

Oracle® Value Chain Planning
Collections Implementation Guide
Release 12.2
Part No. E48791-05

April 2017

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

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

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

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

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

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

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

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

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Contents

Send Us Your Comments

Preface

1 Overview of Value Chain Planning Suite

Overview.....	1-1
Naming Conventions.....	1-1
Value Chain Planning Suite Applications.....	1-2
Value Chain Planning Integration.....	1-3

2 Overview of Value Chain Planning Collections

Overview.....	2-1
Collection Diagrams.....	2-3
Process.....	2-4
VCP Collections Terminology.....	2-6

3 Architecture

Overview.....	3-1
Centralized.....	3-3
Decentralized.....	3-3
General VCP Products Machine Implementation Options.....	3-5
Rapid Planning Machine Implementation Options.....	3-7

4 Initial Setup

Implementation Steps.....	4-1
---------------------------	-----

Implementation Considerations.....	4-2
Tablespace Considerations.....	4-4
Installing Patches.....	4-5
Creating Links.....	4-6
5 Setting Up Partitions	
Overview.....	5-1
Create APS Partition.....	5-3
ODS Tables and Instance Partitions.....	5-3
PDS and Plan Partitions.....	5-5
ODS and PDS Data.....	5-8
Using Drop Partition Request.....	5-10
6 Setting Up Source and Destination Instances	
Overview.....	6-1
Setting Up the EBS Source Instance.....	6-2
Creating a VCP Destination Instance.....	6-6
Cloning an Instance.....	6-11
Moving From a Centralized to a Decentralized Installation.....	6-13
Resolving a Corrupted Instance.....	6-14
7 Configuring Profile Options	
Overview.....	7-1
EBS Source, VCP Destination, and ATP Profile Options.....	7-2
Flex Field Profile Options.....	7-21
Legacy Profile Options.....	7-23
Other Profile Options.....	7-24
8 Running Collections	
Overview.....	8-1
Running Collections for the First Time in a New Instance.....	8-9
Standard Collections.....	8-11
Continuous Collections.....	8-13
ATP Data Collections.....	8-17
Organization-Specific Collections.....	8-17
Category-Specific Collections.....	8-19
Collecting from Single Source to Multiple Destinations.....	8-22
Running Collections after Cleaning Up ODS Data.....	8-24

9 Collecting from Legacy Systems

Overview.....	9-1
Setting Up the Collection of Transaction Data into the Planning Server.....	9-3
Process.....	9-3
Setting Up Batch Uploads.....	9-5
Sequence of Data Uploads.....	9-7
Preprocessing Engine.....	9-9
Setting Up Legacy Instance.....	9-14
Running Legacy Collections Using the Form-Based Application (Collect Flat-File Data)...	9-15
Running Legacy Collections Using the Self-Service Application (Collect Flat-File Data - Self Service).....	9-16
Purging Legacy Data.....	9-18
SRP Streams - Self Service.....	9-22
Shipment History - Self Service.....	9-23

10 Loading Data from Flat Files

Overview.....	10-1
Preparing Transactional Data for Upload.....	10-1
Loading Data from Flat Files to a VCP Destination Instance Using the Form-Based Application.....	10-2
Collecting into a VCP Destination Instance Using the Self-Service Application.....	10-4
Customizations.....	10-4

11 Managing MLOG Tables

Overview.....	11-1
General Guidelines.....	11-2
MLOG Growth.....	11-3
MLOG Terminology.....	11-4
How Rows are Cleared from MLOG\$ Tables.....	11-5
How MLOGs are Used During Data Collection.....	11-6
Managing the MLOG Growth for VCP Data Collections.....	11-7
Most Common Performance Problems.....	11-9
Loading Large Amounts of Data for Implementation.....	11-20

12 Troubleshooting

Overview.....	12-1
First Steps.....	12-1
Basic Performance Tuning for MSC Partitions.....	12-2

Diagnosing Data Collection Problems.....	12-3
Refresh Collection Snapshots or Planning Data Pull Errors.....	12-4
ODS Load Errors.....	12-4
Refresh Snapshot Process Doesn't Start Because of Concurrent Manager Unavailability.....	12-4
Data Collection Fails with Several Messages.....	12-5
Missing Data Collection Entities.....	12-7
Tracing.....	12-7
Other Troubleshooting Resources.....	12-7
Data Field Notes.....	12-8

A Parameters

Overview.....	A-1
Planning Data Pull Parameters.....	A-2
Planning ODS Load Parameters.....	A-10
Legacy Collections.....	A-12

B Concurrent Programs

Overview.....	B-1
Collection Requests.....	B-1
Setup Requests.....	B-7
Running Refresh Collection Snapshots as a Standalone Concurrent Request.....	B-9
Performance or Errors with the Refresh Collection Snapshots Program.....	B-12

C Snapshots

Overview.....	C-1
List of Snapshots.....	C-1
Invalid Snapshots and Manually Refreshing the Snapshot.....	C-9

D SQL Commands

Overview.....	D-1
Partition Table SQL Commands.....	D-1
Snapshot SQL Commands.....	D-9
MLOG SQL Commands.....	D-11
Tablespace SQL Commands.....	D-27

E List of VCP Partitioned Tables

Summary List of Partitioned Tables.....	E-1
ODS and PDS Partitioned Table.....	E-2

Partitioned Staging Tables for Data Collections.....	E-6
MV Partitioned Tables.....	E-9
Partitioned Tables for APCC.....	E-10

F Detailed Integration Flows

Forecast to ASCP Data Flow.....	F-2
MDS to VCP Data Flow.....	F-4
Purchasing Supply to ASCP Data Flow.....	F-5
Sales Orders to ASCP Data Flow.....	F-7
Sales Orders to MRP Data Flow.....	F-9
WIP to ASCP Data Flow.....	F-10

Index

Send Us Your Comments

Oracle Value Chain Planning Collections Implementation Guide, Release 12.2

Part No. E48791-05

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document. Your feedback is important, and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).

Note: Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the new Oracle E-Business Suite Release Online Documentation CD available on My Oracle Support and www.oracle.com. It contains the most current Documentation Library plus all documents revised or released recently.

Send your comments to us using the electronic mail address: appsdock_us@oracle.com

Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and inquire about our Oracle University offerings. A list of Oracle offices is available on our Web site at www.oracle.com.

Preface

Intended Audience

Welcome to Release 12.2 of the *Oracle Value Chain Planning Collections Implementation Guide*.

Implementers of Oracle Value Chain Planning

See Related Information Sources on page xii for more Oracle E-Business Suite product information.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Structure

- 1 Overview of Value Chain Planning Suite**
- 2 Overview of Value Chain Planning Collections**
- 3 Architecture**
- 4 Initial Setup**
- 5 Setting Up Partitions**
- 6 Setting Up Source and Destination Instances**
- 7 Configuring Profile Options**

- 8 Running Collections**
- 9 Collecting from Legacy Systems**
- 10 Loading Data from Flat Files**
- 11 Managing MLOG Tables**
- 12 Troubleshooting**
- A Parameters**
- B Concurrent Programs**
- C Snapshots**
- D SQL Commands**
- E List of VCP Partitioned Tables**
- F Detailed Integration Flows**

Related Information Sources

- Oracle Advanced Planning Command Center User's Guide
- Oracle Advanced Supply Chain Planning Implementation and User's Guide
- Oracle Collaborative Planning Implementation and User's Guide
- Oracle Demand Planning Implementation and User's Guide
- Oracle Demand Signal Repository Users Guide
- Oracle Demantra Analytical Engine Guide
- Oracle Demantra Implementation Guide
- Oracle Demantra Integration Guide
- Oracle Demantra User's Guide
- Oracle Demantra Deductions and Settlement Management User's Guide
- Oracle Demantra Demand Management User's Guide
- Oracle Demantra Predictive Trade Planning User's Guide
- Oracle Demantra Real-Time Sales and Operations Planning User Guide
- Oracle Demantra Security Guide
- Oracle Global Order Promising Implementation and User's Guide
- Oracle Inventory Optimization User's Guide
- Oracle Order Management Implementation Manual
- Oracle Order Management User's Guide
- Oracle Production Scheduling Implementation Guide
- Oracle Rapid Planning Implementation and User's Guide
- Oracle Service Parts Planning Implementation and User's Guide
- Oracle Strategic Network Optimization Implementation Guide

Integration Repository

The Oracle Integration Repository is a compilation of information about the service endpoints exposed by the Oracle E-Business Suite of applications. It provides a complete catalog of Oracle E-Business Suite's business service interfaces. The tool lets users easily discover and deploy the appropriate business service interface for integration with any system, application, or business partner.

The Oracle Integration Repository is shipped as part of the Oracle E-Business Suite. As your instance is patched, the repository is automatically updated with content appropriate for the precise revisions of interfaces in your environment.

Do Not Use Database Tools to Modify Oracle E-Business Suite Data

Oracle STRONGLY RECOMMENDS that you never use SQL*Plus, Oracle Data Browser, database triggers, or any other tool to modify Oracle E-Business Suite data unless otherwise instructed.

Oracle provides powerful tools you can use to create, store, change, retrieve, and maintain information in an Oracle database. But if you use Oracle tools such as SQL*Plus to modify Oracle E-Business Suite data, you risk destroying the integrity of your data and you lose the ability to audit changes to your data.

Because Oracle E-Business Suite tables are interrelated, any change you make using an Oracle E-Business Suite form can update many tables at once. But when you modify Oracle E-Business Suite data using anything other than Oracle E-Business Suite, you may change a row in one table without making corresponding changes in related tables. If your tables get out of synchronization with each other, you risk retrieving erroneous information and you risk unpredictable results throughout Oracle E-Business Suite.

When you use Oracle E-Business Suite to modify your data, Oracle E-Business Suite automatically checks that your changes are valid. Oracle E-Business Suite also keeps track of who changes information. If you enter information into database tables using database tools, you may store invalid information. You also lose the ability to track who has changed your information because SQL*Plus and other database tools do not keep a record of changes.

Overview of Value Chain Planning Suite

This chapter covers the following topics:

- Overview
- Naming Conventions
- Value Chain Planning Suite Applications
- Value Chain Planning Integration

Overview

To deal with the challenges of today's global economy, companies need to transform their supply chains into information-driven value chains. Successful planning relies on the speed of decision making and continuous monitoring of the impact of those decisions at all levels in the organization. Oracle Value Chain Planning enables best in class processes such as demand sensing, shaping for profitability, fast event-driven scenario simulation, supply chain risk management, sales and operations and integrated business planning, comprehensive trade management, integrated order orchestration and promising, and multi-enterprise planning and collaboration. Oracle Value Chain Planning is a complete planning solution built on a common foundation that leverages pre-built integration with both Oracle (E-Business Suite, and JD Edwards EnterpriseOne) and non-Oracle systems and can be deployed in a modular approach to save time and cost.

Naming Conventions

While Value Chain Planning is the current name of this suite of products, it has gone by different names in the past such as Advanced Planning (AP), Advanced Planning Solutions (APS), Supply Chain Planning (SCP), and sometimes Advanced Supply Chain Planning (ASCP), which is a key component of the Value Chain Planning suite. Some of this naming is still present in program names and options. When referred to generally, the name "Value Chain Planning" will be used. When a specific program is being

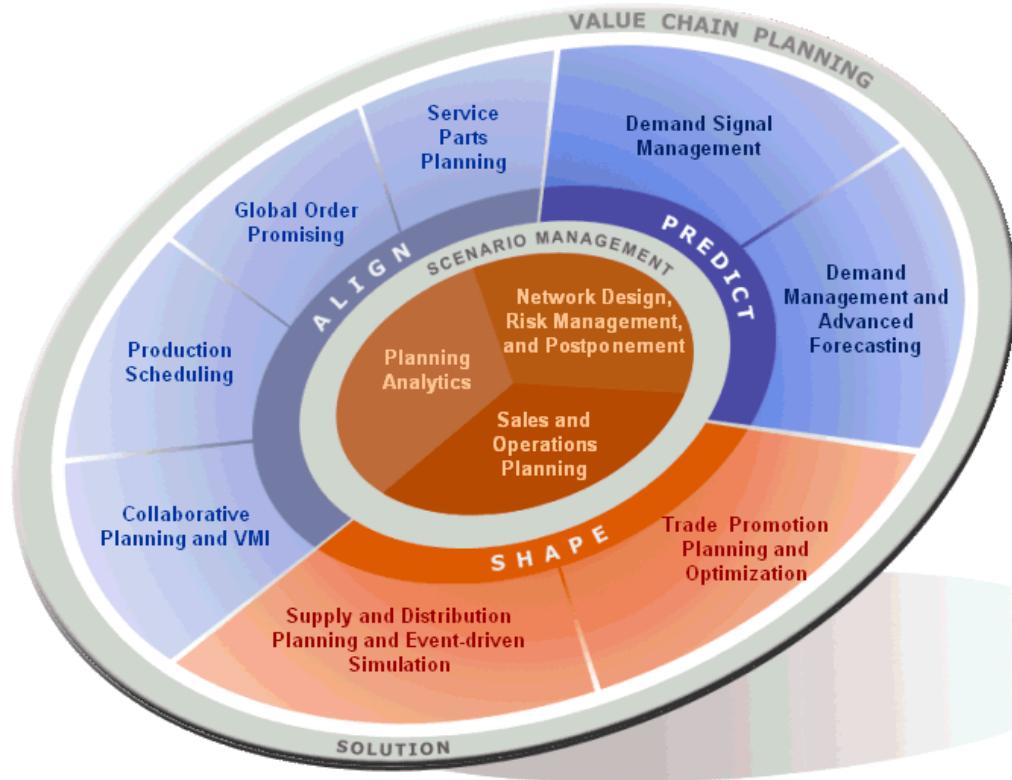
referred to, the actual name will be used, even if it is old terminology such as APS Partitions.

Value Chain Planning Suite Applications

The Oracle Value Chain Planning suite includes the following products:

- Oracle Advanced Planning Command Centre (APCC)
- Oracle Advanced Supply Chain Planning (ASCP)
- Oracle Collaborative Planning
- Oracle Demand Planning
- Oracle Demand Signal Repository
- Oracle Demantra Demand Management and Advanced Forecasting
- Oracle Demantra Sales and Operations Planning
- Oracle Demantra Predictive Trade Planning and Optimization
- Oracle Demantra Deductions and Settlement Management
- Oracle Global Order Promising
- Oracle Inventory Optimization
- Oracle Production Scheduling
- Oracle Rapid Planning
- Oracle Service Parts Planning
- Oracle Strategic Network Optimization

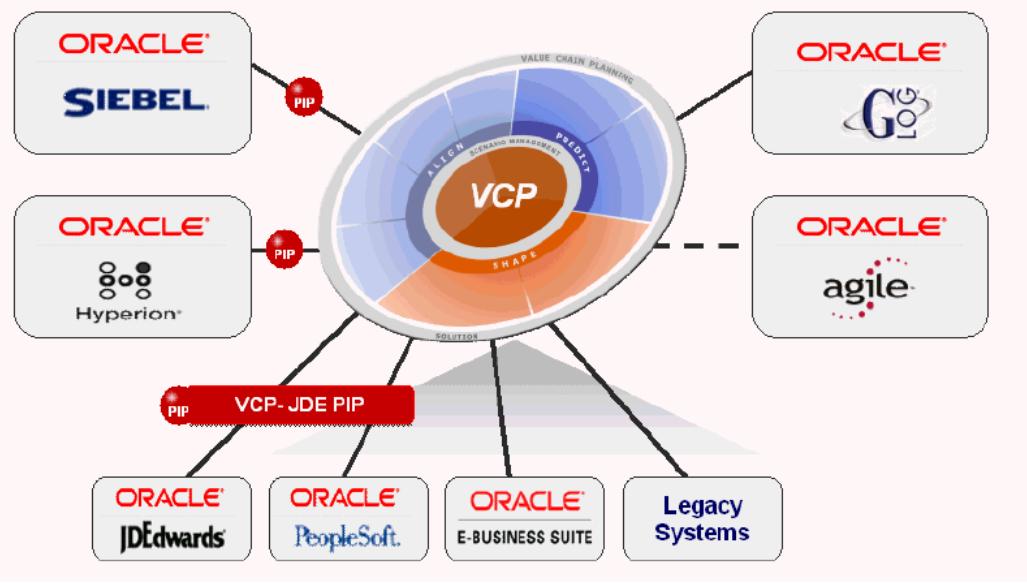
The following graphic shows their relationship and role in Value Chain Planning:



Value Chain Planning Integration

The Value Chain Planning suite leverages pre-built integrations with both Oracle (EBS, JD Edwards, Peoplesoft, Siebel and Hyperion) and non-Oracle ERP and legacy systems. The pre-built integrations with Oracle applications are called Process Integration Packs (PIPs). Oracle's Application Integration Architecture (AIA) makes use of the Oracle Data Integrator (ODI) to transfer and transform data from one application to another according to the PIP.

For most integrations, the VCP collections process is the critical process required to make a subset of EBS data available for planning purposes with other VCP products as well as other ERP systems. The following diagram shows the integrations available between VCP and non-EBS systems:



For documentation about all Value Chain Planning integrations, please see Note #1125094.1, "Value Chain Planning Integration Documentation Library".

For information about integrating with legacy systems, please see Collecting from Legacy Systems, page 9-1.

Overview of Value Chain Planning Collections

This chapter covers the following topics:

- Overview
- Collection Diagrams
- Process
- VCP Collections Terminology

Overview

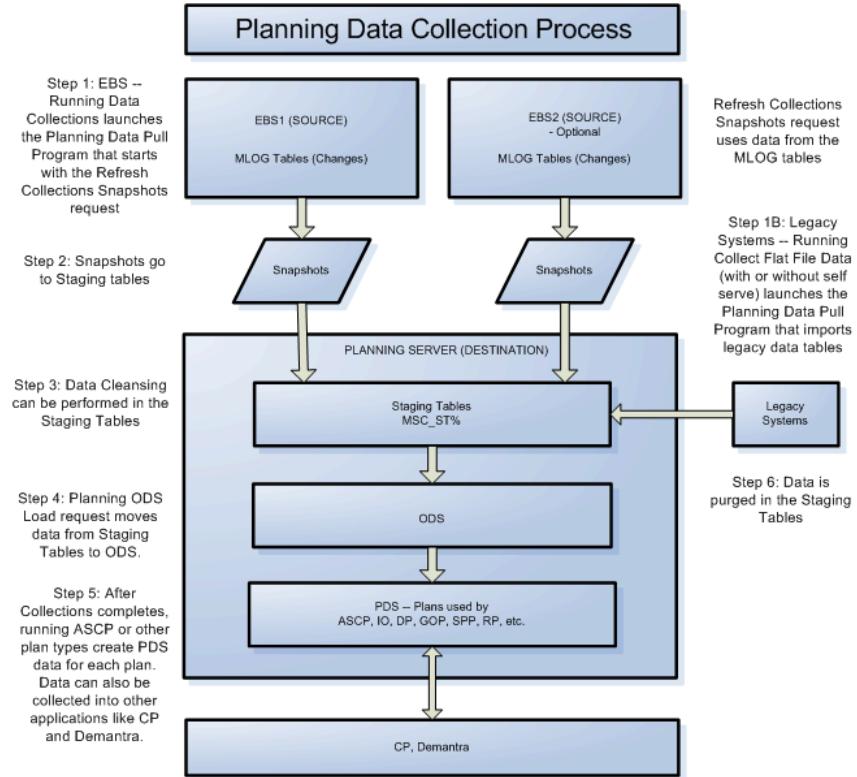
Oracle Value Chain Planning (VCP) has a component architecture that allows a single instance of Oracle VCP to plan one or more EBS source instances. The EBS source instances can be a mixture of releases. The Oracle VCP destination instance (referred to as the planning server) can sit on the same instance as one of the EBS source instances, or be a separate instance altogether. In either case (even if the planning server shares an instance with the EBS source instance to be planned), data to be planned is brought from the EBS source instance(s) to the planning server via a process called Planning Data Collection.

This section describes the architecture used in the collection of planning data from multiple operational sources into Oracle VCP. These sources could be different versions/instances of Oracle applications or other legacy systems. Oracle VCP uses a data store based on the planning data model that is exposed through interface tables. The data is pulled from the designated data sources into its data store; Collections are then responsible for synchronization as changes are made to the data sources. The configurability of the collections is enabled through a pull program based on AOL concurrent program architecture. Thus, for example, different business objects can be collected at different frequencies. Supplies and demands, which change frequently, can be collected frequently. Routings and resources, which change relatively less often, can be collected less frequently.

When collecting from an EBS source instance, the data collection process is performed in two steps: a data pull from the execution module tables on the EBS source instance to the staging tables and a load from the staging tables to Operational Data Store (ODS) load on the APS Planning Destination instance. The collection process lets you collect across several Oracle application versions. It supports several configurations. Types of data collection processes supported:

- Standard: Using the standard collections process, you can manually run three types of data collection methods including a complete refresh, a net change refresh, or a targeted refresh on specific business entities.
- Continuous: The continuous collections process is an automated process of data collection that efficiently synchronizes the data on the planning server by looking up the sources. If you opt for continuous collections, the system automatically determines whether to perform a net change collection of the entity or a targeted collection of the entity on an entity-by-entity basis. The continuous collections process collects data from the sources with the least user intervention. The Continuous Collections concurrent program performs continuous collections.
- ATP: Used by customers using only simple ATP for Order Management in a centralized instance, and not using any other Value Chain Planning applications such as Global Order Promising (GOP). The ATP data collection is a simplified version of the Planning Data Collection with preset program parameters to collect only the data required for ATP based on collected data. Running ATP data collections in complete refresh mode removes the data required for Planning applications. Do NOT use ATP Data Collections when you are using any of the VCP Planning Applications.
- Legacy: The legacy collections process provides an open framework for consulting and system integrators to bring data from legacy systems into Oracle VCP. You can upload data by batch upload of flat files. This is achieved in part by extending the interface table capabilities. A preprocessing engine validates the incoming data from a legacy application and ensures that referential integrity is maintained. Business objects can be imported into VCP using flat files. The legacy collection process can be launched using a form, or through a self-service process.

Collection Diagrams



Important: ATP based on collected data uses the ODS data. ATP based on planning data (GOP) requires that you run a plan to generate PDS data.

More details about the steps:

Step 1B: For Legacy Collections - Collect Flat File data inserts data into the MSC_ST% staging tables the ODS and then launches the pre-processor that validates the data and reports any errors. The pre-processor then launches the ODS Load.

Step 2: After refreshing the snapshots, the Planning Data Pull resumes and uses snapshots and MRP_AP% views to load data into the staging tables.

Step 3: After the Data Pull completes, ODS Load is launched as first calls the custom hook MSC_CL_CLEANSE which can be used to manipulate data for business purposes. See My Oracle Support Note 976328.1 for more information.

Step 4: ODS data is stored in tables with PLAN_ID=-1 and can be viewed in the Collections Workbench.

Step 5: After ODS Load completes, Purge Staging Tables launches. If set up to use Collaborative Planning, the Collaboration ODS Load is launched.

Step 6: After Collections completes, then ASCP or other plan types are launched to create PDS data which is stored in tables with PLAN_ID>0 and can be viewed in the Planner Workbench. The collected data can also be used on other applications like Demantra, Collaborative Planning, Production Scheduling, etc.

Process

There are three key stages to the Data Collections Process:

1. Refresh Collection Snapshots
2. Planning Data Pull
3. Planning ODS Load

Refresh Collections Snapshots

Initially, if the setup profile MSC: Source Setup Required = Yes, then the Refresh Collections Snapshots process launches Setup Requests that create all the objects used by Data Collections in the EBS source applications. When deployed in a distributed installation, these processes are launched on the EBS source instance. When Setup Requests complete successfully, then the profile is set to No by the snapshot process and it continues with standard process flow to refresh the snapshots. In a centralized/single instance installation, all requests are launched on the same instance.

After a complete refresh of a snapshot, when data is subsequently inserted, updated, or deleted in the master table, these changes are inserted into the snapshot log tables (MLOG\$_tables). A snapshot refresh updates the snapshot table with all of the transactions that were logged in the MLOG\$ table. This makes the transaction data available for the Planning Data Pull.

The Planning Data Pull program has a list of parameters (entities) that correspond to one or more of these snapshots. When the Planning Data Pull program is run, it spawns the Refresh Collection Snapshots (RCS) program. Based on the parameters passed to the Planning Data Pull program, the Refresh Collection Snapshots program begins refreshing the snapshots located on the source instance. There are two ways to refresh a snapshot during data collections:

- Fast refresh (most cases): Refreshes the snapshot based on the contents of the MLOG\$_table.
- Complete Refresh: The snapshot is updated by getting all data directly from the application module tables.

When launching Refresh Collection Snapshots as a standalone program, you can also choose type - Automatic Refresh. This truncates the snapshot log when you set the parameter Threshold = 0 and performs a complete refresh of the snapshots. The truncation of the MLOG\$ helps with performance and space allocation for the RDBMS.

