ORACLE_</a> <a href="#">MODULE</a> on page 6-154	<a href="#">OMBRETRIEVE SNAPSHOT</a> on page 6-197
<a href="#">OMBRETRIEVE EXTERNAL_</a> <a href="#">TABLE</a> on page 6-30	<a href="#">OMBRETRIEVE PACKAGE</a> on page 6-163	<a href="#">OMBRETRIEVE TABLE</a> on page 6-200
<a href="#">OMBRETRIEVE FLAT_FILE</a> on page 6-36	<a href="#">OMBRETRIEVE PROCEDURE</a> on page 6-166	<a href="#">OMBRETRIEVE VIEW</a> on page 6-213
<a href="#">OMBRETRIEVE FLAT_FILE_</a> <a href="#">MODULE</a> on page 6-45	<a href="#">OMBRETRIEVE PROCESS_</a> <a href="#">FLOW</a> on page 6-171	
<a href="#">OMBRETRIEVE FUNCTION</a> on page 6-48	<a href="#">OMBRETRIEVE PROCESS_</a> <a href="#">FLOW_MODULE</a> on page 6-178	

---

# OMBRETRIEVE

## Purpose

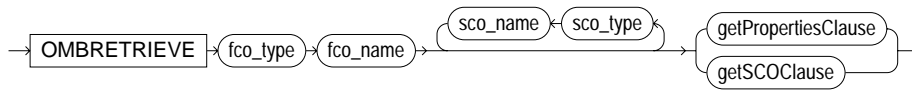
OMBRETRIEVE - Retrieve metadata of a component.

## Prerequisites

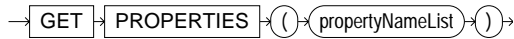
Can be in any context. Component to retrieve can be specified by either absolute or relative path.

## Syntax Diagrams

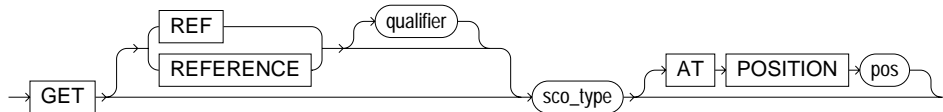
retrieveCommand



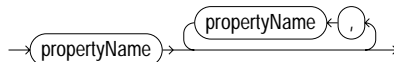
getPropertiesClause



getSCOClause



propertyNameList



propertyName



## Syntax

```
retrieveCommand = OMBRETRIEVE "fco_type" "fco_name" { "sco_type" "sco_name" }
("getPropertiesClause" | "getSCOClause");
```

```
getPropertiesClause = GET PROPERTIES "(" "propertyNameList" ")";
```

```
getSCOClause = GET [ ( REF | REFERENCE ) [ "qualifier" ] ] "sco_type" [ AT
POSITION "pos" ];
```

```
propertyNameList = "propertyName" { "," "propertyName" };
```

```
propertyName = "UNQUOTED_STRING";
```



## Keywords and Parameters

retrieveCommand

Specify the component from which to retrieve metadata.

fco\_type

The type of the component.

fco\_name

The physical name of the component in single quotes.

getPropertiesClause

Retrieve properties of an object.

getSCOClauses

Retrieve a list of child object names of a given type.

qualifier

Specify which reference to set, if there are more than one pointing to the same type.

propertyNameList

A list of property names.

propertyName

An unquoted string representing the name of a property.

## Examples

This is an example for retrieving the description a table:

```
OMBRETRIEVE TABLE 'T1' GET PROPERTIES (DESCRIPTION)
```

The following statement retrieves the datatype and length of a view column:

```
OMBRETRIEVE VIEW 'V1' COLUMN 'COL1'  
GET PROPERTIES (DATATYPE, LENGTH)
```

## See Also

OMBCREATE, OMBALTER

## OMBRETRIEVE ADVANCED\_QUEUE

### Purpose

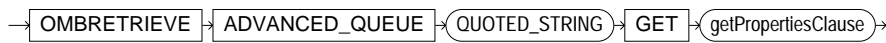
OMBRETRIEVE ADVANCED\_QUEUE - Retrieve details of the Advanced Queue.

### Prerequisites

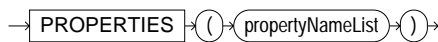
Should be in the context of an Oracle Module.

### Syntax Diagrams

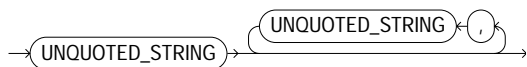
retrieveAQCommand



getPropertiesClause



propertyNameList



### Syntax

```

retrieveAQCommand = OMBRETRIEVE ADVANCED_QUEUE "QUOTED_STRING"
( GET "getPropertiesClause" );
getPropertiesClause = PROPERTIES "(" "propertyNameList" ")";
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
  
```

### Keywords and Parameters

retrieveAQCommand

Retrieves the details of the Advanced Queue with the given name.

getPropertiesClause

Retrieves the values of the given Properties for the Advanced Queue with the given name.

Basic properties for ADVANCED\_QUEUE:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the Advanced Queue

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the Advanced Queue

Name: PAYLOAD\_TYPE

Type: STRING(4000)

Valid Values: N/A

Default: "

PayLoad Type of the Advanced Queue. This has to be the name of an Object Type(OBJECT\_TYPE) existing in the same Oracle Module.

Properties for ADVANCED\_QUEUE:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those objects marked as Deployable = true.

Name: GENERATE\_ADVANCED\_QUEUE

Type: BOOLEAN

Valid Values: true, false

Default: true

Generate the code to create the Advanced Queue.

Name: GENERATE\_OBJECT\_TYPE

Type: BOOLEAN

Valid Values: true, false

Default: true

Generate a script to create the Object Type.

Name: GENERATE\_QUEUE\_TABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Generate code to create the queue table that will persist the messages of this Advanced Queue.

Name: GENERATE\_TEMPORARY\_TABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Generate the code to create the temporary table.

Name: QUEUE\_TABLE\_NAME

Type: STRING(30)

Valid Values: N/A

Default: "

The name of the queue table that is used to persist the messages in this Advanced Queue.

Name: TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Use the Tablespace parameter to specify the name of tablespace.

Note:

1. N/A means any valid character in supported character set.
2. " represents an empty string

propertyNameList

The list of properties.

## Examples

```
OMBRETRIEVE ADVANCED_QUEUE 'SOME_ADVANCED_QUEUE' GET
PROPERTIES (TABLESPACE, QUEUE_TABLE_NAME) This will retrieve the
```

Advanced Queue "SOME\_ADVANCED\_QUEUE"'s properties TableSpace and Queue Table name.

### See Also

OMBRETRIEVE, OMBALTER ADVANCED\_QUEUE, OMBCREATE ADVANCED\_QUEUE, OMBDROP ADVANCED\_QUEUE

## OMBRETRIEVE COLLECTION

### Purpose

OMBRETRIEVE COLLECTION - Retrieve details of the collection, including its shortcuts.

### Prerequisites

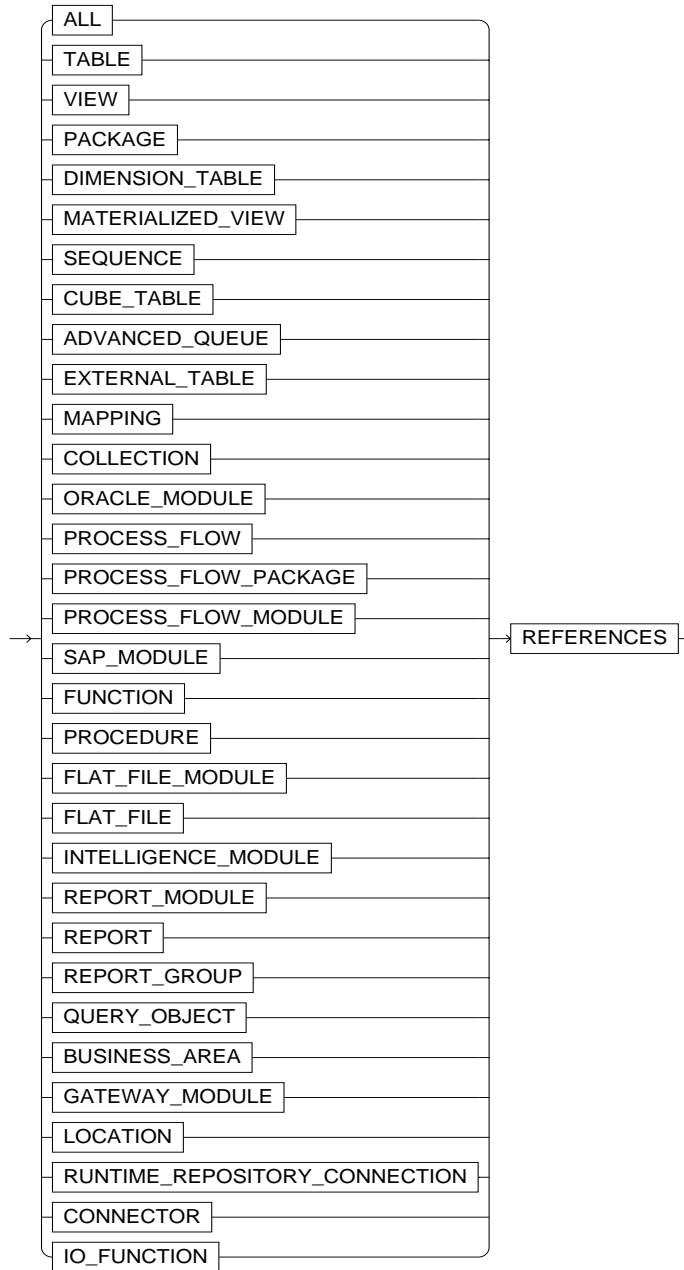
Should be in the context of a project, before retrieving a collection.

### Syntax Diagrams

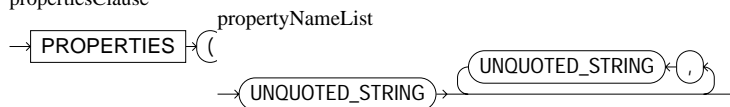
retrieveCollectionCommand



referencesClause



propertiesClause



### Syntax

```
retrieveCollectionCommand = OMBRETRIEVE COLLECTION "QUOTED_STRING"
GET ( ("referencesClause" | "propertiesClause" ));
```

```
referencesClause = ( ALL | TABLE | VIEW | PACKAGE | DIMENSION_TABLE |  
MATERIALIZED_VIEW | SEQUENCE | CUBE_TABLE | ADVANCED_QUEUE |  
EXTERNAL_TABLE | MAPPING | COLLECTION | ORACLE_MODULE |  
PROCESS_FLOW | PROCESS_FLOW_PACKAGE | PROCESS_FLOW_MODULE |  
SAP_MODULE | FUNCTION | PROCEDURE | FLAT_FILE_MODULE | FLAT_FILE  
| INTELLIGENCE_MODULE | REPORT_MODULE | REPORT | REPORT_GROUP |  
QUERY_OBJECT | BUSINESS_AREA | GATEWAY_MODULE | LOCATION |  
RUNTIME_REPOSITORY_CONNECTION | CONNECTOR | IO_FUNCTION )  
REFERENCES;  
propertiesClause = PROPERTIES "(" "propertyNameList" ")";  
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
```

## Keywords and Parameters

retrieveCollectionCommand

Retrieve details regarding a collection of objects.

referencesClause

Specify the type of references to retrieve from the collection.

propertiesClause

Retrieve values for a number of properties.

Basic properties for COLLECTION:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the collection

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the collection

Note:

1. N/A means any valid character in supported character set.
2. " represents an empty string



propertyNameList

Comma separated list of property names to retrieve values. Property names are unquoted.

### Examples

OMBRETRIEVE COLLECTION 'Purchasing Warehouse' GET TABLE REFERENCES

### See Also

OMBLIST

# OMBRETRIEVE CONNECTOR

## Purpose

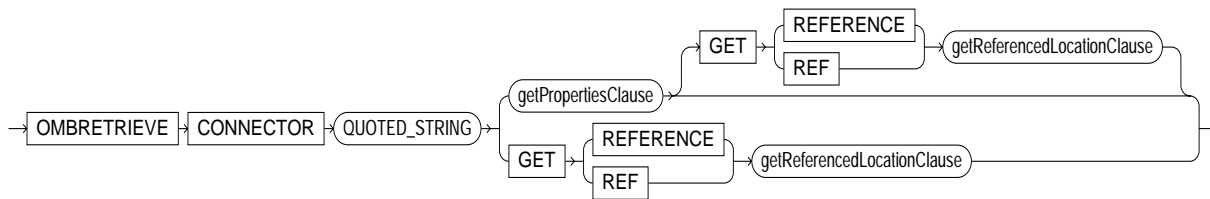
OMBRETRIEVE CONNECTOR - Retrieve details from a connector.

## Prerequisites

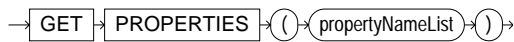
Should be in the context of the connector's owning location.

## Syntax Diagrams

retrieveConnectorCommand



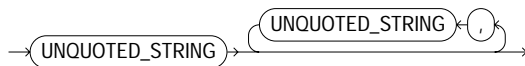
getPropertiesClause



getReferencedLocationClause



propertyNameList



## Syntax

```

retrieveConnectorCommand = OMBRETRIEVE CONNECTOR "QUOTED_STRING" (
  ("getPropertiesClause" [ GET ( REFERENCE | REF ) "getReferencedLocationClause" ]
  ) | GET ( REFERENCE | REF ) "getReferencedLocationClause" );
  
```

```

getPropertiesClause = GET PROPERTIES "(" "propertyNameList" ";";
  
```

```

getReferencedLocationClause = LOCATION;
  
```

```

propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
  
```

## Keywords and Parameters

retrieveConnectorCommand

Retrieve details from the named connector.

getPropertiesClause

Get properties of the connector.

getReferencedLocationClause

Get the name of the location which the connector references.

propertyNameList

The names of the properties whose values you want to set.

Properties for CONNECTOR:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those objects marked as Deployable = true

Note:

1. N/A means any valid character in supported character set.
2. " " represents an empty string

## Examples

```
OMBRETRIEVE CONNECTOR 'A_CONNECTOR' GET PROPERTIES  
(DESCRIPTION, UOID, BUSINESS_NAME) This will retrieve the connector "A_  
CONNECTOR"'s description, uoid, and business name.
```

## See Also

OMBRETRIEVE

# OMBRETRIEVE CUBE\_TABLE

## Purpose

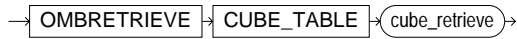
OMBRETRIEVE CUBE\_TABLE - This command retrieves a cube.

## Prerequisites

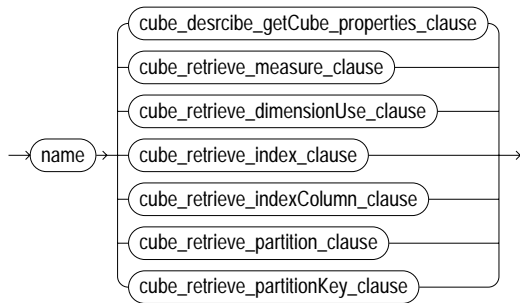
Should be in Oracle Module context.

## Syntax Diagrams

OMBRetrieveCube



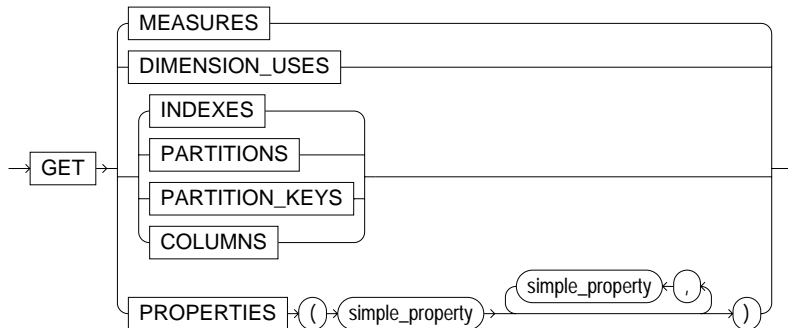
cube\_retrieve



name



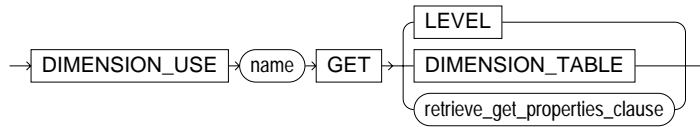
cube\_describe\_getCube\_properties\_clause



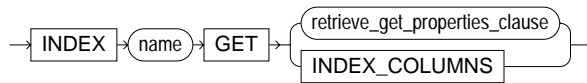
cube\_retrieve\_measure\_clause



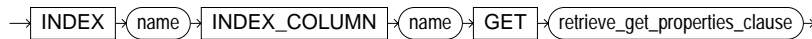
cube\_retrieve\_dimensionUse\_clause



cube\_retrieve\_index\_clause



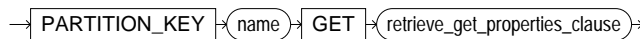
cube\_retrieve\_indexColumn\_clause



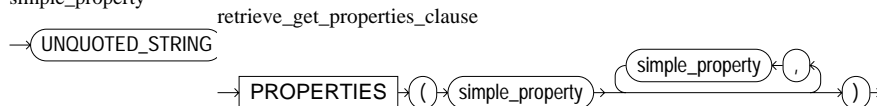
cube\_retrieve\_partition\_clause



cube\_retrieve\_partitionKey\_clause



simple\_property



## Syntax

```

OMBRetrieveCube = OMBRETRIEVE CUBE_TABLE "cube_retrieve";

cube_retrieve = "name" ( "cube_desrcibe_getCube_properties_clause" | "cube_retrieve_measure_clause" | "cube_retrieve_dimensionUse_clause" | "cube_retrieve_index_clause" | "cube_retrieve_indexColumn_clause" | "cube_retrieve_partition_clause" | "cube_retrieve_partitionKey_clause" );

name = ( "QUOTED_STRING" );

cube_desrcibe_getCube_properties_clause = GET ( MEASURES | DIMENSION_USES | ( INDEXES | PARTITIONS | PARTITION_KEYS | COLUMNS ) | ( PROPERTIES "(" ( "simple_property" { "," "simple_property" } ) ) ) );

cube_retrieve_measure_clause = MEASURE "name" GET "retrieve_get_properties_clause";

cube_retrieve_dimensionUse_clause = DIMENSION_USE "name" GET ( LEVEL | DIMENSION_TABLE | "retrieve_get_properties_clause" );

cube_retrieve_index_clause = INDEX "name" GET ( "retrieve_get_properties_clause" | INDEX_COLUMNS );

cube_retrieve_indexColumn_clause = INDEX "name" INDEX_COLUMN "name" GET "retrieve_get_properties_clause";

cube_retrieve_partition_clause = PARTITION "name" GET "retrieve_get_properties_clause";
  
```

```
cube_retrieve_partitionKey_clause = PARTITION_KEY "name" GET "retrieve_get_
properties_clause";
simple_property = "UNQUOTED_STRING";
retrieve_get_properties_clause = PROPERTIES "(" ( "simple_property" { "," "simple_
property" } ) ")";
```

## Keywords and Parameters

OMBRetrieveCube

This command retrieves properties of a cube\_table.

cube\_retrieve

This clause retrieves the properties of a cube\_table, measures, dimensionUses, index, indexColumn, partition, partitionKey.

name

The name has to be a quoted string or an integer, or a decimal number.

cube\_describe\_getCube\_properties\_clause

gets all measures, dimensionUses, partition, partitionKeys etc.

cube\_retrieve\_measure\_clause

This clause gets the measure properties.

cube\_retrieve\_dimensionUse\_clause

This clause retrieves dimensionUses and its properties.

cube\_retrieve\_index\_clause

Get the index properties or IndexColumns.

cube\_retrieve\_indexColumn\_clause

Get the IndexColumn property.

cube\_retrieve\_partition\_clause

Get the partition property.

cube\_retrieve\_partitionKey\_clause

Get the partitionKey property.

simple\_property

gets the simple property.

UNQUOTED\_STRING

Name of the simple property.

retrieve\_get\_properties\_clause

Get the simple properties.

Basic properties for TABLE:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the table

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the table

Name: UOID

Type: STRING

Valid Values: N/A

Default: N/A

Basic properties for INDEX, PARTITION, PARTITION\_KEY, INDEX\_COLUMN:

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the Index, Partition, PartitionKey, IndexColumn

Properties for CUBE\_TABLE:

Name: ANALYZE\_TABLE\_ESTIMATE\_PERCENT

Type: NUMBER

Valid Values: 0 - 100

Default: 99

Value represents the sample size as a percentage of total rows. When set to a nonzero value, Builder generates a DDL script to analyze the table.

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: HASH\_PARTITION\_TABLESPACE\_LIST

Type: STRING

Valid Values: N/A

Default: "

A comma separated list of tablespaces to use for [sub]partition storage.

Name: HASH\_SUBPARTITION\_NUMBER

Type: NUMBER

Valid Values: 2 - 63999

Default: 2

To create Hash partition, specify the number of Hash subpartition.

Name: LOGGING\_MODE

Type: STRING

Valid Values: NOLOGGING, LOGGING

Default: LOGGING

Recovery requirements for a data warehouse : Logging or not logging to Redo Log File.

Name: PARALLEL\_ACCESS\_MODE

Type: STRING

Valid Values: NOPARALLEL, PARALLEL

Default: PARALLEL



---

Enables or disables parallel processing when the table is created. Also enables or disables parallel processing or access.

Name: TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Use the Tablespace parameter to specify the name of tablespace.

Properties for INDEX:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: INDEX\_TYPE

Type: STRING

Valid Values: BITMAP, UNIQUE, NO\_INDEX

Default: UNIQUE

The types of Indexes created on Dimension are BITMAP, UNIQUE or a non-specific index.

Name: LOCAL\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

A local index is constructed so that it reflects the structure of the underlying table. It is equipartitioned with the underlying table, meaning that it is partitioned on the same columns as the underlying table, creates the same number of partitions or

subpartitions, and gives them the same partition bounds as corresponding partitions of the underlying table.

Name: LOGGING\_MODE

Type: STRING

Valid Values: NOLOGGING, LOGGING

Default: LOGGING

Recovery requirements for a data warehouse : Logging or not logging to Redo Log File.

Name: PARALLEL\_ACCESS\_MODE

Type: STRING

Valid Values: NOPARALLEL, PARALLEL

Default: PARALLEL

Enables or disables parallel processing when the table is created. Also enables or disables parallel processing or access.

Name: TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Use the Tablespace parameter to specify the name of tablespace.

Name: USING\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

Use of Index on Dimension Constraint

Properties for PARTITION:

Name: DATE\_LESS\_THAN

Type: STRING

Valid Values: N/A

Default: TO\_DATE('01-JAN-1999', 'DD-MON-YYYY')

Value that represents upper bound of partition stored in warehouse key column for the Days Dimension.

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: EMPTY\_STRING

Use the Tablespace parameter to specify the name of tablespace.

Name: VALUES\_LESS\_THAN

Type: STRING

Valid Values: N/A

Default: TO\_DATE('01-JAN-1999', 'DD-MON-YYYY')

Noninclusive upper bound for the current partition. Type a comma separated list in the same order as the columns in range partitioned key.

Name: VALUE\_LESS\_THAN

Type: STRING

Valid Values: N/A

Default: "

Noninclusive upper bound for the current partition. Type a comma separated list in the same order as the columns in range partitioned key.

Properties for PARTITION\_KEY:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: TYPE

Type: STRING

Valid Values: HASH, RANGE

Default: RANGE

Oracle partitions the storage space and stores rows according to a Hash Algorithm or specified ranges.

Note:

1. N/A means any valid character in supported character set.
2. " represents an empty string

## Examples

```
OMBRETRIEVE CUBE_TABLE 'CUBE1' GET MEASURES OMBRETRIEVE CUBE_
TABLE 'CUBE1' GET DIMENSION_USES
```

## See Also

OMBCREATE CUBE\_TABLE, OMBDROP CUBE\_TABLE, OMBALTER CUBE\_TABLE

## OMBRETRIEVE DEPLOYMENT\_ACTION\_PLAN

### Purpose

OMBRETRIEVE DEPLOYMENT\_ACTION\_PLAN - Retrieve the details of an existing deployment action plan.

### Prerequisites

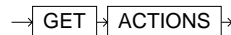
There must be a current working project.

### Syntax Diagrams

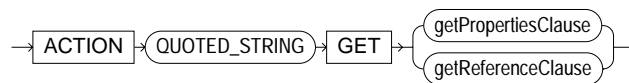
RetrieveActionPlanCommand



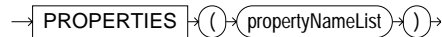
getActionsClause



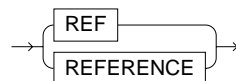
retrieveActionClause



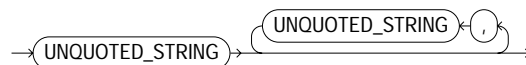
getPropertiesClause



getReferenceClause



propertyNameList



### Syntax

```
RetrieveActionPlanCommand = OMBRETRIEVE ( DEPLOYMENT_ACTION_PLAN )
"QUOTED_STRING" ( "getActionsClause" | "retrieveActionClause" );
```

```
getActionsClause = GET ACTIONS;
```

```
retrieveActionClause = ACTION "QUOTED_STRING" ( GET ( "getPropertiesClause"
| "getReferenceClause" ) );
```

```
getPropertiesClause = PROPERTIES "(" "propertyNameList" ")";
```

```
getReferenceClause = ( REF | REFERENCE );
```

```
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
```

## Keywords and Parameters

RetrieveActionPlanCommand

Retrieve the details of an existing deployment action plan.

getActionsClause

Get a list of actions from an action plan.

retrieveActionClause

Retrieve a set of properties or the associated object of an action.

getPropertiesClause

Retrieve a set of properties that is associated with an action.

PROPERTIES

The only valid property is OPERATION, which specifies the type of action to be taken.

getReferenceClause

Retrieve the object associated with an action.

propertyNameList

Comma separated list of property names. Property names are unquoted.