Planning Data Pull

Once the data in the snapshot tables is refreshed, the Planning Data Pull code on the VCP destination instance "reads" the data from the snapshots, filtered by views on the EBS source instance, into the MSC_ST% staging tables on the VCP destination instance. The Planning Data Pull process spawns workers to help manage the many tasks (40+) that gather the data into the staging tables.

Planning ODS Load

The Planning ODS Load process starts to move the data from the MSC_ST Staging tables to the MSC planning tables.

The Planning ODS Load performs key transformations of critical data into unique keys for the VCP applications. For example, since VCP can collect from multiple instances, the INVENTORY_ITEM_ID from one instance will likely represent a different item than on another instance. Since the planning process needs one unified unique set of INVENTORY_ITEM_IDs, local ID tables are used to track the instance and INVENTORY_ITEM_ID from a given source instance and the corresponding unique value that is used throughout the rest of the planning process. This happens for item-related entities and for trading partner-related entities (customers, suppliers, organizations). The source instance key values are stored in columns with the prefix SR_. For example, SR_INVENTORY_ITEM_ID in MSC_SYSTEM_ITEMS.

Then, the Planning ODS launches workers to handle the many different load tasks that move data from the MSC_ST% staging tables to the base MSC tables. The Alter Temporary ODS Table concurrent program is launched after the Planning ODS Load workers are finished.

There are three collections methods that can be used to update the ODS data:

- Complete: Replaces all data for the entities except items and trading partners. Items are replaced, but item name is kept in the Key ID tables. Trading partners and partner sites can be used across multiple instances are never removed.
- Target: Collects one or more entities in complete refresh mode.
- Net Change: Picks up only the changes to the transactional entities like WIP, PO, OM and some setup entities like items, BOM, etc. See Data Changes That Can Be Collected in Net Change Mode, page 8-2 for more information about the entities that can be collected in Net Change mode.

ODS Loads creates TEMP tables, then copies data from MSC_ST% staging tables into the TEMP tables, then uses Exchange Partition technology to flip this temporary table into the partitioned table.

For example, if your instance code is TST, you can see in the log file where the temp table SYSTEM_ITEMS_TST is created. Data is then moved into the SYSTEMS_ITEMS_TST table from MSC_ST_SYSTEM_ITEMS. The

SYSTEM_ITEMS_TST temporary table is exchanged with the partition of MSC_SYSTEM_ITEMS used for collected data. In this example, if the INSTANCE_ID is 2021, the partition name shown is SYSTEM_ITEMS_2021.

Finally, the Planning ODS Load launches the Planning Data Collections - Purge Staging Tables process to remove data from the MSC_ST% staging tables. If the MSC: Configuration profile is set to 'CP' or 'CP & APS', then the Collaboration ODS Load process is launched to populate data for Collaborative Planning.

Collection Example

For example, the Items parameter corresponds with the snapshot MTL_SYS_ITEMS_SN. When a change is made to an item (MTL_SYSTEM_ITEMS_B table) or a new item is created, data is inserted into MLOG\$_MTL_SYSTEM_ITEMS_B.

When the Refresh Collection Snapshot runs as a fast refresh, the data in MLOG\$_MTL_SYSTEM_ITEMS_B is used to update MTL_SYS_ITEMS_SN and then the data in MLOG\$_MTL_SYSTEM_ITEMS_B is deleted. The Data Pull process moves this data via views (MRP_AP_SYS_ITEMS_V and MRP_AP_SYSTEM_ITEMS_V) into the staging table MSC_ST_SYSTEM_ITEMS, then the Planning ODS Load moves the data from the staging table to the base table MSC_SYSTEM_ITEMS.

VCP Collections Terminology

You should be familiar with the following terms before examining the data collections architecture:

Oracle Applications Data Store (ADS): The set of source data tables in each transaction instance that contain data relevant to planning.

Operational Data Store (ODS): The planning data tables in the planning server that act as the destination for the collected data from each of the data sources (both ADS and Legacy).

Planning Data Store (PDS): The outputs of the planning process. The PDS resides in the same data tables as the ODS. PDS data is stored with plan ID > 0 corresponding to the plan_id assigned to that plan. ODS data is stored with plan ID = -1..

Standard Data Collection: The standard data collection process enables you to select the mode of data collection. You can use a complete refresh, a net change (incremental) refresh, or a targeted refresh. Standard data collection consists of the following processes:

- **Planning Data Pull:** Collects the data from the ADS and stores the data into the staging tables. This pull program is a registered AOL concurrent program that can be scheduled and launched by a system administrator. If you are using a legacy program, you must write your own pull program.
- **Planning ODS Load:** A PL/SQL program which performs the data transformation

and moves the data from the staging tables to the ODS. This collection program is a registered AOL concurrent program that can be scheduled and launched by the system administrator.

Continuous Data Collection: The continuous data collection process automates the process of looking up the sources to populate the tables on the planning server. With the least user intervention, the continuous data collection process determines the type of collection to perform on each type of entity. The Continuous Collections concurrent process performs continuous collections.

Entity: During Planning Data Collection, data is collected and processed together from each EBS application as a business entity. Running a Planning Data Collection Request Set involves selecting which entities will be collected and which will not be collected. There are two types of entities: Setup and Transactional. Calendars are a setup entity. While calendar information is stored across a number of related tables, all of the calendar information is collected together as a single business entity. Similarly, Purchase Orders/Purchase Requisitions are an transactional entity that are always collected together.

Collection Workbench: The Collection Workbench is a user interface for viewing data collected over to the planning server from the transaction instances. The functionality here is similar to Planner Workbench functionality. For more information on the Planner Workbench, see Overview of Planner Workbench, *Oracle Advanced Supply Chain Planning Implementation and User's Guide*.

Architecture

This chapter covers the following topics:

- Overview
- Centralized
- Decentralized
- General VCP Products Machine Implementation Options
- Rapid Planning Machine Implementation Options

Overview

Oracle Value Chain Planning Suite has a component architecture that separates the transaction data and processing, for example, inventory receipts and order entry, in a source instance and the planning calculations done in a destination instance. This results in:

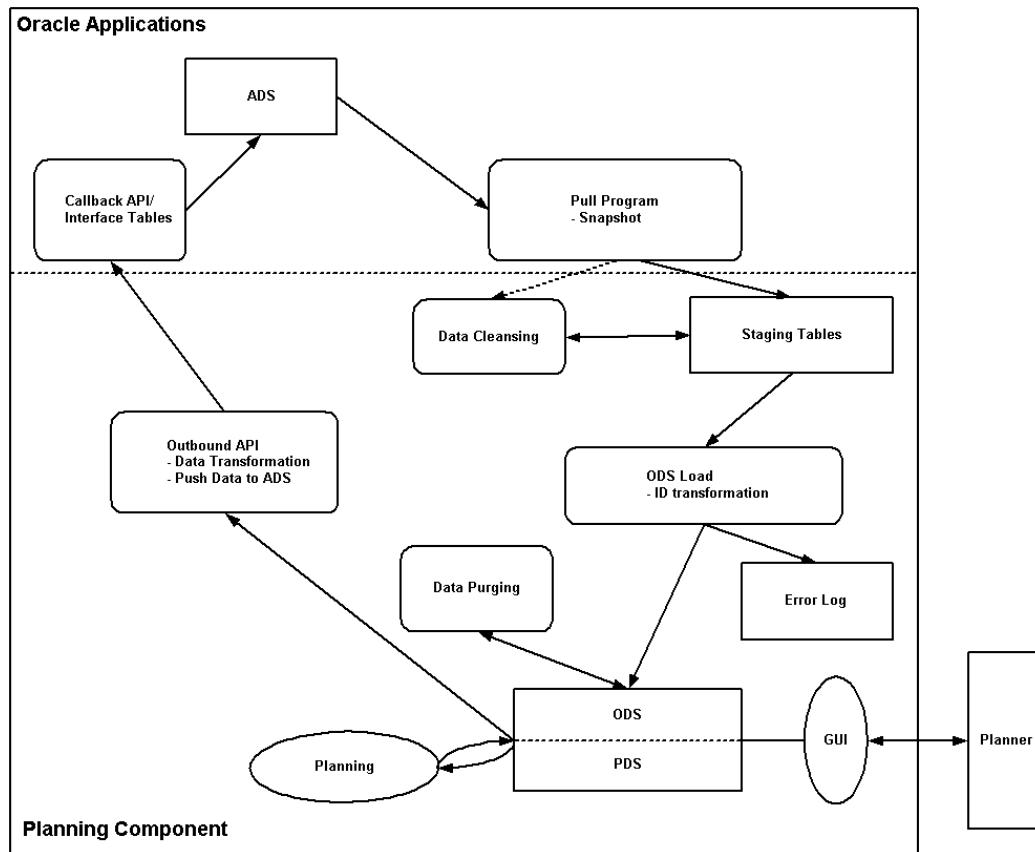
- Better system response
- Planning calculations, for example, demand planning, inventory planning, supply planning and order promising, applying simultaneously to information from multiple source instances, which is useful for a global supply chain spread across multiple source instances.

The source can be any ERP system, but out-of-the-box integration to the Oracle Value Chain Planning Suite destination instance (planning server) exists in some cases but not in all cases.

Both source and destination database instances must be on the same major release of the Oracle database; see Source and Destination Instance Setup, page 6-1.

Oracle Value Chain Planning's data collection architecture, shown in the figure below, depicts the data objects, procedures, and data flow between source data and Oracle planning applications. The major repositories are ADS, ODS, and PDS. Procedures

enable data cleansing, data collecting, data communication, and net-change handling between data repositories.



Supported Configurations

Oracle VCP supports the following planning configurations for installation and deployment:

- Centralized, page 3-3
- Decentralized, page 3-3

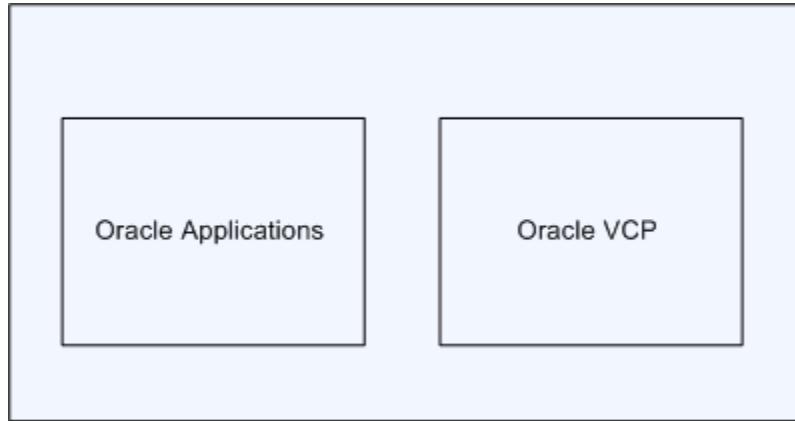
These configurations offer you enough flexibility to design a mode of planning that suits your business objectives. Both configurations are supported using a consistent architecture as outlined in the previous section. The sole distinction is that centralized planning does not use database links to pull data into the Oracle VCP server data store.

To make more detailed machine deployment decisions, review :

- General VCP Products Machine Implementation Options, page 3-5
- Rapid Planning Machine Implementation Options, page 3-7

Centralized

This figure shows the centralized planning configuration.



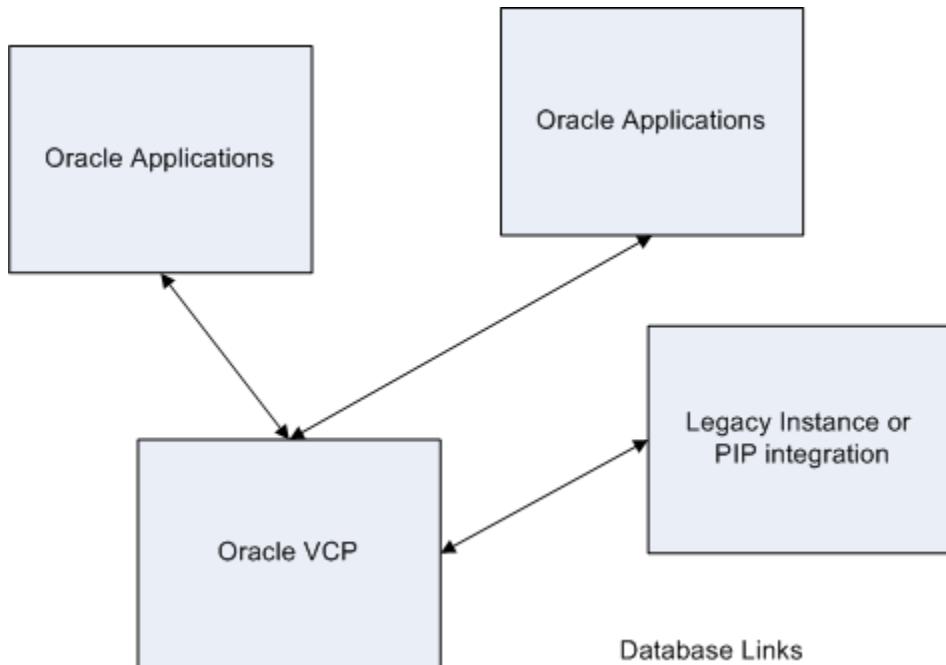
In a centralized scenario, Oracle VCP and its source data reside in the same database. No database link is required in this case. Two components can communicate through the planning object APIs and the interface tables defined in Oracle Applications.

In this configuration, shown in the following figure, a simplified architecture is used, and the data transformation is not required.

When patching for a centralized installation, all patches are applied to the single instance. Ignore references in the patch readme for centralized and distributed installation and apply all patches to your single instance. Most patches mention to just apply the patch to your instance when centralized.

Decentralized

The following figure shows the decentralized planning configuration:



In this scenario, Oracle VCP works as a central Planning Server across one or more source data instances. The collection program is installed on the planning server and the data stripped by instance is moved into staging tables within Oracle VCP during the data collection process.

After the planning process, results can be pushed back to each instance. The planning process is generally done by a planner via the Planners Workbench.

Benefits of distributed installation:

- The processing performed for Planning in ASCP, IO, DRP, etc. can take huge amounts of memory and disk IO. This can slow down the OLTP (OnLine Transaction Processing) processing while the plan processes are running. It is not unusual for large plans to consume anywhere from five to over twenty gigabytes of memory while running and move millions of rows in and out of tables to complete the planning processes.
- It is possible to have VCP destination on a different release. VCP destination could be 12.2 with EBS source running 12.1. You may not have an VCP destination with a lower version than the EBS source. For example, VCP destination on 12.0 or 11.5 with EBS source on 12.1. For more information about compatible versions, please see My Support Oracle Note #1361221.1 -- FAQ - Getting Started With 12.2 for Value Chain Planning (VCP) Applications.
- It is possible to have more than one EBS source connected to the VCP destination. This requires careful consideration as patching for all instances must be maintained. For example, two EBS source instances connected to one VCP destination instance and all are on the same EBS applications release. For VCP patches that affect

collections, all three instances must be patched at the same time in order to prevent breaking this integration. Another example might be a customer with two EBS sources on different releases. VCP destination on 12.1 with an EBS source on 12.1 and another on 11.5.10.

Patching for a distributed installation requires careful attention to the patch readme to make sure the patches are applied to the correct instance and that prerequisite and post-prerequisite patches are applied to the correct instance. Patch readme should be clear on where the patch is to be applied. Usually, the patch refers to the EBS source instance and to the VCP destination instance where the patch is to be applied.

General VCP Products Machine Implementation Options

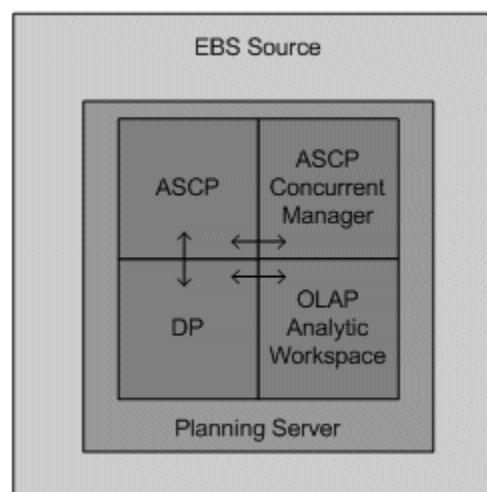
Consider from among these machine implementation options if you are implementing VCP products that use collected data other than Rapid Planning. For Rapid Planning, see Rapid Planning Machine Implementation Options, page 3-7.

Oracle Demand Planning also uses a third instance, an OLAP Analytic Workspace (AW) database, to hold data while multidimensional manipulation of demand data occurs.

One-Machine Implementation

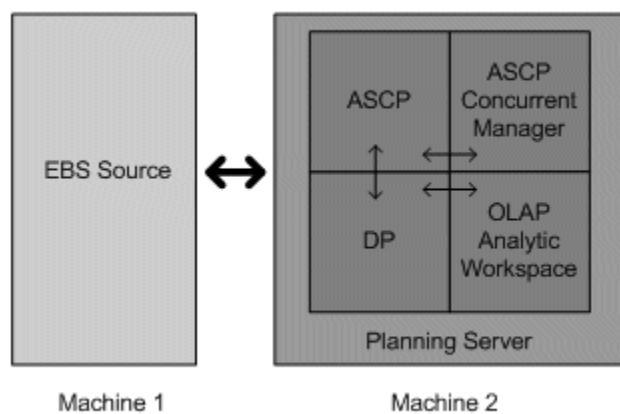
For small implementations, source, destination, and Express can reside on the same machine and be in the same instance. The following figure illustrates this configuration.

Note: This example uses ASCP and DP. ASCP means Advanced Supply Chain Planning and DP means Demand Planning.



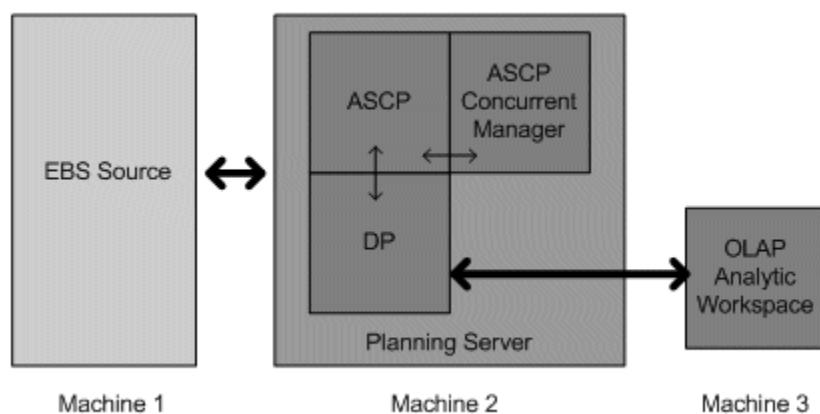
Two-Machine Implementation

For larger implementations where system throughput is more important, the various instances can be deployed on separate machines. A two-machine deployment configuration is appropriate when the size of the demand planning data is relatively small. The following figure illustrates this configuration.



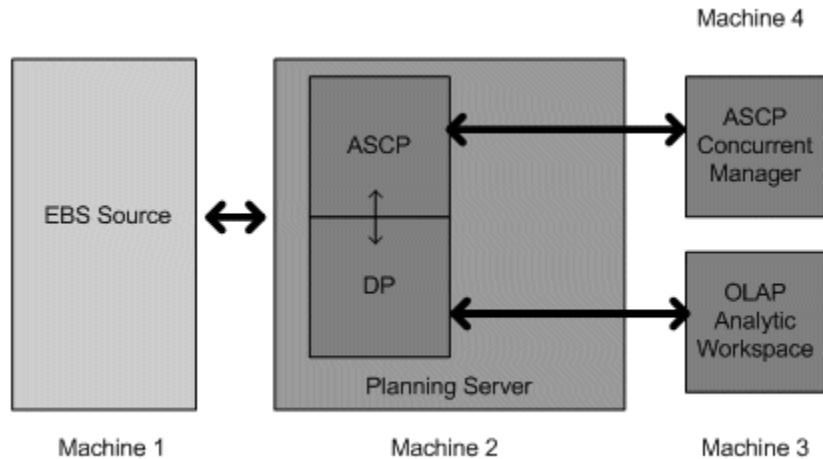
Three-Machine Implementation

A three-machine deployment allows for the manipulation of high-dimensionality, large-scale demand planning data to occur on a machine separate from the planning calculations done on the planning server. The following figure illustrates this configuration.



Four-Machine Implementation

The Advanced Supply Chain Planning concurrent manager may also be deployed on a separate machine. This creates even greater system performance. The following figure illustrates the four-machine implementation.



In all deployment configurations, a collection process brings data from the EBS source tables to the VCP destination tables. A build process brings data from the VCP destination tables to the OLAP Analytic Workspace (AW) for Demand Planning. Finally, a publish process moves data from the AW to the VCP Destination tables to be used for planning and can also publish data back to the EBS source instance.

Global Order Promising and Inventory Optimization planning calculations are also performed on the planning server.

Rapid Planning Machine Implementation Options

Consider from among these machine implementation options if you are implementing Rapid Planning. For VCP products that use collected data other than Rapid Planning, see General VCP Products Machine Implementation Options, page 3-5.

Rapid Planning has these components:

- Source ERP application, for example Oracle e-Business, JD Edwards, and legacy systems: A database tier and an application tier.
- Value Chain Planning planning server: A database tier and an application tier. The database tier supports and manages the Oracle database instance. The application tier includes forms services, web services, and the concurrent processing server.
- WebLogic Server

Single Machine Architecture

All the components are on one machine. The Source ERP application and the Value Chain Planning planning server use the same database tier and application tier.

Use this option with these data volumes:

- Items: Fewer than 10,000
- Demands: Fewer than 50,000
- Planning users: 1 to 5

If you use this configuration with higher data volumes, you can overload your server.

Two Machine (Distributed) Architecture

The source ERP application is on one machine. The Value Chain Planning planning server and the WebLogic Server are on one machine.

This is how the machines communicate for collections and release when the source ERP application is:

- Oracle e-Business Suite: Create a database link.
- JDEdwards 9.0: Use the the JDE-VCP integration Process Integration Pack.
- Another system: Use flat files/legacy collections.

Three Machine Architecture

The source ERP application is on one machine, the Value Chain Planning planning server is on one machine, and the WebLogic Server is on one machine.

This is how the machines communicate for collections and release:

- The same way as the Two Machine (Distributed) Architecture
- Cross mount the location for log files and data files that the concurrent programs on the VCP planning server write with the WebLogic server.

Tier Sizing for Machine Architecture

See:

- Hardware Architecture, *Oracle Rapid Planning Installation Guide*
- Sizing Template

Initial Setup

This chapter covers the following topics:

- Implementation Steps
- Implementation Considerations
- Tablespace Considerations
- Installing Patches
- Creating Links

Implementation Steps

The following steps are recommended when you are planning to implement Value Chain Planning:

- Assess your architecture requirements, page 3-1
- Review implementation considerations, page 4-2.
- Review the tablespace considerations, page 4-4
- Install the software and patches, page 4-5 on both the source and destination
- Set up partitions, page 5-1 on the VCP server
- Set up and test the links between the source and destination databases, page 4-6
- Set up source and destination instances, page 6-1
- Configure profile options, page 7-1 on both the source and destination
- Run refresh collection snapshot, page 8-9 before running data collections

If you are planning to import legacy data, see Loading Data from Flat Files, page 10-1

for more information.

Implementation Considerations

Major features of the collections process include:

- Multiple Source Instances
- Pull Architecture
- Collect Net Changes from Oracle Applications to the Oracle VCP Planning Server
- Multi-Process Collection Architecture
- Data Consolidation
- Projects/Tasks, and Seiban Numbers
- Support for several Oracle Applications Versions and RDBMS Versions
- Support for Several Configurations

Multiple Source Instances

You can register any number of source data instances and non-Oracle data sources on each Oracle VCP installation.

Pull Architecture

You can collect new source data instances into Oracle VCP with minimal impact. The data is pulled from the source data instance by Oracle VCP. Each instance can have its own refresh interval. A failure in one instance will not affect data collections from other instances.

Collect Net Changes from Oracle Applications to the Oracle VCP Planning Server

You can collect net changes in Oracle EBS Source instances and the Oracle VCP Planning server using the Net Change mode. Thus, only the changed source data is collected each time, reducing the computational burden on the collection process.

Note: Only certain business entities can be collected in Net Change mode.

See Data Changes that can be Collected in Net Change Mode, page 8-2

Multi-Process Collection Architecture

You can enhance the performance of the pull program by distributing the tasks to multiple collection workers.

Data Consolidation

The collection program can consolidate the entities shown in the following table across instances based on the corresponding user-defined keys.

Table Name	Entity	User Key
MTL_SYSTEM_ITEMS_B	Items	Concatenated Item Segments
MTL_CATEGORIES	Items	Concatenated Category Name
MTL_CATEGORY_SETS	Items	Category Set Name
PO_VENDORS_VIEW	Suppliers/Customers/Orgs	Vendor Name
PO_VENDOR_SITES_ALL	Suppliers/Customers/Orgs	Vendor Site Code
RA_CUSTOMERS	Suppliers/Customers/Orgs	Customer Name
HZ_CUST_ACCT_SITES_AL L	Suppliers/Customers/Orgs	Customer Name, Site Use Code, Location Operating Unit
MTL_UNITS_OF_MEASURE	Units Of Measure	UOM Code

For all the entities not described in the table, the instance ID together with the entity key in each instance uniquely identifies each row.

Projects/Tasks and Seiban Numbers

You can consider Projects, Tasks, and Seiban Numbers to be unique within the context of an Oracle Applications instance; no consolidation is required.

Support for Several Configurations

You can implement centralized or decentralized configurations based on the scale of the enterprise and specific business needs. Source data applications and Oracle VCP can reside on one server or on separate servers.

Tablespace Considerations

When implementing Value Chain Planning, it is important to understand its space requirements, and how space is consumed in the VCP tablespaces. Because data for MSC tables grows very quickly, you must plan your disk space needs and configure the tables and tablespaces. The Rapid Install of Oracle Application in 11.5.10 and above uses the Oracle Applications Tablespace Model (OATM), a consolidated model of just a couple of dozen tablespaces. For more information about OATM, see Oracle Applications Concepts 11.5.10.2 [http://download.oracle.com/docs/cd/B25284_01/current/acrobat/11iconcepts.pdf].

Evaluating the System:

For a full implementation of the VCP Applications, the Sizer for APS is available internally at <http://www-apps.us.oracle.com/aps/>. Click the APS Sizer button. Use this tool to determine some of the initial sizing for the database. Alternatively, you can download the sizer spreadsheet template [<https://support.oracle.com/CSP/main/article?cmd=show&type=ATT&id=1301231.1:XLS>].

Note: The Sizer for APS is separate from the standard sizing information available for an EBS applications install.

If Data Collections for Order Management ATP is the only part of the VCP implementation being used, the sizing tool will not provide a correct sizing because it is not set up for this type of implementation. In this case, the DBA can use SQL #28 -- Get Table Sizes for Tables Exceeding 10 MB, page D-27 to evaluate tables in the EBS source schemas for INV, BOM, WIP, and ONT for size and then come to a determination on the sizing for the MSC schemas based on the experience of running Data Collections. Generally, a very rough estimate of the sizing that could be used is:

```
(total bytes of MTL_SYSTEM_ITEMS + MTL_ITEM_CATEGORIES )* 3 + 50%
```

Although this should allow for enough space initially, it is still possible that certain system setups will outstrip this estimate and it is not guaranteed. For example, if your BOM or WIP is highly used and very large, then increase this estimate. Once Data Collections has been run, then the system can be evaluated again for sizing and adjustments made as needed.

Note: After Data Collections are run for all organizations to be collected and before any ASCP plans are run, the space used is approximately 40% of the total space needed just for Data Collections to be run successfully. All the data being collected exists more than two times during Data Collections. This is because the MSC_ST% tables are populated during the Planning Data Pull process with all the data that will be in the base tables after the end of Data Collections. Then, during ODS Load, TEMP tables are created to process data in the MSC_ST%

staging tables. Data loaded in the TEMP tables is moved into the MSC base tables using RDBMS Exchange Partition technology. Lastly, the MSC_ST% staging tables are purged at the end of the Data Collections cycle. More space is required for each plan that is run.

If you are loading large amounts of data during the implementation, review the Loading Large Amounts of Data for Implementation, page 11-20 section in the *Managing MLOGs* chapter for additional configuration suggestions.

Appendix E, page D-1 contains a number of SQL commands that are useful for determining how the system has been set up and its space usage:

- Use the SQL statements SQL #30 -- Review Tables and Indices with Zero Percent Increase, page D-27 through SQL #35 -- Check Individual Datafiles for Free Space Available, page D-30 to understand space usage.
- Review Setting Up Partitions, page 5-1 and run SQL #1 -- Show Partitioned Tables in the System, page D-1 through SQL #5 -- Show Plans and Associated Plan Partitions, page D-3 to understand how the system has been set up. If Data Collections has been run, then the system will have data that can be evaluated. If Data Collections has not been run, then the system can be evaluated for initial sizing and then reexamined after the initial Data Collections run.

For ATP based on collected data, remove any extra plan partitions to reduce the number of objects to a minimum. Then run and save the output of SQL #1 and SQL #5 again for future reference.

See Also:

- Setting Up Partitions, page 5-1 to understand how to manage VCP partitions in the MSC schema.
- My Oracle Support Note #396009.1-- Database Initialization Parameters for Oracle Applications R12 to evaluate your database initialization parameters.

Installing Patches

When installing and using VCP applications, you must begin by installing the entire EBS application on the EBS server. Next, you need to install ASCP and your other VCP applications on either the EBS server for centralized installations or the VCP server for decentralized installations. There is no standalone install package limited to VCP applications.

Install the latest release of EBS applications and during the implementation cycle, if a new release of EBS becomes available, plan to install that release as soon as possible. If not running any ASCP plans, then no plan partitions are required and can be dropped. See Drop Partition Request, page 5-10 for more information.

Please refer to My Support Oracle Note #1361221.1 -- FAQ - Getting Started With 12.2 for Value Chain Planning (VCP) Applications for the latest patches for 12.2, the supported integrations to earlier releases, and any patches that need to be installed on a separate EBS source instance.

For a distributed installation, you may have to apply patches on the EBS source instance.

If Demantra integration is required, review My Oracle Support Note #470574.1 - List Of High Priority Patches For Oracle Demantra Including EBS, Siebel and E1 Integrations for the list of high priority patches for Oracle Demantra including EBS integration.

Creating Links

When a separate VCP destination instance is used from the EBS source instance, there must be a database link on the destination that allows programs running on the destination to launch programs (like Refresh Collection Snapshots) on the source or read data from the source (like Planning Data Pull). Similarly, there must be a database link on the source that allows programs running on the source to pull data from the destination to the source. An example of this happens when planned orders generated by the planning process are implemented by releasing them to the source instance. The data flow for such a release includes code that runs on the EBS source instance that pulls released planned order data from the destination to the source so that the requisition or job can be created on the source.

Setting Up TNSNAMES for the Source and Destination Instances

Add the source and destination TNSNAMES to IFILE in the tnsnames.ora file. This file is created for the purpose of maintaining information outside of the AutoConfig managed files.

In this example, our EBS source instance is called NIC and located on the server copernicus.us.oracle.com. The VCP destination instance is called KEPLER and located on server kepler.us.oracle.com. We will create two db-links:

- For NIC on copernicus.us.oracle, we have to create the entry to KEPLER.
- For KEPLER on kepler.us.oracle.com, we have to create an entry to NIC.

1. Sign in as an ORACLE user.

```
[iracke@cioerbucyks ~]$ cd $TNS_ADMIN
[oracle@copernicus admin]$ ls
listener.ora tnsnames.ora
[oracle@copernicus admin]$ more tnsnames.ora
#####
#
# This file is automatically generated by AutoConfig. It will be
read and
# overwritten. If you were instructed to edit this file, or if you
are not
# able to use the settings created by AutoConfig, refer to Metalink
Note
# 387859.1 for assistance.
#
#####
nic=
      (DESCRIPTION=
      ....
```

At the bottom of the file, we see the IFILE info:

```
IFILE=/d04/oracle/nic/inst/apps/nic_copernicus/ora/10.2.0
/network/admin/nic_copernicus_ifile.ora
```

2. Open the IFILE nic_copernicus_ifile.ora and add connection information for the VCP destination instance.

```
more /d04/oracle/nic/db/tech_st/10.2.0
/network/admin/nic_copernicus/nic_copernicus_ifile.ora

kepler=(DESCRIPTION= (ADDRESS=(PROTOCOL=tcp)
                      HOST=kepler.us.oracle.com) PORT=1521))
          (CONNECT_DATA=(SERVICE_NAME=kepler)
          (INSTANCE_NAME=kepler))
```

3. Open the IFILE kepler_kepler_ifile.ora and add connection information for the EBS source instance.

```
[oracle@kepler kepler_kepler]$ more kepler_kepler_ifile.ora

nic= (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=copernicus.us.
oracle.com)(PORT=1520))
      (CONNECT_DATA=(SERVICE_NAME=nic) (INSTANCE_NAME=nic)))
```

Creating and Testing the DB_LINKS

1. Sign in as the SYSTEM User (or sysdba) in SQL*Plus on the EBS source instance NIC and create, query and check the DB_LINK.

A. Create the DB_LINK

```
SQL> create public database link NIC_TO_KEP
  connect to apps
  identified by apps
  using 'kepler';
```

B. Query the setup:

```
SQL> select * from dba_db_links;
select * from dba_db_links where db_link like 'NIC%KEP%'
```

OWNER	DB_LINK	USERNAME	HOST
CREATED			
PUBLIC	NIC_TO_KEP.US.ORACLE.COM	APPS	
kepler	4/3/2009 10:40:29 AM		

C. Check the DB_LINK:

```
SQL> select instance_name from v$instance@NIC_TO_KEP;
```

INSTANCE_NAME
KEPLER

2. Sign in as the SYSTEM User (or sysdba) in SQL*PLUS on the VCP destination instance KEPLER and create, query and check the DB_LINK.

A. Create the DB_LINK

```
SQL> create public database link KEP_TO_NIC
  connect to apps
  identified by apps
  using 'nic';
```

B. Query the setup:

```
SQL> select * from dba_db_links where db_link like 'KEP%NIC%'
```

OWNER	DB_LINK	USERNAME	HOST
CREATED			
PUBLIC	KEP_TO_NIC.US.ORACLE.COM	APPS	nic
4/3/2009 2:57:42 PM			

C. Check the DB_LINK:

```
SQL> select instance_name from v$instance@KEP_TO_NIC;
```

INSTANCE_NAME
NIC

Setting Up Partitions

This chapter covers the following topics:

- Overview
- Create APS Partition
- ODS Tables and Instance Partitions
- PDS and Plan Partitions
- ODS and PDS Data
- Using Drop Partition Request

Overview

Partitions are an RDBMS concept to subdivide a table or index. When a table is partitioned, any index on that table is also partitioned. Partitions are used in VCP to segregate data. This improves overall performance because the data can be manipulated separately by partition. VCP uses partitions for some of the data that is needed by the Operational Data Store (ODS) and the Planning Data Store (PDS). The same tables are generally used for both ODS and PDS data and are segregated by the plan ID and by the instance ID. Each instance defined in the Instances form gets assigned an Instance partition. Each plan defined in a Plan Names form is assigned a plan partition.

The default install creates one instance partition and five plan partitions. This allows users to proceed with standard setups for connecting to a single EBS source instance and create five plans in the application. The information below is required knowledge for understanding how to manage connections when cloning instances and connecting to different EBS source instances. The DBA and VCP Super User MUST understand these concepts and be able to query and understand the use of partitions to manage the post cloning steps, manage connections to different EBS source and Legacy instances, clean up old instance data, and initiate connections to new, different EBS sources.

The ODS is the destination of collected data. After the data collections have completed successfully, the collected data is inserted into the ODS partition for that instance. These

rows have a PLAN_ID = -1. If several Oracle instances are collected, instance data is stored in individual ODS instance partitions. In addition, the tables contain a column called SR_INSTANCE_ID or INSTANCE_ID that identifies the ERP source instance where the data originated.

The PDS is the destination of ASCP planning data. After an ASCP plan has completed successfully, the data is inserted into the PDS plan partition and other tables that are not partitioned. These rows have a PLAN_ID > 0. In the default setup, each plan has its own plan partition. The tables also contain a column called SR_INSTANCE_ID or INSTANCE_ID that reveals the EBS source instance where the data originated.

The default installation of VCP creates one instance partition and five plan partitions. They can be viewed using the following SQL:

```

SELECT
    table_name,
    partition_name,
    num_rows,
    high_value,
    sample_size,
    last_analyzed,
    global_stats
FROM
    all_tab_partitions
WHERE
    table_name like 'MSC%'
    order by substr(partition_name,instr(partition_name,'_',-1,1)+1)
    -- To check the partition count of each table use order by
table_name, partition_name

```

The following table shows the default installation for one table -- MSC_SYSTEM_ITEMS. (use where TABLE_NAME like 'MSC_SYSTEM_ITEMS' for the above SQL). For a full list of tables, see Appendix F -- List of APS Partitioned Tables, page E-1.

TABLE_NAME	PARTITION_NAME	Comments
MSC_SYSTEM_ITEMS	SYSTEM_ITEMS_0	The Template partition - NEVER DELETE THIS PARTITION
MSC_SYSTEM_ITEMS	SYSTEM_ITEMS_1	Instance Partition #1
MSC_SYSTEM_ITEMS	SYSTEM_ITEMS_1	Plan Partition #1
MSC_SYSTEM_ITEMS	SYSTEM_ITEMS_2	Plan Partition #2
MSC_SYSTEM_ITEMS	SYSTEM_ITEMS_3	Plan Partition #3
MSC_SYSTEM_ITEMS	SYSTEM_ITEMS_4	Plan Partition #4

TABLE_NAME	PARTITION_NAME	Comments
MSC_SYSTEM_ITEMS	SYSTEM_ITEMS_5	Plan Partition #5

Note: If there is a drastic mistake made during the manipulation of ODS/PDS data, keep in mind that this data is either source data collected from the ERP source, as in the case of the ODS, or plan data, as in the case of the PDS. The ODS data can be recollected and the PDS data can be reproduced by submitting the ASCP plan.

Create APS Partition

This concurrent program is run from the System Administrator responsibility. Alternatively, the concurrent program may have been assigned to All MSC Reports, which means it can be run from either Advanced Supply Chain Planner OR Advance Planning Administrator. The parameters are:

- Plan Partition Count: The number of new plan partitions to create. Each ASCP plan needs its own plan partition to manage the plan's PDS (output) data.
- Instance Partition Count: The number of new instance partitions to create. Each collection instance in the Instances form requires an instance partition.

Note: There should not be many extra partitions created. This can cause performance issues. For plan partitions, there should not be more than two to three extra plan partitions on the system. For instance partitions, there should not be any extra instance partitions on the system.

ODS Tables and Instance Partitions

To add one new ODS partition, submit the request Create APS Partitions by setting the concurrent program parameters plan_partition_count=0 and inst_partition_count=1.

The ODS partitioned tables have __nn (double underscore and the instance ID) as shown below. Appendix F -- List of APS Partitioned Tables, page E-1 has a complete list of the tables.

TABLE_NAME	PARTITION_NAME	NUM_ROWS
MSC_SUPPLIES	SUPPLIES_21	14700

ODS partitions are created using the sequence MSC_APPS_INSTANCES_S and are visible in the table MSC_INST_PARTITIONS.

SQL #2

```
-- Free_flag
1 = Free
2 = In use in the MSC_APPS_INSTANCES table
select      instance_id,
            free_flag,
            creation_date,
            last_update_date
from        msc_inst_partitions;
```

INSTANCE_ID	FREE_FLAG	CREATION_DATE	LAST_UPDATE_DATE
21	2	5/8/2000 0:40	5/8/2000 0:40
61	2	1/25/2002 11:28	1/25/2002 11:28
81	2	10/14/2002 13:44	10/14/2002 13:44
101	1	9/17/2003 23:34	9/17/2003 23:34

In the above table, instance_id 101 has been created and still has free_flag = 1, which means it can be assigned to a new instance for data collections. Once a partition has been assigned to an ERP source instance using the Application Instance Setup form, the FREE_FLAG = 2 and a line is inserted into MSC_APPS_INSTANCES.

SQL #3

```
-- ST_STATUS
0 = No source pull or collection is in process
1 = Pull and collection process has begun
2 = Collection has ended and waiting for load to begin
3 = Load has begun
4 = Load has ended and staging tables are being purged
select      instance_code,
instance_id,
apps_ver,
a2m_dblink,
m2a_dblink,
st_status
from        msc_apps_instances
```

INSTANCE_CODE	INSTANCE_ID	APPS_VER	A2M_DBLINK	M2A_DBLINK	ST_STATUS
TST	21	3			0
LEG	61	-1			0
V86	81	3	APS	visus86	0

Notes about the above table:

- TST is assigned to the local centralized ERP instance. That is why a2m_dblink and m2a_dblink are NULL. V86 is assigned to an instance on another server and dblinks were created and populated in the Instance Setup form so that the connection could be made for data collections and other operations to be successful. See Defining Source Link, page 6-2 for more information on this process.
- LEG is for custom collections from a non-Oracle data source (apps_ver = -1). An example is when all data is being populated from an external system using Legacy Collections.
- The ODS data can be identified by the PLAN_ID. For ODS data, this is always -1. When more than one instance is being collected, differentiate this data by the SR_INSTANCE_ID or INSTANCE_ID columns in the table. See INSTANCE_ID column above for the number.
- All Columns beginning with SR% in the MSC tables refer to a value obtained from the EBS source instance (for example, SR_INSTANCE_ID, SR_INVENTORY_ITEM_ID, etc.).

PDS and Plan Partitions

The partitioned tables also have partitions for each plan and they have names ending in _nn (single underscore and the plan ID) as shown below. Appendix F -- List of APS Partitioned Tables, page E-1 has a complete list of the tables.

TABLE_NAME	PARTITION_NAME	NUM_ROWS
MSC_SUPPLIES	SUPPLIES_347	900

The PLAN_ID and PARTITION_NUMBER are controlled by the sequence MSC_PLANS_S. The PDS PLAN_ID is identified by selecting the PLAN_ID from MSC_PLANS.

SQL #4:

Plan Type valid values:

- 1 -- Manufacturing Plan (MRP)
- 2 -- Production Plan (MPS)
- 3 -- Master Plan (MPP)
- 4 -- Inventory Plan (IO)
- 5 -- Distribution Plan (DRP)
- 6 -- SNO Schedule
- 7 -- Production Schedule (PS)
- 8 -- Service Parts Plan (SPP)
- 9 -- Service Parts IO Plan (SPIO)
- 101 -- Rapid Plan MRP
- 102 -- Rapid Plan MPS
- 103 -- Rapid Plan MPP

```

select
  mp.COMPILE_DESIGNATOR "Plan Name",
  mp.PLAN_ID "Plan ID",
  mp.SR_INSTANCE_ID "Instance ID",
  mtp.ORGANIZATION_CODE "Owning Org",
  mp.PLAN_COMPLETION_DATE "Last Run Date",
  decode (mp.PLAN_TYPE, 1, 'Manufacturing MRP', 2, 'Production MPS',
          3, 'Master MPP', 4, 'IO Plan', 5, 'Distribution DRP'
          ,
          7, 'PS Production Schedule', 6, 'SNO Schedule',
          8, 'Service Parts SPP', 9, 'Service IO Plan',
          101, 'Rapid Plan MRP', 102, 'Rapid Plan MPS',
          103, 'Rapid Plan MPP', plan_type) "Plan Type",
  decode (md.PRODUCTION, 1, 'Yes', 2, 'No', NULL, 'No') "Production Flag",
  decode (md.LAUNCH_WORKFLOW_FLAG, 1, 'Yes', 2, 'No', NULL, 'No') "Launch Workflow",
  decode (md.INVENTORY_ATP_FLAG, 1, 'Yes', 2, 'No', NULL, 'No') "ATP Plan",
  mp.CURR_START_DATE "Start Date",
  mp.CUTOFF_DATE "End Date"
  from
  msc_designators md,
  msc_plans mp,
  msc_trading_partners mtp
  where
  md.designator=mp.compile_designator and
  md.sr_instance_id=mp.sr_instance_id and
  mtp.sr_instance_id=mp.sr_instance_id and
  md.sr_instance_id=mtp.sr_instance_id and
  mtp.sr_tp_id=md.organization_id and
  mtp.sr_tp_id=mp.organization_id and
  mp.organization_id=md.organization_id
  and mtp.partner_type = 3
  ORDER BY "Plan Name";

```

Plan Name	Plan ID	Instance ID	Owning Org	Last Run Date	Plan Type	Production Flag	Launch Workflow	ATP Plan	Plan Start Date	Plan End Date
JMCVCONALT	5759	64	BOB:D2	3/21/2007 3:54	Manufacturing MRP	No	No	No	3/21/2007	6/18/2007
JMESL30	6759	64	BOB:M1	6/8/2006 14:11	IO Plan	No	No	No	8/14/2006	3/11/2007
JMMRPSHRK	11758	64	BOB:S1	6/1/2006 15:09	Master	No	No	No	6/1/2006	1/14/2007
JMPROD	21758	64	BOB:M1	3/6/2007 1:39	Production MPS	No	No	No	3/6/2007	10/14/2007
JMUNCSS	14759	64	BOB:M1	7/13/2006 17:03	Master	No	No	No	7/13/2006	2/11/2007

SQL #5

MSC_PLAN_PARTITIONS shows how planning partitions are used when the profile MSC: Shared Plan Partition = No

```

select      plan_id,
            plan_name,
            free_flag,
            partition_number
  from      msc_plan_partitions

```

PLAN_ID	PLAN_NAME	FREE_FLAG	PARTITION_NUMBER
347	DG-APS1	2	347
391	PRN_SCP	1	391

PLAN_ID	PLAN_NAME	FREE_FLAG	PARTITION_NUMBER
411	411	1	411

Observations about the above table:

- DG-APS1 is a plan that is being used (FREE_FLAG = 2).
- PRN SCP is a plan that has been purged, but a new plan has not been defined that uses that partition_number. The plan name is replaced when a new plan and the plan options have been defined.
- Plan name 411 is a new partition created by the Create APS Partitions request and has never been used. Therefore, two new plans could be defined in the Names form in ASCP. After the plan options are defined and saved, this table will be updated with the new plan name and the FREE_FLAG changed to 2.
- If no rows with FREE_FLAG = 1 are returned, then you may still create the new plan in the Names form, but when attempting to define and save the plan options, an error will be received similar to 'No free partitions available, Contact the DBA to create partitions'. This is because the Names form only inserts into the table MSC_DESIGNATORS. When plan options are defined, then the process checks MSC_PLAN_PARTITIONS and updates the FREE_FLAG and PLAN_NAME. The tables inserted/updated are MSC_PLANS, MSC_PLAN_ORGANIZATIONS, MSC_DESGINATORS and MSC_SUBINVENTORIES. Use the concurrent request Create APS Partitions with parameters Instance Partition Count = 0 and Plan Partition Count = 1 to create one new plan partition.

ODS and PDS Data

Data within the ODS and PDS is identified by SR_INSTANCE_ID (or INSTANCE_ID) and PLAN_ID. Rows with a PLAN_ID = -1 belong to the ODS and if more than one EBS source instance is being used, the SR_INSTANCE_ID identifies the other EBS source instance. Rows with a PLAN_ID > 0 belong to the PDS and can also be isolated by using SR_INSTANCE_ID.

To count the number of rows in a specific table, use the following SQL: All rows within MSC_SYSTEM_ITEMS, which includes all partitions.

```
select count(*)
from msc_system_items;
```

Count

6876

In a multiple source situation, the following SQL shows the rows and the PLAN_ID as well as the SR_INSTANCE_ID to indicate the origin of each row:

```
select  count(*),
        sr_instance_id,
        plan_id
  from    msc_system_items
 group  by sr_instance_id,
           plan_id
```

COUNT(*)	SR_INSTANCE_ID	PLAN_ID
3132	21	-1
2931	81	-1
56	81	147
1	81	148
10	21	149
36	21	181
2	81	183
2	81	184
5	21	221
1	81	222
10	21	224
43	21	263
108	21	283
6	21	303

COUNT(*)	SR_INSTANCE_ID	PLAN_ID
376	21	323
115	21	325
24	21	347
9	81	347
9	81	349

From the queries above, data has been collected from the TST and V86 instances, INSTANCE_ID 21 and 81 respectively. This ODS data resides under PLAN_ID –1. We have not yet loaded any legacy data for INSTANCE_CODE LEG, INSTANCE_ID 61. Additionally, several plans have been run and items populated according to those plans.

Note: PLAN_ID 347 was run against the ODS data collected from both instances, therefore this PLAN_ID is listed twice in this SQL showing the rows which belong to each SR_INSTANCE_ID.

Using Drop Partition Request

Sometimes, you may need to reorganize your partitions by dropping a partition. For example, you may need to clean up after data corruption or cloning. The following procedures provide instructions for setting security to allow the dropping of plan partitions, as well as specific instructions for dropping both plan and instance partitions.

Important: Never drop the _0 Partition – this template partition is used to create new partitions. This will cause the instance to fail and it will have to be restored from a backup, a clone or reinstalled.