## Examples

```
OMBRETRIEVE DEPLOYMENT_ACTION_PLAN 'MY_PLAN' GET ACTIONS
OMBRETRIEVE DEPLOYMENT_ACTION_PLAN 'MY_PLAN' ACTION 'MY_VIEW_
CREATE'
GET PROPERTIES (OPERATION)
OMBRETRIEVE DEPLOYMENT_ACTION_PLAN 'MY_PLAN' ACTION 'MY_
TABLE_DEPLOY'
GET REFERENCE
```

## See Also

OMBCREATE DEPLOYMENT\_ACTION\_PLAN, OMBDEPLOY

# OMBRETRIEVE EXTERNAL\_TABLE

## Purpose

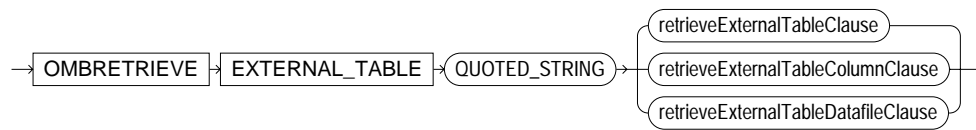
OMBRETRIEVE EXTERNAL\_TABLE - Retrieve details of an external table.

## Prerequisites

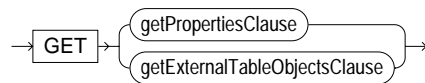
Should be in the context of an Oracle module.

## Syntax Diagrams

retrieveExternalTableCommand



retrieveExternalTableClause



retrieveExternalTableColumnClause



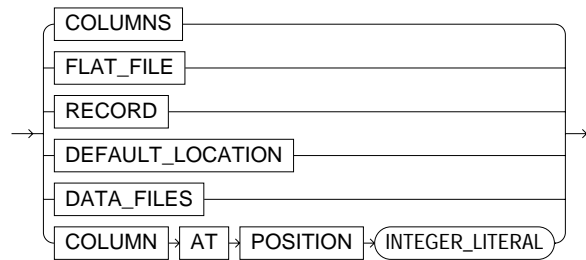
retrieveExternalTableDatafileClause



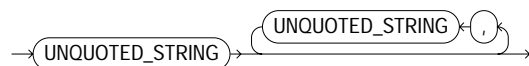
getPropertiesClause



getExternalTableObjectsClause



propertyNameList



## Syntax

```
retrieveExternalTableCommand = OMBRETRIEVE EXTERNAL_TABLE "QUOTED_
STRING" ("retrieveExternalTableClause" | "retrieveExternalTableColumnClause" |
"retrieveExternalTableDatafileClause");

retrieveExternalTableClause = GET ("getPropertiesClause" |
"getExternalTableObjectsClause");

retrieveExternalTableColumnClause = COLUMN "QUOTED_STRING" GET (
"getPropertiesClause" | FIELD);

retrieveExternalTableDatafileClause = DATA_FILE "QUOTED_STRING" GET
"getPropertiesClause";

getPropertiesClause = PROPERTIES "(" "propertyNameList" ")";

getExternalTableObjectsClause = COLUMNS | FLAT_FILE | RECORD | DEFAULT_
LOCATION | DATA_FILES | COLUMN AT POSITION "INTEGER_LITERAL";

propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
```

## Keywords and Parameters

`retrieveExternalTableCommand`

Retrieve details from an external table.

`QUOTED_STRING`

The name of the external table.

`retrieveExternalTableClause`

Retrieve details from an external table.

`retrieveExternalTableColumnClause`

Retrieve details from an external table column.

`retrieveExternalTableDatafileClause`

Retrieve details from one of the external table's data files.

`getPropertiesClause`

Retrieve specified properties.

`getExternalTableObjectsClause`

Retrieve a record reference, flat file reference, columns, or data files.

`propertyNameList`



The names of the properties whose values you want to retrieve.

Properties for EXTERNAL\_TABLE:

Name: BAD\_FILE\_LOCATION

Type: STRING

Valid Values: N/A

Default: USE\_DEFAULT\_LOCATION

Location to use when creating the bad file. When the Access Parameters property is specified for the External Table, this configuration parameter is ignored.

Name: BAD\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: "

Name to use when creating the bad file. When the Access Parameters property is specified for the External Table, this configuration parameter is ignored.

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

When deployable is set to true, a script to create an External Table is generated.

Name: DISCARD\_FILE\_LOCATION

Type: STRING

Valid Values: N/A

Default: USE\_DEFAULT\_LOCATION

Location to use when creating the discard file. When the Access Parameters property is specified for the External Table, this configuration parameter is ignored.

Name: DISCARD\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: "

Name to use when creating the discard file. When the Access Parameters property is specified for the External Table, this configuration parameter is ignored.

Name: ENDIAN

Type: STRING

Valid Values: BIG, LITTLE, PLATFORM

Default: PLATFORM

Data endian should be platform default, little or big. When the Access Parameters property is specified for the External Table, this configuration parameter is ignored.

Name: LOAD\_NULLS\_WHEN\_MISSING\_VALUES

Type: BOOLEAN

Valid Values: true, false

Default: false

If TRUE, then NULLs are loaded for any missing values in the record. If FALSE, then records with missing values are rejected. When the Access Parameters property is specified for the External Table, this configuration parameter is ignored.

Name: LOG\_FILE\_LOCATION

Type: STRING

Valid Values: N/A

Default: USE\_DEFAULT\_LOCATION

Location to use when creating the log file. When the Access Parameters property is specified for the External Table, this configuration parameter is ignored.

Name: LOG\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: "

Name to use when creating the log file. When the Access Parameters property is specified for the External Table, this configuration parameter is ignored.

Name: NUMBER\_OF\_REJECTS\_ALLOWED

Type: NUMBER

Valid Values: 0 - 2147483647

Default: 0

The number of rejects allowed before processing is terminated.

Name: PARALLEL\_ACCESS\_DRIVERS

Type: NUMBER

Valid Values: 1 - 63999

Default: 1

The number of parallel access drivers to enable.

Name: PARALLEL\_ACCESS\_MODE

Type: BOOLEAN

Valid Values: true, false

Default: false

Enable or disable parallel processing.

Name: REJECTS\_ARE\_UNLIMITED

Type: BOOLEAN

Valid Values: true, false

Default: true

Enable or disable limiting the number of rejected records.

Name: STRING\_SIZES\_IN

Type: STRING

Valid Values: CHARACTERS, BYTES

Default: BYTES

String sizes are in bytes or characters. When the Access Parameters property is specified for the External Table, this configuration parameter is ignored.

Name: TRIM

Type: STRING

Valid Values: LEFT, NONE, SQL\*LOADER, BOTH, RIGHT

Default: NONE

Specification from trim option on input fields. When the Access Parameters property is specified for the External Table, this configuration parameter is ignored.

Note:

1. N/A means any valid character in supported character set.
2. " " represents an empty string

## Examples

```
OMBRETRIEVE EXTERNAL_TABLE 'SRC_TABLE' GET PROPERTIES  
(DESCRIPTION, UOID, BUSINESS_NAME) This will retrieve the external table  
"SRC_TABLE"'s description, uoid, and business name.
```

## See Also

OMBRETRIEVE

# OMBRETRIEVE FLAT\_FILE

## Purpose

OMBRETRIEVE FLAT\_FILE - Retrieve details of a flat file.

## Prerequisites

Create and change context to a flat file module.

## Syntax Diagrams

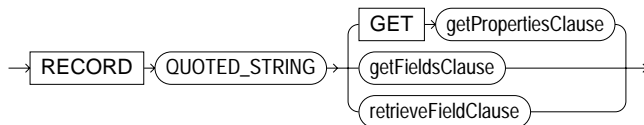
retrieveFlatFileCommand



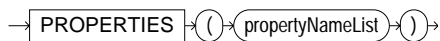
retrieveFlatFileClause



retrieveFlatFileObjectsClause



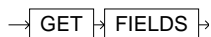
getPropertiesClause



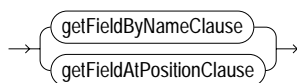
getRecordsClause



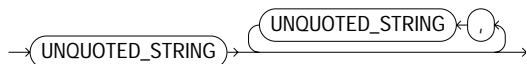
getFieldsClause



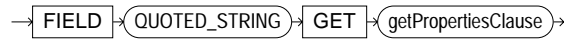
retrieveFieldClause



propertyNameList



getFieldByNameClause



getFieldAtPositionClause



## Syntax

```

retrieveFlatFileCommand = OMBRETRIEVE FLAT_FILE "QUOTED_STRING" (
  "retrieveFlatFileClause" | "retrieveFlatFileObjectsClause");
retrieveFlatFileClause = GET ("getPropertiesClause" | "getRecordsClause");
retrieveFlatFileObjectsClause = RECORD "QUOTED_STRING" (( GET
  "getPropertiesClause") | "getFieldsClause" | "retrieveFieldClause");
getPropertiesClause = PROPERTIES "(" "propertyNameList" ")";
getRecordsClause = RECORDS;
getFieldsClause = GET FIELDS;
retrieveFieldClause = "getFieldByNameClause" | "getFieldAtPositionClause";
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
getFieldByNameClause = FIELD "QUOTED_STRING" GET "getPropertiesClause";
getFieldAtPositionClause = GET FIELD AT POSITION "INTEGER_LITERAL";
  
```

## Keywords and Parameters

**retrieveFlatFileCommand**

Retrieve the details of a flat file.

**QUOTED\_STRING**

The name of a flat file in quotes.

**retrieveFlatFileClause**

Retrieve properties of the flat file.

**retrieveFlatFileObjectsClause**

Retrieve details of the flat file's records and fields.

**getPropertiesClause**

Retrieve specified properties from the flat file, record, or field.

**getRecordsClause**

Retrieve a list of records from the flat file.

**getFieldsClause**

Retrieve a list of fields from the record.

**retrieveFieldClause**

Identify a specific field to retrieve properties from.

**propertyNameList**

The names of the properties whose values you want to retrieve.

Properties for FLAT\_FILE:

Name: DATA\_FILE\_NAME

Type: STRING

Valid Values: N/A Default " The name of the "sampled" file. Also the default data file value used in SQL\*Loader maps and External Tables.

Name: IS\_DELIMITED

Type: BOOLEAN

Valid Values: true, false, 1, 0

Default: true

True indicates that this flat file is delimited. False indicates that its fields are defined by fixed lengths

Name: CHARACTERSET

Type: STRING

Valid Values:

AL24UTFFSS,AR8ARABICMAC,AR8ARABICMACS,AR8ISO8859P6,AR8MSAWIN,A  
R8MSWIN1256,BLT8CP921,BLT8EBCDIC1112,BLT8MSWIN1257,BLT8PC775,CDN8PC  
863,CL8EBCDIC1025,CL8EBCDIC1025X,CL8ISO8859P5,CL8KOI8R,CL8MACCYRILLI  
C,CL8MACCYRILLICS,CL8MSWIN1251,D8EBCDIC273,DK8EBCDIC277,EE8EBCDIC  
870,EE8ISO8859P2,EE8MACCE,EE8MACCES,EE8MACCROATIAN,EE8MACCROATI  
ANS,EE8MSWIN1250,EE8PC852,EL8EBCDIC875,EL8ISO8859P7,EL8MACGREEK,EL8  
MACGREEKS,EL8MSWIN1253,EL8PC437S,EL8PC737,EL8PC869,F8EBCDIC297,I8EBC  
DIC280,IS8MACICELANDIC,IS8MACICELANDICS,IS8PC861,IW8EBCDIC424,IW8IS  
O8859P8,IW8MACHEBREW,IW8MACHEBREWS,IW8MSWIN1255,JA16EBCDIC930,J  
A16EUC,JA16EUCYEN,JA16MACSJS,JA16SJS,JA16SJSYEN,JA16VMS,KO16KSC560  
1,LT8MSWIN921,N8PC865,NEE8ISO8859P4,RU8PC855,RU8PC866,S8EBCDIC278,SE8I  
SO8859P3,TH8MACTHAI,TH8MACTHAIS,TH8TISASCII,TR8EBCDIC1026,TR8MAC  
TURKISH,TR8MACTURKISHS,TR8MSWIN1254,TR8PC857,US7ASCII,US8PC437,UTF  
8,WE8EBCDIC284,WE8EBCDIC285,WE8EBCDIC37,WE8EBCDIC37C,WE8EBCDIC500  
,WE8EBCDIC500C,WE8EBCDIC871,WE8ISO8859P1,WE8ISO8859P9,WE8MACROMA  
N8,WE8MACROMAN8S,WE8MSWIN1252,WE8PC850,WE8PC860,ZHS16CGB231280,  
ZHS16GBK,ZHS16MACCGB231280,ZHT16BIG5,ZHT16MSWIN950,ZHT32EUC

Default: WE8MSWIN1252

The character set of the data file.

Name: RECORD\_DELIMITER

Type: STRING

Valid Values: N/A

Default: "

The character(s) which denote the end of a physical record in a data file. (Please note that this is not the FIELD\_DELIMITER.

Name: RECORD\_LENGTH

Type: NUMBER

Valid Values: 0+

Default: 0 (Records are delimited by default)

The length (in characters) of the records in the data file.

Name: RECORD\_TYPE\_COLUMN\_NUMBER

Type: NUMBER

Valid Values: 0+

Default: 1

The column which contains the record type values for a delimited, multi-record type file.

Name: RECORD\_TYPE\_START\_POSITION

Type: NUMBER

Valid Values: 0+

Default: 0

The starting position of the field (relative to 1) which contains the record type values for a fixed-length, multi-record type file.

Name: RECORD\_TYPE\_END\_POSITION

Type: NUMBER

Valid Values: 0+

Default: 0

The ending position of the field (relative to 1) which contains the record type values for a fixed-length, multi-record type file.

Name: NUMBER\_OF\_RECORDS\_TO\_SKIP

Type: NUMBER

Valid Values: 0 - 1000

Default: 0

The default number of records to skip when loading this file.

Name: FIELD\_DELIMITER

Type: STRING

Valid Values: Any single character

Default: ',' (Comma)

The character to divide the fields in a delimited file.

Name: FIELD\_LEFT\_ENCLOSURE

Type: STRING

Valid Values: Any single character

Default: None

A character to enclose fields which may contain the field delimiter.

Name: FIELD\_RIGHT\_ENCLOSURE

Type: STRING

Valid Values: Any single character

Default: None

A character to enclose fields which may contain the field delimiter.

Name: NUMBER\_OF\_PHYSICAL\_RECORDS\_PER\_LOGICAL

Type: Number

Valid Values: 0+

Default: 0

Set this value if you wish to concatenate a fixed number of physical records to form a single logical record.

Name: CONTINUE\_IF\_ENDS\_WITH

Type: STRING

Valid Values: Any single character

Default: None

Set this value if you wish to concatenate a variable number of physical records to form a single logical record, determined by records ending with this character.

Name: CONTINUE\_IF\_STARTS\_WITH

Type: STRING

Valid Values: N/A

Default: None

Set this value if you wish to concatenate a variable number of physical records to form a single logical record, determined by records beginning with this character.

Properties for RECORD:

Name: RECORD\_TYPE\_VALUE

Type: STRING

Valid Values: N/A

Default: None

This is a mandatory property for each record of a multi-record type file. It is the string which will identify this record type in the data file.

Properties for FIELD:



Name: DATATYPE

Type: STRING

Valid Values: CHAR, DATE, DECIMAL EXTERNAL, FLOAT EXTERNAL, INTEGER EXTERNAL, ZONED EXTERNAL, ZONED

Default: CHAR

This is the SQL\*Loader data type for the field.

Name: LENGTH

Type: NUMBER

Valid Values: 1+

Default: 1

This is the length of the field in a fixed length file. This is the max length of the field in a delimited file.

Name: PRECISION

Type: NUMBER

Valid Values: Depends on data type

Default: 0

Precision of the field.

Name SCALE

Type: NUMBER

Valid Values: Depends on data type

Default: 0

Scale of the field

Name: START\_POSITION

Type: NUMBER

Valid Values: 0+

Default: 0

The starting position of a field for a fixed length file.

Name: SQL\_DATATYPE

Type: STRING

Valid Values: CHAR, DATE, FLOAT, NUMBER, VARCHAR, VARCHAR2, DEFAULT

Default: DEFAULT. This will derive the SQL\_DATATYPE from the value of

DATATYPE. The data type which the field will be treated as in mapping and for External Tables.

Name: SQL\_LENGTH

Type: NUMBER

Valid Values: for 'CHAR' : 1 - 2000 for 'VARCHAR' and 'VARCHAR2' : 1 - 4000

Default: Depends on Sql data type.

Name: SQL\_PRECISION

Type: NUMBER

Valid Values: 1 - 38

Default: 1

Name: SQL\_SCALE

Type: NUMBER

Valid Values: -84 - 127

Default: 0

Name: MASK

Type: STRING

Valid Values: N/A

Default: None

This is the mask used to define the format of DATE fields in the data file.

Name: NULL\_IF

Type: STRING

Valid Values: Either = or != followed by either 'BLANKS', a single quoted string, or a hexadecimal string

Default: None

If this condition is true for a field, the value loaded will be NULL.

Name: DEFAULT\_IF

Type: STRING

Valid Values: Either = or != followed by either 'BLANKS', a single quoted string, or a hexadecimal string

Default: None

If this condition is true for a field, the value loaded will be either NULL or 0, dependent on data type.

`getFieldByNameClause`

Identify a specific field by its name.

`getFieldAtPositionClause`

Identify a specific field by its position in the record.

## Examples

OMBRETRIEVE FLAT\_FILE 'SRC\_FILE' GET PROPERTIES (DESCRIPTION, UOID, BUSINESS\_NAME) This will retrieve the flat file "SRC\_FILE"'s description, uoid, and business name.

OMBRETRIEVE FLAT\_FILE 'SRC\_FILE\_2' GET RECORDS

This will retrieve the records of the flat file "SRC\_FILE\_2".

OMBRETRIEVE FLAT\_FILE 'SRC\_FILE\_3' RECORD 'EMPLOYEE' GET FIELDS

This will retrieve the fields of the "EMPLOYEE" record.

OMBRETRIEVE FLAT\_FILE 'TARGET\_FILE' RECORD 'TARGET\_FILE' GET FIELD AT POSITION 3 GET PROPERTIES (UOID) This will retrieve the UOID of the third field in the single record flat file "TARGET\_FILE".

## See Also

OMBRETRIEVE

## OMBRETRIEVE FLAT\_FILE\_MODULE

### Purpose

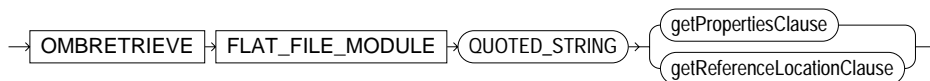
OMBRETRIEVE FLAT\_FILE\_MODULE - Retrieve details from a flat file module.

### Prerequisites

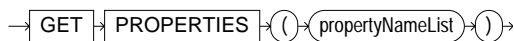
Should be in the context of a project.

### Syntax Diagrams

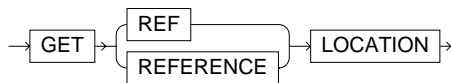
retrieveFlatFileModuleCommand



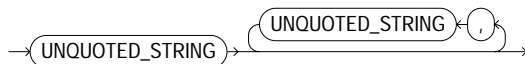
getPropertiesClause



getReferenceLocationClause



propertyNameList



### Syntax

```

retrieveFlatFileModuleCommand = OMBRETRIEVE FLAT_FILE_MODULE
"QUOTED_STRING" ( "getPropertiesClause" | "getReferenceLocationClause" );
getPropertiesClause = GET PROPERTIES "(" "propertyNameList" ";";
getReferenceLocationClause = GET ( REF | REFERENCE ) LOCATION;
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
    
```

### Keywords and Parameters

**retrieveFlatFileModuleCommand**  
 Retrieve details from a flat file module.

**QUOTED\_STRING**  
 The name of the flat file module to retrieve details from.

**getPropertiesClause**  
 Retrieve specified property values from the flat file module.

getReferenceLocationClause

Retrieve the name of the flat file module's location.

propertyNameList

The names of the properties whose values you want to retrieve.

Basic properties for FLAT\_FILE\_MODULE:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: NAME

Business name of the flat file module.

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the flat file module.

Name: UOID

Type: STRING(40)

Valid Values: N/A

Default: N/A

UOID of the flat file module.

## Examples

OMBRETRIEVE FLAT\_FILE\_MODULE 'src\_module' GET PROPERTIES  
(DESCRIPTION, UOID, BUSINESS\_NAME) This will retrieve the flat file module  
"src\_module"'s description, uoid, and business name.

## See Also

OMBRETRIEVE

# OMBRETRIEVE FUNCTION

## Purpose

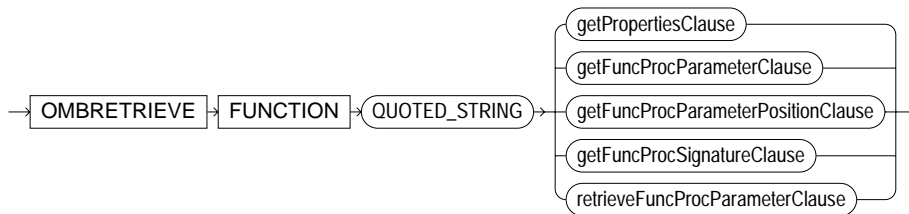
OMBRETRIEVE FUNCTION - Retrieve details of the Function.

## Prerequisites

Should be in the context of a Oracle Module or Package or Transformation Module.

## Syntax Diagrams

retrieveFunctionCommand



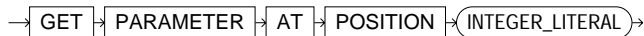
getPropertiesClause



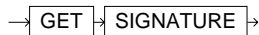
getFuncProcParameterClause



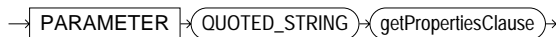
getFuncProcParameterPositionClause



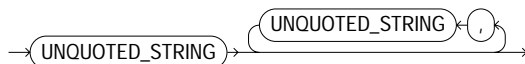
getFuncProcSignatureClause



retrieveFuncProcParameterClause



propertyNameList



## Syntax

```

retrieveFunctionCommand = OMBRETRIEVE FUNCTION "QUOTED_STRING" (
"getPropertiesClause" | "getFuncProcParameterClause" |
"getFuncProcParameterPositionClause" | "getFuncProcSignatureClause" |
"retrieveFuncProcParameterClause" );

getPropertiesClause = GET PROPERTIES "(" "propertyNameList" ");
getFuncProcParameterClause = GET PARAMETERS;
    
```

```

getFuncProcParameterPositionClause = GET PARAMETER AT POSITION
"INTEGER_LITERAL";
getFuncProcSignatureClause = GET SIGNATURE;
retrieveFuncProcParameterClause = PARAMETER "QUOTED_STRING"
"getPropertiesClause";
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };

```

## Keywords and Parameters

retrieveFunctionCommand

This command retrieves the details of a Function

QUOTED\_STRING

Name of the existing Function or path to the Function.

getPropertiesClause

Used to get properties (core, user-defined) for function. Valid properties are shown below:

Basic properties for FUNCTION:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the Function

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the Function

Name: RETURN\_TYPE

Type: STRING

Valid Values: PLS\_INTEGER, BINARY\_INTEGER, BOOLEAN, NUMBER, FLOAT, CHAR, VARCHAR, VARCHAR2, DATE

Default: NUMBER

Set the Return Type for Function

Name: IMPLEMENTATION

Type: STRING

Valid Values: N/A

Default: "

Set the code for Function which is included global variable declaration and code between BEGIN and END.

Name: IS\_DETERMINISTIC

Type: BOOLEAN

Valid Values: true, false

Default: false

This setting helps the optimizer avoid redundant function calls.

Name: IS\_PARALLEL\_ENABLE

Type: BOOLEAN

Valid Values: true, false

Default: false

This option sets flag to a stored function can be used safely in the slave sessions of parallel DML evaluations.

Basic properties for PARAMETER:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the Parameter

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the Parameter

Name: DATATYPE

Type: STRING

Valid Values: PLS\_INTEGER, BINARY\_INTEGER, BOOLEAN, NUMBER, FLOAT, CHAR, VARCHAR, VARCHAR2, DATE



Default: NUMBER

Set the data type for Parameter

Name: IN\_OUT

Type: STRING

Valid Values: IN, OUT, INOUT

Default: 'IN'

Set the parameter mode for Parameter

Name: DEFAULT\_VALUE

Type: STRING

Valid Values: N/A

Default: "

Set the default value for Parameter

Properties for FUNCTION:

Name: AUTHID

Type: STRING

Valid Values: None, Current\_User, Definer

Default: None

Generate the transformation with selected AUTHID option. Function will be executed with the permissions defined by the AUTHID clause rather than the function owner's permissions.

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Note:

1. N/A means any valid character in supported character set.
2. " represents an empty string

getFuncProcParameterClause

Get all the parameter names of the Function

getFuncProcParameterPositionClause  
Get the parameter position of Function

getFuncProcSignatureClause  
Get the complete signature of the Function which includes parameter names, datatype, in/out type and default values

retrieveFuncProcParameterClause  
Get the parameter information such as datatype, default value, in/out type and position

QUOTED\_STRING  
Name of the existing Parameter

propertyNameList  
Comma separated list of property names. Property names are unquoted.

## Examples

**OMBRETRIEVE FUNCTION 'func' GET PROPERTIES (DESCRIPTION, UOID, BUSINESS\_NAME, RETURN\_TYPE, IMPLEMENTATION, IS\_DETERMINISTIC, IS\_PARALLEL\_ENABLE, IS\_IMPORTED)**  
This will retrieve the Function "func's description, uoid, business name, return type, implementation, and boolean values of deterministic parallel\_enable and imported. If Packaged Function is overloaded, first find the Signature by using OMBLIST command, and then use OMBRETRIEVE command using appropriate signature. Example, if OMBLIST FUNCTIONS gives following two signatures, FUNC\_1 (NUMBER) RETURN NUMBER FUNC\_1 (VARCHAR2, NUMBER) RETURN NUMBER The OMBRETRIEVE Syntax to retrieve the first one will be as follows OMBRETRIEVE FUNCTION 'FUNC\_1 \ (NUMBER) RETURN NUMBER' GET PROPERTIES (DESCRIPTION, BUSINESS\_NAME)

## See Also

OMBRETRIEVE

## OMBRETRIEVE GATEWAY\_MODULE

### Purpose

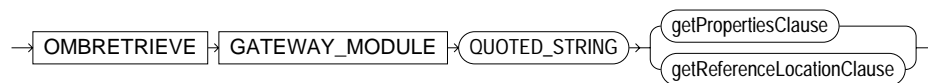
OMBRETRIEVE GATEWAY\_MODULE - Retrieve details of a gateway module.