When to Use this Request

1. You are using ATP based on collected data and not running any VCP plans. Then, the five plan partitions that were created during installation should be removed. In this case, you can proceed to run the request using the parameters described below.
2. A user has created too many plan partitions or instance partitions. These partitions

have never been used and need to be removed from the system to prevent performance issues. In this case, you can proceed to run the request using the parameters described below. SQL #3 -- Show Instance Partitions Created and Assigned to an EBS Source Instance, page D-2 and SQL #5 -- Show Plans and Associated Plan Partitions, page D-3 will show the partitions on the system and which ones are free (free_flag=1).

3. You have old ASCP plans and now you want to remove them and reduce the number of plan partitions on the system. SQL #5 -- Show Plans and Associated Plan Partitions, page D-3 will show which plan partitions are free and which are in use. Before dropping a plan partition that is in use (msc_plan_partitions.free_flag=2), the plan must first be purged via the Names form. Once the plan is purged, then you will see that free_flag=1 and you can drop the plan partition.
4. You have changed/cloned your EBS Source instance and need to purge ALL the old ODS instance and PDS instance data to clean up the system and prepare to use a new EBS Source instance. In this case, all plans for this old instance need to be purged via the Names form prior to dropping the instance partition -- see Instance Cleanup and Reset, page 5-12 below.

Setting Security to Allow Dropping Partitions

The Drop Partition request must be added to the All MSC Reports request group as follows:

1. Using the System Administrator responsibility, navigate to Security, then Responsibility, and then Request.
2. Query for the 'Group'%MSC% to retrieve 'All MSC Reports'.
3. Highlight a line in the form and use /File/New.
4. Enter the following:
 - Type: Program
 - Name: Drop Partition
5. Save your work. The program will now be available for responsibilities that use this request group such as Advanced Supply Chain Planner.

Parameters

- Partition -- [enter number]. Never use 0 [zero], this is the template partition!
- Plan -- Yes/No. Use 'Yes' if this is a plan partition. Use 'No' if this is an instance

partition.

Instance Cleanup and Reset

Check your setups for instance partitions via SQL #3 -- Show Instance Partitions Created and Assigned to an EBS Source Instance, page D-2 and the related plans via SQL #4 -- Show Plans Created in the System, page D-2. Then do the following:

1. Purge all plans in the instance partition before dropping the partition:
 - Navigate to the Plan Names form.
 - Choose your organization (if prompted).
 - Highlight the plan name, then use /edit/delete. Acknowledge the message in the form, then Save. This launches the Purge Designator request. (Note: this request is designed to be launched via the form only.)
 - Change orgs as required to purge all plans that exist for that instance.
2. Drop the partition relating to the previous EBS Source instance using the Drop Partition Request and set the parameter Plan = No.
3. In the EBS Source instance, remove the line from MRP_AP_APPS_INSTANCES_ALL using SQL #10 -- Check MRP_AP_APPS_INSTANCES Table on the EBS Source Instance and Delete Line, page D-6.
4. Dropping the instance partition does not remove all the data that has been collected for that INSTANCE_ID. Use SQL #6 -- Delete All Tables Where INSTANCE_ID Column Appears, page D-5 and SQL #7 -- Delete All Tables Where SR_INSTANCE_ID Column Appears, page D-5 to delete all the VCP destination instance data. For instances with Oracle Demand Planning (ODP) data, also see SQL #13 -- Delete All Tables Where the INSTANCE_ID Column Appears for ODP Data, page D-8, SQL #14 -- Delete All Tables Where the INSTANCE_ID Column Appears for ODP Data, page D-8, and SQL #14B -- Delete All Tables Where the Instance Column Appears for ODP Data, page D-9.
5. Run Clean up Instance Partitions with parameter Mode -- Repair to clean up any partitions in the staging table associated with the old instance.
6. For connecting to a new EBS Source instance, you need one instance partition with free_flag=1. If none exists, then run Create APS partitions with instance partition = 1 and Plan partition = 0. Once completed, then you can begin setup to connect the new EBS source to this APS destination.

Clean Up Instance Partition Request

This request is designed to check the partitions created for the MSC ST Staging tables and if there are any errant partitions or missing partitions, then it can remove or create as required. This program has a single parameter -- Mode:

- List only: This will list any partition problems discovered, but take no action.
- Repair: This will list and then fix any partition problems discovered. This can be run at any time in List mode to check the system. If an Instance partition is dropped using Drop Partition, then this program can be run to ensure that all staging table partitions have been cleaned up.

Setting Up Source and Destination Instances

This chapter covers the following topics:

- Overview
- Setting Up the EBS Source Instance
- Creating a VCP Destination Instance
- Cloning an Instance
- Moving From a Centralized to a Decentralized Installation
- Resolving a Corrupted Instance

Overview

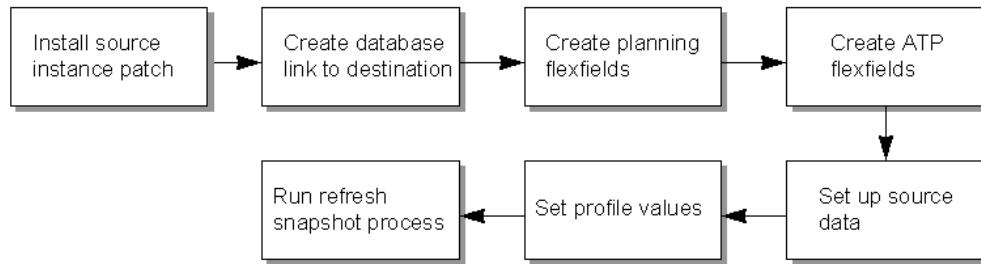
The setup for Value Chain Planning consists of steps for the EBS source instance and steps for the VCP destination instance.

In Oracle Value Chain Planning where you have a distributed installation with separate EBS source and Planning Server (VCP destination instance), there is support for disparate database releases in all supported Oracle RDBMS releases. For instance, you could have an 11.5.10 EBS source instance on 10gR2 and your 12.1 VCP destination on 11gR2.s

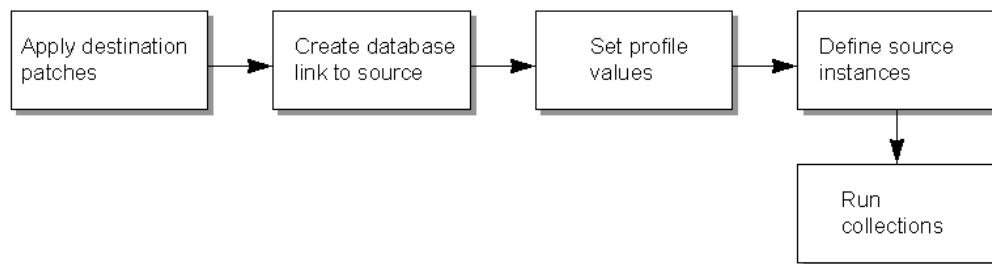
The following figure is a flowchart illustrating the source and destination setup steps.

Source and Destination Instance Setup

Source Instance Setup



Destination Instance Setup



Setting Up the EBS Source Instance

Before data collections can be run, the EBS source must be properly configured. In particular, the following steps need to be addressed before configuring the destination instance:

- Setting security
- Creating a link to the destination instance
- Setting up planning flexfields
- Entry of source data
- Configuring EBS profile options
- Running the Planning Manager in certain business cases to collect the data for PO and WIP entities. Please review My Oracle Support #790125.1 -- "When Does the Planning Manager Need to be Running?" for more information.

In a centralized install, you can assign the user Advanced Supply Chain Planner responsibility and they will be able to perform all steps to run collections, run ASCP plans, and release recommendations. For other types of planning (IO, DRP, SPP), their

respective responsibilities also perform the same functions. The VCP super user would need the Advanced Planning Administrator responsibility to define the instance and set organization security for each responsibility. In a distributed installation where the EBS source instance is release 12.1 or higher, users must have the responsibility APS COLLECTIONS on the source instance to run collections and APS RELEASE on the source instance to release planned orders.

1. Create a database link pointing to the planning server.

A database link must be established on the EBS Source instance that points to the APS Destination (planning) instances, MRP: ATP Database Link. This database link will also be used, along with a database link established on the destination instance, in setting up the instances on the planning server.

2. Create Advanced Supply Chain Planner responsibilities in the source instance related to the collections process. See Defining Source Link.
3. The Create Planning Flexfields concurrent program creates new segment definitions in existing descriptive flexfields to hold data that may be required for constrained and/or optimized planning. The program also populates profile values with the value corresponding to the descriptive flexfield attribute number for each attribute (planning parameter) created. The collections process uses the profile values for the flexfield attribute to collect any values for entities (such as items and bills of material) when you populate them in the source instance.

Note: These values are not required at the item level. If they are not set at the item level, they will be defaulted at the plan level by the constrained planning engine. Oracle recommends that you use the plan level constraints until you have found a specific need to modify at a lower level.

See Appendix A.3 -- Flex Field Profile Options, page 7-21 for more details.

Launch the Create Planning Flexfields report from the System Administration responsibility. The parameters that must be set for the report are the attributes that you wish to utilize for the new flexfield definitions. The list of values for each parameter lists only the available attributes in the subject descriptive flexfield.

Note: Keep track of the attribute number that you select for each flexfield segment. You will need to verify that each corresponding profile option was populated with the correct attribute number when the process completes.

After submitting the program, eleven additional processes should be spawned. These jobs are compiling the descriptive flexfield views.

Check that the profile values corresponding to each flexfield attribute were

populated with the correct attribute number. Some profile values may retain the value of "unassigned" after the Create Planning Flexfield program completed. You must change any unassigned profiles to the attribute number corresponding to the flexfield attribute where the new segment was defined. For more details, see Appendix A.4 -- Flex Field Profile Options, page 7-21.

4. Set profile values. For more details about the key profile options, see Appendix A.2 -- Profile Options, page 7-2.

Important: Verify two profiles: MSC: Share Plan Partition set to No (set on the APS destination in a distributed install), and the MSC: Source Setup Required profile is set to Yes (set on EBS Source in a distributed install)

5. Set up source data with BOMs, resources, routings, supplier data, flexfields, purchasing information, item masters, Oracle BIS targets, and any other data required by your plans.

Defining Source Link

This describes the details of Advanced Supply Chain Planner responsibilities in the source instance related to the collections process.

Source Instance Prior to Release 12.1

You must assign this Advanced Supply Chain Planner Responsibility with the specifications listed below to the User who will launch Collections and Release from the workbench:

- Responsibility Name: Advanced Supply Chain Planner
- Application: Advanced Supply Chain Planning
- Description: Used for running Refresh Collection Snapshots
- Menu: MSC_TOP_4.0
- Data Group Name: Standard
- Application: Advanced Supply Chain Planning
- Request Group: All MSC Reports
- Application: Advanced Supply Chain Planning
- Oracle Advanced Supply Chain Planning uses this responsibility to run the concurrent program Refresh Snapshot during all data collection runs.

Warning: If you create custom responsibilities for your users on the APS Destination to launch Data Collection or Release planned orders from the Planner Workbench, then you must also define those responsibilities on the EBS Source instance. When creating those responsibilities, the following fields must match exactly or the Release process will fail: Responsibility Name, Application, Responsibility Key, and Description.

For other fields, you should use the same fields used for Advanced Supply Chain Planner. The user will not use this responsibility for any data processing. It must exist only for processes launched on the APS destination which must have this information on the EBS Source to complete processing successfully.

Source Instance At Release 12.1 or Later

You must assign this Advanced Supply Chain Planner Responsibility, with the specifications listed below, to the User who will launch Collections and Release from the workbench:

- First Responsibility Name: APS Collections
- Application: Advanced Supply Chain Planning
- Description: APS Collections Responsibility
- Menu: MSC_COLL_REL_MENU
- Data Group Name: Standard
- Application: Advanced Supply Chain Planning
- Request Group: All MSC Reports
- Application: Advanced Supply Chain Planning
- Assign the responsibility to user names in the source instance that are authorized to run the APS collection processes.
- Second Responsibility Name: APS Release
- Application: Advanced Supply Chain Planning
- Description: APS Collections & Release
- Menu: MSC_COLL_REL_MENU
- Data Group Name: Standard

- Application: Advanced Supply Chain Planning
- Request Group: All MSC Reports
- Application: Advanced Supply Chain Planning
- Assign the responsibility to user names in the source instance that are authorized to run the APS release processes.

Creating a VCP Destination Instance

After setting up and configuring the EBS source instance, define the VCP destination instance.

To set up a VCP destination instance:

1. Use either of these GUIs:

To use Oracle Forms, select responsibility Advanced Planning Administrator and select Admin > Instances. Do not access this form while the collections process is running; it locks a table that the collections process needs to complete successfully. The Application Instances window appears

To use the Advanced Supply Chain Planning Supply Planning Work Area, see chapter Supply Planning Work Area, Manage Instances in *Oracle Advanced Supply Chain Planning Implementation and User's Guide*.

2. Enter for each of the EBS source instances that you want the Planning Server planning engine to plan.
3. Complete the fields and flags in the Application Instances window as shown in the table below.

Note: You are only required to set up VCP destination instances before the first time you perform data collection.

Field/Flag	Description
Instance Code	<p>This is a user-defined 1-3 character name for the EBS source instance to be planned. It appears in front of organization names and other designators. For example, if two EBS source instances TA1 and TA2 are to be planned, and both instances have an internal organization named M1, ASCP will display TA1's organization as TA1:M1 and TA2's organization as TA2:M1.</p> <p>In some of the SQL examples in this guide, V86 was used when setting up INSTANCE_ID 81.</p>
Instance Type	<p>The valid values here are Discrete, Process, Discrete and Process, and Others (for Legacy collection instances). This controls whether discrete manufacturing data or process manufacturing data (or both) are collected from the EBS source instance to the planner server for planning. Only use choices that include Process if you have implemented Oracle Process Manufacturing on this instance. Legacy is used when collecting from the applications other than E-Business Suite (EBS) application.</p>
Version	<p>The Oracle Application version of the EBS source instance. The Planning Data Pull program adjusts how it interacts with the source based on this setting.</p>
From Source to APS	<p>Application Database Link Name: Used in a distributed installation. Define a link to connect the EBS source instance to the VCP destination instance. This link is determined by the database administrator. This database link is defined on the EBS source transaction instance and points to the VCP destination planning server. This link is used to publish releases from the VCP destination instance back to the EBS source instance such as purchase orders or work orders.</p>
	<p>For legacy and centralized installations, leave blank.</p>

Field/Flag	Description
From APS to Source	<p>Planning Database Link Name: Used in a distributed installation. Enter the link created on the VCP destination server that points to the EBS source transaction server. This link is determined by the database administrator. This link is used to find and collect the EBS source transaction data to be collected for planning.</p> <p>For legacy and centralized installations, leave blank.</p>
Enable Flag	Select this option to enable the collection process.
Allow ATP	<p>When enabled, the planning engine performs the ATP process in the destination instance specified, taking into consideration the plans selected for ATP in the Plan Names form.</p> <p>You can select the Allow ATP check box for only one of the destinations that are associated with a single source. The planning engine displays an error message if you attempt to select more than one destination for ATP.</p> <p>For more details about how this field works in a multiple destination scenario, see Example: Using the Allow ATP Option With Multiple Destination Instances, page 6-10.</p> <p>ATP-able instances are displayed in the EBS source table MRP_AP_APPS_INSTANCES_ALL.</p>
Staging Instance	This flag enables dual collection processes from an EBS source instance to both an VCP destination instance and a RP (Rapid Planning) instance at the same time. Only for use with Rapid Planning collections.

Field/Flag	Description
Allow Release	<p>Select the Allow Release check box for the destination instances that are allowed to release automatically to the source instance. The Allow Release check box is used for both auto release and manual release processes.</p>
	<p>To trigger release, the Production checkbox must be selected in the plan options.</p>
	<p>For more details about how the Allow Release option works in a multiple destination scenario, see Example: Using the Allow Release Option With Multiple Destination Instances, page 6-10.</p>
GMT Difference	<p>The difference between planning server time zone and GMT. For example, if your server is in the Eastern Standard Time zone (EST) , enter -5.</p>
	<p>Note: This setting is ignored in the 12.2 Planning Data Collection releases.</p>
Currency	<p>Enter your currency code. For example, for US currency, enter 'USD'.</p>
Assignment Set	<p>Optionally, enter the default assignment set for this instance. The assignment is used to resolve sourcing for order promising inquiries.</p>
	<p>Leave blank if you are working with multiple assignment sets.</p>
Collect All Category Sets	<p>Use this field to specify the category sets to be collected for each instance. See Category-Specific Collections.</p>

4. Enter the organizations on each of the defined EBS source instances from which to collect planning data by clicking Organizations.

The Organizations window appears.

5. Select the organizations for **each** instance. Depending on your Family Pack level, all of the organizations for **each** source instance should be visible or available by clicking on the Organization lookup.

- Collection group -- a free form field to assign a name to be used to collect only specific organizations when running data collections. The option to choose a collection group is one of the Planning Data Pull Parameters, page A-2.
- Be sure to select the master organization and enable the organization to be used for ATO and inline forecast explosion during the ASCP plan.

6. Close the Organization window.
7. Save the Instance Definition setup.
8. Verify that the new destination instance is ready for collections by using SQL #3 -- Show Instance Partitions Created and Assigned to an EBS Source Instance, page D-2 to make sure that MSC_APPS_INSTANCES.ST_STATUS=0 for the INSTANCE_CODE.. If the ST_STATUS is not 0 for the INSTANCE_CODE, run Planning Data Collection - Purge Staging Tables with the parameter Validation = No to reset the status and clean up the collection tables.
ST_STATUS values:
0 = No source pull or collection is in process.
1 = Pull and collection process has begun.
2 = Pull has ended and is waiting for load to begin.
3 = Load has begun.
4 = Load has ended and staging tables are being purged.

Example: Using the Allow ATP Option With Multiple Destination Instances

Consider:

We allow a single EBS source to be connected to multiple APS destination instances. For example: A single source instance S that is associated to two destination instances D1 and D2:

- When you define the instance S in D1 destination instance, the Allow ATP check box is selected for destination instance D1.
- Now you go to instance D2 and also define connection to the same source S, and then check the Allow ATP flag. Validation occurs and you get the error message "Source instance S already has a designated ATP destination instance. Please uncheck first the 'Allow ATP' flag in the destination instance corresponding to the source APS".

Conclusion: You cannot set up two different APS instances with Allow ATP for the same EBS source.

Example: Using the Allow Release Option With Multiple Destination Instances

Consider:

- A single source instance S that is associated to two destination instances D1 and D2.
- The Allow Release check box is selected for D2 and not for D1.

The plans in D2 with the Production check boxes selected in the Plan Options form trigger the Auto Release process to source.

You can select the Production check boxes for plans in D1 but they will not trigger any Auto Release process to source. This is because the Allow Release check box is not checked for D1 in the Application Instances form.

Conclusion: You can select the Allow Release check box for multiple destinations that are associated to a single source.

Cloning an Instance

There are times when you want to clone an instance, like when you want to create an duplicate instance for testing purposes. The procedures differ depending on whether it is a centralized or decentralized installation.

Centralized Installation

When you have a centralized installation, then you do not have to consider any steps after cloning your instance and can proceed with normal operations of running data collections and most VCP applications. However, if after cloning you want to change the instance code in the Applications Instances form, you must follow steps in Instance Cleanup and Reset, page 5-12 to clean up all the old data before you create a new instance.

Warning: If you just change the instance code in the form, then this will cause data corruption.

Decentralized Installation

There are many different strategies to cloning when you have a distributed installation such as using separate databases for VCP and EBS applications.

When you create the line in the Applications Instances form on the VCP destination (navigation: Advanced Planning Administrator/Admin/Instances), you are:

- Using an INSTANCE_ID from MSC_INST_PARTITIONS where FREE_FLAG = 1 to store information on these setups you define in the form.
- Inserting a line into MSC_APPS_INSTANCES on the VCP destination instance with the Instance Code and INSTANCE_ID.
- Inserting a line into the MRP_AP_APPS_INSTANCES_ALL table on the EBS source

instance.

When creating orgs in the Organizations region of the Applications Instances form, you are inserting rows in the VCP destination table MSC_APPS_INSTANCE_ORGS for that INSTANCE_ID.

In MSC_INST_PARTITIONS, FREE_FLAG = 1 means that the instance_id is not yet used. Once the INSTANCE_ID is used, the FREE_FLAG = 2. You should not manually change the FREE_FLAG using SQL. Create a new instance partition using the Create APS Partition request with parameters Instance partition = 1 and Plan partition = 0. We do not recommend creating extra instance partitions, since this creates many objects in the database that are not required. If the previous collected data and plan data from the old instance is not required, then it should be removed from the system.

Key Points:

- You will have to manually manipulate the EBS source instance table MRP_AP_APPS_INSTANCES_ALL.
- You cannot have two instances which have the combination of: Instance_id, Instance_code, and a2m_dblink in the EBS source table MRP_AP_APPS_INSTANCES_ALL
- You cannot have two instances which have the same Instance_id, Instance_code, m2a_dblink in the VCP destination table MSC_APPS_INSTANCES.

Cloning Your Existing Instance

In most cases, the clone is performed before any data is purged to avoid downtime. Complete the following steps:

1. On both the EBS source instance and VCP destination instances, purge all ASCP plans in the system, drop the plan and instance partitions. See Instance Cleanup and Reset, page 5-12 for more information.
2. On the VCP destination instance, finish the cleanup of the VCP destination instance using SQL #6 -- Delete All Tables Where the INSTANCE_ID Column Appears, page D-5 and SQL 7 -- Delete All Tables Where the SR_INSTANCE_ID Column Appears, page D-5. For instances with Oracle Demand Planning (ODP) data, also see SQL #13 -- Delete All Tables Where the INSTANCE_ID Column Appears for ODP Data, page D-8, SQL #14 -- Delete All Tables Where INSTANCE_ID Column Appears for ODP Data, page D-8, and SQL #14b -- Delete All Tables Where the Instance Column Appears for ODP Data, page D-9.
3. Add an ODS partition by submitting the request Create APS Partitions with plan_partition_count = 0 and inst_partition_count = 1. See ODS Tables and Instance Partitions, page 5-3 for more information.

4. Set up the new instance to be collected from within the application. See [Setting Up the EBS Source Instance](#), page 6-2 for more details.
5. Launch Gather Schema Statistics on the VCP destination instance. The Gather Schema Statistics program generates statistics that quantify the data distribution and storage characteristics of tables, columns, indexes, and partitions.
6. Set profile MSC: Source Setup Required = Yes on the EBS source instance.
7. Launch data collections with a complete refresh to collect data to the new APS destination.
8. Create and launch your ASCP plans. If the plan fails, then re-run it once more. If failure continues and it was working fine with the same code on the previous instance, then run Gather Schema Statistics again with 50% and then try the plan once more.

Tip: Once you have a very clear understanding of setting up partitions and the ways to manipulate instance codes and instance ids, you may want to refer to Section XV of My Oracle Support Note 137293.1 - "How to Manage APS Partitions in the MSC Schema" which provides details of an untested advanced cloning approach.

Moving From a Centralized to a Decentralized Installation

There are two options for this installation.

- Perform a fresh installation of the application on the VCP destination machine and set up VCP. The advantage is that it takes much less disk space and there's less chance of corrupting the instance. The disadvantage is that additional setup time is needed.
- Clone your existing instance onto a new server and then clean up the data on the EBS source and VCP destination instances in the MSC schema. The advantage is that much of the setup is accomplished through the cloning process. The disadvantage is all the disk space needed for all the unused data residing in other schemas. See [Cloning an Instance](#), page 6-11 for more information about cloning.

To perform a fresh installation of the application on the VCP destination machine, do the following:

1. On the EBS source instance, purge all ASCP plans in the system, drop the plan and instance partitions. See [Using Drop Partition Request](#), page 5-10 for more information.
2. Install applications in the new instance and set up your employees, users, etc. for

basic applications access.

3. Set up the VCP destination instance as if it was a brand new installation. See Defining Destination Link, page 6-6 for more information.

Resolving a Corrupted Instance

If the backup of the instance does not seem to have stable data, then all partitions should be dropped and then recreated.

1. Clean up all the data using SQL #6 -- Delete All Tables Where the INSTANCE_ID Column Appears, page D-5 and SQL #7 -- Delete All Tables Where the SR_INSTANCE_ID Column Appears, page D-5.
2. Check the tables using SQL #1 -- List Partitioned Tables in the System, page D-1 and keep a copy of that information.
3. Use Drop Partition with parameter Plan = No to remove the instance partition.
4. Submit Create APS Partition request with instance partitions = 1 and plan partitions = 0.

Note: Do not create more partitions than necessary.

5. Use SQL #10 -- Check MRP_AP_APPS_INSTANCES Table on the EBS Source Instance and Delete Line, page D-6 to remove the line from MRP_AP_APPS_INSTANCES_ALL in the EBS source instance.
6. Set up a destination instance. See Creating a VCP Destination Instance, page 6-6 for more information.
7. Run a complete refresh of the data collection. See Running Standard Collections, page 8-11.

Configuring Profile Options

This chapter covers the following topics:

- Overview
- EBS Source, VCP Destination, and ATP Profile Options
- Flex Field Profile Options
- Legacy Profile Options
- Other Profile Options

Overview

Once EBS and VCP have been installed on the source and destination servers, you must configure the software to support the collections process. This is done by configuring the following profile options:

- EBS Source, VCP Destination, and ATP Profile Options
- Flex Field Profile Options
- Legacy Profile Options (only if applicable)
- Other Profile Options

The EBS Source and VCP Destination profile options, in conjunction with the Flex Field profile options, are the basic profile options that need to be set for VCP collections. The EBS source and VCP destination profile options are used to customize the ASCP and EBS planning applications. These profile options can be modified at any time. The MSC: Source Setup Required profile option is critical for establishing the EBS source instance database objects used for Refresh Collections Snapshots and data collection. The flex field profile options are only used for the VCP planning applications and are run after the EBS source instance link is defined.

If you are using ATP with Oracle Order Management alone, you configure the set of

profile options specified. If you are planning to use GOP with ASCP, you will need to configure additional profile options. Please refer to Other Profile Options, page 7-24 for information.

If you are integrating with legacy systems or other VCP applications, you will need to configure additional profile options.

When all profile options are set, enter any data that may be required for VCP.

EBS Source, VCP Destination, and ATP Profile Options

The following table indicates which profile options need to be installed on the EBS Source and VCP Destination. If you are planning to use ATP, the table also indicates which profile options need to be set to make use of ATP functionality.

Note: ATP can be run in conjunction with Oracle Order Management alone, or with ASCP and GOP. The profile options specified in the table are the basic profile options required for ATP. Additional profile options may need to be set for ATP to function with ASCP or GOP.

Profile Options	Set in VCP	Set in EBS	Used for ATP	Description/Value
Dest ination	Sour ce	Sour ce	Data	
BIS: Primary Rate Type	Y	Y		Lookup from GL tables. If populated, then can be used for price information of supplies and demands from GL_DAILY_CONVERSION_TYPES. Profile DB Name: BIS_PRIMARY_RATE_TYPE
BOM: Hour UOM	Y	Y	Y	Used for collecting Resource Requirements. HR is the usual setting. Profile DB Name: BOM:HOUR_UOM_CODE
CRP: Spread discrete requirements	Y	Y		Used in ASCP for resource horizontal plan. Not used for collections. Valid values: Yes/No. Profile DB Name: CRP_SPREAD_LOAD

Profile Options	Set in VCP	Set in EBS	Use d for ATP	Description/Value
Dest ination	Sour ce	Data		
GMD: Yield Type		Y		<p>Used for OPM - UOM Class for common conversions from SY_UOMS_TYP. Values come from a lookup for OPM conversion. Typically MASS or something similar.</p> <p>Profile DB Name: FM_YIELD_TYPE</p>
INV: Capable to promise	Y	Y	Y	<p>Determines the source of ATP/CTP information.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • ATP Based on Collected Data: If ASCP is not installed or if the ATP will not be using ASCP planning data. Default. • ATP/CTP Based on Planning Data: If ASCP is installed and the ATP will be using ASCP planning data. <p>Profile DB Name: INV_CP</p>

Profile Options	Set in VCP	Set in EBS	Use ATP	Description/Value
Dest ination	Sour ce	Data		
MRP: Compute Standard Mandatory Components for ATO Models	Y	Y		Determines whether to call the CTO API to calculate the dependent demand for ATO items during the Refresh Snapshots process. This will cause performance to be slower when Yes. Setting this to Yes would be normal if ATO models are present in the system. Collections code will default this profile to Yes if it is Null. Valid values: Yes/No.
				Profile DB Name: MRP_EXPLODE_ATO
				Note: When a new CTO/ATO item is created from a model, a * (asterisk) is inserted into the new item number. The CTO BMO not found for the item error prompt is a setup violation since the requirement for CTO is to configure and create the * (asterisk) item using CTO Auto-Create or by progressing the sales order.
				Warning: ITEM_ID = 781553, ORG_ID=5002; No BOM found for configured item in ship-from org: You may safely disregard this warning if the configured item is transferred across orgs, and the warning points to an org in which no assembly of the configured item takes place.
				If you are using ODS-based ATP, check that you have the BOM in shipping org. If you are using PDS-based ATP, check that item attribute "Create Configured Item, BOM" equals "Based on Model" and BOM exists in ship-from org.
MRP: Consume MPS	Y	Y		Specifies if the Planning Manager must be running in EBS Source instance. Valid Values: Yes (default) /No.
				Profile DB Name: MRP_MPS_CONSUMPTION

Profile Options	Set in VCP	Set in EBS	Use ATP	Description/Value
Dest	Sour	Data		
inatiate	ce	on		
MRP: Cutoff Data Offset Months	Y	Y		<p>Determines how many months of resource availability should be computed for resources and simulation sets. No resource availability is calculated beyond this horizon. For constrained plans beyond this horizon, resource capacity is viewed as infinite. Valid values: Integer/Null. Null is seen as 99999. This should be set between 12-24 in most cases.</p> <p>System performance can be degraded when there is a high value for this profile option and a high volume of source transaction data.</p>
				<p>Profile DB Name: MRP_CUTOFF_HISTORY_DAYS</p>
MRP: Debug Mode	Y	Y		<p>Specifies whether more complete messaging is required for Refresh Collections Snapshots requests. Valid values: Yes/No (default).</p>
				<p>Profile DB Name: MRP_DEBUG</p> <p>Note: Do not set at site level, only at user level. For Collections, information written into MSRFWOR Refresh Collections Snapshots log file. For ASCP Planning: this profile makes very large log files and could slow down performance.</p>
MRP: Plan Revenue Discount Percent	Y	Y		<p>Optional. Specifies the average discount to be applied when calculating revenue of a plan as part of the plan performance indicators. This value is a key performance indicator used to calculate performance in a plan for Business Intelligence reports. Valid values: Integer 0-100.</p>
				<p>Profile DB Name: MRP_BIS_AV_DISCOUNT</p>

Profile Options	Set in VCP	Set in EBS	Used for Dest Sour Data	Description/Value
MRP: Plan Revenue Price List		Y		<p>Optional. The Price List is used to calculate revenue of a plan as part of plan performance indicators from SO_PRICE_LISTS. Key performance indicator used to calculate revenue of a plan for Business Intelligence reports. Valid values: Lookup for Price List in Order Management.</p> <p>Profile DB Name: MRP_BIS_PRICE_LIST</p>
MRP: Purge Batch Size	Y	Y	Y	<p>This setting is used in Data Collections:</p> <ul style="list-style-type: none"> When MSC: Purge Stg Tbl Cntrl = No. For non-partitioned tables repopulated with data during ODS Load for a Complete Refresh. <p>Valid values: Integer; default 75000. Generally, there is no reason to change this profile from the default setting.</p> <p>Profile DB Name: MRP_PURGE_BATCH_SIZE</p> <p>Note: In some circumstances, a delete process is taking too much time due to millions of rows of data. In the case of millions of records, Oracle suggests a setting of 250000. A higher value is not recommended since this profile option value is also used as a limit for BULK FETCH in ODS Load and a higher value could cause errors if the memory limit is reached.</p>
MRP: Purchasing By Revision	Y	Y	Y	<p>Determines if ASCP Planning is released from the workbench. Valid values: Yes/No.</p> <p>Profile DB Name: MRP_PURCHASING_BY_REVISION</p>

Profile Options	Set in VCP	Set in EBS	Use d for ATP	Description/Value
Dest ination	Sour ce	Data		
MRP: Retain Dates Within Calendar Boundary	Y	Y	Y	<p>Some applications do not look at the calendar assigned to the org and allow you to enter a date that is outside the calendar date. For example: Latest Acceptable Date in the Order Management application. This setting avoids failures related to dates that are outside the calendars defined in the system. Valid values: Yes/No (default). It is recommended that you set this profile to Yes.</p> <p>Profile DB Name: MRP_RETAIN_DATES_WTIN_CAL_BOUNDARY</p>
MRP: Use Ship Arrived Flag	Y	Y	Y	<p>Used when collecting Intransit Supplies. Valid values:</p> <ul style="list-style-type: none"> • Yes: the wsh_new_deliveries. ultimate_dropoff_date is checked to populate MRP_AP_INTRANSPORT_SUPPLIES_V. ARRIVED_FLAG. If Arrived, then the supply is not collected. • No or Null: Intransit supply collected. <p>Profile DB Name: MRP_SHIP_ARRIVE_FLAG</p>
MSC: Calendar Reference for Bucketing	Y			Calendar used for bucketing KPIs.
MSC: Category for Jobs without an Item Reference	Y			<p>Displays category for jobs without an item reference in the ASCP planner workbench. Valid values are NULL (unspecified items) or specific category for jobs without an item reference.</p> <p>Profile DB Name: MSC_NO_ACTIVITY_ITEM_CATEGORY</p>

Profile Options	Set in VCP Dest ination	Set in EBS Sour ce	Use d for ATP Data	Description/Value
MSC: Category Set for Jobs without an Item Reference		Y		<p>Displays category set for jobs without an item reference in the ASCP planner workbench. Valid values are NULL (NEW.MISC) or specific category set.</p> <p>Profile DB Name: MSC_NO_ACTIVITY_ITEM_CATEGORY_SET</p>
MSC: Coll Error Debug Mode		Y		<p>Collects error messages for debugging purposes. Valid values: Off (default), Medium, High, 384-do not use, 128-do not use, 0-do not use. Leave this as Off if you have a performance issue. Unless instructed by Support or Development, do not use the other options.</p> <p>Profile DB Name: MSC_COLL_ERR_DEBUG</p>
MSC: Coll Performance Debug Mode		Y		<p>Collects performance information for debugging purposes. Valid values: Off (default), Medium, High, 384-do not use, 128-do not use, 0-do not use. Leave this as Off if you have a performance issue. Unless instructed by Support or Development, do not use the other options.</p> <p>Profile DB Name: MSC_COL_PERF_DEBUG</p>
MSC: Collect CMRO Work Order Demands for PS		Y		<p>Supports Production Scheduling application and Complex Maintenance, Repair and Overhaul). Valid values: Yes/No (default).</p> <p>Profile DB Name: MSC_CMRO_WO_DEMAND_PS</p>

Profile Options	Set in VCP Dest ination	Set in EBS Sour ce	Use d for ATP Data	Description/Value
MSC: Collect Completed Jobs	Y	Y	Y	<p>Specifies whether completed jobs are collected. Valid values:</p> <ul style="list-style-type: none"> • Yes: Collect jobs with the status "complete". If some component demands have not been fully issued to the job, then the component demands are collected and will be seen in the ASCP plan and for ATP. • No: Do not collect requirements for completed jobs. This option was created to stop the collection of requirements left when the job is completed and not yet set to Complete - No Charges Allowed or Closed. <p>Profile DB Name: MSC_COLLECT_COMPLETED_JOBS</p>
MSC: Collect Item, Material and Resource Costs	Y	Y	Y	<p>Specifies whether item, material and resource costs are collected. Valid values are:</p> <ul style="list-style-type: none"> • Yes: Collects item, material and resource costs. Will join with CST_ITEM_COSTS table or for OPM items, will launch OPM package to get costs from OPM tables. Item cost is seen in MSC_SYSTEM_ITEMS table. If using Inventory Optimization or Constrained Optimized plan, Yes is required. Null is seen as Yes. • No: Uses a different view and collects costs as 0. <p>Tip: Using No can be faster for collections to complete, so if not required, set to No.</p> <p>Profile DB Name: MSC_COLLECT_COSTS_FOR_ITEM</p>

Profile Options	Set in VCP Dest ination	Set in EBS Sour ce	Use d for ATP Data	Description/Value
MSC: Collect Routing	Y	Y		<p>Specifies whether routings are collected. Valid values:</p> <ul style="list-style-type: none"> • Yes: Collects routings. If profile INV: Capable to Promise = ATP/CTP based on Planning output, then will always be seen as Yes and collect data. Null seen as Yes. • No: Doesn't collect routings. It is useful to avoid collecting routing information when using ATP based on Collected Data or if you have large routing tables. <p>Tip: Using No can be faster for collections to complete, so if not required, set to No.</p>
MSC: Collection Window for Trading Partner Changes (Days)	Y			<p>Profile DB Name: MSC_COLLECT_ROUTING</p> <p>Specifies the number of days backwards to check for changes in customer names, customer sites, vendors and vendor sites to be collected. Use when suppliers, customers, orgs are set to yes for collections when using ATP and when running ASCP plans. Valid values: Integer/Null (default). Null is the equivalent of 0, which will collect all records. Null must be used for the first run. After the first run, you should set up for a smaller number. Generally, 7 days should be sufficient. As long as you run a complete refresh at least once per 7 days, then this will collect the changes.</p> <p>Profile DB Name: MSC_COLLECTION_WINDOW_FOR_TP_CHANGES</p> <p>Caution: Setting to Null or 0 after the first run of data collection can cause performance issues.</p>

Profile Options	Set in VCP	Set in EBS	Use ATP	Description/Value
Dest ination	Sour ce	Data		
MSC: Collections Snapshot Threshold	Y	Y		Checks the numbers of rows in table and number of rows in the MLOG. If ratio exceeds the threshold, then a Complete Refresh of the Snapshot is launched. Valid values: Integers 0-100/Null (default).
				Tip: It is not recommended to set this profile unless directed by Development. It is faster to perform a complete refresh when a large number of rows exist in the MLOG, than reference the MLOG table for a fast refresh. This profile is usually triggered when a large data load is performed to load data into setup tables.
				Profile DB Name: MSC_SNAPSHOT_THRESHOLD
MSC: Configuration	Y	Y		<p>Specifies whether the Collaboration ODS Load for Collaborative Planning is launched at the end of Data Collections for ASCP. This process does not affect ASCP Planning. You can launch the ASCP plan while this process is still running. If using CP, then the process does need to complete before you can launch any other CP processes. Valid values:</p> <ul style="list-style-type: none"> • APS: Will not launch the Collaboration ODS Load for Collaborative Planning. • CP: Launches the Collaboration ODS Load for Collaborative Planning. Only used for standalone implementation of CP. • APS & CP: Launches Collaboration ODS Load for Collaborative Planning. <p>Important: This option is required for Demantra.</p>
				Profile DB Name: MSC_X_CONFIGURATION

Profile Options	Set in VCP Dest inati on	Set in EBS Sour ce	Use d for ATP Data	Description/Value
MSC: Cost Type		Y		<p>Specifies the cost type that collections should use when collecting the item costs from CST COST TYPES. Valid Values:</p> <ul style="list-style-type: none"> • Yes: Lookup of cost type in the Costing application. • No: Doesn't lookup cost type for collections. <p>Profile DB Name: MSC_COST_TYPE</p>
MSC: Currency Conversion Type		Y		<p>Set this profile option on the source instance to specify currency conversion type. If you change the value, the next time you run collections, the collections process purges the data with the previous currency conversion type and replaces it with the new currency conversion type. Valid values: A currency conversion type; Corporate is the default.</p>
MSC: Days of Resource Availability before sysdate		Y		<p>Calculates resource availability for the number of days specified before sysdate. Valid values: Integer/Null (equals 1).</p> <p>Tip: If a high number, then resource availability calculation during ODS Load could take a longer time.</p> <p>Profile DB Name: MSC_RES_AVAIL_BEFORE_SYSDAT</p>

Profile Options	Set in VCP Dest ination	Set in EBS Sour ce	Use d for ATP Data	Description/Value
MSC: Degree of Parallelism for Index Creation	Y	Y	Y	<p>Controls the use of parallel processes when creating indices on the temporary tables created during the ODS Load. Valid values: Integer 0 (default) to 10. Required setting of 1 if fails with NULL or 0. The value can be increased if there are performance problems with the ODS Load and system resources can be allotted.</p> <p>Profile DB Name: MSC_INDEX_PARALLEL_THREADS</p>
MSC: Enable ATP Summary Mode	Y	Y	Y	<p>Enables summarization of supply and demand data following the data collection and plan launch processes. These tables are only for ATP based on collected data. Valid values:</p> <ul style="list-style-type: none"> • Yes: Enable ATP Summary Mode. <p>Note: If this profile set to Yes and INV: Capable to Promise = ATP/CTP based on Planning Data, then the request Load ATP Summary Tables will run at the end of Data Collections and will error. This can be ignored or you can set the profile to No in order to avoid the error.</p> <ul style="list-style-type: none"> • No: Disable ATP Summary Mode. <p>Profile DB Name: MSC_ENABLE_ATP_SUMMARY</p>

Profile Options	Set in VCP	Set in EBS	Use d for Dest Sour Data	Description/Value
MSC: Future Days for Currency Conversion	Y			<p>Number of days in the future to be considered for collecting currency conversions. Collects currency conversion data for display in APCC. Valid values: Integer. Set this to a reasonable number of future days for displaying currency conversion data. A very high number or NULL with long calendar end date 10 years out could cause performance issues. If you are not using APCC, do not set this profile.</p> <p>Profile DB Name: MSC_FUTURE_DAYS_FOR_CURR_CON</p>
MSC: History Days for Currency Conversion	Y			<p>Number of days in the past to be considered for collecting currency conversions and displayed in APCC. Valid values: Integer. A high number could cause longer collection times when using Currency Conversion = Yes. In most cases, 365 days should be sufficient.</p> <p>Profile DB Name: MSC_HISTORY_DAYS_FOR_CURR_CON</p>
MSC: Inflate WIP Resource Requirements	Y	Y		<p>Determines whether ASCP should inflate the Work in Process resource requirements based on resource efficiency and utilization during collections from Oracle Work in Process. Used for ASCP Planning. See "Appendix A" in the ASCP User's Guide for more information. Valid values: Yes/No.</p> <p>Profile DB Name: MSC_INFLATE_WIP</p>

Profile Options	Set in VCP	Set in EBS	Use d for ATP	Description/Value
Dest ination	Sour ce	Data		
MSC: Operator Company Name	Y	Y	Y	<p>Specifies the operator company name. Valid values:</p> <ul style="list-style-type: none"> • My Company is defaulted at application level for ASCP during the installation. MSCXUPG.sql is SQL that inserts MSC_COMPANIES. COMPANY_ID = 1, COMPANY_NAME = My Company during installation. • Null: Data Collections will fail if not specified. <p>Profile DB Name: MSC_X_COMPANY_NAME</p>
MSC: Organization containing generic BOM for forecast explosion	Y			<p>Used by Collections to identify and populate the Bill of Material validation organization. This is used for ASCP Planning to explode the BOM for forecast explosion and consumption. In a distributed installation, this profile is setup on the EBS Source. Then Data Collections will populate this organization_id in MSC_APPS_INSTANCES in column VALIDATION_ORG_ID. Valid values: Null (default)/Organization code of organization that will hold the BOM for ATO models (usually the master org).</p>
MSC: Planning Hub Currency Code	Y			<p>Use this profile option on the destination instance to specify reporting currency. If you change the value, the next time you run collections, the collections process purges the data with the previous currency code and replaces it with the new currency code. Valid values: Currency code; blank is the default.</p>

Profile Options	Set in VCP	Set in EBS	Use d for Dest Sour Data	Description/Value
MSC: Project Task Collection Window Days	Y			<p>Used to limit the collection of completed projects and tasks. Valid values: Integer. Defaults as NULL (seen as 0) which will collect all records. A value of 100 means that projects and tasks marked as completed for 100 days are collected. On day 101, those completed projects and tasks are no longer collected. This profile can be set to help performance if you are using projects and tasks and have many records.</p> <p>Profile DB Name: MSC_COLLECTION_WINDOW_FOR_PROJ_TASKS</p>
MSC: Purchase Order Dock Date Calculation Preference	Y	Y		<p>Specifies the purchase order line date that ATP and ASCP Plan use as the material arrival (dock) date. Valid values: Promised Date or Need By Date.</p> <p>Profile DB Name: MSC_PO_DOCK_DATE_CALC_PREF</p>

Profile Options	Set in VCP Dest inati on	Set in EBS Sour ce	Use d for ATP Data	Description/Value
MSC: Purge Staging and Entity Key Translation Tables	Y			<p>Specifies whether the staging tables will be truncated or deleted when Data Collections retrieve data from an instance (usually a single instance).</p>
(formerly called MSC: Purge Stg Tbl Cntrl pre-R12. 0.6)				<p>Valid values:</p> <p>Yes -- Truncates the staging table during the data collection process to improve performance and conserve disk space. Set to Yes if you are collecting from only one EBS source instance. For multiple instances, separate the processes.</p> <p>No -- Causes staging tables to be deleted instead of using the Truncate command. Since the tables are not truncated, their size can increase over time without space being recovered when deleted, ultimately leading to slower collections performance. In this case, it is recommended that the DBA periodically schedule a truncate of the MSC_ST% tables when Data Collections are not running to recover space and help performance.</p>
				Profile DB Name: MSC_PURGE_ST_CONTROL

Profile Options	Set in VCP Dest ination	Set in EBS Sour ce	Use d for ATP Data	Description/Value
MSC: Refresh Site to Region Mappings Event		Y		<p>This profile performs mapping of regions to supplier sites based on setups in the Order Management Regions and Zones form. Valid values:</p> <ul style="list-style-type: none"> • Collection of Sourcing Rules: When Parameter Sourcing Rules is set to Yes, then run Region to Site mapping (recommended). This just collect supplier information, which is not performance-intensive. • Running of Refresh Collection Snapshots Request: Only run this integration when you launch Refresh Collection Snapshots as a standalone request. <p>Note: This profile is only used for vendor site to region mapping, which is used for Transit Times setup in ASCP with Region and Zones. After initial collection, only changes to current regions are collected. But when new regions or zones are defined, all the vendor sites are remapped, which can be time-consuming and affect performance.</p>
MSC: Refresh Snapshots Pending Timeout		Y		<p>Profile DB Name: MSC_REFRESH_REGION_SITE</p> <p>Specifies timeout when Refresh Collections Snapshots is launched. Valid values: Integer. 10 is the default (minutes). No changes are recommended unless there is a problem with launching Refresh Snapshots.</p> <p>Profile DB Name: MSC_REF_SNAP_PENDING_TIMEOUT</p>

Profile Options	Set in VCP	Set in EBS	Use d for ATP	Description/Value
Dest ination	Sour ce	Data		
MSC: Sales Orders Offset Days	Y	Y		<p>Specifies the number of days of closed sales orders to be collected after the SYSDATE. Uses Shipped Sales Order line information during ASCP Planning to indicate the forecast for the demand schedule of a plan. Designed to collect only closed sales orders where open_flag = no or order_quantity = shipped_quantity. Valid values: Integer/NULL. The smaller the number, the faster Data Collections can complete Sales Order loading. The number required depends on forecasts consumption Forward Days setting. For example, if Maximum Forward Days setting for any Collected Forecast or ODP forecast is 30 days, then the setting should be at least 30 days. NULL is (not recommended as it will collect all closed sales orders after the SYSDATE -99999 (or 10,000 days). Used in the MRP_AP2_SALES_ORDERS_V collections view.</p> <p>Note: Calculated using calendar days. Must be the same on both the Source and Destination.</p>
MSC: Share Plan Partitions	Y			<p>Profile DB Name: MSC_SO_OFFSET_DAYS</p> <p>Specifies whether to share the plan partitions.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • Yes: Causes performance issues in any implementation where any volume testing is required. Can only be used in small test instances. • No: Required for all implementations.

Profile Options	Set in VCP Dest ination	Set in EBS Sour ce	Use d for ATP Data	Description/Value
MSC: Source Setup Required		Y		<p>Automatically sets the application when a Collections Patch is applied. Valid values:</p> <ul style="list-style-type: none"> • Yes: Setup requests are run during Data Collections. The profile is read by the request Refresh Collection Snapshots. These requests create all the EBS source instance database objects that are needed for data collections to complete successfully. • No: Does not run setup request during Data Collections. <p>Note: This profile is not considered when you run Refresh Collection Snapshots as a standalone request and use Automatic Refresh.</p>
MSC: Sourcing History Start Date Offset (in months)		Y		<p>Profile DB Name: MSC_SOURCE_SETUP</p> <p>Offsets the starting date of sourcing history calculations and is used with collections parameters Purge Sourcing History = Yes and Recalculate Sourcing History = Yes. These settings delete, then calculate Sourcing History from the Start date determined by the profile option value (in months). Used to calculate planned orders when the sourcing rules are split for ranks. Valid values: Integer/NULL (default). System performance can be degraded when left as NULL or set to a high value with a high volume of source transaction data. Set 0 if you have no sourcing splits defined in the sourcing rules. For sourcing rules with splits, choose a setting of between 3 and 12 depending on your business requirements.</p> <p>Profile DB Name: MSC_START_SOURCING_HISTORY</p>

Profile Options	Set in VCP	Set in EBS	Use ATP	Description/Value
Dest ination	Sour	Data		
MSC: Use Shipping/Receiving Calendar	Y	Y		<p>Specifies whether the Shipping/Receiving Calendar is collected for use with ASCP and Global Order Promising (GOP). Valid values:</p> <ul style="list-style-type: none"> • Yes: Collect Calendars assigned via Order management/Shipping/Setup/Calendars/Assignment. See the ASCP User's Guide and the GOP User's Guide for more information. • No: Default. <p>Profile DB Name: MSC_USE_SHIP_REC_CAL</p>

Flex Field Profile Options

The table below shows the descriptive flexfield attributes that are required, the name of the attributes that are created, the tables in which the data resides, and the profile options that correspond to each attribute.