### Prerequisites

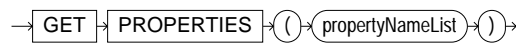
Can be in any context. Use absolute/relative path notation to locate a gateway module.

### Syntax Diagrams

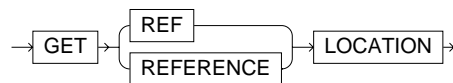
retrieveGatewayModuleCommand



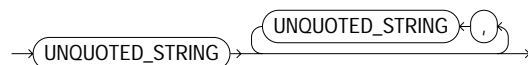
getPropertiesClause



getReferenceLocationClause



propertyNameList



### Syntax

```
retrieveGatewayModuleCommand = OMBRETRIEVE GATEWAY_MODULE
"QUOTED_STRING" ( "getPropertiesClause" | "getReferenceLocationClause" );
```

```
getPropertiesClause = GET PROPERTIES "(" "propertyNameList" ")";
```

```
getReferenceLocationClause = GET ( REF | REFERENCE ) LOCATION;
```

```
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
```

### Keywords and Parameters

retrieveGatewayModuleCommand

Specify the gateway module from which to retrieve details.

getPropertiesClause

Retrieve the properties of a gateway module.

Basic properties for GATEWAY\_MODULE:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the gateway module

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the gateway module

Name: UOID

Type: STRING

Valid Values: N/A

Default: N/A

Universal object identifier

Name: GATEWAY\_TYPE

Type: STRING

Valid Values: N/A

Default: N/A

Type of gateway module

getReferenceLocationClause

Retrieve the name of the location referenced by this gateway module.

propertyNameList

A list of property names.

## Examples

The following line retrieves the description of a gateway module:

OMBRETRIEVE GATEWAY\_MODULE 'INFX1' GET PROPERTIES(DESCRIPTION)

**See Also**

OMBRETRIEVE

## OMBRETRIEVE LOCATION

### Purpose

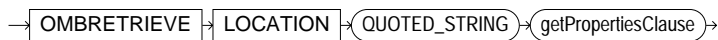
OMBRETRIEVE LOCATION - Retrieve details of the location.

### Prerequisites

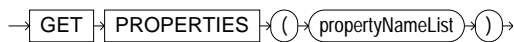
Should be in the context of a project.

### Syntax Diagrams

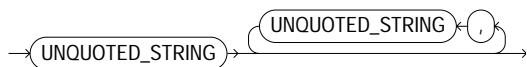
retrieveLocationCommand



getPropertiesClause



propertyNameList



### Syntax

```

retrieveLocationCommand = OMBRETRIEVE LOCATION "QUOTED_STRING"
"getPropertiesClause";
getPropertiesClause = GET PROPERTIES "(" "propertyNameList" ")";
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
  
```

### Keywords and Parameters

retrieveLocationCommand

Retrieve details of the named location.

getPropertiesClause

Get specified properties of the location.

propertyNameList

The names of the properties whose values you want to retrieve.

Properties for LOCATION:

Name: TYPE

Type: STRING

Valid Values: 'Oracle Gateway', 'File System', 'Oracle Database', 'OEM Agent', 'Oracle Workflow', 'SAP'

Default: N/A

The type of system the location represents.

Name: VERSION

Type: STRING

Valid Values:

for 'Oracle Gateway' : '0'

for 'File System' : '0'

for 'Oracle Database' : '9.2','9.0','8.1','8.0','7.3.4'

for 'OEM Agent' : '9.2','9.0'

for 'Oracle Workflow' : '2.6'

for 'SAP' : '4.x','3.x'

Default: N/A

The version of the system(s) the location represents.

Basic properties for LOCATION:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the location.

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the location.

Note:

1. N/A means any valid character in supported character set.
2. " represents an empty string

## Examples

OMBRETRIEVE LOCATION 'A\_LOCATION' GET PROPERTIES (TYPE, VERSION, DESCRIPTION, UOID, BUSINESS\_NAME) This will retrieve the location "A\_LOCATION"'s type, version, description, uoid, and business name.

## See Also

OMBRETRIEVE



# OMBRETRIEVE MAPPING

## Purpose

OMBRETRIEVE MAPPING - Retrieve mapping details such as the number of operators and their connections.

## Prerequisites

The current context must be in an Oracle Module.

## Syntax Diagrams

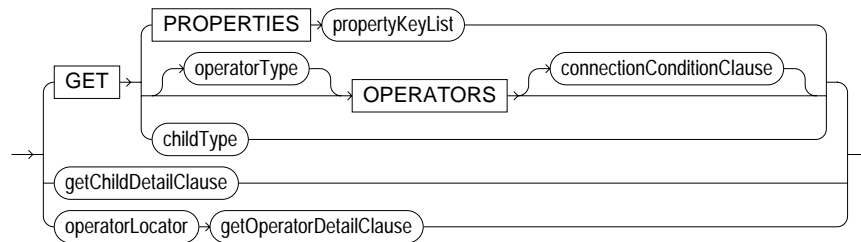
retrieveMappingCommand



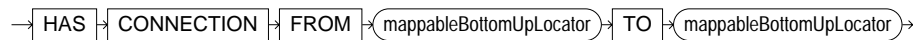
mappingName



getMappingDetailClause



testConnectionClause



propertyKeyList



operatorType



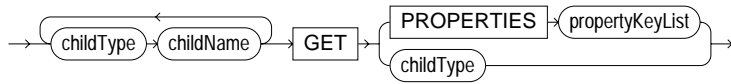
connectionConditionClause



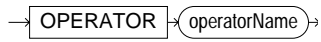
childType



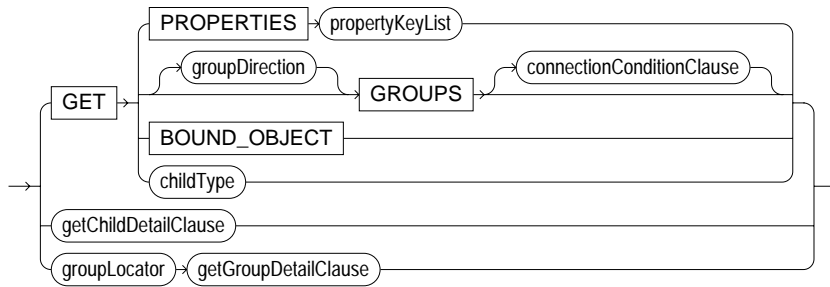
getChildDetailClause



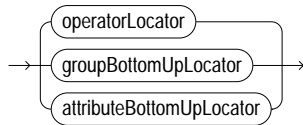
operatorLocator



getOperatorDetailClause



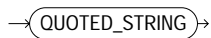
mappableBottomUpLocator



propertyKey



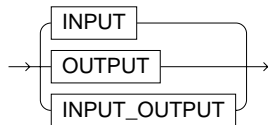
childName



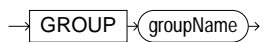
operatorName



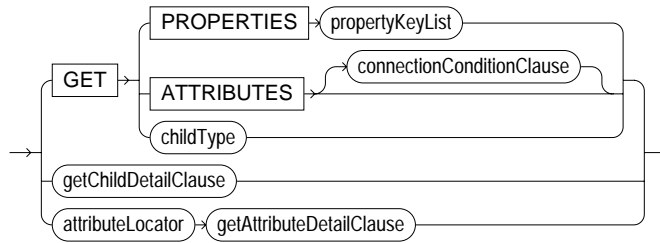
groupDirection



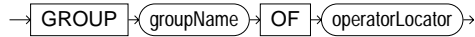
groupLocator



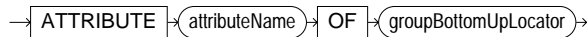
getGroupDetailClause



groupBottomUpLocator



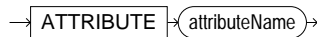
attributeBottomUpLocator



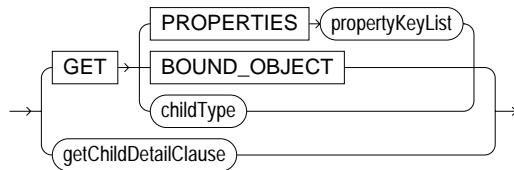
groupName



attributeLocator



getAttributeDetailClause



attributeName



## Syntax

```

retrieveMappingCommand = OMBRETRIEVE MAPPING "mappingName" (
  "getMappingDetailClause" | "testConnectionClause");
mappingName = "QUOTED_STRING";
getMappingDetailClause = GET ( PROPERTIES "propertyKeyList" | [ "operatorType"
] OPERATORS [ "connectionConditionClause" | "childType" ) |
"getChildDetailClause" | "operatorLocator" "getOperatorDetailClause";
testConnectionClause = HAS CONNECTION FROM "mappableBottomUpLocator"
TO "mappableBottomUpLocator";
propertyKeyList = "(" "propertyKey" { "," "propertyKey" } ")";
operatorType = "UNQUOTED_STRING";
connectionConditionClause = CONNECTED ( FROM "mappableBottomUpLocator" |
TO "mappableBottomUpLocator" );
  
```

```
childType = "UNQUOTED_STRING";
getChildDetailClause = ( "childType" "childName" )+ GET ( PROPERTIES
"propertyKeyList" | "childType" );
operatorLocator = OPERATOR "operatorName";
getOperatorDetailClause = GET ( PROPERTIES "propertyKeyList" | [
"groupDirection" ] GROUPS [ "connectionConditionClause" ] | BOUND_OBJECT |
"childType" ) | "getChildDetailClause" | "groupLocator" "getGroupDetailClause";
mappableBottomUpLocator = "operatorLocator" | "groupBottomUpLocator" |
"attributeBottomUpLocator";
propertyKey = "UNQUOTED_STRING";
childName = "QUOTED_STRING";
operatorName = "QUOTED_STRING";
groupDirection = INPUT | OUTPUT | INPUT_OUTPUT;
groupLocator = GROUP "groupName";
getGroupDetailClause = GET ( PROPERTIES "propertyKeyList" | ATTRIBUTES [
"connectionConditionClause" ] | "childType" ) | "getChildDetailClause" |
"attributeLocator" "getAttributeDetailClause";
groupBottomUpLocator = GROUP "groupName" OF "operatorLocator";
attributeBottomUpLocator = ATTRIBUTE "attributeName" OF
"groupBottomUpLocator";
groupName = "QUOTED_STRING";
attributeLocator = ATTRIBUTE "attributeName";
getAttributeDetailClause = GET ( PROPERTIES "propertyKeyList" | BOUND_
OBJECT | "childType" ) | "getChildDetailClause";
attributeName = "QUOTED_STRING";
```

## Keywords and Parameters

retrieveMappingCommand

Retrieve the detail of a mapping such as how many mapping operators are there or which mapping operators are connected to each other.

mappingName

Name of the mapping.

getMappingDetailClause

Get the desired detail of the mapping.

testConnectionClause

Verify if there is a connection between mapping operators, mapping groups or mapping attributes.

**propertyKeyList**

The list of property keys

**operatorType**

Type of a mapping operator. The following operator types are available: ADVANCED\_QUEUE, AGGREGATOR, CONSTANT, CUBE, DATA\_GENERATOR, DEDUPLICATOR, DIMENSION, EXPRESSION, EXTERNAL\_PROCESS, EXTERNAL\_TABLE, FILTER, FLAT\_FILE, INPUT\_PARAMETER, JOINER, KEY\_LOOKUP, MATCHMERGE, MATERIALIZED\_VIEW, NAME\_AND\_ADDRESS, OUTPUT\_PARAMETER, PIVOT, POSTMAPPING\_PROCESS, PREMAPPING\_PROCESS, SEQUENCE, SET\_OPERATION, SORTER, SPLITTER, TABLE, TRANSFORMATION, UNPIVOT, VIEW.

**connectionConditionClause**

List objects only if they are connected from or to objects specified in the connection condition.

**childType**

Type of a child that belongs to map, mapping operator, mapping group or mapping attribute.

**getChildDetailClause**

Get the desired detail of a child object that belongs to the mapping, mapping operator, mapping group or mapping attribute.

**operatorLocator**

Location of a mapping operator.

**getOperatorDetailClause**

Get the desired detail of a mapping operator.

**mappableBottomUpLocator**

Location of the object to be bound to a mapping mapping operator or mapping attribute.

**propertyKey**

A property key for an object.

Basic properties for MAPPING:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the mapping

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the mapping

Basic properties for OPERATOR:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the operator

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the operator

Basic properties for GROUP:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the group

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the group

Basic properties for ATTRIBUTE:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the attribute

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the attribute

Name: DATATYPE

Type: STRING(20)

Valid Values: NUMBER, VARCHAR2, VARCHAR, DATE, LONG

Default: "

Datatype of the Attribute

Name: LENGTH

Type: NUMBER

Valid Values: N/A

Default: 0

Length of the attribute.

Name: PRECISION

Type: NUMBER

Valid Values: N/A

Default: 0

Precision of the attribute.

Name: SCALE

Type: NUMBER

Valid Values: N/A

Default: 0

Scale of the attribute.

Properties for MAPPING:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: STEP\_TYPE

Type: STRING

Valid Values: ABAP, SQLLOADER, PLSQL

Default: UNKNOWN

The step type used to generate this mapping

Properties for MAPPING ABAP STEP:

Name: CONTROL\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: owb.dat

Control File Name

Name: DATA\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: owb.dat

Data File Name

Name: FILE\_DELIMITER\_FOR\_STAGING\_FILE

Type: STRING

Valid Values: N/A

Default: ~



---

**File Delimiter for Staging File**

Name: LOG\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: owb.log

Log File Name

Name: NESTED\_LOOP

Type: STRING

Valid Values: TRUE\_DEFAULT, TRUE, FALSE

Default: TRUE\_DEFAULT

Nested Loop

Name: PRIMARY\_FOREIGN\_KEY\_FOR\_JOIN

Type: STRING

Valid Values: N/A

Default: DEFAULT

Primary Foreign Key for Join

Name: SAP\_SYSTEM\_VERSION

Type: STRING

Valid Values: SAP\_R3\_4X, SAP\_R3\_3X

Default: SAP\_R3\_4X

SAP System Version

Name: SQL\_JOIN\_COLLAPSING

Type: STRING

Valid Values: TRUE\_DEFAULT, TRUE, FALSE

Default: TRUE\_DEFAULT

Sql Join Collapsing

Name: STAGING\_FILE\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: C:\temp\

Staging File Directory

Name: USE\_SELECT\_SINGLE

Type: STRING

Valid Values: TRUE\_DEFAULT, TRUE, FALSE

Default: TRUE\_DEFAULT

Use Select Single

Properties for MAPPING PLSQL STEP:

Name: ANALYZE\_TABLE\_SAMPLE\_PERCENTAGE

Type: NUMBER

Valid Values: N/A

Default: 90

The default percentage of rows to be sampled when the target tables are analyzed for statistics to improve performance during insertion.

Name: ANALYZE\_TABLE\_STATEMENTS

Type: BOOLEAN

Valid Values: true, false

Default: true

Generate statistics collection statement if this is true.

Name: BULK\_PROCESSING\_CODE

Type: BOOLEAN

Valid Values: true, false

Default: true

Generate bulk processing code if this is true.

Name: BULK\_SIZE

Type: NUMBER

Valid Values: N/A

Default: 50

The default number of rows to be fetched in batch during cursor processing.

Name: COMMIT\_FREQUENCY

Type: NUMBER

Valid Values: N/A

Default: 1000

The default number of rows processed before a commit is issued.

Name: CORRELATED\_COMMIT

Type: BOOLEAN

Valid Values: true, false

Default: false

The mapping commits or rolls back correlated rows together.

Name: DEFAULT\_AUDIT\_LEVEL

Type: STRING

Valid Values: NONE, STATISTICS, ERROR\_DETAILS, COMPLETE

Default: ERROR\_DETAILS

The default audit level when the step is executed.

Name: DEFAULT\_OPERATING\_MODE

Type: STRING

Valid Values: SET\_BASED, ROW\_BASED, ROW\_BASED\_TARGET\_ONLY, SET\_BASED\_FAIL\_OVER\_TO\_ROW\_BASED, SET\_BASED\_FAIL\_OVER\_TO\_ROW\_BASED\_TARGET\_ONLY

Default: SET\_BASED\_FAIL\_OVER\_TO\_ROW\_BASED

The default operating mode.

Name: DEFAULT\_PURGE\_GROUP

Type: STRING

Valid Values: N/A

Default: WB

The default purge group to be used when the step is executed.

Name: MAXIMUM\_NUMBER\_OF\_ERRORS

Type: NUMBER

Valid Values: N/A

Default: 50

The default maximum number of errors encountered before aborting the step execution.

Name: OPTIMIZED\_CODE

Type: BOOLEAN

Valid Values: true, false

Default: true

Attempt to generate optimized code if this is true.

Name: PARALLEL\_ROW\_CODE

Type: BOOLEAN

Valid Values: true, false

Default: false

Generate parallel row code if this is true.

Properties for MAPPING SQLLOADER STEP:

Name: AUDIT

Type: BOOLEAN

Valid Values: true, false

Default: true

Perform audit when the step is executed.

Name: BIND\_SIZE

Type: NUMBER

Valid Values: N/A

Default: 50000

Bind Size

Name: CONTINUE\_LOAD

Type: BOOLEAN

Valid Values: true, false

Default: false

Continue Load

Name: CONTROL\_FILE\_LOCATION

Type: STRING

Valid Values: N/A

Default: N/A

The Control File Location

Name: CONTROL\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The control file name used in TCL generation

Name: DATABASE\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Database file name from which to allocate extents

Name: DEFAULT\_PURGE\_GROUP

Type: STRING

Valid Values: N/A

Default: WB

The default purge group to be used when the step is executed.

Name: DELIMITED\_FILE\_RECORD\_TERMINATION

Type: STRING

Valid Values: N/A

Default: N/A

This property has been deprecated. Please set the record delimiter in the Flat File Sample Wizard or Property Sheet.

Name: DIRECT\_MODE

Type: BOOLEAN

Valid Values: true, false

Default: true

Direct Mode

Name: ERRORS\_ALLOWED

Type: NUMBER

Valid Values: N/A

Default: 50

Number of errors to allow

Name: LOG\_FILE\_LOCATION

Type: STRING

Valid Values: N/A

Default: N/A

The log file location.

Name: LOG\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The log file name.

Name: NLS\_CHARACTERSET

Type: STRING

Valid Values: AL24UTFFSS, AR8ARABICMAC, AR8ARABICMACS, AR8ISO8859P6, AR8MSAWIN, AR8MSWIN1256, BLT8CP921, BLT8EBCDIC1112, BLT8MSWIN1257, BLT8PC775, CDN8PC863, CL8EBCDIC1025, CL8EBCDIC1025X, CL8ISO8859P5, CL8KOI8R, CL8MACCYRILLIC, CL8MACCYRILLICS, CL8MSWIN1251, D8EBCDIC273, DK8EBCDIC277, EE8EBCDIC870, EE8ISO8859P2, EE8MACCE, EE8MACCES, EE8MACCROATIAN, EE8MACCROATIANS, EE8MSWIN1250, EE8PC852, EL8EBCDIC875, EL8ISO8859P7, EL8MACGREEK, EL8MACGREEKS, EL8MSWIN1253, EL8PC437S, EL8PC737, EL8PC869, F8EBCDIC297, I8EBCDIC280, IS8MACICELANDIC, IS8MACICELANDICS, IS8PC861, IW8EBCDIC424, IW8ISO8859P8, IW8MACHEBREW, IW8MACHEBREWS, IW8MSWIN1255, JA16EBCDIC930, JA16EUC, JA16EUCYEN, JA16MACSJIS, JA16SJIS, JA16SJISYEN, JA16VMS, KO16KSC5601, LT8MSWIN921, N8PC865, NEE8ISO8859P4, RU8PC855, RU8PC866, S8EBCDIC278, SE8ISO8859P3, TH8MACTHAI, TH8MACTHAIS, TH8TISASCII, TR8EBCDIC1026, TR8MACTURKISH, TR8MACTURKISHS, TR8MSWIN1254, TR8PC857, US7ASCII, US8PC437, UTF8, WE8EBCDIC284, WE8EBCDIC285, WE8EBCDIC37, WE8EBCDIC37C, WE8EBCDIC500, WE8EBCDIC500C, WE8EBCDIC871, WE8ISO8859P1, WE8ISO8859P9, WE8MACROMAN8, WE8MACROMAN8S, WE8MSWIN1252, WE8PC850, WE8PC860, ZHS16CGB231280, ZHS16GBK, ZHS16MACCGB231280, ZHT16BIG5, ZHT16MSWIN950, ZHT32EUC

Default: WE8MSWIN1252

Nls Characterset

Name: OPERATION\_RECOVERABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Operation Recoverable

Name: PERFORM\_PARALLEL\_LOAD

Type: BOOLEAN  
Valid Values: true, false  
Default: false  
Perform Parallel Load

Name: PRESERVE\_BLANKS  
Type: BOOLEAN  
Valid Values: true, false  
Default: false  
Preserve Blanks

Name: READ\_BUFFERS  
Type: NUMBER  
Valid Values: N/A  
Default: 4  
The Number of Buffers

Name: READ\_SIZE  
Type: NUMBER  
Valid Values: N/A  
Default: 65536  
The size of the read buffer

Name: RECORDS\_TO\_LOAD  
Type: NUMBER  
Valid Values: N/A  
Default: N/A  
Number of logical records to load. The default value of 0 indicates to load all records.

Name: RECORDS\_TO\_SKIP  
Type: NUMBER  
Valid Values: N/A  
Default: N/A  
Number of logical records to skip

Name: ROWS\_PER\_COMMIT  
Type: NUMBER

Valid Values: N/A

Default: 200

Rows per Commit

Properties for MAPPING SQLLOADER STEP SOURCE\_DATA\_FILE CHILD:

Name: BAD\_FILE\_LOCATION

Type: STRING

Valid Values: N/A

Default: N/A

Bad file location.

Name: BAD\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Bad file name. If the Bad File Location is set then this must be a relative file name. Otherwise this should contain a fully qualified path.

Name: DATA\_FILE\_LOCATION

Type: STRING

Valid Values: N/A

Default: N/A

The location for this component.

Name: DATA\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The data file name for this component. If the Data File Location is set then this must be a relative file name. Otherwise this should contain a fully qualified path.

Name: DISCARD\_FILE\_LOCATION

Type: STRING

Valid Values: N/A

Default: N/A

Discard file location



Name: DISCARD\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Discard file name. If the Discard File Location is set then this must be a relative file name. Otherwise this should contain a fully qualified path.

Name: DISCARD\_MAX

Type: NUMBER

Valid Values: N/A

Default: 0

Discard Max

Properties for ADVANCED\_QUEUE OPERATOR:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Properties for AGGREGATOR OPERATOR:

Name: GROUP\_BY\_CLAUSE

Type: STRING

Valid Values: N/A

Default: N/A

The Group By clause for the aggregation

Name: HAVING\_CLAUSE

Type: STRING

Valid Values: N/A

Default: N/A

The Having clause for the aggregation

Properties for CUBE OPERATOR:

Name: `ADVANCED_MATCH_BY_CONSTRAINT`

Type: `STRING`

Valid Values: `N/A`

Default: `No constraints`

This property is used by public API and scripting to influence how columns are used for UPDATE or DELETE DMLs. If this property is set with the name of a primary or unique key, all the columns in the key will be used for matching during UPDATE or DELETE; and all the columns not in the key are used for loading. The property can also be assigned the value "All constraints" or "No constraints". If the DML type is INSERT, TRUNCATE/INSERT, or CHECK/INSERT, setting this property causes no effect.

Name: `BOUND_NAME`

Type: `STRING`

Valid Values: `N/A`

Default: `N/A`

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: `DATABASE_FILE_NAME`

Type: `STRING`

Valid Values: `N/A`

Default: `N/A`

Database file name to allocate extents from

Name: `DATABASE_LINK`

Type: `STRING`

Valid Values: `N/A`

Default: `N/A`

The database link used to access this entity during mapping.

Name: `DATA_COLLECTION_FREQUENCY`

Type: `STRING`

Valid Values: `UNKNOWN, YEAR, QUARTER, MONTH, DAY, HOUR, MINUTE`

Default: `UNKNOWN`

New Data Granularity

Name: `DIRECT`

Type: `BOOLEAN`

Valid Values: true, false

Default: false

Directly swap source into target as a partition without first creating a staging table.

Name: ENABLE\_CONSTRAINTS

Type: BOOLEAN

Valid Values: true, false

Default: true

Enable Constraints

Name: EXCEPTIONS\_TABLE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Exceptions Table Name

Name: EXTRACTION\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when extracting from this table using SQL

Name: LOADING\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when loading into this table using SQL

Name: LOADING\_TYPE

Type: STRING

Valid Values: INSERT, UPDATE, INSERT\_UPDATE, UPDATE\_INSERT, DELETE, NONE, TRUNCATE\_INSERT, DELETE\_INSERT, CHECK\_INSERT

Default: INSERT

The loading operation to be performed when this is a target.

Name: MATCH\_BY\_CONSTRAINT

Type: STRING

Valid Values: N/A

Default: YES

A property to indicate whether unique or primary key information on this target will override the matching criteria obtained from Match by constraint property on the attributes of this target.

Name: PARTITION\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Partition Name

Name: PEL\_ENABLED

Type: BOOLEAN

Valid Values: true, false

Default: false

PEL Enabled

Name: PRIMARY\_SOURCE

Type: STRING

Valid Values: YES, NO

Default: NO

A boolean value to indicate whether this is a primary source or not (only used in EDW).