Note: These values are not required at the item level. If not set at the item level, they will be defaulted at the plan level by the constrained planning engine. We recommend that you use the plan level constraints until you have found a specific need to modify at a lower level.

Descriptive Flexfield Names <i>(In flexfield form and report parameter)</i>	Parameter Name	Base Table Name	Profile Option Name
Items	Late Demands Penalty (Item)	MTL_SYSTEM_ITEM S	MSO: Penalty Cost Factor for Late Item Demands Flexfield Attribute

Descriptive Flexfield Names	Parameter Name (In flexfield form and report parameter)	Base Table Name	Profile Option Name
Items	Material Over-Capacity Penalty (Item)	MTL_SYSTEM_ITEM S	MSO: Penalty Cost Factor for Exceeding Item Material Capacity Flexfield Attribute
Organization Parameters	Late Demands Penalty (Org)	MTL_PARAMETERS	MSO: Penalty Cost Factor for Late Demands (Organization) flexfield attribute
Organization Parameters	Material Over-Capacity Penalty (Org)	MTL_PARAMETERS	MSO: Penalty cost factor for exceeding material capacity (Organization) flexfield
Organization Parameters	Resource Over-Capacity Penalty (Org)	MTL_PARAMETERS	MSO: Penalty cost factor for exceeding resource capacity (Organization) flexfield
Organization Parameters	Transport Over-Capacity Penalty (Org)	MTL_PARAMETERS	MSO: Penalty cost factor for exceeding transportation capacity (Organization)
Department Resource Information	Aggregate Resources	BOM_DEPARTMENT_T_RESOURCES	MSC: Aggregate Resource Name Flexfield Attribute
Department Resource Information	Resource Over-Capacity Penalty (Resource)	BOM_DEPARTMENT_T_RESOURCES	MSO: Penalty cost factor for exceeding resource capacity (Resource) flexfield

Descriptive Flexfield Names	Parameter Name (In flexfield form and report parameter)	Base Table Name	Profile Option Name
Attribute (Purchasing)	Material Over-Capacity Penalty (Supplier)	PO_ASL_ATTRIBUT_ES	MSO: Penalty Cost Factor for Exceeding Material Capacity Flexfield Attribute
Substitute Component Information	Substitute Items Priority	BOM_SUBSTITUTE_COMPONENT	MSC: Priority for Substitute Items Flexfield Attribute
MTL Interorg ship methods	Transport Over-Capacity Penalty (Ship Method)	MTL_INTERORG_SHIP_METHOD	MSO: Penalty cost factor for exceeding transportation capacity flexfield attribute
Bill of Material Information	BOM/Routing Cost	BOM_BILL_OF_MATERIALS	MSC: Cost of Using a BOM/ Routing Flexfield Attribute
MRP Forecast Dates	Late Forecasts	MRP_FORECAST_DATES	MSO: Penalty cost factor for late forecasts
Additional Line Information	Late Sales Order Penalty	SO_LINES	MSO: Penalty cost factor for late sales orders
Production Line	Resource Group (Line)	WIP_LINES	MSC: Resource Group for a Line Flexfield Attribute

Legacy Profile Options

If you are planning to upload legacy data to EBS for collection purposes, the following profile options need to be set:

Profile Options	Set in APS Dest ination	Set in EBS Sour ce	Used for ATP Data	Description/Value
MSC: Batch Size for Self Service Loads		Y		<p>Batch size for self-service loads. Valid values: Integer; default is 1000. If you have many rows to import, set to a higher value.</p> <p>Profile DB Name: MSC_SELF_SERVICE_LOADS_BATCH_SIZE</p>
MSC: Number of Workers for Self Service Loads		Y		<p>Number of workers involved in self-service loads. Valid values: Integer; default is 3. If you have many rows to import, set higher to match increased batch size profile option.</p> <p>Profile DB Name: MSC_SELF_SERVICE_LOADS_WORKERS_NUM</p>
MSC: Self Service Loads Delimiter		Y		<p>Delimiter used as the default for the DAT files provided in OATemplate.exe. The templates can be downloaded from the Collect Flat File Data -- Self Service, and then unzipped. Review the resulting file OATemplateReadme.html. Valid values: ~ (tilde; default) or , (comma).</p> <p>Profile DB Name: MSC_SELF_SERVICE_LOADS_DELIMITER</p>

Other Profile Options

Each implementation of VCP is unique, and may involve a range of VCP products. After you have set your profile options for EBS and VCP including the flex fields, you need to configure the other VCP products you want to use. This may involve configuring additional profile options.

The following table provides the source of the profile option documentation for the following VCP applications:

Application	Documentation Source
Oracle Advanced Planning Command Center (APCC)	"Setting Up Profile Options", <i>Oracle Advanced Planning Command Center User's Guide</i> .
Oracle Advanced Supply Chain Planning (ASCP)	"Appendix A: Profile Options", <i>Oracle Advanced Supply Chain Planning Implementation and User's Guide</i> .
Oracle Collaborative Planning Implementation and User's Guide (CP)	<i>Oracle Collaborative Planning Implementation and User's Guide</i> .
Oracle Demantra Demand Management	"General Integration Guidelines", "Demantra Data Tables and Integration Processes", and "Demantra Demand Management to EBS Integration", <i>Oracle Demantra Integration Guide</i> .
	If you are also integrating EBS Service Parts Planning with Demantra, see "Demantra Demand Management to EBS Service Parts Planning Integration", <i>Oracle Demantra Integration Guide</i> .
Oracle Demantra Sales and Operations Planning	Same as Demantra Demand Management plus "Demantra Sales and Operations Planning to EBS Integration", <i>Oracle Demantra Integration Guide</i>
Oracle Global Order Promising (GOP)	<i>Oracle Global Order Promising Implementation and User's Guide</i>
Oracle Inventory Optimization (IO)	<i>Oracle Inventory Optimization User's Guide</i>
Oracle JD Edwards EnterpriseOne	"Configuring Oracle Demantra", <i>Oracle Value Chain Planning Integration Base Pack 3.1 - Implementation Guide</i>
Oracle Production Scheduling (PS)	"Integrating Production Scheduling", <i>Oracle Advanced Supply Chain Planning Implementation and User's Guide</i> .
Oracle Rapid Planning (RP)	<i>Oracle Rapid Planning Implementation and User's Guide</i>

Application	Documentation Source
Oracle Service Parts Planning (SPP)	<i>Oracle Service Parts Planning Implementation and User's Guide</i>
Oracle Strategic Network Optimization (SNO)	"Integrating Strategic Network Optimization", <i>Oracle Advanced Supply Chain Planning Implementation and User's Guide.</i>

Running Collections

This chapter covers the following topics:

- Overview
- Running Collections for the First Time in a New Instance
- Standard Collections
- Continuous Collections
- ATP Data Collections
- Organization-Specific Collections
- Category-Specific Collections
- Collecting from Single Source to Multiple Destinations
- Running Collections after Cleaning Up ODS Data

Overview

The data collections process by which data from EBS is captured and replicated on the VCP server for planning purposes. Profiles and parameters are used to customize the data collected.

There are three types of collections:

- Standard Collections: Manual process that collects data specified by the parameters.
- Continuous Collections: Automated process that synchronizes certain snapshot-enabled data entities between the source(s) and planning server.
- ATP Collections: Collects only data that supports the ATP feature in Order Management

Collecting data can take a significant amount of time compared to the time for the overall planning cycle. Oracle Advanced Supply Chain Planning (ASCP) provides a

collection method that allows the collections duration to be reduced in cases where information about some - but not all - planning-related business entities on the planning server needs to be updated.

In addition to the three types of collections, there are three collection methods:

- The Complete Refresh method clears all previously collected planning data for all business entities from the planning server (for the source instance being collected), then copies over information about the user-selected entities. This method can be time consuming because it copies all of the planning information from the EBS application tables.
- The Targeted Refresh method clears transaction data for only the user-selected business entities from the planning server, and then copies the entity information over from the transaction instance. Information about unselected entities remains intact on the planning server. All planning business entities are supported by Targeted Refresh collections. When running targeted refresh for entity Trading partners, also collect entity Calendars. This ensures that the Oracle Project Manufacturing organizations are set up as organization type process so that the release process from Oracle Advanced Supply Chain Planning to Oracle Project Manufacturing succeeds.
- The Net Change Refresh method copies only incremental changes to business entities to the planning server (and is thus faster), but is supported mainly for demand and supply business entities only. Process manufacturing (OPM) is not supported using Net Change Refresh. Instead, use a targeted refresh for OPM. See Data Changes That Can Be Collected in Net Change Mode, page 8-2 for more information.

Data Changes That Can Be Collected in Net Change Mode

When the net change mode for collections is selected (by setting the collections parameter Complete Refresh to No), the data changes shown in the following table can be collected. If you set the collections parameter Complete Refresh to yes, the collections program collects the entire data for the entity.

All other data changes must be collected by running full collections (by setting the collections parameter Complete Refresh to Yes). Net change collections run more quickly than full collections.

You can run data collections in net change mode for these transactions:

Data Element	Comments
Sales orders	Cancellations of or modifications to sales orders and sales order reservations are captured. Net change collections always collects sales orders; it is not dependent on any parameters or settings, including Pull collection parameter Sales Orders.
Reservations against demands	Reservations against both external and internal sales order demands are captured. Set the Planning Data Pull parameter Reservations to Yes.
Master production schedule demands	MPS demands that are added, modified or relieved in the source instance are captured. Set the Planning Data Pull parameter Master Production Schedules to Yes to collect just these changes during a Net Change collection.
Master demand schedule	Set the Planning Data Pull parameter Master Demand Schedules to Yes.
WIP component demands	Demand changes due to cancellation of WIP jobs, changes in the state of WIP jobs (for example, operations within a job have been performed or cancelled), and changes to WIP jobs because of changes in item information are captured. Set the Planning Data Pull parameter Work in Process to Yes.
WIP repetitive item demands	Demand changes due to cancellation of WIP repetitive schedules, changes in the state of WIP repetitive schedules, and changes to WIP repetitive schedules because of changes in item information are captured. Set the Planning Data Pull parameter Work in Process to Yes.
Forecast demands	Changes and deletions in forecasts are captured. Set the Planning Data Pull parameter Forecasts to Yes.

Data Element	Comments
Master production schedule supplies	Changes in supply schedules or item information are captured. Set the Planning Data Pull parameter Master Production Schedules to Yes.
User supplies and demands	Changes to user supplies and demands because of changes to item information are captured. Set the Planning Data Pull parameter User Supplies and Demands to Yes.
Purchase order supplies	Changes to PO supplies because of rejections, returns, or cancellations or changes to item information are captured. Set the Planning Data Pull parameter Purchase Orders/Purchase Requisitions to Yes.
On-hand supplies	Set the Planning Data Pull parameter On Hand to Yes.
Work orders in Oracle Work in Process	Changes in WIP Jobs are captured. Set the Planning Data Pull parameter Work in Process to Yes.
Resource availability	<p>You can use net change mode in discrete manufacturing organizations. Set the Planning Data Pull parameter Resource Availability to Collect and Regenerate Data.</p> <p>If you are a process manufacturing organization, use complete refresh.</p>
Supplier capacity	Set the Planning Data Pull parameter Approved Supplier Lists to Yes.
Bills of material	All BOM changes are captured: new components, disabled components, component quantities, effectivity dates, BOM revisions, and component substitutes. Set the Planning Data Pull parameter Bills of Materials / Routings / Resources to Yes.

Data Element	Comments
Routing operations	Changes to and deletions of routing operations as a result of changes to operation sequences (for example, the addition of new operations, the disabling of operations, or the changing of operation dates), the disabling of a routing, the changing of routing dates, or changes to item information (for example, the disabling of an item, the creation of a new item) are captured. Set the Planning Data Pull parameter Bills of Materials / Routings / Resources to Yes.
Components needed for a routing operation	Changes to and deletions of components needed for a routing operation are captured. Set the Planning Data Pull parameter Bills of Materials / Routings / Resources to Yes.
Resources attached to a routing operation	Changes to and deletions of operation resources or operation resource sequences within a routing are captured. Set the Planning Data Pull parameter Bills of Materials / Routings / Resources to Yes.
Resource requirements for WIP jobs	Changes in resource requirements of WIP jobs because of completion of the WIP jobs, completion of operations within the WIP jobs, or changes in item information are captured. Set the Planning Data Pull parameter Work in Process to Yes.
Items or Item categories	Changes in items and items categories are captured. Set the Planning Data Pull parameter Items to Yes.

Transactions (supply and demand) change more frequently than setup entities. After data collections, the Advanced Supply Chain Planning program maintains logs of entity transactions. Each time you run data collections, the collections program applies the logged transactions to a snapshot. Net Change collections selects the transactions applied to the snapshot since the last Planning Data Collection and pulls only those new transactions to the staging tables. As setup entities change less frequently, the collections process does not keep snapshots for these and cannot perform net change collections on them. Schedule either a targeted or a complete refresh for setup entities.

You cannot run data collections in net change mode for the following setup entities:

- Simulation sets
- Resource shift setup
- Projects or project tasks
- Units of measure (class conversion, conversions)
- Sourcing information
- Bills of resources
- Calendar information (start dates calendar dates, calendar week, calendar shifts, shift Dates, shift exceptions, shift times, period start dates)
- Interorganization ship methods
- Planning parameters
- Planners
- Business intelligence systems periods
- Resource groups
- Demand classes
- Available to promise (ATP) rules
- Trading partners (customer or customer sites, suppliers, supplier sites, organization, organization sites, location associations, customer, vendor, buyer, contacts)

Handling of Currency During Collections

Currency: Currencies collected from Oracle Order Management and Oracle Purchasing are posted in functional currency even if their source is in transactional currency. Functional currency is the currency defined in the Instances form during the setup of the instance in the Advanced Planning Administrator/Admin Instances form.

Currency Conversion Rates: Collect currency and currency conversion rates for a past, current and future period from the source in order to support displaying the cost in functional currency, which is defined when the instance is set up. The default value is No. Currency conversion rates are collected only when the profile option MSC: Planning Hub Currency Code is Yes for standard collections.

For more information, see profile options:

- MSC: Currency Conversion Type, page 7-12
- MSC: Planning Hub Currency Code, page 7-15
- MSC: History Days for Currency Conversion, page 7-14
- MSC: Future Days for Currency Conversion, page 7-14

Details About What is Collected

Discrete Jobs: A discrete job that is complete and not closed appears in the collected data with quantity zero. A discrete job that is complete and closed does not appear in the collected data. Therefore, if you have a job that is closed but still has resource and/or material requirements, they can be collected and planned. You can control this with profile option MSC: Collect Completed Jobs

Drop ship purchase orders: The collections process does not collect drop ship purchase orders because the planning engine does not plan drop ship sales orders. (A drop ship purchase order is one where your business directs the supplier to deliver the goods to a third party and not to your business.)

End-Item Substitutes: You can see end-item substitutes in Collections Workbench as long as you have not defined a substitute set. If you have defined a substitute set, you must first run the ASCP plan and then use the Planner Workbench Items window to see the substitute set and end-item substitute. For more information about end-item substitutes and substitution sets, see "End-Item-Level Substitution" in the *Advanced Supply Chain Planning Implementation and User Guide*.

Global Forecasts: You can review global forecast entries in Collections Workbench horizontal plan using rows Original and Cumulative Original.

To set up the horizontal plan to show rows:

1. Open the Workbench.
2. From the Tools menu, choose Preferences.
3. In the Material Plan tab, Supplies/Demands region, change lookup to Demands.
4. Select the Global Forecast.
5. Save.

Inactive forecasts: The collections process does not collect the demands of inactive forecasts if run in Full Refresh or Targeted Refresh collections mode

Intransit, Purchase Order, and Requisition: Oracle recommends that you always collect purchase order, requisition and intransit together. To do so, set Planning Data Pull parameter Purchase Orders/Purchase Requisitions to Yes.

Reservations: This table shows the reservations that collect. The rows are supply types and the columns are demand types. An X in the cell indicates a reservation that collects.

Routings: The collections process does not collect routing operation resources with Schedule = No and does not display them in their routings. It also does not collect alternate resources of primary resources that are Schedule = No, even if the alternate resources themselves are Schedule = Yes.

Supply Type	Sales Order Demand	Internal Sales Order Demand	Process Batch Demand	Maintenance Work Order Demand
On Hand	X	X	X	X
Discrete Job	X	X	--	--
Lot based Job	X	X	X	--
Process Batch	X	X	X	--
Flow Schedule	X	X	X	--
Maintenance Work Order	--	--	--	X
External Purchase Requisition	X	X	X	--
Internal Purchase Requisition	X	X	X	X
Purchase Order	X	X	X	--
ASN	X	X	X	--
Intransit Shipment	X	X	X	--
PO in Receiving	X	X	X	--

Running Collections for the First Time in a New Instance

Expect that data collections will take an additional amount of time for the first run. Collecting what can be millions of rows from the OLTP tables into the MSC tables -- which are empty -- is a very I/O intensive process for the database.

It is not unusual for the first run to require that you use the timeout parameter of 600 to 1600 minutes to get all data collected from the EBS source tables to the VCP destination MSC tables. It does not matter if you are using database links and two instances or a single instance. Be sure to set the timeout parameter to a high value for both the Planning Data Pull and ODS Load.

Do not use a Collection Group during the first run; collect using parameter = All. If you are not collecting from all Orgs defined in the source instance, then only define the required orgs in the Instances form. Be sure to include the Master org in the Instances form.

Once the data collections have completed successfully the first time, run Gather Schema Stats for the MSC schema. The timing should settle down and be reasonable. The default timeout of 180 for Planning Data Pull and 60 minutes for the ODS Load can be maintained.

To run the initial collection:

Important: Before performing the initial Planning Data Collection, ensure that fresh statistics have been gathered on the EBS Source instance for INV, BOM, WIP, PO, ONT, MRP, MSC, APPS, APPLSYS, and FND schemas. Inaccurate statistics on these schemas can cause the RDBMS Cost-Based Optimizer to make poor performing execution plan choices, which may take days to complete.

1. On your EBS Source instance, check that profile MSC: Source Setup Required = Yes.
2. Review the parameters and setup requests that create all the objects used for Data Collections. See Appendix B: Parameters, page A-1 and Appendix C: Concurrent Programs, page B-1.
3. Navigate to Collections/Oracle Systems/Standard Collections.
4. Run data collections using Targeted Refresh for Planners only. This is a one-time setup step for the very first run of Collections. Key parameters are:
 - Purge Previously Collected Data: No.
 - Collection Method: Targeted.
 - Set only Planners = Yes and all others No (or equivalent).

5. If you have a lot of data to collect, then the Refresh Snapshots process could take several hours to run for this initial run.
6. Planning Data Pull Parameters special considerations:
 - Set Timeout parameter very high -- 900 to 1600 suggested for the first run.
 - Set Sales Order parameter explicitly to Yes if using Order Management application.

Note: When running a Complete Refresh, this parameter default to No and runs net change collection of Sales Order for performance reasons. This is normal and recommended during normal operations.

 - All other parameters should be set to default for the first run of collections.
 - If there are other parameters that will need to be Yes for your implementation, then plan to set those to Yes after you get the initial run of collections completed successfully.
7. ODS Load Parameters, special considerations:
 - Set Timeout parameter very high -- Oracle suggests 900 to 1600 for the first run.
 - Set all other Parameters to No for this first run.
 - If you need to set these to Yes, then do this later after the first run has completed successfully and timing for data collections is stable.
8. When the first run is completed, run Gather Schema Statistics for the MSC schema on the VCP Planning Destination instance.
9. If a setup request(s) fails, then run Planning Data Collections - Purge Staging Tables with parameter Validation = No. Then launch again. Many times, initial failures in the setup requests are resolved by running two or three times.
10. If Planning Data Pull or ODS Load fails, check for a Timeout error or other specific error in the log file of the main request and all the log files for the workers also. You must run Planning Data Collections - Purge Staging Tables with parameter Validation = No before you launch the data collections again. If you fail to run this request, then data collections will error with one of the following messages:
 - "Either another Planning ODS Load process is running for this instance or an earlier ODS Load may have been terminated abnormally. If it is the latter case, you need to launch the 'Planning Data Collection - Purge Staging Table'

program before launching the Planning Data Collections again."

- "Either another planning data pull process is running for this instance, or an earlier planning data pull process may have terminated abnormally. In the latter case, you need to launch the 'Planning Data Collection - Purge Staging Table'".
- "Another Data Pull process is running."

Standard Collections

Using the standard collections process, you can manually run three types of data collection methods including a complete refresh, a net change refresh, or a targeted refresh on specific business entities.

To collect data from an Oracle EBS source instance

1. Sign on using the Advanced Supply Chain Planner responsibility or the Advanced Planning Administrator responsibility.
2. Navigate to the Planning Data Collection window by selecting Collections > Oracle Systems > Standard Collection.

The Planning Data Collection window appears.

This window shows you that the collections process consists of two sequentially executed concurrent programs. The first program, Planning Data Pull, copies information from the source instance into the MSC staging tables on the planning server. Here, against the data held in the staging tables, the Planning ODS Load performs some basic cleansing. For example, it ensures that items with the same names that are collected from different source instances will be assigned the same internal IDs (recognized as the same item). Further cleansing operations on the staging tables (if desired) may be done at this point via any custom concurrent program. This custom concurrent program would need to be inserted into the Planning Data Collection request set, in between Planning Data Pull and Planning ODS Load. The second program, Planning ODS Load, copies information from the VCP staging tables into the operation data store on the planning server, where it becomes available for use during planning.

3. Select the Parameters field for the Planning Data Pull program.

The Planning Data Pull Parameters window appears.

4. Specify your Planning Data Pull Paramters in this window. For more information about the specific fields and options, see Planning Data Pull Parameters, page A-2

Note: If you are running Standard Collections in Complete Refresh mode, then setting No will remove all data for the entity. Use Targeted Collection if you require to collect only certain data entities and are not going to use default values populated for Complete Refresh.

5. Select OK.
6. Select the Parameters field for the Planning ODS Load program.
The Parameters window appears.
7. Specify your Planning ODS Load Paramters in this window. For more information about the specific fields and options, see Planning ODS Load Parameters, page A-10.
8. Select OK.
9. Select Submit in the Planning Data Collection window to run collections immediately, or select Schedule to schedule collections for some later time.

If you select Schedule, the Schedule window appears.

Note: If you want to perform an incremental refresh frequently, use this feature.

You have complete control over the timing and frequency of the collection of data from the transaction systems, and the timing and frequency of planning. You can manage the balance between network traffic and the need to monitor current status in your plans.

To select a frequency for running the job:

- Select a frequency for running the job in the left pane. Complete any additional fields that appear based on your selection.
- Click OK.
- Choose Submit in the Planning Data Collection window.

10. From the toolbar, choose View > Requests to view the status of the collection process.
The Find Requests window appears.
11. Select a type of requests to view then select Find.

The Requests Window displays data collection progress.

12. After the collection process completes, view your results.

If concurrent process Refresh Collections Snapshot ends in a warning and you have configure-to-order items, there may be a setup issue with those items. Concurrent process Refresh Snapshot calls the configure-to-order applications programming interface to explode the configured bills of material and create demand in the proper manufacturing organizations. Check the log file of the configure-to-order applications programming interface for details of the setup issues and take corrective action.

13. From the Navigator window, choose Collection > Workbench.

Notice that data is brought over from selected instances.

Note: Users can collect forecasts into the planning server. If you want the collections program to collect a forecast set, select the Advanced Planning Collections checkbox while defining the forecast set.

Continuous Collections

Continuous collection is an automated process that synchronizes snapshot-enabled data entities (supply and demand) and snapshot-disabled setup entities (suppliers, customers and supplier rules) between the sources and the planning server. You can schedule separate collection programs for collecting data entities and setup entities.

The Continuous Collections concurrent program performs the process of continuous collections. You have to select only those business entities for which the collections process needs to run automatically. The Continuous Collections concurrent program determines the appropriate mode of performing collections for the selected business entities. You can run continuous collections on the following entities:

- For entities that have snapshots associated with the source, you need to specify a threshold value (as a percent). Based on this value, the Continuous Collections concurrent program determines whether the collections should run in the Targeted mode or the Net Change refresh mode. If continuous collections are run frequently, then for most entities, the data collections are performed in the Net Change Refresh mode.
- If the changed records percent is below the threshold percent, the concurrent process collects only the changed records (Net Change Refresh) from the snapshot log.
- If the changed records percent is higher than the threshold percent, the concurrent process collects all of the rows (Targeted Change Refresh) from the snapshot.

- If there are no changed records, the concurrent process does not collect any data.

The following table details whether or not snapshots are associated for the entities supported by continuous collections:

Entities	Snapshot Associated
Approved supplier lists (Supplier capacity)	Yes
Bills of material	Yes
Routings	Yes
Resources	Yes
Bills of resources	Yes
Forecasts	Yes
Items	Yes
Master demand schedule	Yes
Master production schedule	Yes
On hand quantity	Yes
Purchase orders	Yes
Purchase requisitions	Yes
Sales orders	Yes
User supplies and demands	Yes
Work in process	Yes
Available to promise rules	No
Calendars	No
Demand classes	No

Entities	Snapshot Associated
End item substitution	No
Key performance indicator targets	No
Planning parameters	No
Planners	No
Projects and tasks	No
Reservations	No
Resource availability	No
Safety stock	No
Sourcing history	No
Sourcing rules	No
Subinventories	No
Trading partners (customers and suppliers)	No
Unit numbers	No
Units of measure	No
User company association	No

For entities without snapshots, the concurrent program always initiates targeted refresh.

You can plan to use continuous collections when extensive transactions are involved. For example, a manufacturing company with extensive work in process transactions might setup continuous collections to run every 20 minutes to collect on hand balance. Similarly, Oracle Collaborative Planning users might schedule continuous collections every 2 minutes if they want to view the current supplies status.

To run continuous collections

To collect data from an Oracle EBS source instance:

1. Sign on using the Advanced Supply Chain Planner responsibility or the Advanced Planning Administrator responsibility.

2. From the Navigator, select Collections > Oracle Systems > Continuous Collection.

The Continuous Collections window appears.

This window enables you to schedule the process of data collection, set parameters that are required for running Continuous collections, select language preferences, and specify the notification tasks that need to be triggered on completion of Continuous collections.

3. Click in the Parameters field to set values that the concurrent program would require to perform Continuous collections.

The Parameters window appears.

Specify Yes for the entities that you want the Continuous Collections concurrent program to consider for collection. Most of the fields in this window are similar to the parameter fields for the Standard collections process. The parameter that distinguishes the Continuous collections process from the Standard collections process is Snapshot Threshold (%). By default, the threshold value is set to 40%. You can change this value.

4. Select OK.

5. Select Schedule in the Continuous Collections window to schedule collections.

The Schedule window appears.

6. Select the frequency for running collections in the left pane. Complete any additional fields that appear based on your selection.

7. Click OK.

8. Select Submit in the Continuous Collections window.

9. From the toolbar, choose View > Requests to view the status of the collections process.

The Find Requests window appears.

10. Specify the type of request you want to view.

11. Select Find.

The Requests window displays the status of the data collection process.

After the collection process completes, view the result in the Collection Workbench.

ATP Data Collections

Important: Used by customers using only simple ATP for Order Management in a centralized instance, and not using any other Value Chain Planning applications such as Global Order Promising (GOP).

The ATP Data Collection is a simplified version of the Planning Data Collection with only the program parameters required for ATP based on collected data. Running ATP data collections in complete refresh mode will corrupt the data in the MSC tables, requiring a complete refresh of the standard data collection to recover.

For instructions for running ATP Data Collections, see "Functional Setup for ATP Based on Collected Data" in the *Oracle Global Order Promising Implementation and User's Guide*

Organization-Specific Collections

Oracle Value Chain Planning supports organization-specific collections, which helps you in maintaining independent planning processes in independent business units of a company that is running a single instance of the Oracle e-Business Suite.

You can run collections for collecting data from only specified organizations of a source instance. This helps you in elimination of unnecessary data collection from organizations that are not planned in your planning process and reduce the processing time.

Oracle Value Chain Planning allows organization-specific collection for:

- Standard collections:
 - Complete refresh method
 - Net change refresh method
 - Targeted refresh method
- Continuous collections

Note: Organization-specific collection is not available for legacy collections.

To set up organization specific collections

1. Select Advanced Planning Administrator responsibility.
2. Navigate to Admin > Instances.

The Application Instances form appears.

3. Define the instance.

4. Click Organizations.

The Organizations form appears.

5. Select the Enable check box for the organizations of the defined instance.

6. Specify Collection Group for the organizations.

Assign all related organizations to the same collection group to ensure that group organizations are collected together.

See [Creating a VCP Destination Instance, page 6-6](#) for more information on organization-specific collections.

To perform organization specific collections

1. Select the Advanced Supply Chain Planner responsibility.
2. Navigate to Collections > Oracle Systems > Standard Collections.
3. The Planning Data Collection form appears.
4. Click the Parameters field for the Planning Data Pull program.
5. The Parameters window appears.
6. Select the Collection Group.

Regardless of the Collection Group value, Oracle Value Chain Planning always collects the following:

- ATP Rules (Global)
- Demand Classes (Global)
- Trading Partners (Customers)(Global)
- Trading Partners (Suppliers)(Global)
- Units of Measure
- User Company Association (Global)
- Sourcing Rules and Assignment Sets
- Shipping Networks (Org-Specific) and Shipping Methods (Global)

In addition, the collection programs collect all calendars to accommodate all trading partner shipping and receiving calendars.

Category-Specific Collections

You can filter collected data by category sets for each instance in both Oracle Standard and Continuous Collections.

Set Up

Navigate to form Application Instances.

Navigate to checkbox Collect All Category Sets. The default is selected.

If you want to collect all category sets for collection, select Collect All Category Sets.

If you want to specify only some category sets for collection:

- Clear Collect All Category Sets
- Click Category Sets
- In window Category Sets, select the category sets that you want the collections process to collect.

Processing Collections Type Complete Refresh

The process:

- Removes all collected data
- Collects the category sets that you specify in window Category Sets

For example:

- Instance A contains collected data for category sets CS-1, CS-2, CS-3, and CS-4.
- You specify category filtering for category sets CS-4 and CS-5.
- You run Complete Refresh.
- Instance A contains collected data for category sets CS-4 and CS-5. It removed all of the earlier data from category sets CS-1, CS-2, CS-3, and CS-4.

Processing Collections Type Targeted Refresh

Oracle recommends that if you change the list of category sets and run Targeted Refresh, run it for all entities.

The process:

- Removes all collected data except for the data of the category sets in window Category Sets

- Collects the category sets that you specify in window Category Sets

For example:

- Instance A contains collected data for category sets CS-1, CS-2, CS-3, and CS-4 and only items and purchase orders.
- You specify category filtering for category sets CS-4 and CS-5 and only items.
- You run Targeted Refresh

Old Collection		New Collection	
Category Set	Entities	Category Set	Entities
CS-1	Items, Purchase Orders	CS-4	Items
CS-2		CS-5	
CS-3			
CS-4			

Instance A contains:

- Purchase order data for category sets CS-1, CS-2, and CS-3, but not item data
- Both item and purchase order data for category set CS-4
- Item data for category set CS-5, but not purchase order data

Collected Data – Old Collection		
Category Set	Item	Purchase Order
CS-1	Item1	PO1
CS-2	Item2	PO2
CS-3	Item3	PO3
CS-4	Item4	PO4

Collected Data – New Collection		
Category Set	Item	Purchase Order
CS-1		PO1
CS-2		PO2
CS-3		PO3
CS-4	Item4	PO4
CS-5	Item5	

Therefore, item data is missing for category sets CS-1, CS-2, and CS-3 and purchase order data is missing for category set CS-5. The collections data has unwanted data in the old category sets and missing data in the new category sets.

Processing Collections Type Net Change Refresh

Oracle recommends that if you change the list of category sets, do not run this collections type.

The process:

- Keeps all collected data
- Collects the category sets that you specify in window Category Sets

For example:

- Instance A contains all previously collected data for category sets CS-1, CS-2, CS-3,

and CS-4.

- You specify category filtering for category sets CS-4 and CS-5.
- You run Net Change Refresh.

Instance A contains:

- All previously collected data for category sets CS-1, CS-2, and CS-3, and CS-4.
- Net change data for category sets CS-4 and CS-5.

The collections data has unwanted data in the old category sets and missing data in the new category sets. In addition, you cannot fix the issue with this type of collection by collecting all entities.

Collecting from Single Source to Multiple Destinations

Oracle Value Chain Planning allows you to collect a single source instance to multiple destination instances. This is useful when you need to perform volume testing, where you need to collect from a source production instance to a test destination instance while still generating production plans out of a production planning server instance. Both of the planning server instances can share the same production source transaction instance.

Collection from a single source to multiple destinations allows you to leverage the setup and transaction data in the EBS source instance to do volume testing on the test planning server instance. You can avoid duplicating the source instance and reduce substantial amount of storage and maintenance costs.

Note: The source and destination (planning server) instances do not need to be on the same release.

For example, you can have a test R12.1 destination instance and R11*i* other destination instance. Both can collect from the same R11*i*EBS Source instance.

The critical factor in such a design is the patching management. If you have to apply patch to one instance, you may have to apply the same patch to all three instances. For example, you may patch the R12.1 destination instance, which will require a patch on your R11*i* e-Business Suite Source instance. Then, you may need to apply the same patch to your R11*i* other destination instance.

Note: A source instance can be enabled for ATP for only one destination instance. The ATP engine needs to know which destination instance to connect with when performing ATP from the source

instance.

To setup collection from a single source to multiple destinations

Perform these steps in each In each of your multiple destination instances.

1. Select Advanced Planning Administrator responsibility.
2. Navigate to Admin > Instances.

The Application Instances form appears.

3. Associate destination instances to source instance in each destination instance separately.

4. Select the Allow ATP check box for the destination instance that you want to consider in the ATP process from the source instance.

You can select the Allow ATP check box for only one of the destinations that are associated to a single source.

The planning engine displays an error message if you attempt to select more than one destination for ATP.

5. Select the Allow Release ATP check box for the destination instances that are allowed to release to the source instance.

The Allow Release Flag check box is used for both auto release and manual release processes.

You can select the Allow Release check box for multiple destinations that are associated to a single source.

To trigger the Auto Release process for a plan:

1. Select the Allow Release check box for the instance where the plan is defined in the Application Instances form.
2. Select the Production check box for the plan in the Plan Names form.

Example: Using the Allow ATP option

Consider:

- A single source instance S that is associated to two destination instances D1 and D2.
- The Allow ATP check box is selected for destination instance D1 related to source S.

The ATP request from source S points to D1. The planning engine performs the ATP process in D1 taking into consideration the plans that have ATP check box selected in

the Plan Names form.

In destination instance D2, there could be plans with the ATP check boxes selected in the Plan Names form. The ATP process from D2 uses these plans. There are no restrictions in checking the ATP check boxes in the Plan Names form in destinations.

Conclusion: You can select the Allow ATP check box in the Application Instances form for only one of the destinations associated to a single source

Example: Using the Allow Release option

Consider:

- A single source instance S that is associated to two destination instances D1 and D2.
- The Allow Release check box is selected for D2 and not for D1.

The plans in D2 with the Production check boxes selected in the Plan Options form trigger the Auto Release process to source.

You can select the Production check boxes for plans in D1 but they will not trigger any Auto Release process to source. This is because the Allow Release check box is not checked for D1 in the Application Instances form.

Conclusion: You can select the Allow Release check box for multiple destinations that are associated to a single source.

Running Collections after Cleaning Up ODS Data

After cloning a VCP destination instance, you will need to clean up your ODS data before running a standard collection.

If you have cleaned up your ODS data including dropping an old ODS partition, creating a new partition, and creating a new source instance, follow these steps when running your first post-cleanup collection:

Prerequisite Steps:

1. Drop the old ODS partition. See Using Drop Partition or Drop Plan Partition, page 5-10 for more details.
2. Add an ODS partition by submitting the request Create APS Partitions with plan_partition_count = 0 and inst_partition_count = 1. See ODS Tables and Instance Partitions, page 5-3 for more details.
3. Set up the instance to be collected from within the application. See Creating a VCP Destination Instance, page 6-6 for more details.

Running First Post-Cleanup Collection Steps:

1. Launch Gather Schema statistics on the VCP destination instance.
2. Set profile MSC: Source Setup Required = Y on the EBS source instance.

3. Make sure statistics have been gathered on the EBS source instance.
4. (Optional) Refresh the collections snapshots as follows:

Note: By running this concurrent program first, it ensures that the collections will run faster because the snapshots have already been refreshed. However, when you run a complete collections process, the refresh collections snapshots concurrent program is launched automatically.

1. From the Advanced Supply Chain Planner, Standard responsibility, navigate to Setup, and then Run Requests.
The Submit Request form appears.
2. In the Name field, enter Refresh Collection Snapshots.
3. In the Parameters field, set as following:
 - Refresh Mode: Complete
 - Snapshot Name: All Snapshots
 - Threshold for Truncating Logs: 0 (default).
 - Degree: 0 (default). Snapshot log is truncated either if it has more rows than threshold parameter or if the snapshot is empty.
5. Launch a complete refresh of the data collection. For more information about running a data collection, see Running Standard Collections, page 8-11.
6. Set profile MSC: Calendar Reference for Bucketing to the correct value for the new calendars collected with the new instance definition or null.
Important: If you do not follow this step, you may not be able to define plans.
7. After collecting Trading Partners (Organizations in particular), set up the Organization Security for the Responsibilities. Use the Advanced Planning Administrator, Standard and navigate to Admin, then Organization Security.
Important: If you do not set the Organization Security, errors will be caused in the Plan Options/Organization tab as there will not be a list of values available. Also, viewing plans after they have run will not be possible.

8. Run Gather Schema Statistics for the MSC schema with the default parameters.
9. Run the ASCP plans. If the plan fails, then rerun it once more. If failure continues and it was working fine on the same code on the previous instance, try step 3 again and run the plan again.

Collecting from Legacy Systems

This chapter covers the following topics:

- Overview
- Setting Up the Collection of Transaction Data into the Planning Server
- Process
- Setting Up Batch Uploads
- Sequence of Data Uploads
- Preprocessing Engine
- Setting Up Legacy Instance
- Running Legacy Collections Using the Form-Based Application (Collect Flat-File Data)
- Running Legacy Collections Using the Self-Service Application (Collect Flat-File Data - Self Service)
- Purging Legacy Data
- SRP Streams - Self Service
- Shipment History - Self Service

Overview

Legacy Collection provides an open framework for consulting and system integrators to bring data from legacy systems into Oracle VCP. You can upload data by batch upload of flat files. This is achieved in part by extending the interface table capabilities. A preprocessing engine validates the incoming data from legacy application and ensures that referential integrity is maintained. All business objects can be imported into VCP using flat files.

In addition to collecting data from your EBS instance to your planning instance, you can collect data to the Planning instance from:

- Your non-Oracle (legacy) systems.
- Your trading partners' non-Oracle systems

To collect data from your non-Oracle ERP systems or your trading partners' systems, you can either model each non-Oracle ERP system or trading partner as:

- an Oracle Applications organization and store their setup and transaction data there.
- a 100% Legacy collection, in which case no standard EBS instance is defined or collected from.

Setup information includes organization setup, items, bills of material, resources, routings, and sourcing information. Transaction data is of the following types:

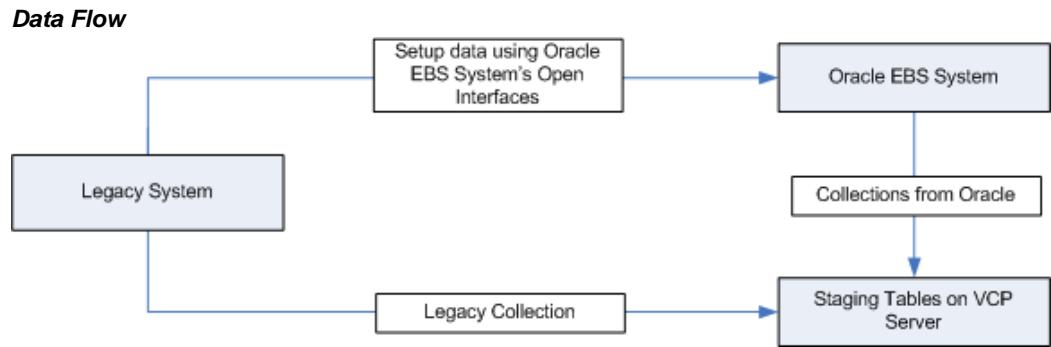
- On-hand balance
- Purchase orders
- Purchase requisition
- Work orders
- Work Order component demand
- Intransit shipment and receipt
- Planned orders
- Local forecasts
- Demand schedules
- Sales orders
- Item suppliers
- Supplier capacities
- Supplier flex fences

You can perform the following steps to collect data from your trading partners' non-Oracle systems to your planning instance:

- Load setup data (such as items, BOMs, Trading Partners) from the trading partner's system to flat files. Load the flat files to the source (Oracle EBS) instance using the standard interfaces and use standard collections to move it to the destination (planning) server.

- Load transaction data from flat files to the staging tables (MSC_ST%) on the planning server. Use legacy collections to move data from the legacy instance to the planning server.

The following diagram illustrates the flow of data from non-Oracle ERP (legacy) systems to an Oracle EBS application and the planning server.



Setting Up the Collection of Transaction Data into the Planning Server

To set up the collection of transaction data into the Planning Server

- Define two types of organizations in the source instance. The first organization is for OEM and the second is for supplier and customer. You also need to define sourcing rules between the OEM organization and the supplier and customer organization.
- Import the setup data from your non-Oracle ERP system into the supplier and customer organization using the Oracle EBS open interfaces such as Items open interface, BOM open interface.

To load the organizations for trading partners, you must first define the trading partners in previous loads .

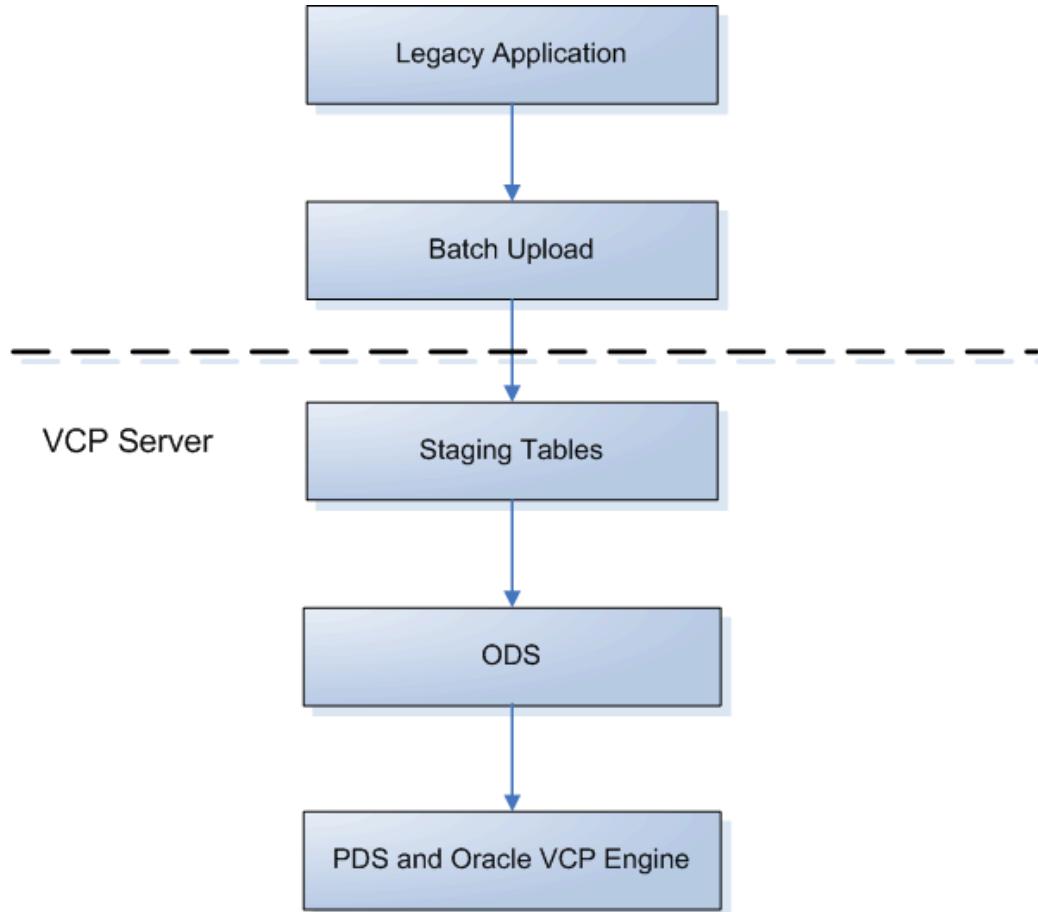
- Collect all data from the OEM and supplier and customer organizations into your destination instance (planning server).
- Load transaction data into each supplier and customer organization's flat files. Using either an Oracle Applications form or the self-service application, you can load data from the non-Oracle ERP system into the organizations. If you are using the self-service method, you can upload a zip file containing all the data.

Process

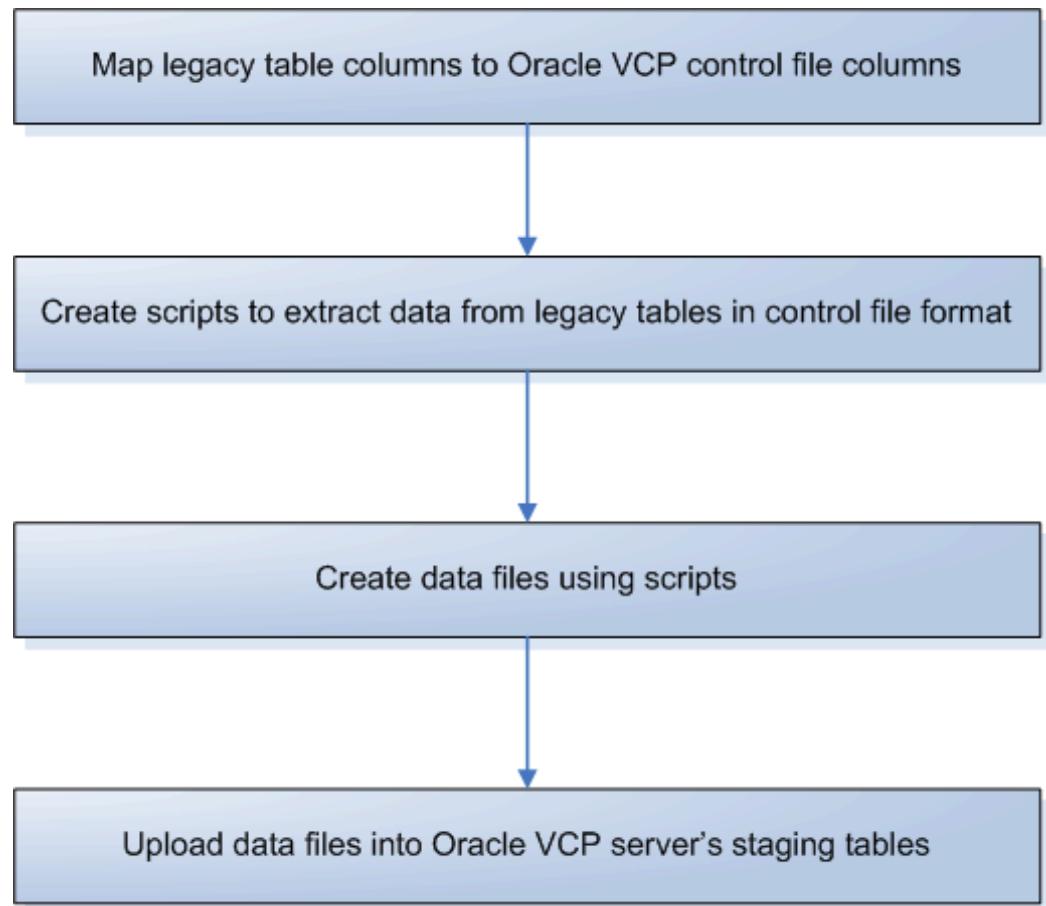
You push legacy data, such as items, bills of materials, routings, etc. into Oracle VCP staging tables using batch upload. Batch upload is done using Oracle SQL*Loader. SQL*Loader requires that data is brought over in a format described in a control file. Oracle has provided control files for all the staging tables. The list of control files is available in Oracle iSupport.