Name: RECORDS\_TO\_SKIP

Type: NUMBER

Valid Values: N/A

Default: N/A

Number of records to skip

Name: REPLACE\_DATA

Type: BOOLEAN

Valid Values: true, false

Default: false

Replace existing data in target partition if there is any.

Name: SCHEMA

Type: STRING  
Valid Values: N/A  
Default: N/A  
Schema

Name: SINGLEROW  
Type: BOOLEAN  
Valid Values: true, false  
Default: false  
Singlerow

Name: SORTED\_INDEXES\_CLAUSE  
Type: STRING  
Valid Values: N/A  
Default: N/A  
Sorted Indexes Clause

Name: SUBPARTITION\_NAME  
Type: STRING  
Valid Values: N/A  
Default: N/A  
Subpartition Name

Name: TARGET\_FILTER\_FOR\_DELETE  
Type: STRING  
Valid Values: N/A  
Default: N/A

A condition on the rows in the target and if evaluated to true, that row will participate in the delete loading operation.

Name: TARGET\_FILTER\_FOR\_UPDATE  
Type: STRING  
Valid Values: N/A  
Default: N/A

A condition on the rows in the target and if evaluated to true, that row will participate in the update loading operation.

Name: TRAILING\_NULLCOLS

Type: BOOLEAN

Valid Values: true, false

Default: false

Trailing Nullcols

Properties for DIMENSION OPERATOR:

Name: ADVANCED\_MATCH\_BY\_CONSTRAINT

Type: STRING

Valid Values: N/A

Default: No constraints

This property is used by public API and scripting to influence how columns are used for UPDATE or DELETE DMLs. If this property is set with the name of a primary or unique key, all the columns in the key will be used for matching during UPDATE or DELETE; and all the columns not in the key are used for loading. The property can also be assigned the value "All constraints" or "No constraints". If the DML type is INSERT, TRUNCATE/INSERT, or CHECK/INSERT, setting this property causes no effect.

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DATABASE\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Database file name to allocate extents from

Name: DATABASE\_LINK

Type: STRING

Valid Values: N/A

Default: N/A

The database link used to access this entity during mapping.

Name: DATA\_COLLECTION\_FREQUENCY

Type: STRING

Valid Values: UNKNOWN, YEAR, QUARTER, MONTH, DAY, HOUR, MINUTE

Default: UNKNOWN

New Data Granularity

Name: DIRECT

Type: BOOLEAN

Valid Values: true, false

Default: false

Directly swap source into target as a partition without first creating a staging table.

Name: ENABLE\_CONSTRAINTS

Type: BOOLEAN

Valid Values: true, false

Default: true

Enable Constraints

Name: EXCEPTIONS\_TABLE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Exceptions Table Name

Name: EXTRACTION\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when extracting from this table using SQL

Name: LOADING\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when loading into this table using SQL

Name: LOADING\_TYPE

Type: STRING

Valid Values: INSERT, UPDATE, INSERT\_UPDATE, UPDATE\_INSERT, DELETE, NONE, TRUNCATE\_INSERT, DELETE\_INSERT, CHECK\_INSERT

Default: INSERT

The loading operation to be performed when this is a target.

Name: MATCH\_BY\_CONSTRAINT

Type: STRING

Valid Values: N/A

Default: YES

A property to indicate whether unique or primary key information on this target will override the matching criteria obtained from Match by constraint property on the attributes of this target.

Name: PARTITION\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Partition Name

Name: PEL\_ENABLED

Type: BOOLEAN

Valid Values: true, false

Default: false

PEL Enabled

Name: PRIMARY\_SOURCE

Type: STRING

Valid Values: YES, NO

Default: NO

A boolean value to indicate whether this is a primary source or not (only used in EDW).

Name: RECORDS\_TO\_SKIP

Type: NUMBER

Valid Values: N/A

Default: N/A

Number of records to skip



Name: REPLACE\_DATA

Type: BOOLEAN

Valid Values: true, false

Default: false

Replace existing data in target partition if there is any.

Name: SCHEMA

Type: STRING

Valid Values: N/A

Default: N/A

Schema

Name: SINGLEROW

Type: BOOLEAN

Valid Values: true, false

Default: false

Singlerow

Name: SORTED\_INDEXES\_CLAUSE

Type: STRING

Valid Values: N/A

Default: N/A

Sorted Indexes Clause

Name: SUBPARTITION\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Subpartition Name

Name: TARGET\_FILTER\_FOR\_DELETE

Type: STRING

Valid Values: N/A

Default: N/A

A condition on the rows in the target and if evaluated to true, that row will participate in the delete loading operation.

Name: TARGET\_FILTER\_FOR\_UPDATE

Type: STRING

Valid Values: N/A

Default: N/A

A condition on the rows in the target and if evaluated to true, that row will participate in the update loading operation.

Name: TRAILING\_NULLCOLS

Type: BOOLEAN

Valid Values: true, false

Default: false

Trailing Nullcols

Properties for EXTERNAL\_TABLE OPERATOR:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DATABASE\_LINK

Type: STRING

Valid Values: N/A

Default: N/A

The database link used to access this entity during mapping.

Name: EXTRACTION\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when extracting from this table using SQL

Name: LOADING\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

---

Hint used when loading into this table using SQL

Name: PRIMARY\_SOURCE

Type: STRING

Valid Values: YES, NO

Default: NO

A boolean value to indicate whether this is a primary source or not (only used in EDW).

Name: SCHEMA

Type: STRING

Valid Values: N/A

Default: N/A

Schema

Properties for FILTER\_OPERATOR:

Name: FILTER\_CONDITION

Type: STRING

Valid Values: N/A

Default: N/A

The boolean filtering condition that identifies what data is to be processed. Any row with a false condition will be ignored.

Properties for FLAT\_FILE\_OPERATOR:

Name: CONCATENATE\_RECORDS

Type: NUMBER

Valid Values: N/A

Default: 0

Number of Physical Records per Logical Record.

Name: CONTINUATION\_CHARACTER

Type: STRING

Valid Values: N/A

Default: N/A

Character that indicates the record is continued on the next line.

Name: CONTINUATION\_CHARACTER\_ON\_NEXT\_LINE

Type: BOOLEAN

Valid Values: true, false

Default: FALSE

If there is a continuation character, is it at the start of the line.

Name: FIELD\_ENCLOSURE\_CHARACTERS

Type: STRING

Valid Values: N/A

Default: N/A

Characters that wrap fields. Example ' or ".

Name: FIELD\_NAMES\_IN\_THE\_FIRST\_ROW

Type: BOOLEAN

Valid Values: true, false

Default: FALSE

Indicates whether to write the field names in the first row of the output file.

Name: FIELD\_TERMINATION\_CHARACTER

Type: STRING

Valid Values: N/A

Default: ,

Character that separates the fields of a delimited file.

Name: FILE\_FORMAT

Type: STRING

Valid Values: N/A

Default: DELIMITED

File Format (Fixed or Delimited).

Name: LOADING\_TYPE

Type: STRING

Valid Values: INSERT, UPDATE, NONE

Default: INSERT

The loading operation to be performed

Name: NLS\_CHARACTERSET

Type: STRING  
Valid Values: N/A  
Default: WE8MSWIN1252  
NLS Characterset

Name: RECORD\_DELIMITER  
Type: STRING  
Valid Values: N/A  
Default: N/A  
Character that indicates the end of the record.

Name: RECORD\_SIZE  
Type: NUMBER  
Valid Values: N/A  
Default: 0  
Size of a fixed length record.

Name: RECORD\_TYPE\_LENGTH  
Type: NUMBER  
Valid Values: N/A  
Default: 0  
If this is a multi record file, this will indicate the length of the data that identifies the type of record. It is used with the Record Type Position.

Name: RECORD\_TYPE\_POSITION  
Type: NUMBER  
Valid Values: N/A  
Default: 0  
If this is a multi record file, this will indicate the position of the field that identifies the type of record.

Name: SAMPLED\_FILE\_NAME  
Type: STRING  
Valid Values: N/A  
Default: N/A  
The name of the file that was sampled to get the metadata for this file.

Name: TARGET\_DATA\_FILE\_LOCATION

Type: STRING

Valid Values: N/A

Default: N/A

The name of the location in which to open the target data file. Make sure the root path of this location, as registered in the Runtime Platform, is exactly specified in the initialization file (INIT.ORA) of your runtime database using the UTL\_FILE\_DIR parameter.

Name: TARGET\_DATA\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Name of the target data file, including extension (file type). This name should not include the file path. To specify where the target data file will be created/appended, set the Target Data File Location.

Properties for INPUT\_PARAMETER OPERATOR:

Properties for JOINER OPERATOR:

Name: JOIN\_CONDITION

Type: STRING

Valid Values: N/A

Default: N/A

The Join Condition for the join operator

Properties for KEY\_LOOKUP OPERATOR:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DATABASE\_LINK

Type: STRING

Valid Values: N/A

Default: N/A

---

The database link used to access this entity during mapping.

Name: LOOKUP\_CONDITION

Type: STRING

Valid Values: N/A

Default: N/A

Key lookup condition based on the source inputs. This condition is used to lookup a value in the bound table. If the condition is not met, the default value expression will be returned. If a default expression is not defined, null is used.

Name: SCHEMA

Type: STRING

Valid Values: N/A

Default: N/A

Schema

Properties for MATCHMERGE OPERATOR:

Name: MATCH\_KEYS

Type: STRING

Valid Values: N/A

Default: N/A

Ordered list of attributes that control the set of records to be matched at any at any particular time.

Name: MERGED\_PREFIX

Type: STRING

Valid Values: N/A

Default: N/A

Set the prefix used for the merged attributes in the cross-reference group.

Properties for MATERIALIZED\_VIEW OPERATOR:

Name: ADVANCED\_MATCH\_BY\_CONSTRAINT

Type: STRING

Valid Values: N/A

Default: No constraints

This property is used by public API and scripting to influence how columns are used for UPDATE or DELETE DMLs. If this property is set with the name of a primary or unique key, all the columns in the key will be used for matching during UPDATE or DELETE; and all the columns not in the key are used for loading. The property can also be assigned the value "All constraints" or "No constraints". If the DML type is INSERT, TRUNCATE/INSERT, or CHECK/INSERT, setting this property causes no effect.

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DATABASE\_LINK

Type: STRING

Valid Values: N/A

Default: N/A

The database link used to access this entity during mapping.

Name: EXTRACTION\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when extracting from this table using SQL

Name: LOADING\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when loading into this table using SQL

Name: LOADING\_TYPE

Type: STRING

Valid Values: INSERT, UPDATE, INSERT\_UPDATE, UPDATE\_INSERT, DELETE, NONE, TRUNCATE\_INSERT, DELETE\_INSERT, CHECK\_INSERT

Default: INSERT

The loading operation to be performed when this is a target.



Name: MATCH\_BY\_CONSTRAINT

Type: STRING

Valid Values: N/A

Default: YES

A property to indicate whether unique or primary key information on this target will override the matching criteria obtained from Match by constraint property on the attributes of this target.

Name: PRIMARY\_SOURCE

Type: STRING

Valid Values: YES, NO

Default: NO

A boolean value to indicate whether this is a primary source or not (only used in EDW).

Name: SCHEMA

Type: STRING

Valid Values: N/A

Default: N/A

Schema

Name: TARGET\_FILTER\_FOR\_DELETE

Type: STRING

Valid Values: N/A

Default: N/A

A condition on the rows in the target and if evaluated to true, that row will participate in the delete loading operation.

Name: TARGET\_FILTER\_FOR\_UPDATE

Type: STRING

Valid Values: N/A

Default: N/A

A condition on the rows in the target and if evaluated to true, that row will participate in the update loading operation.

Properties for NAME\_AND\_ADDRESS OPERATOR:

Name: ADDRESS\_LINE\_1

Type: STRING

Valid Values: N/A

Default: N/A

Address line 1

Name: ADDRESS\_LINE\_2

Type: STRING

Valid Values: N/A

Default: N/A

Address line 2

Name: ADDRESS\_LINE\_3

Type: STRING

Valid Values: N/A

Default: N/A

Address line 3

Name: ADDRESS\_LINE\_4

Type: STRING

Valid Values: N/A

Default: N/A

Address line 4

Name: DUAL\_ADDRESS\_ASSIGNMENT

Type: STRING

Valid Values: NA\_DUALADDR\_STREET, NA\_DUALADDR\_POBOX, NA\_DUALADDR\_CLOSESTTOLASTLINE

Default: NA\_DUALADDR\_STREET

A dual address refers to two address lines for the same destination. For example, a record contains both a street address and a P.O. Box; this is common with business data. Select which of the two address lines should be assigned in these cases.

Name: GENERATE\_CASS\_REPORT

Type: STRING

Valid Values: NA\_YES, NA\_NO

Default: NA\_NO

Select "Yes" to generate CASS (Coding Accuracy Support System) report. CASS report is a text file specified by the United States Postal Service. The report is written to the [nas/bin/admin/reports] folder under the home folder of the name/address server.

Name: LIST\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The list name is optional and provides a reference for tracking multiple CASS reports.

Name: PARSING\_TYPE

Type: STRING

Valid Values: NA\_NAMEONLY, NA\_ADDRESSONLY, NA\_NAMEANDADDRESS

Default: NA\_NAMEANDADDRESS

Select a name-address parsing type to be performed on the input data

Name: PRIMARY\_COUNTRY

Type: STRING

Valid Values: NA\_ARG, NA\_AUS, NA\_BEL, NA\_BRA, NA\_CAN, NA\_CHL, NA\_COL, NA\_DNK, NA\_FRA, NA\_DEU, NA\_HKG, NA\_IND, NA\_IRL, NA\_ITA, NA\_MEX, NA\_MYS, NA\_NLD, NA\_NZL, NA\_PER, NA\_PHL, NA\_PRT, NA\_SGP, NA\_ZAF, NA\_ESP, NA\_SWE, NA\_CHE, NA\_ARE, NA\_GBR, NA\_USA, NA\_VEN

Default: NA\_USA

Select the primary parsing country which best represents the input data. Input addresses having the same country as the primary parsing country will only need to be parsed once. Input addresses having a different country than the primary parsing country may be reparsed by a different parser. For performance reasons, it is best to minimize the percentage of 2-pass parses by selecting the optimal parser.

Name: PROCESSOR\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The processor name is the name of the organization submitting the CASS report.

Properties for OUTPUT\_PARAMETER OPERATOR:

Properties for PIVOT OPERATOR:

Name: PIVOT\_GROUP\_SIZE

Type: NUMBER

Valid Values: 1 - 1000

Default: 2

A number specifying the pivot group size for the pivot operation. Pivot group size determines the number of output rows that are produced from each input row.

Properties for POSTMAPPING\_PROCESS OPERATOR:

Name: FUNCTION\_CALL

Type: STRING

Valid Values: N/A

Default: N/A

Expression template for procedure call

Name: FUNCTION\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Name of the transformation to be called.

Name: POSTMAPPING\_PROCESS\_RUN\_CONDITION

Type: STRING

Valid Values: ALWAYS, ON\_SUCCESS, ON\_ERROR, ON\_WARNING

Default: ON\_SUCCESS

Indicates under what condition of the mapping the post-mapping process will be run.

Name: ROWBASED\_ONLY

Type: BOOLEAN

Valid Values: true, false

Default: N/A

Indicates if this transformation must be used only Row Based mode. Some transformations can be used in SQL mode as well as Row Based mode.

Properties for PREMAPPING\_PROCESS OPERATOR:

Name: FUNCTION\_CALL

Type: STRING

Valid Values: N/A

Default: N/A

Expression template for procedure call

Name: FUNCTION\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Name of the transformation to be called.

Name: MAPPING\_RUN\_CONDITION

Type: STRING

Valid Values: ALWAYS, ON\_SUCCESS, ON\_ERROR

Default: ON\_SUCCESS

Indicates under what condition of the pre-mapping process the mapping will be run.

Name: ROWBASED\_ONLY

Type: BOOLEAN

Valid Values: true, false

Default: N/A

Indicates if this transformation must be used only Row Based mode. Some transformations can be used in SQL mode as well as Row Based mode.

Properties for SEQUENCE OPERATOR:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DATABASE\_LINK

Type: STRING

Valid Values: N/A

Default: N/A

The database link used to access this entity during mapping.

Name: SCHEMA

Type: STRING

Valid Values: N/A

Default: N/A

## Schema

### Properties for SET\_OPERATION OPERATOR:

Name: SET\_OPERATION

Type: STRING

Valid Values: UNION, UNIONALL, INTERSECT, MINUS

Default: UNION

Specifies the set operation that is to be performed by this operator.

### Properties for SORTER OPERATOR:

Name: ORDER\_BY\_CLAUSE

Type: STRING

Valid Values: N/A

Default: N/A

The Order By Clause

### Properties for TABLE OPERATOR:

Name: ADVANCED\_MATCH\_BY\_CONSTRAINT

Type: STRING

Valid Values: N/A

Default: No constraints

This property is used by public API and scripting to influence how columns are used for UPDATE or DELETE DMLs. If this property is set with the name of a primary or unique key, all the columns in the key will be used for matching during UPDATE or DELETE; and all the columns not in the key are used for loading. The property can also be assigned the value "All constraints" or "No constraints". If the DML type is INSERT, TRUNCATE/INSERT, or CHECK/INSERT, setting this property causes no effect.

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DATABASE\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Database file name to allocate extents from

Name: DATABASE\_LINK

Type: STRING

Valid Values: N/A

Default: N/A

The database link used to access this entity during mapping.

Name: DATA\_COLLECTION\_FREQUENCY

Type: STRING

Valid Values: UNKNOWN, YEAR, QUARTER, MONTH, DAY, HOUR, MINUTE

Default: UNKNOWN

New Data Granularity

Name: DIRECT

Type: BOOLEAN

Valid Values: true, false

Default: false

Directly swap source into target as a partition without first creating a staging table.

Name: ENABLE\_CONSTRAINTS

Type: BOOLEAN

Valid Values: true, false

Default: true

Enable Constraints

Name: EXCEPTIONS\_TABLE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Exceptions Table Name

Name: EXTRACTION\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when extracting from this table using SQL

Name: LOADING\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when loading into this table using SQL

Name: LOADING\_TYPE

Type: STRING

Valid Values: INSERT, UPDATE, INSERT\_UPDATE, UPDATE\_INSERT, DELETE, NONE, TRUNCATE\_INSERT, DELETE\_INSERT, CHECK\_INSERT

Default: INSERT

The loading operation to be performed when this is a target.

Name: MATCH\_BY\_CONSTRAINT

Type: STRING

Valid Values: N/A

Default: YES

A property to indicate whether unique or primary key information on this target will override the matching criteria obtained from Match by constraint property on the attributes of this target.

Name: PARTITION\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Partition Name

Name: PEL\_ENABLED

Type: BOOLEAN

Valid Values: true, false

Default: false

PEL Enabled



Name: PRIMARY\_SOURCE

Type: STRING

Valid Values: YES, NO

Default: NO

A boolean value to indicate whether this is a primary source or not (only used in EDW).

Name: RECORDS\_TO\_SKIP

Type: NUMBER

Valid Values: N/A

Default: N/A

Number of records to skip

Name: REPLACE\_DATA

Type: BOOLEAN

Valid Values: true, false

Default: false

Replace existing data in target partition if there is any.

Name: SCHEMA

Type: STRING

Valid Values: N/A

Default: N/A

Schema

Name: SINGLEROW

Type: BOOLEAN

Valid Values: true, false

Default: false

Singlerow

Name: SORTED\_INDEXES\_CLAUSE

Type: STRING

Valid Values: N/A

Default: N/A

Sorted Indexes Clause

Name: SUBPARTITION\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Subpartition Name

Name: TARGET\_FILTER\_FOR\_DELETE

Type: STRING

Valid Values: N/A

Default: N/A

A condition on the rows in the target and if evaluated to true, that row will participate in the delete loading operation.

Name: TARGET\_FILTER\_FOR\_UPDATE

Type: STRING

Valid Values: N/A

Default: N/A

A condition on the rows in the target and if evaluated to true, that row will participate in the update loading operation.

Name: TRAILING\_NULLCOLS

Type: BOOLEAN

Valid Values: true, false

Default: false

Trailing Nullcols

Properties for TABLE\_FUNCTION OPERATOR:

Name: TABLE\_FUNCTION\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name of the Table Function

Properties for TRANSFORMATION OPERATOR:

Name: FUNCTION\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Name of the transformation to be called.

Name: ROWBASED\_ONLY

Type: BOOLEAN

Valid Values: true, false

Default: N/A

Indicates if this transformation must be used only Row Based mode. Some transformations can be used in SQL mode as well as Row Based mode.

Properties for VIEW OPERATOR:

Name: ADVANCED\_MATCH\_BY\_CONSTRAINT

Type: STRING

Valid Values: N/A

Default: No constraints

This property is used by public API and scripting to influence how columns are used for UPDATE or DELETE DMLs. If this property is set with the name of a primary or unique key, all the columns in the key will be used for matching during UPDATE or DELETE; and all the columns not in the key are used for loading. The property can also be assigned the value "All constraints" or "No constraints". If the DML type is INSERT, TRUNCATE/INSERT, or CHECK/INSERT, setting this property causes no effect.

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DATABASE\_FILE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Database file name to allocate extents from

Name: DATABASE\_LINK

Type: STRING

Valid Values: N/A

Default: N/A

The database link used to access this entity during mapping.

Name: EXTRACTION\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when extracting from this table using SQL

Name: LOADING\_HINT

Type: STRING

Valid Values: N/A

Default: N/A

Hint used when loading into this table using SQL

Name: LOADING\_TYPE

Type: STRING

Valid Values: INSERT, UPDATE, INSERT\_UPDATE, UPDATE\_INSERT, DELETE, NONE, TRUNCATE\_INSERT, DELETE\_INSERT, CHECK\_INSERT

Default: INSERT

The loading operation to be performed when this is a target.

Name: MATCH\_BY\_CONSTRAINT

Type: STRING

Valid Values: N/A

Default: YES

A property to indicate whether unique or primary key information on this target will override the matching criteria obtained from Match by constraint property on the attributes of this target.

Name: PARTITION\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Partition Name

Name: PRIMARY\_SOURCE

Type: STRING

Valid Values: YES, NO

Default: NO

A boolean value to indicate whether this is a primary source or not (only used in EDW).

Name: RECORDS\_TO\_SKIP

Type: NUMBER

Valid Values: N/A

Default: N/A

Number of records to skip

Name: SCHEMA

Type: STRING

Valid Values: N/A

Default: N/A

Schema

Name: SINGLEROW

Type: BOOLEAN

Valid Values: true, false

Default: false

Singlerow

Name: SORTED\_INDEXES\_CLAUSE

Type: STRING

Valid Values: N/A

Default: N/A

Sorted Indexes Clause

Name: SUBPARTITION\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Subpartition Name

Name: TARGET\_FILTER\_FOR\_DELETE

Type: STRING

Valid Values: N/A

Default: N/A

A condition on the rows in the target and if evaluated to true, that row will participate in the delete loading operation.

Name: TARGET\_FILTER\_FOR\_UPDATE

Type: STRING

Valid Values: N/A

Default: N/A

A condition on the rows in the target and if evaluated to true, that row will participate in the update loading operation.

Name: TRAILING\_NULLCOLS

Type: BOOLEAN

Valid Values: true, false

Default: false

Trailing Nullcols

Properties for CUBE,DIMENSION,KEY\_LOOKUP,MATERIALIZED\_VIEW,TABLE,VIEW OPERATOR KEYS\_READONLY CHILD:

Name: KEY\_COLUMNS

Type: STRING

Valid Values: N/A

Default: N/A

Local columns that define this key (Comma separated if more than one).

Name: KEY\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Name of the primary, foreign or unique key (primary, foreign, or unique).

Name: KEY\_TYPE

Type: STRING

Valid Values: N/A

Default: UNIQUE

Type of key - primary, foreign or unique.

Name: REFERENCED\_KEYS

Type: STRING

Valid Values: N/A

Default: N/A

If the key is a foreign key, this will contain the key or keys used of the referenced object.

Properties for MATCHMERGE OPERATOR MATCH\_RULES CHILD:

Name: ADDRESS\_ADDRESS\_LINE\_SIMILARITY\_SCORE

Type: NUMBER

Valid Values: N/A

Default: N/A

Similarity score for address line in Address Match Rule.

Name: ADDRESS\_ALLOW\_DIFFERING\_SECONDARY\_ADDRESSES

Type: BOOLEAN

Valid Values: true, false

Default: false

Allow differing secondary addresses to match in Address Match Rule.

Name: ADDRESS\_LAST\_LINE\_SIMILARITY\_SCORE

Type: NUMBER

Valid Values: N/A

Default: N/A

Similarity score for last line in Address Match Rule.

Name: ADDRESS\_MATCH\_ON\_ADDRESS\_LINE\_SIMILARITY

Type: BOOLEAN

Valid Values: true, false

Default: false

Check address line similarity in Address Match Rule.

Name: ADDRESS\_MATCH\_ON\_BLANK\_SECONDARY\_ADDRESSES

Type: BOOLEAN

Valid Values: true, false

Default: false

Match on blank secondary address in Address Match Rule.

Name: ADDRESS\_MATCH\_ON\_LAST\_LINE\_SIMILARITY

Type: BOOLEAN

Valid Values: true, false

Default: false

Check last line similarity in Address Match Rule.

Name: ADDRESS\_MATCH\_ON\_STREET\_OR\_PO\_BOX

Type: BOOLEAN

Valid Values: true, false

Default: false

Match on Street or Post Office (PO) Box in Address Match Rule.

Name: CUSTOM\_RULE

Type: STRING

Valid Values: N/A

Default: N/A

Custom Merge Rule

Name: DESCRIPTION

Type: STRING

Valid Values: N/A

Default: N/A

Description of match rule.

Name: FIRM\_CROSS\_MATCH\_FIRM1\_AND\_FIRM2

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a firm match rule, should Warehouse Builder cross match firm 1 and firm 2?

Name: FIRM\_MATCH\_ON\_ABBREVIATIONS

Type: BOOLEAN

Valid Values: true, false



Default: false

If this is a firm match rule, should Warehouse Builder match on abbreviations?