The following diagram shows the movement of data from legacy systems into the Oracle VCP server via staging tables using the batch upload process.

Legacy Application



Setting Up Batch Uploads



To set up batch uploads

You must do the following to set up the batch uploads:

1. Map the Oracle VCP staging tables' control files (a control file is a template that specifies the input data format) to the legacy system's tables. All the control files (*.ctl) are installed in \$MSC_TOP/patch/115/import. All files are named for the staging table to be populated. For example, the Items Base table is MSC_SYSTEM_ITEMS, so we populate MSC_ST_SYSTEM_ITEMS using MSC_ST_SYSTEM_ITEMS.ctl file.

Note: Control files like MSC*_F.ctl are for the Advanced Planning Command Center application and not relevant to the Advanced Supply Chain Planning application.

2. Create scripts to extract data from the legacy system in the format prescribed by the

control files.

When loading for trading partner sites, provide values for the location for organizations (Partner Type = 3); do not provide values for the location for customer and supplier sites (Partner Type = 1 or 2).

For example, the following is the control file for Purchase Order Supplies (MSC_ST_SUPPLIES_PO.ctl)

```
OPTIONS (BINDSIZE=1000000, ROWS=1000, SILENT=(FEEDBACK,DISCARDS))
LOAD DATA
INFILE 'MSC_ST_SUPPLIES_PO.DAT'
APPEND
INTO TABLE MSC.MSC_ST_SUPPLIES
FIELDS TERMINATED BY '-'
(
ITEM_NAME,
ORGANIZATION_CODE,
NEW_SCHEDULE_DATE,
SUPPLIER_NAME,
FIRM_PLANNED_TYPE "NVL(:FIRM_PLANNED_TYPE,1)",
SUPPLIER_SITE_CODE,
PURCH_LINE_NUM,
ORDER_NUMBER,
SR_INSTANCE_CODE,
REVISION "NVL(:REVISION,1)",
UNIT_NUMBER,
NEW_ORDER_QUANTITY,
NEW_DOCK_DATE,
PROJECT_NUMBER,
TASK_NUMBER,
PLANNING_GROUP,
DELIVERY_PRICE,
QTY_SCRAPPED,
FROM_ORGANIZATION_CODE,
ORDER_TYPE CONSTANT '1',
DELETED_FLAG "DECODE(:DELETED_FLAG,1,1,2,2,2)",
COMPANY_NAME "NVL(:COMPANY_NAME,-1)",
END_ORDER_NUMBER,
END_ORDER_RELEASE_NUMBER,
END_ORDER_LINE_NUMBER
ORDER_RELEASE_NUMBER,
COMMENTS,
SHIP_TO_PARTY_NAME,
SHIP_TO_SITE_CODE,
SR_INSTANCE_ID CONSTANT '0',
PROCESS_FLAG CONSTANT '1',
DATA_SOURCE_TYPE CONSTANT 'BATCH',
LAST_UPDATE_LOGIN CONSTANT '-1',
LAST_UPDATE_DATE SYSDATE,
CREATION_DATE SYSDATE
)
```

The script to extract Purchase Order data for this format from a legacy system hosted on an Oracle database could look like the following:

```

SET HEAD OFF';
SET LINESIZE 200;
SET PAGESIZE 50000;
SPOOL ON;
SPOOL MSC_ST_SUPPLIES_PO.dat;
SELECT
DISTINCT
ITEM_TAB.ITEM_NAME||'~'|||
ITEM_TAB.ORGANIZATION_CODE||'~'|||
PO_TAB.EXPECTED_DELIVERY_DATE||'~'|||
SITES_TAB.TP_NAME||'~'|||
1||'~'||/* All orders are treated as Firmed */
SITES_TAB.TP_SITE_CODE||'~'|||
PO_TAB.LINE_NUM||'~'|||
PO_TAB.PO_NUMBER||'~'|||
&&SR_INSTANCE_CODE||'~'|||
NVL(ITEM_TAB.ITEM_REVISION,1)||'~'|||
YES||'~'|||
PO_TAB.MRP_PRIMARY_QUANTITY||'~'|||
PO_TAB.EXPECTED_DOCK_DATE||'~'|||
PO_TAB.PROJECT_ID||'~'|||
PO_TAB.TASK_ID||'~'|||
YES||'~'|||
PO_TAB.UNIT_PRICE||'~'|||
0||'~'|||
1||'~'||/* All records are either for Insert/Change. No deletions are
being uploaded */
YES||'~'|||
-1||'~'|||
YES||'~'|||
YES||'~'|||
YES||'~'|||
YES||'~'|||
YES||'~'|||
YES||'~'|||
YES||'~'|||
YES||'~'|||
0||'~'|||
1||'~'|||
'BATCH'||'~'|||
-1||'~'|||
SYSDATE||'~'|||
SYSDATE
FROM <LEGACY_SUPPLY_TABLE> PO_TAB,
<LEGACY_ITEMS> ITEM_TAB,
<LEGACY_PARTNER_SITES> SITES_TAB
WHERE PO_TAB.ORGANIZATION_ID = ITEM_TAB.ORGANIZATION_ID
AND PO_TAB.ITEM_ID = ITEM_TAB.INVENTORY_ITEM_ID
AND PO_TAB.VENDOR_ID = SITES_TAB.SR_TP_ID
AND PO_TAB.VENDOR_SITE_ID = SITES_TAB.SR_TP_SITE_ID;

```

1. Run the scripts to get the data files and ftp these to the concurrent manager node. The steps to upload these files into Oracle VCP are described below under Running Legacy Collections, page 9-15.

Sequence of Data Uploads

Data must be uploaded in a specific sequence because fields from some dat files may use values that are defined in other dat files. For example, since there are calendar_code references in MSC_ST_TRADING_PARTNERS.ctl, calendar information needs to be loaded before trading partner information.

Load all this information either together or in the following order:

1. Upload calendar data information. All the calendar data files corresponding to calendar's control files (MSC_ST_CALENDARS.ctl, MSC_ST_WORKDAY_PATTERNS.ctl, MSC_ST_SHIFT_TIMES.ctl, MSC_ST_CALENDAR_EXCEPTIONS.ctl, MSC_ST_SHIFT_EXCEPTIONS.ctl) need to be uploaded in one single run. Based on the information provided, the calendar is built on the VCP server. If calendar already exists in ODS tables on the VCP server and you want to rebuild the calendar again, then the entire information (all the above mentioned files) must be sent again. Also, in this case for MSC_ST_CALENDARS.ctl the OVERWRITE_FLAG should be sent as Y.
2. Upload the UOM information. The control file for this is MSC_ST_UNITS_OF_MEASURE.ctl.
3. Upload the Demand Class information.
4. Upload the Trading Partner information. The control files for setting up trading partners are MSC_ST_TRADING_PARTNERS.ctl, MSC_ST_TRADING_PARTNER_SITES.ctl, MSC_ST_LOCATION_ASSOCIATIONS.ctl, MSC_ST_SUB_INVENTORIES.ctl and MSC_ST_PARTNER_CONTACTS.

The trading partner sites, location associations, sub inventories and contacts can be uploaded along with the trading partner information and also in subsequent runs. Only MSC_ST_TRADING_PARTNERS.ctl can be uploaded in the first run.

MSC_ST_TRADING_PARTNERS.ctl has CALENDAR_CODE field. This should refer to a valid calendar code existing on the planning server or to a calendar code that you are uploading in this run of collections. If calendar does not exist on the planning server and has not been uploaded either, then the trading partner record is not accepted and is marked as error.

5. Upload the category sets information. The control file for setting up category sets is MSC_ST_CATEGORY_SETS.ctl
6. Upload the designators information for forecast, MDS and MPS. The control files required are: MSC_ST_DESIGNATORS_MDS.ctl, MSC_ST_DESIGNATORS_FORECAST.ctl and MSC_ST_DESIGNATORS_PLAN_ORDERS.ctl. The forecast, MDS and MPS records can be uploaded now or in subsequent runs.
7. Upload the projects and tasks information. The control file name is MSC_ST_PROJECT_TASKS.ctl
8. Upload the items information as per the MSC_ST_SYSTEM_ITEMS.ctl file. If the UOM_CODE of the data file has an invalid value (that is, a value that does not exist in the planning system and is also not being uploaded along with items as per the

MSC_ST_UNITS_OF_MEASURE.ctl in this upload) the item records are errored out.

9. Upload the item related information; for example, supplier capacity, supplies and demands, categories, UOM conversions, and sourcing rules. Upload the data as per the preprocessing diagram shown below and make sure that the items are valid; that is, the items exist in the planning system or are being uploaded in this run of legacy collections.
10. Upload categories using control file MSC_ST_ITEM_CATEGORIES.ctl.
11. Upload sourcing rules using control file MSC_ST_ITEM_SOURCING.ctl.
12. Upload UOM conversions using MSC_ST_UOM_CONVERSIONS.ctl, MSC_ST_UOM_CLASS_CONVERSIONS.ctl.
13. Upload resources using control file MSC_ST_DEPARTMENT_RESOURCES.ctl.
14. Upload bill of materials using the following control files: MSC_ST_BOMS.ctl, MSC_ST_BOM_COMPONENTS.ctl, and MSC_ST_COMPONENT_SUBSTITUTES.ctl. You can upload BOM components and substitutes to BOM at the same time or upload these in later runs.
15. Upload routings using the following control files: MSC_ST_ROUTINGS.ctl, MSC_ST_ROUTING_OPERATIONS.ctl, and MSC_ST_OPERATION_RESOURCES.ctl. You can upload resources to operations at the same time or upload these in later runs.
16. Upload supplier capacity using the following control files: MSC_ST_ITEM_SUPPLIERS.ctl, MSC_ST_SUPPLIER_CAPACITIES.ctl, and MSC_ST_SUPPLIER_FLEX_FENCES.ctl. You can upload MSC_ST_SUPPLIER_CAPACITIES.ctl with MSC_ST_ITEM_SUPPLIERS.ctl or in subsequent runs. You can also upload MSC_ST_SUPPLIER_FLEX_FENCES.ctl with MSC_ST_ITEM_SUPPLIERS.ctl or in subsequent runs.
17. Load material supply for work order after routings are loaded because there is a field ROUTING_NAME in MSC_ST_SUPPLIES_WO.ctl.
18. Upload resource demand using the control file MSC_ST_RESOURCE_REQUIREMENTS.ctl. If WIP_ENTITY_NAME is not valid (it was not previously loaded using the MSC_ST_SUPPLIES_WO.ctl and also is not loaded in this run using this control file) the record is errored out.

Preprocessing Engine

After data from the legacy application has been loaded onto the planning server, it

undergoes preprocessing before it can be used by the planning engine.

Preprocessing generates IDs for the entities coming into the planning system based on a set of user-defined keys (UDKs). For example, to identify an item record in the planning system, the UDK is Instance Code, Organization code, Item Name and Company Name. (Company Name is required only if Collaborative Planning is being deployed, which requires that the profile MSC: Configuration is set to the Advanced Supply Chain Planner responsibility to either 'APS and CP' or 'CP'. For standalone VCP, the profile is set to 'APS' and this value defaults to -1.) A UDK uniquely identifies an existing record in the staging tables. UDKs are used as reference to update existing records in the staging tables.

The preprocessing program is a concurrent program that runs independently from the Planning engine and Global ATP engine.

After the data files have been brought over to the concurrent manager node, the legacy collection request sets the pre-processing engine to read and load the data files into the staging tables. It can then preprocess the data and load the data into the main planning tables (ODS), all in a single run.

The preprocessing engine has the intelligence to handle scenarios wherein transaction data and any prerequisite setup data needed to perform this transaction co-exist in a single data load.

The figure below shows the sequence in which the uploaded data is processed by the preprocessing engine. The preprocessing engine possesses parallel processing capabilities. Parallel processing is enabled for processing items and item-related entities as shown in the diagram. Items, supplies and demand records can further be broken into sub-batches and processed in parallel.

Preprocessing Architecture

Preprocessing Architecture



The above architecture also makes it necessary to ensure that all the setup related data is sent to the planning server to avoid errors while processing the transactions. For example, a purchase order line coming into the planning server referring to an item that has not been sent to the server is flagged as an error. Also, the supplier for the item should have been defined as well.

Records in the staging tables are checked for multiple occurrences of the same UDK combination. For instance, in the case of data coming in via XML, if two or more item records are found in the staging table having the same combination of instance code, organization code, item name and company name, preprocessing picks the latest record for further processing and the older records are flagged as errors. For instance, for data coming in via batch upload, if two or more item records are found in the staging table having same combination of instance code, organization code, item name and company

name, preprocessing flags those records as errors because preprocessing is not able to determine which is the correct record to be picked up. Error messages are written to the staging table in column ERROR_TEXT in each MSC_ST Staging table.

Data Cleansing

Preprocessing cleanses the incoming data to make it fit for the planning engine. Data that are invalid but are not critical for planning purposes are automatically replaced by default values. This is done to avoid the overhead of resending data. In such cases warning messages are generated. Based on these warning messages, you can resend corrected data. These corrected values then replace the default values.

Error Processing

Errors occur in preprocessing if any of the critical validations fail. In such cases the records are immediately flagged with an error status and do not undergo further processing. These errors are logged in the Oracle VCP table, MSC_ERRORS. These records can be resubmitted after correcting the errors reported by the preprocessing engine.

If XML Gateway is installed, error and warning messages encountered during preprocessing can be emailed to the System Administrator, whose email id can be specified in a XML Gateway profile, ECX: System Administrator Email Address.

Net Change

Preprocessing works on the net change principle. Users need to send only new and changed data to the planning server between successive plan runs. However, a purge program has been provided to carry out a complete refresh. After a complete refresh, all setup and transaction data have to be pushed again into the Oracle VCP server. The purge program also allows partial deletion of supplies and demands. Supply and demand records that fall earlier than a user-specified date can be deleted from the VCP server.

Data Transformation

Character fields have been provided in some cases even where the planning engine requires a numeric value for easy adaptation to legacy data. If numeric values are provided in these fields then the preprocessing engine will pass on the same value to the planning engine. Otherwise it will either generate a new numeric value corresponding to the character value or default a pre-determined constant. For example, the planning engine expects the calendar's shift identifier to be a numeric value. The legacy application may define its workday shifts as A, B and C. The preprocessing engine would then generate a new numeric value corresponding to these character values. It then passes these shift numbers to the planning engine for subsequent processing. Other examples of this kind of flexibility are operation sequence number

and resource sequence number. If these fields get a nonnumeric value, instead of not accepting the record, preprocessing replaces these with a numeric constant - 1.

Denormalized Tables

To simplify mapping of the staging tables to the legacy application's tables, wherever possible the staging tables have been denormalized. For instance, the sourcing rules can be uploaded using a single table. This information is then broken down by preprocessing and sent to four different tables within the VCP server. Denormalization has been provided for:

- sourcing rules
- projects and tasks
- resource availability and simulations sets
- forecast sets and forecasts
- routings

Workday Calendar Build

You have the option of building the workday calendar in the VCP server by providing just *summary* information: calendar start date, end date, shift information, workday pattern and exceptions. The preprocessor has an embedded calendar build program, which will explode this data to get the working and nonworking days and also the shifts available for each working day. You must specify the workday calendar code for each planning organization that you upload into the VCP server. Otherwise, the record for the planning organization will not be accepted. You may want to rebuild the calendar if there is a change in the summary information at the source. This can be easily achieved by just uploading the new summary information with the calendar's *OVERWRITE_FLAG* set to 'Y'.

Purge Program

You can purge all uploaded data or selectively purge supply and demands with dates on or before a date that you specify. However, calendars, planning organizations, suppliers and customers cannot be purged. Except for the calendar, these can be disabled if required.

Customization

You may want to add custom validations to enable preprocessing to filter out unwanted incoming data. The preprocessing engine provides hooks for each entity that can be used to plug in custom validations.

Setting Up Legacy Instance

To set up legacy instance

The system default installation creates one instance partition and five plan partitions. Use this process if you need to create an instance partition.

1. From the System Administrator responsibility, select Requests > Run.
The Submit a New Request screen appears.
2. Select Single Request and select the OK button.
The Submit Request form appears.
3. In the Name field, select Create APS Partitions and select the OK button.
The Parameters screen appears.
4. Enter the number of plan partitions and instance partitions and select the OK button.
The partitions are created.
5. Change to the Advanced Planning Administrator responsibility. From the Navigator, select Admin > Instances.
The Application Instances screen appears.
6. Specify the Instance Code for the legacy instance and set the Instance Type as Other. Leave the fields From Source to APS and From APS To Source blank. Fill the other fields for the instance as specified in the online help.
You are now set to use the batch upload solution. Using the Running Legacy Collections process described below, upload the Workday Calendar data and Planning Organizations for this instance. This data can be uploaded along with the other entities' data. Preprocessing has the intelligence to consider the new organizations that have come in the same batch upload. After Legacy Collection is completed, you can view these organizations using the Organizations button at the bottom of the Instance Setup form.
- Note:** Setting up batch uploads and setting up legacy instance steps can occur in parallel up to creation of scripts for data uploads. However, for getting the data files from the scripts, the instance code is required.
7. Error messages for errors that occurred during preprocessing can be sent to the system administrator if Oracle XML Gateway is installed. You can specify the email

address for sending the error messages using the Profile ECS: System Administrator Email Address.

8. For enabling XML inbound/outbound for the planning system, you need to install the Oracle XML Gateway. After the planning organizations have been collected, the valid transactions for each of these organizations can be registered in the XML Gateway. To configure the XML Gateway, please refer to the Oracle XML Gateway User Guide.

Running Legacy Collections Using the Form-Based Application (Collect Flat-File Data)

Using either an Oracle Applications form or the self-service application page, you can upload data from flat files to the legacy instance and finally to the planning engine. Using the form, you upload each data file separately.

Using the self-service method, you can upload a zip file containing all data files. Each type of data file, such as work order supply or BOM header, is identified using a tag in the file name. Ensure that you do not zip the entire directory but add individual files to the zip file.

To collect into a legacy instance using the form-based application

1. Copy all the data files conforming to the control files in the \$MSC_TOP/patch/<version>/import in a directory on the concurrent manager node. If there are more than one concurrent manager nodes and if these are not NFS mounted, then the data files need to be copied to all the nodes in same directory structure. This directory (or all the directories in case of multiple non-NFS mounted concurrent manager nodes) should have read/write privileges to all users, because SQL*Loader discards files for the data that could not be uploaded due to errors.
2. Choose the Advanced Supply Chain Planner, Standard responsibility.
3. In the Navigator, choose Collections > Legacy Systems > Collect Flat File Data.

The Planning Data Collection screen appears showing three programs: Flat File Loader, Pre-Process Monitor, and Planning ODS Load. Planning ODS Load moves the data from the staging tables to the ODS.

4. Choose the Parameters field for Flat File Loader.
The Parameters screen appears.
5. Enter the required information and the File Names for all the data files that you want to upload. You can either enter the directory path in the Data File's Directory field and then enter the file names for each entity to be uploaded in the File Name fields, or you can leave the Data File's Directory field blank and enter the complete

path and file name of each entity in the File Name fields. The second option is useful if all the data files are not kept in the same directory.

The Total Number of Workers field specifies the number of maximum number of loader workers that should be running in parallel at any given point in time. A loader worker is launched for each file name specified.

6. When finished entering information for this screen, choose the OK button.
7. Choose the Parameters field for Pre-Process Monitor.

The Parameters screen appears.

8. Specify the entities that you want to be preprocessed for the legacy instance.

The Processing Batch Size field determines the size of batches while processing the records in the staging tables. A larger batch size is faster but requires more system resources. The current default batch size is 1000.

The Total Number of Workers field specifies the number of concurrent processes to be launched to process the data in parallel.

9. When finished entering information for this screen, choose the OK button.
10. Choose the Parameters field for Planning ODS Load.

The Parameters screen appears.

11. This program moves data from the staging tables to Oracle Value Chain Planning's main tables (ODS). Specify whether you want the Sourcing History to be recalculated or purged after the data has been moved, and whether you want to refresh the collected data in the Advanced Planning Command Center (APCC).

12. When finished entering information for this screen, choose the OK button.

The Planning Data Collection screen appears.

13. Press the Submit button to allow the concurrent manager to schedule the request as per the schedule options that you specify.
14. Use the View Requests Form to monitor the progress of the different programs.
15. Using the Advanced Supply Chain Planner responsibility, navigate to Collections > View Collected Data menu option to view the data coming into the planning server.

Running Legacy Collections Using the Self-Service Application (Collect Flat-File Data - Self Service)

The self-service application is used for small amounts of data.

To collect into a legacy instance using the self-service application

1. From the Advanced Supply Chain Planner (or Advanced Planning Administrator) responsibility, navigate to Collections > Legacy Systems > Collect Flat File Data - Self Service.

The Oracle Collaborative Planning page appears.

2. Click the Download link to download the Oracle Applications (OA) template.

All zipped .dat files, for example, bills of material and calendar appear.

You can read the OATemplateReadme.html file for information on how to load various entities into Oracle VCP using flat files. Open the ExcelLoad.xlt file, and import your data files from the APS menu to view and modify them.

Note: Resource Balance Flag: Indicates whether a resource is load balanced. Valid values are:

- 1: Yes
- 2: No

This flag is only for Oracle Process Manufacturing. Since you cannot use legacy collections with Oracle Process Manufacturing, always leave this field null.

Unit of Measure: Load all base unit of measure conversions without an item name. This creates rows in MSC_UOM_CONVERSIONS with INVENTORY_ITEM_ID = 0, for example:

- Base UOM: LB
- Standard conversions: KG - LB, Gram - LB, Kiloton - LB, mg - LB, Oz - LB, Ton - LB,

For specific conversions across UOM Class or specific intra-class unit of measure conversions for some items, load them using the item name

3. When all files have been imported to the templates, zip the files.
4. From the Oracle Collaborative Planning page, File Name field, click Browse to navigate to the zip file location.
5. Select the zip file containing the data files to be uploaded.
6. Click Start Load Now.

The concurrent request starts. You can note down the request id for your reference.

After the completion of this request, navigate to Collections Workbench to view the collected data.

Purging Legacy Data

There are two options for purging legacy data:

- Purge Collected Data deletes the data from the ODS table as well as the local id table (MSC_LOCAL_ID_XXX).
- Purge StagingTables purges Oracle Applications and legacy data from the staging tables. Use it before launching new collections when a previous data collection run failed, or if you find data corruption in the staging tables.

To purge collected data

1. Choose the Advanced Supply Chain Planner responsibility.
2. From the Navigator, choose Collections > Legacy System > Purge Collected Data. The Purge Parameters window appears.

The following table shows the values for this screen.

Field	Value
Instance	Legacy source instance against which the purge program is to be run.
Complete Refresh	Specify whether to perform a complete refresh or not. Valid values: No, Yes.
Delete Records Up to Date	The date range up to which all the records will be deleted from the collected data. Defaults to the current date.
Delete Supplies	Specify whether to delete supply-related entries from the collected data. Valid values: No, Yes (will always be Yes if complete refresh is Yes).

Field	Value
Delete Demands	Specify whether to delete demand-related entries from the collected data. Valid values: No, Yes (will always be Yes if complete refresh is Yes).

Example: Complete Refresh = Yes, Delete Supplies and Demands

In this case, the following tables get purged from ODS:

MSC_SYSTEM_ITEMS
 MSC_BOMS
 MSC_BOM_COMPONENTS
 MSC_COMPONENT_SUBSTITUTES
 MSC_ROUTINGS
 MSC_ROUTING_OPERATIONS
 MSC_OPERATION_RESOURCES
 MSC_OPERATION_COMPONENTS
 MSC_OPERATION_RESOURCE_SEQS
 MSC_PROCESS_EFFECTIVITY
 MSC_DEPARTMENT_RESOURCES
 MSC_RESOURCE_SHIFTS
 MSC_RESOURCE_CHANGES
 MSC_SIMULATION_SETS
 MSC_PROJECTS
 MSC_PROJECT_TASKS
 MSC_ITEM_CATEGORIES
 MSC_DESIGNATORS (Here program updates disable date as current date instead of deleting)
 MSC_DEMANDS
 MSC_SALES_ORDERS
 MSC_SUPPLIES
 MSC_INTERORG_SHIP_METHODS
 MSC_ABC_CLASSES

MSC_ST_RESOURCE_GROUPS

MSC_ST_DEMAND_CLASSES

MSC_ST_RESERVATIONS MSC_ST_SAFETY_STOCKS

In addition, the entities listed in the following table, which are stored in the LID table will be deleted.

Entity Name	LID Table Name	Business Object
SR_INVENTORY_ITEM_ID	MSC_LOCAL_ID_ITEM	Item
ABC_CLASS_ID	MSC_LOCAL_ID_MISC	Item
BILL_SEQUENCE_ID	MSC_LOCAL_ID_SETUP	BOM
COMPONENT_SEQUENCE_