Name: FIRM\_MATCH\_ON\_ACRONYMS

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a firm match rule, should Warehouse Builder match on acronyms?

Name: FIRM\_MATCH\_ON\_PARTIAL\_NAMES

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a firm match rule, should Warehouse Builder match on partial names?

Name: FIRM\_MATCH\_ON\_SIMILARITY

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a firm match rule, should Warehouse Builder check firm similarity?

Name: FIRM\_SIMILARITY\_SCORE

Type: NUMBER

Valid Values: N/A

Default: N/A

Similarity score for firm in Firm Match Rule.

Name: FIRM\_STRIP\_NOISE\_WORDS

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a firm match rule, should Warehouse Builder strip noise words?

Name: PERSON\_DETECT\_SWITCHED\_NAME\_ORDER

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person rule, should Warehouse Builder detect the switched name order.

Name: PERSON\_FN\_DETECT\_COMPOUND\_NAME

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder detect compound names?

Name: PERSON\_FN\_MATCH\_ON\_INITIALS

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder match on initials.

Name: PERSON\_FN\_MATCH\_ON\_SIMILARITY

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder match on first name similarity?

Name: PERSON\_FN\_MATCH\_ON\_SOUNDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder match on first name soundex?

Name: PERSON\_FN\_MATCH\_ON\_SUBSTRINGS

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder match on first name substrings?

Name: PERSON\_FN\_MRS\_MATCH

Type: BOOLEAN

Valid Values: true, false

Default: false

---

If this is a person match rule, should Warehouse Builder detect MRS? (For example, should Mrs John Smith match Mrs Smith)

Name: PERSON\_FN\_SIMILARITY\_SCORE

Type: NUMBER

Valid Values: N/A

Default: N/A

If matching in first name similarity, what is the similarity score?

Name: PERSON\_LN\_DETECT\_MISSING\_HYPHEN

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder detect missing hyphens?

Name: PERSON\_LN\_MATCH\_HYPHENATED\_NAMES

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder match on hyphenated last name?

Name: PERSON\_LN\_MATCH\_ON\_SIMILARITY

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder match on last name similarity?

Name: PERSON\_LN\_MATCH\_ON\_SOUNDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder match on last name soundex?

Name: PERSON\_LN\_SIMILARITY\_SCORE

Type: NUMBER

Valid Values: N/A

Default: N/A

If Warehouse Builder is matching on last name similarity, what is the similarity score?

Name: PERSON\_MN\_MATCH\_ON\_INITIALS

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder match on middle name initials?

Name: PERSON\_MN\_MATCH\_ON\_SIMILARITY

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder match on middle name similarity?

Name: PERSON\_MN\_MATCH\_ON\_SOUNDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder match on middle name soundex?

Name: PERSON\_MN\_MATCH\_ON\_SUBSTRINGS

Type: BOOLEAN

Valid Values: true, false

Default: false

If this is a person match rule, should Warehouse Builder match on middle name substrings?

Name: PERSON\_MN\_SIMILARITY\_SCORE

Type: NUMBER

Valid Values: N/A

Default: N/A

If Warehouse Builder is matching on middle name similarity, what is the similarity score?

Name: TOTAL\_WEIGHT\_SCORE

Type: NUMBER

Valid Values: N/A

Default: N/A

If this is a weight rule, the sum of all weights must be equal to or greater than this score for the records to pass the weght rule.

Name: TYPE

Type: STRING

Valid Values: MM\_TRUE, MM\_FALSE, MM\_WEIGHT, MM\_CONDITIONAL, MM\_CUSTOM, MM\_PERSON, MM\_FIRM, MM\_ADDRESS

Default: MM\_TRUE

What type of match rule is this? Possible types are: MM\_TRUE, MM\_FALSE, MM\_WEIGHT, MM\_CONDITIONAL, MM\_CUSTOM, MM\_PERSON, MM\_FIRM, MM\_ADDRESS

Name: USAGE

Type: STRING

Valid Values: MM\_ACTIVE, MM\_PASSIVE

Default: MM\_ACTIVE

Is this an active or passive rule? Possible usages are: MM\_ACTIVE, MM\_PASSIVE.

Properties for MATCHMERGE OPERATOR MERGE\_RULES CHILD:

Name: ATTRIBUTE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Merge Attribute

Name: COPY\_ATTRIBUTE

Type: STRING

Valid Values: N/A

Default: N/A

Copy this merged attribute to merge rule target attribute.

Name: CUSTOM\_TEXT

Type: STRING

Valid Values: N/A

Default: N/A

Implementation text for custom merge rule. Include "BEGIN and END statements.

Name: DESCRIPTION

Type: STRING

Valid Values: N/A

Default: N/A

Description

Name: MATCH\_ID\_SEQUENCE\_MODULE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Physical sequence module name for match id merge rule. This sequence will be used to generate the match id.

Name: MATCH\_ID\_SEQUENCE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Physical sequence name for match id merge rule. This sequence will be used to generate the match id.

Name: MIN\_MAX\_ATTRIBUTE

Type: STRING

Valid Values: N/A

Default: N/A

Selecting attribute for min/max merge rule

Name: MIN\_MAX\_TYPE

Type: STRING

Valid Values: MM\_MIN, MM\_MAX, MM\_SHORTEST, MM\_LONGEST

Default: MM\_MAX

Select record where attribute is min,max, shortest, longest. Possible values are MM\_MIN, MM\_MAX, MM\_SHORTEST, MM\_LONGEST.

Name: SEQUENCE\_MODULE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Physical sequence module name for sequence merge rule.

Name: SEQUENCE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Physical sequence name for sequence merge rule.

Name: TYPE

Type: STRING

Valid Values: MM\_ANY, MM\_MATCH\_ID, MM\_RANK, MM\_SEQUENCE, MM\_MIN\_MAX, MM\_COPY, MM\_CUSTOM, MM\_RECORD\_ANY, MM\_RECORD\_RANK, MM\_RECORD\_MIN\_MAX, MM\_RECORD\_CUSTOM

Default: MM\_ANY

Merge Rule Type. Possible values are:MM\_ANY,MM\_MATCH\_ID,MM\_RANK,MM\_SEQUENCE,MM\_MIN\_MAX,MM\_COPY,MM\_CUSTOM,MM\_RECORD\_ANY,MM\_RECORD\_RANK,MM\_RECORD\_MIN\_MAX,MM\_RECORD\_CUSTOM.

Properties for MATCHMERGE OPERATOR MATCH\_RULES CHILD ADDRESS\_ROLES CHILD:

Name: ATTRIBUTE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

What is the attribute for this role?

Name: TYPE

Type: STRING

Valid Values: MM\_PRIMARY\_ADDR, MM\_UNIT\_NUM, MM\_PO\_BOX, MM\_DUAL\_PRIMARY\_ADDR, MM\_DUAL\_UNIT\_NUM, MM\_DUAL\_PO\_BOX, MM\_CITY, MM\_STATE, MM\_POSTAL\_CODE, MM\_IS\_FOUND

Default: MM\_PRIMARY\_ADDR

What role is this attribute? Possible values are: MM\_PRIMARY\_ADDR, MM\_UNIT\_NUM, MM\_PO\_BOX, MM\_DUAL\_PRIMARY\_ADDR, MM\_DUAL\_UNIT\_NUM, MM\_DUAL\_PO\_BOX, MM\_CITY, MM\_STATE, MM\_POSTAL\_CODE, MM\_IS\_FOUND.

Properties for MATCHMERGE OPERATOR MATCH\_RULES CHILD CONDITIONS CHILD:

Name: ALGORITHM

Type: STRING

Valid Values: MM\_EXACT, MM\_STD\_EXACT, MM\_SOUNDEX, MM\_SIMILARITY, MM\_STD\_SIMILARITY, MM\_PARTIAL\_NAME, MM\_ABBREVIATION, MM\_ACRONYM

Default: MM\_EXACT

Algorithm of this condition. Possible values are: MM\_EXACT, MM\_STD\_EXACT, MM\_SOUNDEX, MM\_SIMILARITY, MM\_STD\_SIMILARITY, MM\_PARTIAL\_NAME, MM\_ABBREVIATION, MM\_ACRONYM.

Name: ATTRIBUTE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

To which attribute does this condition apply?

Name: BLANK\_MATCHING

Type: STRING

Valid Values: MM\_MATCH\_BOTH\_BLANK, MM\_MATCH\_EITHER\_BLANK, MM\_NO\_MATCH\_IF\_BLANK

Default: MM\_MATCH\_BOTH\_BLANK

How do you want blanks to be handled? Possible values are: MM\_MATCH\_BOTH\_BLANK, MM\_MATCH\_EITHER\_BLANK, MM\_NO\_MATCH\_IF\_BLANK.

Name: SIMILARITY\_SCORE

Type: NUMBER

Valid Values: N/A

Default: N/A

If this is a similarity condition, what score must the similarity equal or exceed for records to meet the condition?

Properties for MATCHMERGE OPERATOR MATCH\_RULES CHILD FIRM\_ROLES CHILD:

Name: ATTRIBUTE\_NAME

Type: STRING

Valid Values: N/A



Default: N/A

What is the attribute for this role?

Name: TYPE

Type: STRING

Valid Values: MM\_FIRM1, MM\_FIRM2

Default: MM\_FIRM1

What role is this attribute? Possible values are: MM\_FIRM1,MM\_FIRM2.

Properties for MATCHMERGE OPERATOR MATCH\_RULES CHILD PERSON\_ROLES CHILD:

Name: ATTRIBUTE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

What is the attribute for this role.

Name: TYPE

Type: STRING

Valid Values: MM\_PRENAME, MM\_FIRST\_NAME\_STD, MM\_MIDDLE\_NAME\_STD, MM\_MIDDLE\_NAME\_2\_STD, MM\_MIDDLE\_NAME\_3\_STD, MM\_LAST\_NAME, MM\_MATURITY\_POST\_NAME

Default: MM\_PRENAME

What role is this attribute? Possible values are: MM\_PRENAME, MM\_FIRST\_NAME\_STD, MM\_MIDDLE\_NAME\_STD, MM\_MIDDLE\_NAME\_2\_STD, MM\_MIDDLE\_NAME\_3\_STD, MM\_LAST\_NAME, MM\_MATURITY\_POST\_NAME.

Properties for MATCHMERGE OPERATOR MATCH\_RULES CHILD WEIGHTS CHILD:

Name: BLANK\_SCORE

Type: NUMBER

Valid Values: N/A

Default: N/A

Score if either attribute is blank

Name: WEIGHT\_ATTRIBUTE

Type: STRING

Valid Values: N/A

Default: N/A

Attribute

Name: WEIGHT\_SCORE

Type: NUMBER

Valid Values: N/A

Default: N/A

Score used for this weight if the attributes in the two match records are identical. Similarity will be used to generate this score.

Properties for MATCHMERGE OPERATOR MERGE\_RULES CHILD ATTRIBUTES CHILD:

Name: ATTRIBUTE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

Record merge rule attribute

Properties for MATCHMERGE OPERATOR MERGE\_RULES CHILD RANK\_RULES CHILD:

Name: EXPRESSION

Type: STRING

Valid Values: N/A

Default: N/A

Rank expression.

Properties for FLAT\_FILE OPERATOR INOUT GROUP:

Name: RECORD\_TYPE\_VALUES

Type: STRING

Valid Values: N/A

Default: N/A

Record Type Values.

Properties for JOINER OPERATOR INPUT GROUP:

---

Properties for PIVOT OPERATOR OUTPUT GROUP:

Name: ROW\_LOCATOR

Type: STRING

Valid Values: N/A

Default: N/A

An expression indicating which attribute within the output group is the row locator.

Properties for SPLITTER OPERATOR OUTPUT GROUP:

Name: SPLIT\_CONDITION

Type: STRING

Valid Values: N/A

Default: N/A

Condition that defines when to perform the attribute maps for the attributes in this group.

Properties for TABLE\_FUNCTION OPERATOR INPUT GROUP:

Name: INPUT\_PARAMETER\_TYPE

Type: STRING

Valid Values: REF\_CURSOR, SCALAR

Default: REF\_CURSOR

This property specifies whether the input parameter is a scalar or a ref cursor type

Name: PARAMETER\_POSITION

Type: NUMBER

Valid Values: N/A

Default: N/A

The position of the argument in the table function signature corresponding to this parameter group

Properties for TABLE\_FUNCTION OPERATOR OUTPUT GROUP:

Name: RETURN\_TABLE\_OF\_SCALAR

Type: BOOLEAN

Valid Values: true, false

Default: false

This property specifies whether the return of the table function is a TABLE of SCALAR or not.

Properties for UNPIVOT OPERATOR INPUT GROUP:

Name: ROW\_LOCATOR

Type: STRING

Valid Values: N/A

Default: N/A

An expression indicating which attribute within the input group is the row locator.

Name: ROW\_LOCATOR\_VALUES

Type: STRING

Valid Values: N/A

Default: NULL, NULL

A comma-separated expressions that gives the possible values of the row locator within a unpivot group.

Properties for ADVANCED\_QUEUE OPERATOR INOUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Properties for CUBE OPERATOR INOUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: LOAD\_COLUMN\_WHEN\_INSERTING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will participate in the insert load operation.

Name: LOAD\_COLUMN\_WHEN\_UPDATING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will participate in the update load operation.

Name: MATCH\_COLUMN\_WHEN\_DELETING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will be used to construct the matching criteria between the incoming data and the existing data on the target during the delete load operation.

Name: MATCH\_COLUMN\_WHEN\_UPDATING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will be used to construct the matching criteria between the incoming data and the existing data on the target during the update load operation.

Name: UPDATE\_OPERATION

Type: STRING

Valid Values: =, +=, -=, =-, =|, ||=

Default: =

The computation to be performed on this attribute between the incoming data and the existing data on the target during the update load operation.

Properties for DIMENSION OPERATOR INOUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: LOAD\_COLUMN\_WHEN\_INSERTING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will participate in the insert load operation.

Name: LOAD\_COLUMN\_WHEN\_UPDATING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will participate in the update load operation.

Name: MATCH\_COLUMN\_WHEN\_DELETING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will be used to construct the matching criteria between the incoming data and the existing data on the target during the delete load operation.

Name: MATCH\_COLUMN\_WHEN\_UPDATING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will be used to construct the matching criteria between the incoming data and the existing data on the target during the update load operation.

Name: UPDATE\_OPERATION

Type: STRING

Valid Values: =, +=, -=, =-, =||, ||=

Default: =

The computation to be performed on this attribute between the incoming data and the existing data on the target during the update load operation.

Properties for EXTERNAL\_TABLE OPERATOR INOUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Properties for INPUT\_PARAMETER OPERATOR OUTPUT GROUP ATTRIBUTE:

Name: DEFAULT\_VALUE

Type: STRING

Valid Values: N/A

Default: N/A

The default value applies when the map is not given a value for this attribute. The value can be specified only by the calling program.

Properties for KEY\_LOOKUP OPERATOR OUTPUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DEFAULT\_VALUE

Type: STRING

Valid Values: N/A

Default: N/A

The default value used for this attribute if no key lookup table row exists for a given input row.

Properties for MATCHMERGE OPERATOR OUTPUT GROUP ATTRIBUTE:

Name: MERGE\_ATTR

Type: STRING

Valid Values: N/A

Default: N/A

Related merge attribute

Properties for MATERIALIZED\_VIEW OPERATOR INOUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: LOAD\_COLUMN\_WHEN\_INSERTING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will participate in the insert load operation.

Name: LOAD\_COLUMN\_WHEN\_UPDATING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will participate in the update load operation.

Name: MATCH\_COLUMN\_WHEN\_DELETING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will be used to construct the matching criteria between the incoming data and the existing data on the target during the delete load operation.

Name: MATCH\_COLUMN\_WHEN\_UPDATING\_ROW



Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will be used to construct the matching criteria between the incoming data and the existing data on the target during the update load operation.

Name: UPDATE\_OPERATION

Type: STRING

Valid Values: =, +=, -=, =-, =| |, || =

Default: =

The computation to be performed on this attribute between the incoming data and the existing data on the target during the update load operation.

Properties for NAME\_AND\_ADDRESS OPERATOR INPUT GROUP ATTRIBUTE:

Name: INPUT\_ROLE

Type: STRING

Valid Values: NA\_NONE, NA\_FIRSTNAME, NA\_MIDDLENAME, NA\_MIDDLENAME2, NA\_MIDDLENAME3, NA\_LASTNAME, NA\_FIRSTPARTNAME, NA\_LASTPARTNAME, NA\_PRENAME, NA\_POSTNAME, NA\_PERSON, NA\_PERSON2, NA\_PERSON3, NA\_FIRMNAME, NA\_PRIMARYADDRESS, NA\_SECONDARYADDRESS, NA\_ADDRESS, NA\_ADDRESS2, NA\_NEIGHBORHOOD, NA\_LASTLINE, NA\_CITY, NA\_STATE, NA\_POSTALCODE, NA\_COUNTRYNAME, NA\_COUNTRYCODE, NA\_LINE1, NA\_LINE2, NA\_LINE3, NA\_LINE4, NA\_LINE5, NA\_LINE6, NA\_LINE7, NA\_LINE8, NA\_LINE9, NA\_LINE10

Default: NA\_NONE

Assigns a name-address input role to the selected input attribute

Properties for NAME\_AND\_ADDRESS OPERATOR OUTPUT GROUP ATTRIBUTE:

Name: ADDRESS\_TYPE

Type: STRING

Valid Values: NA\_ADDRTYPE\_NORMAL, NA\_ADDRTYPE\_DUAL

Default: NA\_ADDRTYPE\_NORMAL

You can designate an address type as Normal or Dual. For example, a dual address occurs when a record contains both a street address and a P.O. Box; this is common with business data. A normal address contains only one type of address.

Name: INSTANCE

Type: STRING

Valid Values: NA\_INSTANCE\_FIRST, NA\_INSTANCE\_SECOND, NA\_INSTANCE\_THIRD, NA\_INSTANCE\_FOURTH, NA\_INSTANCE\_FIFTH, NA\_INSTANCE\_SIXTH

Default: NA\_INSTANCE\_FIRST

The instance option is used when an address contains multiple names, you can specify which name in the group should be used. In addition, you can use this option to assign an address type to a miscellaneous address component.

Name: OUTPUT\_COMPONENT

Type: STRING

Valid Values: NA\_NONE, NA\_PRENAME, NA\_FIRSTNAMESTD, NA\_MIDDLENAMESTD, NA\_MIDDLENAME2STD, NA\_MIDDLENAME3STD, NA\_POSTNAME, NA\_OTHERPOSTNAME, NA\_NAMEDESIGNATOR, NA\_RELATIONSHIP, NA\_PERSON, NA\_FIRSTNAME, NA\_MIDDLENAME, NA\_MIDDLENAME2, NA\_MIDDLENAME3, NA\_LASTNAME, NA\_GENDER, NA\_PERSONCOUNT, NA\_FIRMNAME, NA\_FIRMCOUNT, NA\_ADDRESS, NA\_PRIMARYADDRESS, NA\_STREETNUMBER, NA\_PREDIRECTIONAL, NA\_STREETNAME, NA\_STREETTYPE, NA\_POSTDIRECTIONAL, NA\_SECONDARYADDRESS, NA\_UNITDESIGNATOR, NA\_UNITNUMBER, NA\_BOXNAME, NA\_BOXNUMBER, NA\_ROUTENAME, NA\_ROUTENUMBER, NA\_BUILDINGNAME, NA\_COMPLEX, NA\_MISCADDRESS, NA\_LASTLINE, NA\_NEIGHBORHOOD, NA\_CITY, NA\_STATE, NA\_POSTALCODE, NA\_POSTALCODEFORMATTED, NA\_DELIVERYPOINT, NA\_COUNTRYCODE, NA\_COUNTRYCODE3, NA\_COUNTRYNAM, NA\_ISGOODGROUP, NA\_ISPARSED, NA\_PARSESTATUS, NA\_PARSESTATUSDESC, NA\_ISGOODNAME, NA\_NAMEWARNING, NA\_ISGOODADDRESS, NA\_ISFOUND, NA\_CITYMATCH, NA\_STREETNAMEMATCH, NA\_STREETNUMBERMATCH, NA\_STREETCOMPMATCH, NA\_NONAMBIGUOUSMATCH, NA\_CITYWARNING, NA\_STREETWARNING, NA\_ISADDRESSVERIFIABLE, NA\_ADDRESSCORRECTED, NA\_POSTALCODECORRECTED, NA\_CITYCORRECTED, NA\_STREETCORRECTED, NA\_STREETCOMPCORRECTED, NA\_ADDRESSTYPE, NA\_PARSINGCOUNTRY, NA\_INSTALLATIONTYPE, NA\_INSTALLATIONNAME, NA\_DELIVERYOFFICECODE, NA\_DELIVERYBEATCODE, NA\_ADDRESS2, NA\_LOCALITYCODE, NA\_LOCALITYNAME, NA\_COUNTYNAME, NA\_ZIP5, NA\_ZIP4, NA\_URBANIZATIONNAME, NA\_LACS, NA\_CART, NA\_CHECKDIGIT, NA\_MSA, NA\_MCD, NA\_LATITUDE, NA\_LONGITUDE, NA\_FIPSCOUNTY, NA\_FIPS, NA\_CENSUSID

Default: NA\_NONE

Assigns a Name and Address output component to the selected output attribute.

Properties for PIVOT OPERATOR INPUT GROUP ATTRIBUTE:

Name: GROUP\_KEY

Type: BOOLEAN

Valid Values: true, false

Default: false

A boolean value to indicate whether this input attribute is a part of the pivot group key.

---

Properties for PIVOT OPERATOR OUTPUT GROUP ATTRIBUTE:

Name: GROUP\_KEY

Type: BOOLEAN

Valid Values: true, false

Default: false

A boolean value to indicate whether this output attribute is a part of the pivot group key, which obtains its value from its corresponding input attribute.

Name: PIVOT\_EXPRESSION

Type: STRING

Valid Values: N/A

Default: N/A

A comma-separated expression that gives the input attribute to be used for each output row in the pivot group.

Properties for POSTMAPPING\_PROCESS OPERATOR INPUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DEFAULT\_VALUE

Type: STRING

Valid Values: N/A

Default: N/A

The Default Value for the function input parameter

Name: IS\_OPTIONAL

Type: BOOLEAN

Valid Values: true, false

Default: N/A

If true, the input is not required to be connected

Properties for POSTMAPPING\_PROCESS OPERATOR OUTPUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: FUNCTION\_RETURN

Type: BOOLEAN

Valid Values: true, false

Default: N/A

Specifies whether this output is the return value of this function

Properties for PREMAPPING\_PROCESS OPERATOR INPUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DEFAULT\_VALUE

Type: STRING

Valid Values: N/A

Default: N/A

The Default Value for the function input parameter

Name: IS\_OPTIONAL

Type: BOOLEAN

Valid Values: true, false

Default: N/A

If true, the input is not required to be connected

Properties for PREMAPPING\_PROCESS OPERATOR OUTPUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: FUNCTION\_RETURN

Type: BOOLEAN

Valid Values: true, false

Default: N/A

Specifies whether this output is the return value of this function

Properties for TABLE OPERATOR INOUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: LOAD\_COLUMN\_WHEN\_INSERTING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will participate in the insert load operation.

Name: LOAD\_COLUMN\_WHEN\_UPDATING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will participate in the update load operation.

Name: MATCH\_COLUMN\_WHEN\_DELETING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will be used to construct the matching criteria between the incoming data and the existing data on the target during the delete load operation.

Name: MATCH\_COLUMN\_WHEN\_UPDATING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will be used to construct the matching criteria between the incoming data and the existing data on the target during the update load operation.

Name: UPDATE\_OPERATION

Type: STRING

Valid Values: =, +=, -=, =-, =| |, || =

Default: =

The computation to be performed on this attribute between the incoming data and the existing data on the target during the update load operation.

Properties for TABLE\_FUNCTION OPERATOR INPUT GROUP ATTRIBUTE:

Name: PARAMETER\_POSITION

Type: NUMBER

Valid Values: N/A

Default: N/A

The position of the argument in the table function signature corresponding to this parameter

Properties for TABLE\_FUNCTION OPERATOR OUTPUT GROUP ATTRIBUTE:

Name: TYPE\_ATTRIBUTE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name of the field of the PLS Record or attribute of the Object Type or column of the ROWTYPE that corresponds to this attribute. This property is not applicable if the return type is TABLE or SCALAR.

---

Properties for TRANSFORMATION OPERATOR INPUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: DEFAULT\_VALUE

Type: STRING

Valid Values: N/A

Default: N/A

The Default Value for the function input parameter

Name: IS\_OPTIONAL

Type: BOOLEAN

Valid Values: true, false

Default: N/A

If true, the input is not required to be connected

Properties for TRANSFORMATION OPERATOR OUTPUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: FUNCTION\_RETURN

Type: BOOLEAN

Valid Values: true, false

Default: N/A

Specifies whether this output is the return value of this function

Properties for TRANSFORMATION OPERATOR INOUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Properties for UNPIVOT OPERATOR INPUT GROUP ATTRIBUTE:

Name: GROUP\_KEY

Type: BOOLEAN

Valid Values: true, false

Default: false

A boolean value to indicate whether this input attribute is a part of the unpivot group key.

Properties for UNPIVOT OPERATOR OUTPUT GROUP ATTRIBUTE:

Name: GROUP\_KEY

Type: BOOLEAN

Valid Values: true, false

Default: false

A boolean value to indicate whether this output attribute is a part of the unpivot group key, which obtains its value from its corresponding input attribute.

Name: MATCHING\_ROW

Type: NUMBER

Valid Values: 1 - 1000

Default: 1

An positive integer to indicate from which row within the unpivot group this output attribute obtains its data.

Name: UNPIVOT\_EXPRESSION

Type: STRING

Valid Values: N/A

Default: NULL

An expression that gives the input attribute to be used as the output of this attribute.



---

Properties for VIEW OPERATOR INOUT GROUP ATTRIBUTE:

Name: BOUND\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The name to be used by the code generator to identify this item. By default it is the same physical name as the item.

Name: LOAD\_COLUMN\_WHEN\_INSERTING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will participate in the insert load operation.

Name: LOAD\_COLUMN\_WHEN\_UPDATING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will participate in the update load operation.

Name: MATCH\_COLUMN\_WHEN\_DELETING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will be used to construct the matching criteria between the incoming data and the existing data on the target during the delete load operation.

Name: MATCH\_COLUMN\_WHEN\_UPDATING\_ROW

Type: STRING

Valid Values: YES, NO

Default: YES

A boolean value to indicate whether this attribute will be used to construct the matching criteria between the incoming data and the existing data on the target during the update load operation.

Name: UPDATE\_OPERATION

Type: STRING

Valid Values: =, +=, -=, =-, =| |, || =

Default: =

The computation to be performed on this attribute between the incoming data and the existing data on the target during the update load operation.

Note:

1. N/A means any valid character in supported character set.
2. " " represents an empty string

childName

Name of a child that belongs to map, mapping operator, mapping group or mapping attribute.

operatorName

Name of a mapping operator.

groupDirection

Direction of a mapping group.

groupLocator

Location of a mapping group.

getGroupDetailClause

Get the desired detail of a mapping group.

groupBottomUpLocator

Location of a mapping group.

attributeBottomUpLocator

Location of a mapping attribute.

groupName

Name of a mapping group.

attributeLocator

Location of a mapping attribute.

`getAttributeDetailClause`

Get the desired detail of a mapping attribute.

`attributeName`

Name of a mapping attribute.

## Examples

```
OMBRETRIEVE MAPPING 'MAP1' GET OPERATORS
```

```
OMBRETRIEVE MAPPING 'MAP1' OPERATOR 'SRC1' ATTRIBUTE_GROUP  
'INOUTGRP1'
```

```
GET ATTRIBUTE CONNECTED TO OPERATOR 'target1'
```

```
OMBRETRIEVE MAPPING 'MAP1' OPERATOR 'SRC1' GET PROPERTIES  
(BUSINESS_NAME, DESCRIPTION)
```

## See Also

OMBRETRIEVE, OMBCREATE MAPPING, OMBALTER MAPPING, OMBDROP MAPPING

# OMBRETRIEVE MATERIALIZED\_VIEW

## Purpose

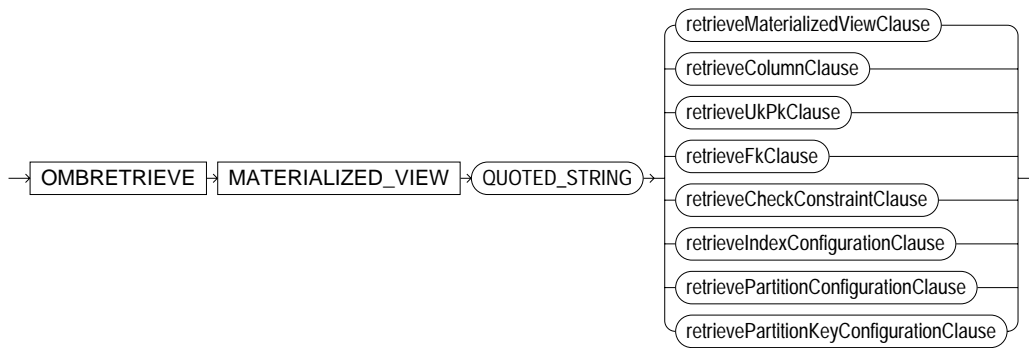
OMBRETRIEVE MATERIALIZED\_VIEW - To retrieve properties of a materialized view.

## Prerequisites

In the context of an Oracle Module

## Syntax Diagrams

retrieveMaterializedViewCommand



retrieveMaterializedViewClause



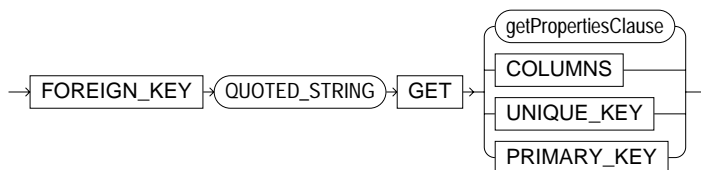
retrieveColumnClause



retrieveUkPkClause



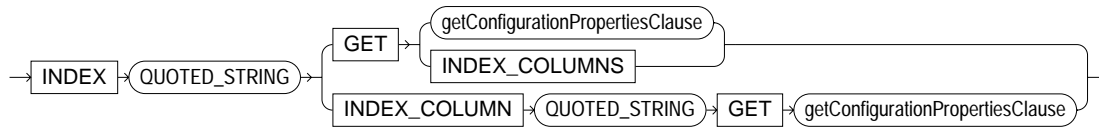
retrieveFkClause



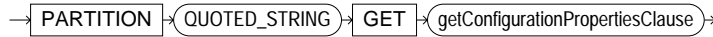
retrieveCheckConstraintClause



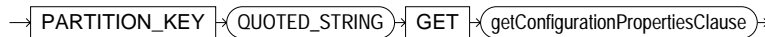
retrieveIndexConfigurationClause



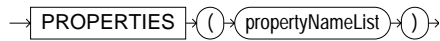
retrievePartitionConfigurationClause



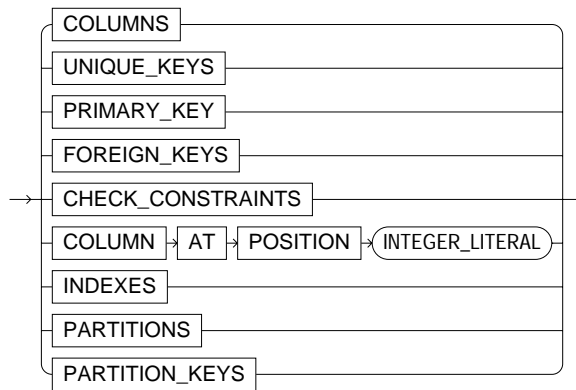
retrievePartitionKeyConfigurationClause



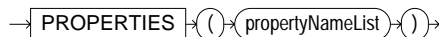
getPropertiesClause



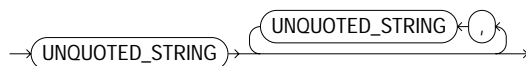
getTableSCOClauses



getConfigurationPropertiesClause



propertyNameList



## Syntax

```

retrieveMaterializedViewCommand = OMBRETRIEVE MATERIALIZED_VIEW
"QUOTED_STRING" ( "retrieveMaterializedViewClause" | "retrieveColumnClause" |
"retrieveUkPkClause" | "retrieveFkClause" | "retrieveCheckConstraintClause" |
"retrieveIndexConfigurationClause" | "retrievePartitionConfigurationClause" |
"retrievePartitionKeyConfigurationClause" );
  
```

```

retrieveMaterializedViewClause = GET ( "getPropertiesClause" |
"getTableSCOClauses" );
  
```

```

retrieveColumnNameClause = COLUMN "QUOTED_STRING" GET "getPropertiesClause";
retrieveUkPkClause = ( UNIQUE_KEY | PRIMARY_KEY ) "QUOTED_STRING" GET
( "getPropertiesClause" | COLUMNS );
retrieveFkClause = FOREIGN_KEY "QUOTED_STRING" GET ( "getPropertiesClause"
| COLUMNS | UNIQUE_KEY | PRIMARY_KEY );
retrieveCheckConstraintClause = CHECK_CONSTRAINT "QUOTED_STRING" GET
"getPropertiesClause";
retrieveIndexConfigurationClause = INDEX "QUOTED_STRING" ( GET (
"getConfigurationPropertiesClause" | INDEX_COLUMNS ) | ( INDEX_COLUMN
"QUOTED_STRING" GET "getConfigurationPropertiesClause" ) );
retrievePartitionConfigurationClause = PARTITION "QUOTED_STRING" GET
"getConfigurationPropertiesClause";
retrievePartitionKeyConfigurationClause = PARTITION_KEY "QUOTED_STRING"
GET "getConfigurationPropertiesClause";
getPropertiesClause = PROPERTIES "(" "propertyNameList" ")";
getTableSCOClauses = COLUMNS | UNIQUE_KEYS | PRIMARY_KEY | FOREIGN_
KEYS | CHECK_CONSTRAINTS | COLUMN AT POSITION "INTEGER_LITERAL" |
INDEXES | PARTITIONS | PARTITION_KEYS;
getConfigurationPropertiesClause = PROPERTIES "(" "propertyNameList" ")";
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };

```

## Keywords and Parameters

**retrieveMaterializedViewCommand**

This clause retrieves a materialized view.

**QUOTED\_STRING**

name of the materialized view.

**retrieveMaterializedViewClause**

This clause retrieves a properties of a materialized view.

**retrieveColumnNameClause**

This clause will retrieve columns

**QUOTED\_STRING**

Name of the column

**retrieveUkPkClause**

This will get the unique key or primary key clause.

**retrieveFkClause**

Name of the foreign key

retrieveCheckConstraintClause

This clause gets the check constraint

QUOTED\_STRING

name of the check constraint.

retrieveIndexConfigurationClause

Gets the index in this clause.

QUOTED\_STRING

Name of the index.

retrievePartitionConfigurationClause

Gets the partition.

QUOTED\_STRING

The partition name.

retrievePartitionKeyConfigurationClause

This clause gets the partition key.

QUOTED\_STRING

The name of the partition key.

getPropertiesClause

This clause retrieves all the properties.

Basic properties for VIEW, MATERIALIZED\_VIEW:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the View, MaterializedView

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the View, MaterializedView

Name: VIEW\_QUERY

Type: STRING(4000)

Valid Values: N/A

Default: "

Sets the query definition in View and MaterializedView.

Name: UOID

Type: STRING

Valid Values: N/A

Default: N/A

Basic properties for INDEX, PARTITION, PARTITION\_KEY, INDEX\_COLUMN:

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the Index, Partition, PartitionKey, IndexColumn in a MaterializedView.

Properties for MATERIALIZED\_VIEW:

Name: BASE\_TABLES

Type: STRING

Valid Values: N/A

Default: "

Comma separated list of base tables.

Name: BUILD

Type: STRING

Valid Values: DEFERRED, IMMEDIATE



Default: IMMEDIATE

Immediate : populates the view when it is created. Deferred : delays population until the next refresh operation.

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: HASH\_PARTITION\_TABLESPACE\_LIST

Type: STRING

Valid Values: N/A

Default: "

A comma separated list of tablespaces to use for [sub]partition storage.

Name: HASH\_SUBPARTITION\_NUMBER

Type: NUMBER

Valid Values: 2 - 63999

Default: 2

Hash SubPartition Number

Name: LOGGING\_MODE

Type: STRING

Valid Values: NOLOGGING, LOGGING

Default: LOGGING

Recovery requirements for a data warehouse : Logging or not logging to Redo Log File.

Name: PARALLEL\_ACCESS\_MODE

Type: STRING

Valid Values: NOPARALLEL, PARALLEL

Default: PARALLEL

Enables or disables parallel processing when the table is created. Also enables or disables parallel processing or access.

Name: QUERY\_REWRITE

Type: STRING

Valid Values: DISABLE, ENABLE

Default: ENABLE

Enable marks the View eligible for query rewrite and disable marks the View ineligible for query rewrite

Name: REFRESH

Type: STRING

Valid Values: COMPLETE, FAST, ON\_COMMIT, ON\_DEMAND, FORCE

Default: COMPLETE

Complete : specifies the complete refresh method implemented by executing the query of the view. Fast : specifies the incremental refresh method which refreshes the view according to changes that have occurred to the master tables. Force : specifies that when a refresh occurs, Oracle performs a fast refresh if possible or a complete refresh otherwise.

Name: TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Use the Tablespace parameter to specify the name of tablespace.

Properties for UNIQUE\_KEY:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: USING\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

Use of Index on Dimension Constraint

Properties for FOREIGN\_KEY:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: USING\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

Use of Index on Dimension Constraint

Properties for CHECK\_CONSTRAINT:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: USING\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

Use of Index on Dimension Constraint

Properties for INDEX:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: INDEX\_TYPE

Type: STRING

Valid Values: BITMAP, UNIQUE, NO\_INDEX

Default: UNIQUE

The types of Indexes created on Dimension are BITMAP, UNIQUE or a non-specific index.

Name: LOCAL\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

A local index is constructed so that it reflects the structure of the underlying table. It is equipartitioned with the underlying table, meaning that it is partitioned on the same columns as the underlying table, creates the same number of partitions or subpartitions, and gives them the same partition bounds as corresponding partitions of the underlying table.

Name: LOGGING\_MODE

Type: STRING

Valid Values: NOLOGGING, LOGGING

Default: LOGGING

Recovery requirements for a data warehouse : Logging or not logging to Redo Log File.

Name: PARALLEL\_ACCESS\_MODE

Type: STRING

Valid Values: NOPARALLEL, PARALLEL

Default: PARALLEL

Enables or disables parallel processing when the table is created. Also enables or disables parallel processing or access.

Name: TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Use the Tablespace parameter to specify the name of tablespace.

Name: USING\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

Use of Index on Dimension Constraint

Properties for PARTITION:

Name: DATE\_LESS\_THAN

Type: STRING

Valid Values: N/A

Default: TO\_DATE('01-JAN-1999', 'DD-MON-YYYY')

Value that represents upper bound of partition stored in warehouse key column for the Days Dimension.

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: EMPTY\_STRING

Use the Tablespace parameter to specify the name of tablespace.

Name: VALUES\_LESS\_THAN

Type: STRING

Valid Values: N/A

Default: TO\_DATE('01-JAN-1999', 'DD-MON-YYYY')

Noninclusive upper bound for the current partition. Type a comma separated list in the same order as the columns in range partitioned key.

Name: VALUE\_LESS\_THAN

Type: STRING

Valid Values: N/A

Default: "

Noninclusive upper bound for the current partition. Type a comma separated list in the same order as the columns in range partitioned key.

Properties for PARTITION\_KEY:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: TYPE

Type: STRING

Valid Values: HASH, RANGE

Default: RANGE

Oracle partitions the storage space and stores rows according to a Hash Algorithm or specified ranges.

Note:

1. N/A means any valid character in supported character set.

2. " represents an empty string

getTableSCOClaude

This clause retrieves components like columns, indexes etc. of a table.

getConfigurationPropertiesClause

This clauses gets the configuration properties of the object.

propertyNameList

The list of properties.

## Examples

```
OMBRETRIEVE MATERIALIZED_VIEW 'NEW_MATERIALIZED_VIEW' GET
PROPERTIES (DESCRIPTION, BUSINESS_NAME) This will retrieve its description
and business name.
```

## See Also

OMBRETRIEVE

## OMBRETRIEVE OBJECT\_TYPE

### Purpose

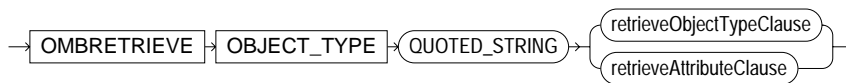
OMBRETRIEVE OBJECT\_TYPE - Retrieve details of the Object Type.

### Prerequisites

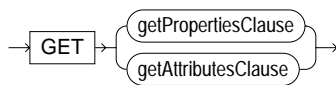
Should be in the context of an Oracle Module.

### Syntax Diagrams

retrieveObjectTypeCommand



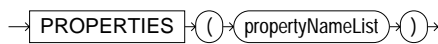
retrieveObjectTypeClause



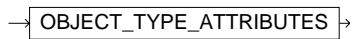
retrieveAttributeClause



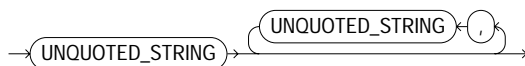
getPropertiesClause



getAttributesClause



propertyNameList



### Syntax

```

retrieveObjectTypeCommand = OMBRETRIEVE OBJECT_TYPE "QUOTED_STRING"
("retrieveObjectTypeClause" | "retrieveAttributeClause" );
retrieveObjectTypeClause = GET ("getPropertiesClause" | "getAttributesClause" );
retrieveAttributeClause = OBJECT_TYPE_ATTRIBUTE "QUOTED_STRING" GET
"getPropertiesClause";
getPropertiesClause = PROPERTIES "(" "propertyNameList" ")";
getAttributesClause = OBJECT_TYPE_ATTRIBUTES;
    
```



---

```
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
```

## Keywords and Parameters

**retrieveObjectTypeCommand**

Retrieves the details of the Object Type with the given name.

**retrieveObjectTypeClause**

Gets the properties or the Attribute names.

**retrieveAttributeClause**

Gets the properties of the Attribute with the given name.

**getPropertiesClause**

This clause retrieves all the properties.

Basic properties for OBJECT\_TYPE:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the Object Type

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the Object Type

Basic properties for OBJECT\_TYPE\_ATTRIBUTE:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the Attribute

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the Attribute

Name: DATATYPE

Type: STRING(20)

Valid Values: NUMBER, VARCHAR2, VARCHAR, DATE, FLOAT

Default: "

Datatype of the Attribute

Note:

1. N/A means any valid character in supported character set.
2. " represents an empty string

getAttributesClause

Gets the names of all the Attributes of the Object Type with the given name.

propertyNameList

The list of properties.

## Examples

```
OMBRETRIEVE OBJECT_TYPE 'SOME_OBJECT_TYPE' OBJECT_TYPE_ATTRIBUTE  
'ATTR1' GET PROPERTIES (DATATYPE) This will retrieve the Object Type "SOME_  
OBJECT_TYPE"'s attribute "ATTR1"'s datatype.
```

## See Also

OMBRETRIEVE, OMBALTER OBJECT\_TYPE, OMBCREATE OBJECT\_TYPE,  
OMBDROP OBJECT\_TYPE

# OMBRETRIEVE ORACLE\_MODULE

## Purpose

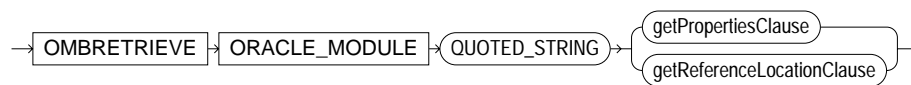
OMBRETRIEVE ORACLE\_MODULE - Retrieve details of the Oracle module.

## Prerequisites

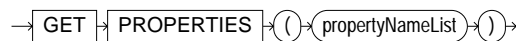
Should be in the context of project.

## Syntax Diagrams

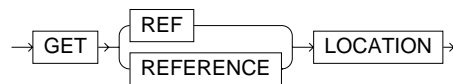
retrieveOracleModuleCommand



getPropertiesClause



getReferenceLocationClause



propertyNameList



## Syntax

```
retrieveOracleModuleCommand = OMBRETRIEVE ORACLE_MODULE "QUOTED_
STRING" ("getPropertiesClause" | "getReferenceLocationClause");
```

```
getPropertiesClause = GET PROPERTIES "(" "propertyNameList" ")";
```

```
getReferenceLocationClause = GET ( REF | REFERENCE ) LOCATION;
```

```
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
```

## Keywords and Parameters

retrieveOracleModuleCommand

This command retrieves the details of an Oracle Module

QUOTED\_STRING

Name of the existing Oracle module or path to the Oracle module.

**getPropertiesClause**

Retrieve a set of properties that is associated with an Oracle Module.

**Properties for ORACLE\_MODULE:**

**Name:** ABAP\_DIRECTORY

**Type:** STRING

**Valid Values:** N/A

**Default:** abap\

Location where ABAP scripts are stored

**Name:** ABAP\_EXTENSION

**Type:** STRING

**Valid Values:** N/A

**Default:** .abap

File name extension for ABAP scripts

**Name:** ABAP\_RUN\_PARAMETER\_FILE

**Type:** STRING

**Valid Values:** N/A

**Default:** \_run.ini

Run Parameter File Suffix for the parameter script in a ABAP job.

**Name:** ABAP\_SPOOL\_DIRECTORY

**Type:** STRING

**Valid Values:** N/A

**Default:** abap\log\

Location where ABAP scripts are buffered during script generation processing.

**Name:** APPLICATION\_SHORT\_NAME

**Type:** STRING

**Valid Values:** N/A

**Default:** WB

Application Short Name

**Name:** ARCHIVE\_DIRECTORY

Type: STRING  
Valid Values: N/A  
Default: archive\  
Archive Directory

Name: CONNECT\_STRING

Type: STRING  
Valid Values: N/A  
Default: "

A Net\*8 style connection string to the remote database. Alternatively, you can specify machine, port, service name of the remote database.

Name: DDL\_DIRECTORY

Type: STRING  
Valid Values: N/A  
Default: ddl\  
Location where scripts for database objects for the target schema are stored.

Name: DDL\_EXTENSION

Type: STRING  
Valid Values: N/A  
Default: .ddl  
File name extension for DDL scripts.

Name: DDL\_SPOOL\_DIRECTORY

Type: STRING  
Valid Values: N/A  
Default: ddl\log\  
Location where DDL scripts are buffered during script generation processing.

Name: DEFAULT\_INDEX\_TABLESPACE

Type: STRING(30)  
Valid Values: N/A  
Default: "  
Default name of tablespace to install indexes into.

Name: DEFAULT\_OBJECT\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Default name of tablespace to install objects into.

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: END\_OF\_LINE

Type: STRING

Valid Values: N/A

Default: \r\n

End of Line

Name: INPUT\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: input\

Input Directory

Name: INVALID\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: invalid\

Directory for SQL\*Loader errors and rejected records

Name: LIB\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: lib\

LIB Directory

Name: LIB\_EXTENSION

Type: STRING  
Valid Values: N/A  
Default: .lib  
LIB Extension

Name: LIB\_SPOOL\_DIRECTORY  
Type: STRING  
Valid Values: N/A  
Default: lib\log\  
LIB Spool Directory

Name: LOADER\_DIRECTORY  
Type: STRING  
Valid Values: N/A  
Default: ctl\  
Location where control files are stored.

Name: LOADER\_EXTENSION  
Type: STRING  
Valid Values: N/A  
Default: .ctl  
Suffix for the loader scripts

Name: LOADER\_RUN\_PARAMETER\_FILE  
Type: STRING  
Valid Values: N/A  
Default: \_run.ini  
Suffix for the parameter initialization file.

Name: LOG\_DIRECTORY  
Type: STRING  
Valid Values: N/A  
Default: log\  
Log Directory for the SQL\*Loader

Name: MAIN\_APPLICATION\_SHORT\_NAME  
Type: STRING

Valid Values: N/A

Default: ora

Main Application Short Name

Name: PLSQL\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: pls\

Location where PL/SQL scripts are stored.

Name: PLSQL\_EXTENSION

Type: STRING

Valid Values: N/A

Default: .pls

File name extension for PL/SQL scripts.

Name: PLSQL\_GENERATION\_MODE

Type: STRING

Valid Values: Oracle9i, Oracle8i

Default: Oracle9i

Generation mode controls validation and generation for version specific features.

Name: PLSQL\_RUN\_PARAMETER\_FILE

Type: STRING

Valid Values: N/A

Default: \_run.ini

Suffix for the parameter script in a PL/SQL job.

Name: PLSQL\_SPOOL\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: pls\log\

Location where PL/SQL scripts are buffered during script generation processing.

Name: PORT

Type: STRING

Valid Values: N/A



Default: "

The port number on the machine where the database listens to.

Name: RECEIVE\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: receive\

Receive Directory

Name: REMOTE\_HOST\_NAME

Type: STRING

Valid Values: N/A

Default: "

The machine where the remote database resides on.

Name: SCHEMA\_OWNER

Type: STRING

Valid Values: N/A

Default: OWB

Schema Owner

Name: SERVICE\_NAME

Type: STRING

Valid Values: N/A

Default: "

The service name (global DB name) of the database instance on the remote machine.

Name: SORT\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: sort\

Sort Directory

Name: TCL\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: tcl\

Location for TCL scripts that are generated after registration with Oracle Enterprise Manager

Name: TOP\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: ..\..\codegen\

Top Directory where generated code will get stored

Name: WORK\_DIRECTORY

Type: STRING

Valid Values: N/A

Default: work\

Work Directory

Note:

1. N/A means any valid character in supported character set.
2. " " represents an empty string

getReferenceLocationClause

Retrieve the location that is set to the Oracle module.

propertyNameList

Comma separated list of property names. Property names are unquoted.

## Examples

```
OMBRETRIEVE ORACLE_MODULE 'src_module' GET PROPERTIES  
(DESCRIPTION, UOID, BUSINESS_NAME) This will retrieve the Oracle module "src_  
module"'s description, uoid, and business name.
```

## See Also

OMBRETRIEVE

## OMBRETRIEVE PACKAGE

### Purpose

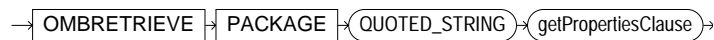
OMBRETRIEVE PACKAGE - Retrieve details of the Package.

### Prerequisites

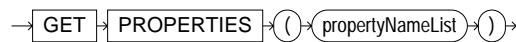
Should be in the context of a Oracle Module or Transformation Module.

### Syntax Diagrams

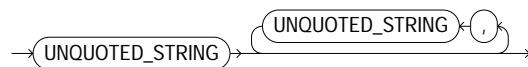
retrievePackageCommand



getPropertiesClause



propertyNameList



### Syntax

```

retrievePackageCommand = OMBRETRIEVE PACKAGE "QUOTED_STRING"
"getPropertiesClause";
getPropertiesClause = GET PROPERTIES "(" "propertyNameList" ")";
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
  
```

### Keywords and Parameters

retrievePackageCommand

This command retrieves the details of a Package

QUOTED\_STRING

Name of the existing Package or path to the Package.

getPropertiesClause

Used to get properties (core, user-defined) for packages. Valid properties are shown below:

Name: BUSINESS\_NAME  
Type: STRING(200)  
Valid Values: N/A  
Default: "  
Business name of the package

Name: DESCRIPTION  
Type: STRING(4000)  
Valid Values: N/A  
Default: "  
Description of the package

Name: PACKAGE\_SPEC  
Type: STRING  
Valid Values: N/A  
Default: "  
Retrieves the Package Spec of Imported Package

Name: PACKAGE\_BODY  
Type: STRING  
Valid Values: N/A  
Default: "  
Retrieves the Package Body of Imported Package

Name: IS\_IMPORTED  
Type: BOOLEAN  
Valid Values: N/A  
Default: "  
Retrieves 'true' if the Package is Imported otherwise 'false'

Properties for PACKAGE:

Name: AUTHID  
Type: STRING  
Valid Values: None, Current\_User, Definer  
Default: None

Generate the package with selected AUTHID option. Function will be executed with the permissions defined by the AUTHID clause rather than the function owner's permissions.

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Note:

1. N/A means any valid character in supported character set.

2. " represents an empty string

propertyNameList

Comma separated list of property names. Property names are unquoted.

## Examples

OMBRETRIEVE PACKAGE 'package\_1' GET PROPERTIES (DESCRIPTION, UOID, BUSINESS\_NAME,) This will retrieve the Package "package\_1's description, uoid, and business name.

## See Also

OMBRETRIEVE

# OMBRETRIEVE PROCEDURE

## Purpose

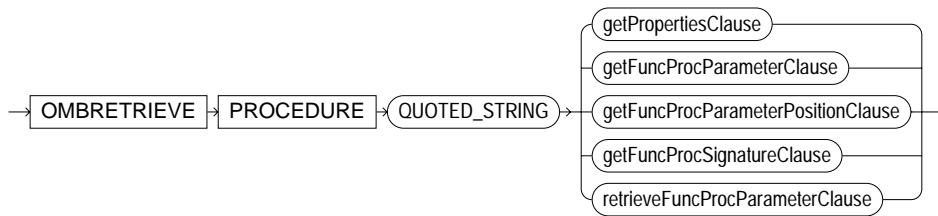
OMBRETRIEVE PROCEDURE - Retrieve details of the Procedure.

## Prerequisites

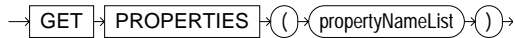
Should be in the context of a Oracle Module or Package or Transformation Module.

## Syntax Diagrams

retrieveProcedureCommand



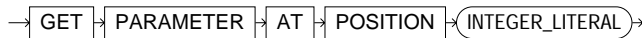
getPropertiesClause



getFuncProcParameterClause



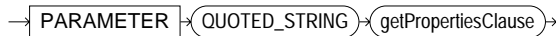
getFuncProcParameterPositionClause



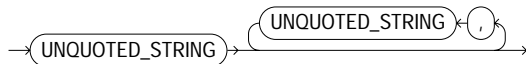
getFuncProcSignatureClause



retrieveFuncProcParameterClause



propertyNameList



## Syntax

```

retrieveProcedureCommand = OMBRETRIEVE PROCEDURE "QUOTED_STRING" (
  "getPropertiesClause" | "getFuncProcParameterClause" |

```

```

"getFuncProcParameterPositionClause" | "getFuncProcSignatureClause" |
"retrieveFuncProcParameterClause" );
getPropertiesClause = GET PROPERTIES "(" "propertyNameList" ")";
getFuncProcParameterClause = GET PARAMETERS;
getFuncProcParameterPositionClause = GET PARAMETER AT POSITION
"INTEGER_LITERAL";
getFuncProcSignatureClause = GET SIGNATURE;
retrieveFuncProcParameterClause = PARAMETER "QUOTED_STRING"
"getPropertiesClause";
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };

```

## Keywords and Parameters

retrieveProcedureCommand

This command retrieves the details of a Procedure

QUOTED\_STRING

Name of the existing Procedure or path to the Procedure.

getPropertiesClause

Used to get properties (core, user-defined) for procedure. Valid properties are shown below:

Basic properties for PROCEDURE:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the Procedure

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the Procedure

Name: IMPLEMENTATION

Type: STRING

Valid Values: N/A

Default: "

Set the code for Procedure which is included global variable declaration and code between BEGIN and END.

Basic properties for PARAMETER:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the Parameter

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the Parameter

Name: DATATYPE

Type: STRING

Valid Values: PLS\_INTEGER, BINARY\_INTEGER, BOOLEAN, NUMBER, FLOAT, CHAR, VARCHAR, VARCHAR2, DATE

Default: NUMBER

Set the data type for Parameter

Name: IN\_OUT

Type: STRING

Valid Values: IN, OUT, INOUT

Default: 'IN'

Set the parameter mode for Parameter

Name: DEFAULT\_VALUE

Type: STRING

Valid Values: N/A

Default: "

Set the default value for Parameter

Properties for PROCEDURE:



Name: AUTHID

Type: STRING

Valid Values: None, Current\_User, Definer

Default: None

Generate the transformation with selected AUTHID option. Function will be executed with the permissions defined by the AUTHID clause rather than the function owner's permissions.

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Note:

1. N/A means any valid character in supported character set.
2. " " represents an empty string

getFuncProcParameterClause

Get all the parameter names of the Procedure

getFuncProcParameterPositionClause

Get the parameter position of Function

getFuncProcSignatureClause

Get the complete signature of the Function which includes parameter names, datatype, in/out type and default values

retrieveFuncProcParameterClause

Get the parameter information such as datatype, default value, in/out type and position

QUOTED\_STRING

Name of the existing Parameter

propertyNameList

Comma separated list of property names. Property names are unquoted.

## Examples

```
OMBRETRIEVE PROCEDURE 'proc' GET PROPERTIES (DESCRIPTION, UOID,  
BUSINESS_NAME, IMPLEMENTATION, IS_IMPORTED)
```

## See Also

OMBRETRIEVE

# OMBRETRIEVE PROCESS\_FLOW

## Purpose

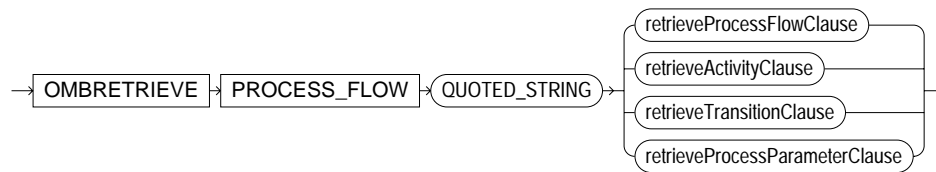
OMBRETRIEVE PROCESS\_FLOW - Retrieves the details of the Process Flow.

## Prerequisites

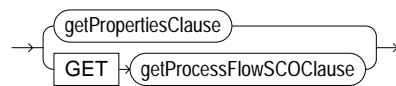
Should be in the context of a Process Flow Package.

## Syntax Diagrams

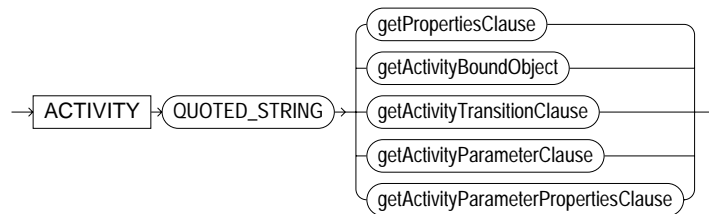
retrieveProcessFlowCommand



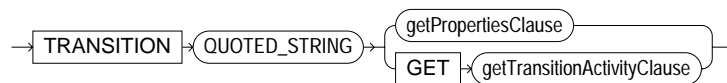
retrieveProcessFlowClause



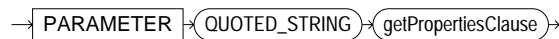
retrieveActivityClause



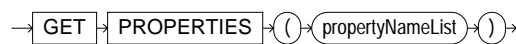
retrieveTransitionClause



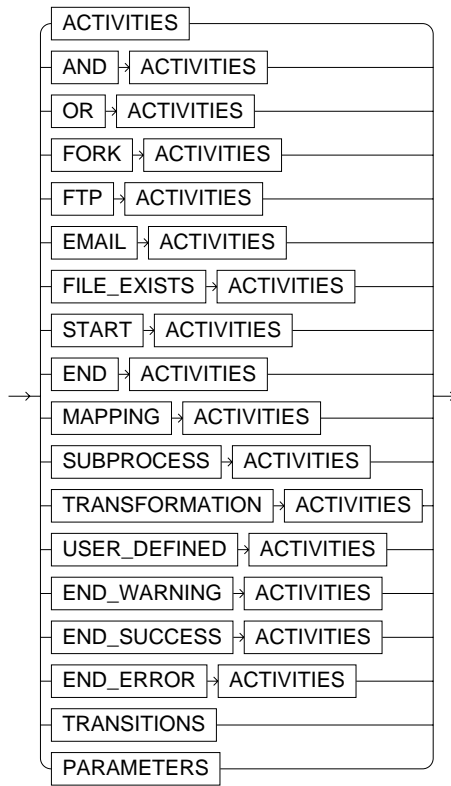
retrieveProcessParameterClause



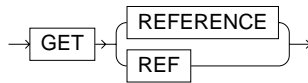
getPropertiesClause



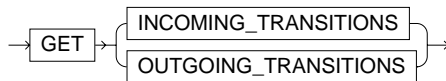
getProcessFlowSCOClause



getActivityBoundObject



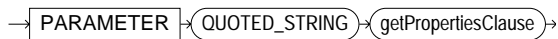
getActivityTransitionClause



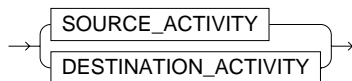
getActivityParameterClause



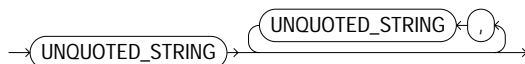
getActivityParameterPropertiesClause



getTransitionActivityClause



propertyNameList



## Syntax

```

retrieveProcessFlowCommand = OMBRETRIEVE PROCESS_FLOW "QUOTED_
STRING" ("retrieveProcessFlowClause" | "retrieveActivityClause" |
"retrieveTransitionClause" | "retrieveProcessParameterClause");

retrieveProcessFlowClause = "getPropertiesClause" | GET
"getProcessFlowSCOClause";

retrieveActivityClause = ACTIVITY "QUOTED_STRING" ("getPropertiesClause" |
"getActivityBoundObject" | "getActivityTransitionClause" |
"getActivityParameterClause" | "getActivityParameterPropertiesClause");

retrieveTransitionClause = TRANSITION "QUOTED_STRING" (
"getPropertiesClause" | GET "getTransitionActivityClause");

retrieveProcessParameterClause = PARAMETER "QUOTED_STRING"
"getPropertiesClause";

getPropertiesClause = GET PROPERTIES "(" "propertyNameList" ")";

getProcessFlowSCOClause = ACTIVITIES | AND ACTIVITIES | OR ACTIVITIES |
FORK ACTIVITIES | FTP ACTIVITIES | EMAIL ACTIVITIES | FILE_EXISTS
ACTIVITIES | START ACTIVITIES | END ACTIVITIES | MAPPING ACTIVITIES |
SUBPROCESS ACTIVITIES | TRANSFORMATION ACTIVITIES | USER_DEFINED
ACTIVITIES | END_WARNING ACTIVITIES | END_SUCCESS ACTIVITIES | END_
ERROR ACTIVITIES | TRANSITIONS | PARAMETERS;

getActivityBoundObject = GET (REFERENCE | REF);

getActivityTransitionClause = GET (INCOMING_TRANSITIONS | OUTGOING_
TRANSITIONS);

getActivityParameterClause = (GET PARAMETERS);

getActivityParameterPropertiesClause = PARAMETER "QUOTED_STRING"
"getPropertiesClause";

getTransitionActivityClause = (SOURCE_ACTIVITY | DESTINATION_ACTIVITY);

propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };

```

## Keywords and Parameters

**retrieveProcessFlowCommand**

Retrieve the details of an existing process flow.

**retrieveProcessFlowClause**

This clause retrieve the Process Flow.

**retrieveActivityClause**

This clause retrieves the Activity of a Process Flow.

**retrieveTransitionClause**

This clause retrieves the Transition of a Process Flow.

**retrieveProcessParameterClause**

This clause retrieves the Parameter of a Process Flow.

**getPropertiesClause**

Used to get properties (core, user-defined) for process flow. Valid properties are shown below:

Base properties for Process Flow, Activity, Transition and Parameter:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: NAME

Business name of a Process Flow

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of a Process Flow

Name: UOID

Type: STRING(40)

Valid Values: N/A

Default: N/A

UOID of a Process Flow

Basic properties for Transition:

Name: TRANSITION\_CONDITION

Type: STRING

Valid Values: ", SUCCESS, ERROR, WARNING

Default: " i.e. Unconditional

Gets the Transition Condition of a Transition Basic properties for Activity Parameter :

Name: DATATYPE

Type: STRING

Valid Values: INTEGER, FLOAT, DATE, STRING, BOOLEAN

Default: STRING

Gets the datatype of a Activity Parameter

Name: DIRECTION

Type: STRING

Valid Values: IN

Default: IN

Gets the direction of a Activity Parameter

Name: VALUE

Type: STRING

Valid Values: Examples '123', '123.456', 'Jan-08-2003', 'I am String', 'true'

Default: "

Gets the value of a Activity Parameter

Name: BINDING

Type: STRING

Valid Values: Examples 'PROCESS\_PARAM1', 'PROCESS\_PARAM2'

Default: "

Get the name of the process parameter that this parameter is bound to, empty if not bound.

getProcessFlowSCOClause

For the current process flow, retrieve all activities or only activities of a specific type.

getActivityBoundObject

Retrieve the name of the object that the current activity refers to, pertinent to only MAPPING, SUBPROCESS and TRANSFORMATION activities.

getActivityTransitionClause

This clause retrieves all Activities of a Process Flow.

getActivityParameterClause

Retrieve the activity parameters for the current activity.

`getActivityParameterPropertiesClause`

Retrieve the details of an activity parameter.

`getTransitionActivityClause`

This clause retrieves all Transitions of a Process Flow.

`propertyNameList`

A comma delimited set of property names to set.

## Examples

`OMBRETRIEVE PROCESS_FLOW 'process_flow' GET PROPERTIES (DESCRIPTION, UOID, BUSINESS_NAME) GET ACTIVITIES` This will retrieve the Process Flow "process\_flow"s description, uoid, and business name, and gets all activities of the process flow.

## See Also

`OMBRETRIEVE`



## OMBRETRIEVE PROCESS\_FLOW\_MODULE

### Purpose

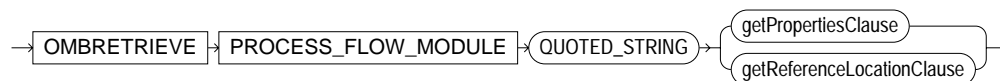
OMBRETRIEVE PROCESS\_FLOW\_MODULE - Retrieve details of the Process Flow Module.

### Prerequisites

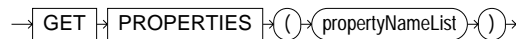
Should be in the context of a project.

### Syntax Diagrams

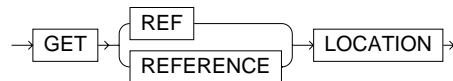
retrieveProcessFlowModuleCommand



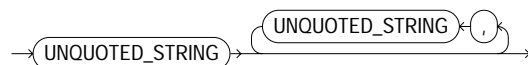
getPropertiesClause



getReferenceLocationClause



propertyNameList



### Syntax

```
retrieveProcessFlowModuleCommand = OMBRETRIEVE PROCESS_FLOW_
MODULE "QUOTED_STRING" ( "getPropertiesClause" |
"getReferenceLocationClause" );
```

```
getPropertiesClause = GET PROPERTIES "(" "propertyNameList" ")";
```

```
getReferenceLocationClause = GET ( REF | REFERENCE ) LOCATION;
```

```
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
```

### Keywords and Parameters

retrieveProcessFlowModuleCommand

Retrieve the details of an existing process flow module.

**getPropertiesClause**

Used to get properties (core, user-defined) for process flow module.

Base properties for PROCESS\_FLOW\_MODULE:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: NAME

Business name of a Process Flow Module

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of a Process Flow Module

Name: UOID

Type: STRING(40)

Valid Values: N/A

Default: N/A

UOID of a Process Flow Module

**getReferenceLocationClause**

Retrieve the name of the Workflow engine location referenced by this process flow module.

**propertyNameList**

Comma separated list of property names. Property names are not in quotation marks.

**Examples**

OMBRETRIEVE PROCESS\_FLOW\_MODULE 'process\_module' GET PROPERTIES (DESCRIPTION, UOID, BUSINESS\_NAME) This will retrieve the Process Flow Module "process\_module"'s description, uoid, and business name.

**See Also**

OMBRETRIEVE

## OMBRETRIEVE PROCESS\_FLOW\_PACKAGE

### Purpose

OMBRETRIEVE PROCESS\_FLOW\_PACKAGE - Retrieve details of the Process Flow Package.

### Prerequisites

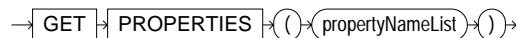
Should be in the context of a Process Flow Module.

### Syntax Diagrams

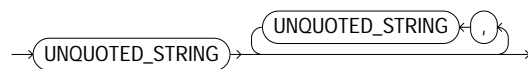
retrieveProcessFlowPackageCommand



getPropertiesClause



propertyNameList



### Syntax

```
retrieveProcessFlowPackageCommand = OMBRETRIEVE PROCESS_FLOW_
PACKAGE "QUOTED_STRING" "getPropertiesClause";
```

```
getPropertiesClause = GET PROPERTIES "(" "propertyNameList" ")";
```

```
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
```

### Keywords and Parameters

retrieveProcessFlowPackageCommand

Retrieve the details of an existing process flow package.

getPropertiesClause

Used to get properties (core, user-defined) for process flow packages. Valid properties are shown below:

Basic properties for PROCESS\_FLOW\_PACKAGE:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: NAME

Business name of a Process Flow Package

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of a Process Flow Package

Name: UOID

Type: STRING(40)

Valid Values: N/A

Default: N/A

UOID of a Process Flow Package

propertyNameList

A comma delimited set of property names to set.

## Examples

OMBRETRIEVE PROCESS\_FLOW\_PACKAGE 'process\_package' GET PROPERTIES (DESCRIPTION, UOID, BUSINESS\_NAME) This will retrieve the Process Flow Package "process\_package"'s description, uoid, and business name.

## See Also

OMBRETRIEVE

# OMBRETRIEVE PROJECT

## Purpose

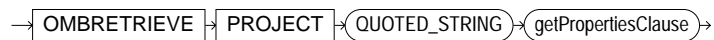
OMBRETRIEVE PROJECT - Retrieve details of the project.

## Prerequisites

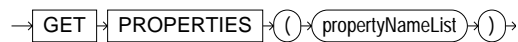
Should be in the top level context.

## Syntax Diagrams

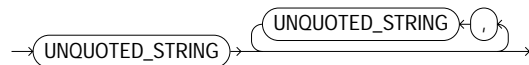
retrieveProjectCommand



getPropertiesClause



propertyNameList



## Syntax

```

retrieveProjectCommand = OMBRETRIEVE PROJECT "QUOTED_STRING"
"getPropertiesClause";
getPropertiesClause = GET PROPERTIES "(" "propertyNameList" ")";
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
  
```

## Keywords and Parameters

retrieveProjectCommand

Retrieve the details of a Project

QUOTED\_STRING

Name of the existing project or path to the project.

getPropertiesClause

Retrieve a set of properties that is associated with a Project.

propertyNameList

Comma separated list of property names. Property names are unquoted.

### Examples

OMBRETRIEVE PROJECT 'New Project' GET PROPERTIES (DESCRIPTION, UID,  
BUSINESS\_NAME) This will retrieve the project "New Project"'s description, uid,  
and business name.

### See Also

OMBRETRIEVE

## OMBRETRIEVE\_RUNTIME\_REPOSITORY\_CONNECTION

### Purpose

OMBRETRIEVE\_RUNTIME\_REPOSITORY\_CONNECTION - Retrieve details from a runtime repository connection.

### Prerequisites

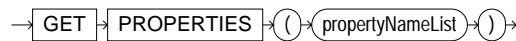
Should be in the context of a project.

### Syntax Diagrams

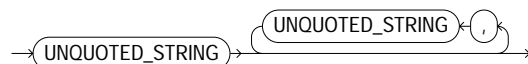
retrieveRuntimeRepositoryCommand



getPropertiesClause



propertyNameList



### Syntax

```
retrieveRuntimeRepositoryCommand = OMBRETRIEVE_RUNTIME_REPOSITORY_
CONNECTION "QUOTED_STRING" "getPropertiesClause";
getPropertiesClause = GET PROPERTIES "(" "propertyNameList" ")";
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
```

### Keywords and Parameters

`retrieveRuntimeRepositoryCommand`

Retrieve details from a runtime repository connection.

`getPropertiesClause`

Get the specified properties of the runtime repository connection.

`propertyNameList`

The names of the properties whose values you want to retrieve.

Properties for RUNTIME\_REPOSITORY\_CONNECTION:

Name: HOST

Type: STRING

Valid Values: N/A

Default: N/A

The host machine the runtime repository is installed on.

Name: PORT

Type: NUMBER

Valid Values: 0 - 65535

Default: N/A

The port number of the database in which the runtime repository is installed.

Name: SERVICE\_NAME

Type: STRING

Valid Values: N/A

Default: N/A

The service name of the database in which the runtime repository is installed.

Name: CONNECT\_AS\_USER

Type: STRING

Valid Values: N/A

Default: N/A

The name of the database user you wish to connect to the runtime repository as.

Name: RUNTIME\_REPOSITORY\_OWNER

Type: STRING

Valid Values: N/A

Default: N/A

The name of the schema in which the runtime repository is installed.

All of the preceding properties are mandatory for OMBCREATE RUNTIME\_REPOSITORY\_CONNECTION.

Basic properties for RUNTIME\_REPOSITORY\_CONNECTION:



Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the runtime repository connection.

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the runtime repository connection.

Note:

1. N/A means any valid character in supported character set.
2. " represents an empty string

## Examples

```
OMBRETRIEVE RUNTIME_REPOSITORY_CONNECTION 'MY_CONNECTION'  
GET PROPERTIES (DESCRIPTION, UOID, BUSINESS_NAME) This will retrieve the  
runtime repository connection "MY_CONNECTION"'s description, uoid, and business  
name.
```

## See Also

OMBRETRIEVE

# OMBRETRIVE SAP\_MODULE

## Purpose

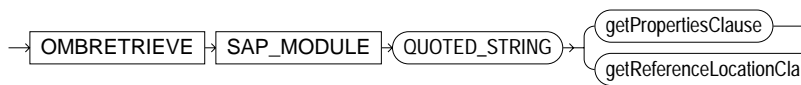
OMBRETRIVE SAP\_MODULE - Retrieve details of the SAP module.

## Prerequisites

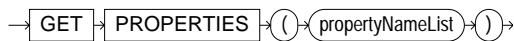
You must open a project to retrieve a SAP module.

## Syntax Diagrams

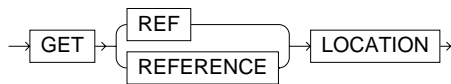
retrieveSAPModuleCommand



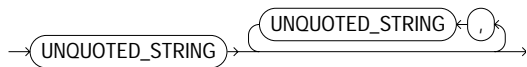
getPropertiesClause



getReferenceLocationClause



propertyNameList



## Syntax

```
retrieveSAPModuleCommand = OMBRETRIVE SAP_MODULE "QUOTED_
STRING" ("getPropertiesClause" | "getReferenceLocationClause");
```

```
getPropertiesClause = GET PROPERTIES "(" "propertyNameList" ")";
```

```
getReferenceLocationClause = GET ( REF | REFERENCE ) LOCATION;
```

```
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
```

## Keywords and Parameters

retrieveSAPModuleCommand

Retrieve the details of an SAP Module

getPropertiesClause

Retrieve a set of properties that is associated with an SAP Module.

Base properties for SAP\_MODULE:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: NAME

Business name of a SAP Module

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of an SAP Module

Name: UOID

Type: STRING(40)

Valid Values: N/A

Default: N/A

UOID of an SAP Module

getReferenceLocationClause

Retrieve the name of the SAP location referenced by this SAP module.

propertyNameList

Comma separated list of property names. Property names are unquoted.

## Examples

OMBRETRIEVE SAP\_MODULE 'src\_module' GET PROPERTIES (DESCRIPTION, UOID, BUSINESS\_NAME) This will retrieve the SAP module "src\_module"'s description, uoid, and business name.

## See Also

OMBRETRIEVE

## OMBRETRIEVE SEQUENCE

### Purpose

OMBRETRIEVE SEQUENCE - To retrieve properties of a sequence.

### Prerequisites

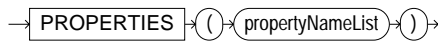
In the context of an Oracle Module

### Syntax Diagrams

retrieveSequenceCommand



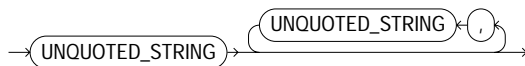
getPropertiesClause



retrieveColumnClause



propertyNameList



### Syntax

```

retrieveSequenceCommand = OMBRETRIEVE SEQUENCE "QUOTED_STRING" (
    GET "getPropertiesClause" | "retrieveColumnClause" );
getPropertiesClause = PROPERTIES "(" "propertyNameList" ");
retrieveColumnClause = COLUMN "QUOTED_STRING" GET "getPropertiesClause";
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
    
```

### Keywords and Parameters

retrieveSequenceCommand

This clause retrieves properties of a sequence.

getPropertiesClause

This clause retrieves all the properties.

## Basic properties for SEQUENCE:

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the sequence.

Name: CURRVAL

Type: NUMBER

Valid Values: N/A

Default: 1

current increment value.

Name: NEXTVAL

Type: NUMBER

Valid Values: N/A

Default: 1

next increment value. next increment value.

## Properties for SEQUENCE:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INCREMENT\_BY

Type: NUMBER

Valid Values: -2147483648 - 2147483647

Default: 1

Sequence Incremented By

Name: START\_WITH

Type: NUMBER

Valid Values: -2147483648 - 2147483647

Default: 1

Sequence Starts With

Note:

1. N/A means any valid character in supported character set.
2. " " represents an empty string

retrieveColumnClause

This clause will retrieve columns

QUOTED\_STRING

Name of the column

propertyNameList

The list of properties.

## Examples

OMBRETRIEVE SEQUENCE 'NEW\_SEQUENCE' GET PROPERTIES (DESCRIPTION)  
This will retrieve its description.

## See Also

OMBRETRIEVE

## OMBRETRIEVE SNAPSHOT

### Purpose

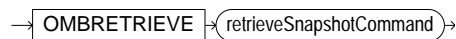
OMBRETRIEVE SNAPSHOT - Since the snapshot may contain many components, this command lets the user view all the contents in a snapshot.

### Prerequisites

Snapshot contents can be retrieved from any context.

### Syntax Diagrams

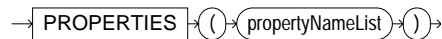
parseRetrieveCommand



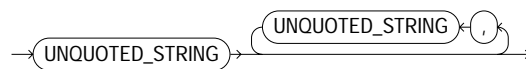
retrieveSnapshotCommand



getPropertiesClause



propertyNameList



### Syntax

```

parseRetrieveCommand = OMBRETRIEVE "retrieveSnapshotCommand";
retrieveSnapshotCommand = ( SNAPSHOT "QUOTED_STRING" [ GET
"getPropertiesClause" ] );
getPropertiesClause = PROPERTIES "(" "propertyNameList" ")";
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
  
```

### Keywords and Parameters

parseRetrieveCommand

Root production of OMBRETRIEVE SNAPSHOT.

retrieveSnapshotCommand

To view contents of snapshot.

**QUOTED\_STRING**

Name of snapshot whose contents are to be retrieved.

**getPropertiesClause**

Gets the property of snapshot which are DESCRIPTION, TYPE.

Basic properties for SNAPSHOT:

**Name:** TYPE

**Type:** STRING(200)

**Valid Values:** FULL,SIGNATURE

**Default:** FULL

This is the type of snapshot

**Name:** DESCRIPTION

**Type:** STRING(4000)

**Valid Values:** N/A

**Default:** "

Description of the snapshot

**PROPERTIES**

Valid set of properties are DESCRIPTION and TYPE.

**propertyNameList**

Property names for SNAPSHOT that can be retrieved.

**Examples**

```
OMBRETRIEVE SNAPSHOT 'S1'
```

This command gets all the contents of snapshot.

```
OMBRETRIEVE SNAPSHOT 'S1' GET  
PROPERTIES(DESCRIPTION,TIMESTAMP,TYPE)
```

This command gets the properties DESCRIPTION,TIMESTAMP, and TYPE of snapshot S1.

**See Also**

OMBCREATE SNAPSHOT, OMBALTER SNAPSHOT, OMBDROP SNAPSHOT,  
OMBRESTORE SNAPSHOT, OMBCOMPARE SNAPSHOT, OMBLIST SNAPSHOT



# OMBRETRIEVE TABLE

## Purpose

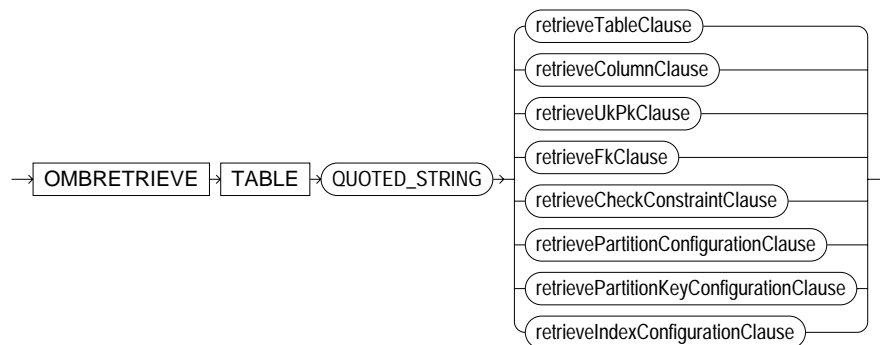
OMBRETRIEVE TABLE - To retrieve properties of a table.

## Prerequisites

In the context of an Oracle Module

## Syntax Diagrams

retrieveTableCommand



retrieveTableClause



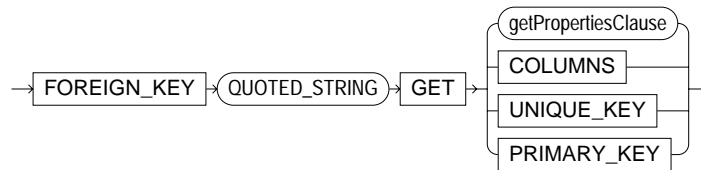
retrieveColumnClause



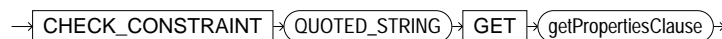
retrieveUkPkClause



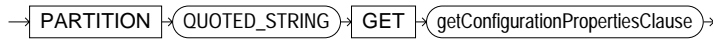
retrieveFkClause



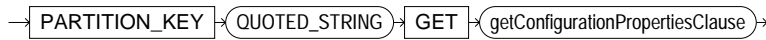
retrieveCheckConstraintClause



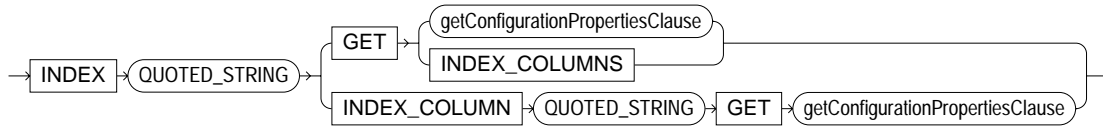
retrievePartitionConfigurationClause



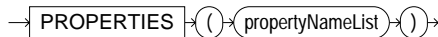
retrievePartitionKeyConfigurationClause



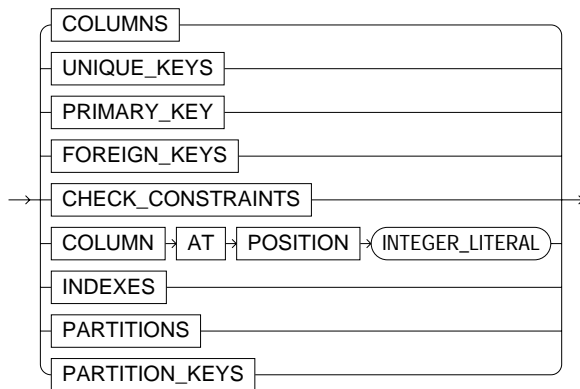
retrieveIndexConfigurationClause



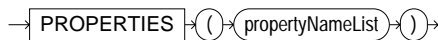
getPropertiesClause



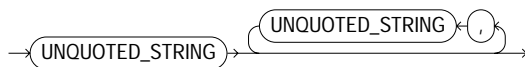
getTableSCOClause



getConfigurationPropertiesClause



propertyNameList



## Syntax

```

retrieveTableCommand = OMBRETRIEVE TABLE "QUOTED_STRING" (
  "retrieveTableClause" | "retrieveColumnClause" | "retrieveUkPkClause" |
  "retrieveFkClause" | "retrieveCheckConstraintClause" |
  "retrievePartitionConfigurationClause" | "retrievePartitionKeyConfigurationClause" |
  "retrieveIndexConfigurationClause");
  
```

```

retrieveTableClause = GET ("getPropertiesClause" | "getTableSCOClause");
  
```

```

retrieveColumnClause = COLUMN "QUOTED_STRING" GET "getPropertiesClause";
  
```

```

retrieveUkPkClause = ( UNIQUE_KEY | PRIMARY_KEY ) "QUOTED_STRING" GET
  ("getPropertiesClause" | COLUMNS);
  
```

```

retrieveFkClause = FOREIGN_KEY "QUOTED_STRING" GET ( "getPropertiesClause"
| COLUMNS | UNIQUE_KEY | PRIMARY_KEY );
retrieveCheckConstraintClause = CHECK_CONSTRAINT "QUOTED_STRING" GET
"getPropertiesClause";
retrievePartitionConfigurationClause = PARTITION "QUOTED_STRING" GET
"getConfigurationPropertiesClause";
retrievePartitionKeyConfigurationClause = PARTITION_KEY "QUOTED_STRING"
GET "getConfigurationPropertiesClause";
retrieveIndexConfigurationClause = INDEX "QUOTED_STRING" ( GET (
"getConfigurationPropertiesClause" | INDEX_COLUMNS ) | ( INDEX_COLUMN
"QUOTED_STRING" GET "getConfigurationPropertiesClause" ) );
getPropertiesClause = PROPERTIES "(" "propertyNameList" ")";
getTableSCOClauses = COLUMNS | UNIQUE_KEYS | PRIMARY_KEY | FOREIGN_
KEYS | CHECK_CONSTRAINTS | COLUMN AT POSITION "INTEGER_LITERAL" |
INDEXES | PARTITIONS | PARTITION_KEYS;
getConfigurationPropertiesClause = PROPERTIES "(" "propertyNameList" ")";
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };

```

## Keywords and Parameters

retrieveTableCommand

This clause retrieves a table.

QUOTED\_STRING

Name of the table.

retrieveTableClause

This clause retrieves a table.

retrieveColumnClause

This clause will retrieve columns

QUOTED\_STRING

Name of the column

retrieveUkPkClause

This will get the unique key or primary key clause.

retrieveFkClause

Name of the foreign key

retrieveCheckConstraintClause

This clause gets the check constraint

QUOTED\_STRING

name of the check constraint.

retrievePartitionConfigurationClause

Gets the partition.

QUOTED\_STRING

The partition name.

retrievePartitionKeyConfigurationClause

This clause gets the partition key.

QUOTED\_STRING

The name of the partition key.

retrieveIndexConfigurationClause

Gets the index in this clause.

QUOTED\_STRING

Name of the index.

getPropertiesClause

This clause retrieves all the properties.

Basic properties for TABLE:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the table

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the table

Name: UOID

Type: STRING

Valid Values: N/A

Default: N/A

Basic properties for COLUMN:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the table

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the table

Name: DATATYPE

Type: STRING

Valid Values: NUMBER, VARCHAR, VARCHAR2, DATE, FLOAT

Default: NUMBER

The datatype of a column

Name: LENGTH

Type: NUMBER

Valid Values:

Default: 1

The length of a number

Name: PRECISION

Type: NUMBER

Valid Values: 0 - 39

Default: 1

The precision of a number.

Name: SCALE

Type: NUMBER

Valid Values: -85 - 125

Default: 1

The scale of a number.

Name: UOID

Type: STRING

Valid Values: N/A

Default: N/A

Basic properties for INDEX, PARTITION, PARTITION\_KEY, INDEX\_COLUMN:

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the Index, Partition, PartitionKey, IndexColumn.

Properties for TABLE:

Name: ANALYZE\_TABLE\_ESTIMATE\_PERCENT

Type: NUMBER

Valid Values: 0 - 100

Default: 99

Value represents the sample size as a percentage of total rows. When set to a nonzero value, Builder generates a DDL script to analyze the table.

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: HASH\_PARTITION\_TABLESPACE\_LIST

Type: STRING

Valid Values: N/A

Default: "

A comma separated list of tablespaces to use for [sub]partition storage.

Name: HASH\_SUBPARTITION\_NUMBER

Type: NUMBER

Valid Values: 2 - 63999

Default: 2

To create Hash partition, specify the number of Hash subpartition.

Name: LOGGING\_MODE

Type: STRING

Valid Values: NOLOGGING, LOGGING

Default: LOGGING

Recovery requirements for a data warehouse : Logging or not logging to Redo Log File.

Name: PARALLEL\_ACCESS\_MODE

Type: STRING

Valid Values: NOPARALLEL, PARALLEL

Default: PARALLEL

Enables or disables parallel processing when the table is created. Also enables or disables parallel processing or access.

Name: TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Use the Tablespace parameter to specify the name of tablespace.

Properties for UNIQUE\_KEY:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: USING\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

Use of Index on Dimension Constraint

Properties for FOREIGN\_KEY:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: USING\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false



### Use of Index on Dimension Constraint

#### Properties for CHECK\_CONSTRAINT:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: USING\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

### Use of Index on Dimension Constraint

#### Properties for INDEX:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: INDEX\_TYPE

Type: STRING

Valid Values: BITMAP, UNIQUE, NO\_INDEX

Default: UNIQUE

The types of Indexes created on Dimension are BITMAP, UNIQUE or a non-specific index.

Name: LOCAL\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

A local index is constructed so that it reflects the structure of the underlying table. It is equipartitioned with the underlying table, meaning that it is partitioned on the same columns as the underlying table, creates the same number of partitions or subpartitions, and gives them the same partition bounds as corresponding partitions of the underlying table.

Name: LOGGING\_MODE

Type: STRING

Valid Values: NOLOGGING, LOGGING

Default: LOGGING

Recovery requirements for a data warehouse : Logging or not logging to Redo Log File.

Name: PARALLEL\_ACCESS\_MODE

Type: STRING

Valid Values: NOPARALLEL, PARALLEL

Default: PARALLEL

Enables or disables parallel processing when the table is created. Also enables or disables parallel processing or access.

Name: TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Use the Tablespace parameter to specify the name of tablespace.

Name: USING\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

Use of Index on Dimension Constraint

Properties for PARTITION:

Name: DATE\_LESS\_THAN

Type: STRING

Valid Values: N/A

Default: TO\_DATE('01-JAN-1999', 'DD-MON-YYYY')

Value that represents upper bound of partition stored in warehouse key column for the Days Dimension.

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: EMPTY\_STRING

Use the Tablespace parameter to specify the name of tablespace.

Name: VALUES\_LESS\_THAN

Type: STRING

Valid Values: N/A

Default: TO\_DATE('01-JAN-1999', 'DD-MON-YYYY')

Noninclusive upper bound for the current partition. Type a comma separated list in the same order as the columns in range partitioned key.

Name: VALUE\_LESS\_THAN

Type: STRING

Valid Values: N/A

Default: "

Noninclusive upper bound for the current partition. Type a comma separated list in the same order as the columns in range partitioned key.

Properties for PARTITION\_KEY:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: TYPE

Type: STRING

Valid Values: HASH, RANGE

Default: RANGE

Oracle partitions the storage space and stores rows according to a Hash Algorithm or specified ranges.

Note:

1. N/A means any valid character in supported character set.
2. " " represents an empty string

getTableSCOClause

This clause retrieves components like columns, indexes etc. of a table.

getConfigurationPropertiesClause

This clauses gets the configuration properties of the object.

propertyNameList

The list of properties.

## Examples

OMBRETRIEVE TABLE 'NEW\_TABLE' GET PROPERTIES (DESCRIPTION, BUSINESS\_NAME) GET COLUMNS This will retrieve its description and business name, and get columns.

## See Also

OMBRETRIEVE

# OMBRETRIEVE VIEW

## Purpose

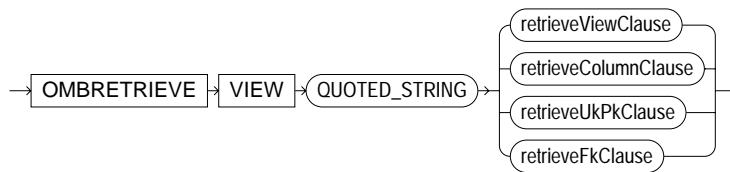
OMBRETRIEVE VIEW - To retrieve properties of a view.

## Prerequisites

In the context of an Oracle Module

## Syntax Diagrams

retrieveViewCommand



retrieveViewClause



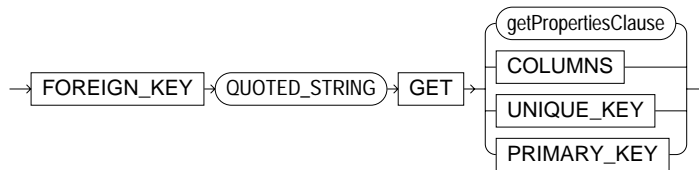
retrieveColumnClause



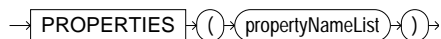
retrieveUkPkClause



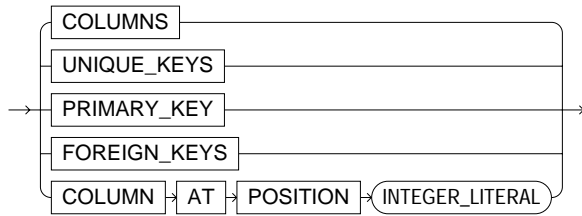
retrieveFkClause



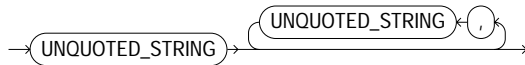
getPropertiesClause



getViewSCOClause



propertyNameList



## Syntax

```

retrieveViewCommand = OMBRETRIEVE VIEW "QUOTED_STRING" (
"retrieveViewClause" | "retrieveColumnClause" | "retrieveUkPkClause" |
"retrieveFkClause" );
retrieveViewClause = GET ( "getPropertiesClause" | "getViewSCOClause" );
retrieveColumnClause = COLUMN "QUOTED_STRING" GET "getPropertiesClause";
retrieveUkPkClause = ( UNIQUE_KEY | PRIMARY_KEY ) "QUOTED_STRING" GET
( "getPropertiesClause" | COLUMNS );
retrieveFkClause = FOREIGN_KEY "QUOTED_STRING" GET ( "getPropertiesClause"
| COLUMNS | UNIQUE_KEY | PRIMARY_KEY );
getPropertiesClause = PROPERTIES (" "propertyNameList" ");
getViewSCOClause = COLUMNS | UNIQUE_KEYS | PRIMARY_KEY | FOREIGN_
KEYS | COLUMN AT POSITION "INTEGER_LITERAL";
propertyNameList = "UNQUOTED_STRING" { "," "UNQUOTED_STRING" };
  
```

## Keywords and Parameters

**retrieveViewCommand**

This clause retrieves a view.

**QUOTED\_STRING**

name of the view.

**retrieveViewClause**

This clause will retrieve a view.

**retrieveColumnClause**

This clause will retrieve columns

---

**QUOTED\_STRING**

Name of the column

**retrieveUkPkClause**

This will get the unique key or primary key clause.

**retrieveFkClause**

Name of the foreign key

**getPropertiesClause**

This clause retrieves all the properties.

Basic properties for VIEW, MATERIALIZED\_VIEW:

Name: BUSINESS\_NAME

Type: STRING(200)

Valid Values: N/A

Default: "

Business name of the View, MaterializedView

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the View, MaterializedView

Name: VIEW\_QUERY

Type: STRING(4000)

Valid Values: N/A

Default: "

Sets the query definition in View and MaterializedView.

Name: UOID

Type: STRING

Valid Values: N/A

Default: N/A

Basic properties for INDEX, PARTITION, PARTITION\_KEY, INDEX\_COLUMN:

Name: DESCRIPTION

Type: STRING(4000)

Valid Values: N/A

Default: "

Description of the Index, Partition, PartitionKey, IndexColumn in a MaterializedView.

Properties for VIEW:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Properties for UNIQUE\_KEY:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: USING\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

Use of Index on Dimension Constraint



## Properties for FOREIGN\_KEY:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: USING\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

Use of Index on Dimension Constraint

## Properties for CHECK\_CONSTRAINT:

Name: DEPLOYABLE

Type: BOOLEAN

Valid Values: true, false

Default: true

Warehouse Builder generates a set of scripts to create an object only for those object marked as Deployable = true

Name: INDEX\_TABLESPACE

Type: STRING(30)

Valid Values: N/A

Default: "

Tablespace to store Index on Table Constraint

Name: USING\_INDEX

Type: BOOLEAN

Valid Values: true, false

Default: false

Use of Index on Dimension Constraint

Note:

1. N/A means any valid character in supported character set.
2. " " represents an empty string

getViewSCOClause

This clause will retrieves components like columns, keys, etc. of a view.

propertyNameList

The list of properties.

## Examples

OMBRETRIEVE VIEW 'NEW\_VIEW' GET PROPERTIES (DESCRIPTION, BUSINESS\_NAME) This will retrieve its description and business name.

## See Also

OMBRETRIEVE

---

---

## Running OMB Plus in Oracle JDeveloper

This appendix outlines the steps to install and run OMB Plus in Oracle Database JDeveloper. For more information, see the Oracle Database JDeveloper documentation.

This appendix includes the following topics:

- [Installing OMB Plus in Oracle JDeveloper](#) on page A-1
- [Opening the Syntax Highlighting Editor in JDeveloper](#) on page A-1
- [Invoking Keyword Auto Completion](#) on page A-2
- [Invoking the OMBPlus Interpreter](#) on page A-2
- [Viewing the OMBPlus Console](#) on page A-2
- [Viewing Help Documentation](#) on page A-2

### Installing OMB Plus in Oracle JDeveloper

Use the following steps to install OMB Plus in Oracle JDeveloper. You must install Oracle JDeveloper and Warehouse Builder client on the same machine.

To install OMB Plus in JDeveloper:

1. Drop `shiphome/owb/lib/int/OMBPlus_jdev.jar` in the JDeveloper installation directory/`lib/ext` directory.
2. Start JDeveloper.
3. From the Tools menu, select **Preferences, OMBPlus**, and then **OMBPlus Installation** to set the Warehouse Builder installation directory. For example, `c:\oracle\ora81`. Do not include the Warehouse Builder directory at the end of this path.
4. Restart JDeveloper.

Repeat steps 3 and 4 each time you change the Warehouse Builder installation directory.

### Opening the Syntax Highlighting Editor in JDeveloper

To open the syntax highlighting editor:

1. From the File menu, select **New** and then **OMBPlus**, if you want to create a new OMBPlus script. To open an existing OMBPlus script, select **File**, and then **Open**.
2. From the Tools menu, select **Preferences**, then **Editor**, and then **Syntax Colors**.
3. Select the OMBPlus Category to configure the syntax highlight styles.

## Invoking Keyword Auto Completion

Use any of the following methods to invoke keyword auto completion:

- Wait for a predefined number of seconds after typing the first few characters of the keyword.

You can configure the predefined number of seconds for invoking the auto completion by changing the Auto-Popup Delay for Completion Insight scale. From the Tools menu, select **Preferences**, then **Editor**, and then **Code Insight**.

- Press **Ctrl** and **Space** after typing the first few characters of the keyword.
- Type any number of the first few characters of the keyword.

## Invoking the OMBPlus Interpreter

**To invoke the OMBPlus Interpreter:**

1. From the View menu, select **Log Window** to view the OMBPlus Log.
2. From the Run menu, select **Run OMBPlus** to run the current OMBPlus script.  
 You can also invoke Run OMBPlus by right-clicking a specific script node on the system navigator and selecting **Run OMBPlus**.  
 You can terminate the OMBPlus program by selecting **Run**, then **Terminate**, and then **OMBPlus**.
3. If you want to provide input to the OMBPlus script, select **Project**, then **Project Settings**, then **Runner**, and then **Options** to set Allow Program Input.

## Viewing the OMBPlus Console

**To open the OMBPlus Console:**

1. Select OMBPlus Console to open the OMBPlus Console window.
2. From the Tools menu, select **Preferences**, then **OMBPlus**, and then **OMBPlus Console** to configure the display in the console window.

## Viewing Help Documentation

Use any of the following methods to invoke the help documentation:

- Invoke the help document for a specific keyword by placing the cursor anywhere in the keyword and then pressing F2.
- Invoke the help document for a specific keyword by selecting the specific keyword and then pressing F2.
- Locate the general help document for OMBPlus scripting language by selecting **OMBPlus**, and then **OMBPlus Help Topics**.

---



---

## Default Groups and Parameters

When you use OMB Plus to add an operator to a mapping, Warehouse Builder adds the operator and assigns default groups and parameters. This appendix lists the default groups and parameters for each operator.

This appendix includes the following topic:

- [Default Group Names and Attribute Names](#) on page B-1

### Default Group Names and Attribute Names

[Table B-1](#) lists the default groups and attributes that are created when an operator is added.

**Table B-1** *Default Names for Groups and Attributes*

Operator Type	Default Operator Name	Default Group Name	Default Attribute Name
TABLE VIEW EXTERNAL TABLE MATERIALIZED VIEW CUBE DIMENSION	Same as bound object name	INOUTGRP1	Same as column names
FLAT FILE	Object name	Same as file record name	Same as field names in each record
SEQUENCE	Same as sequence name	OUTGRP1	NEXTVAL CURRVAL
DATAGENERATOR	DATAGENERATOR	OUTGRP1	RECNUM SYS_DATE SEQUENCE
CONSTANT	CONST	OUTGRP1	No defaults
KEY LOOKUP	Object name	INGRP1 OUTGRP1	In LOOKUP_OUT object column names
SET	SET	INGRP1 INGRP2 OUTGRP1	None

**Table B-1 (Cont.) Default Names for Groups and Attributes**

Operator Type	Default Operator Name	Default Group Name	Default Attribute Name
JOINER	JOIN	INGRP1 INGRP2 OUTGRP1	None
SPLITTER	SPLIT	INGRP1 OUTGRP1 OUTGRP2 REMAINING_ ROWS	None
DEDUPLICATOR	DEDUP	INOUTGRP1	None
AGGREGATOR	AGG	INGRP1 OUTGRP1	None
FILTER	FLTR	INOUTGRP1	None
SORTER	SORT	INOUTGRP1	None
NAME AND ADDRESS	NAMEADDR	INGRP1 OUTGRP1	Within Group INGRP1: Line 1, Line 2, Line 3  Within Group OUTGRP1: First Name, Last Name, Primary Address, Secondary Address, City, State, Postal Code, Is Good Group
PROCEDURES	Procedure name	If input parameter exists, an input group will be created with the name INGRP1.  If output parameter exists, an output group will be created with the name OUTGRP1.  If inout parameter exists, an input-output group will be created with the name INGRP1.	Same as parameter names
FUNCTIONS	Same as function name	INGRP1, RETURN	An attribute
TRIGGER	PREMAP POSTMAP	Naming see PROCEDURES and FUNCTIONS	None
INPUT_PARAMETER	MAP_INPUTS	MAP_INPUTS	None
OUTPUT_PARAMETER	MAP_OUTPUTS	MAP_OUTPUTS	None
EXTERNAL_PROCESS	EXTERNALPROCES S	None	None

**Table B-1 (Cont.) Default Names for Groups and Attributes**

<b>Operator Type</b>	<b>Default Operator Name</b>	<b>Default Group Name</b>	<b>Default Attribute Name</b>
EXPRESSION	EXPR	INGRP1 OUTGRP1	None

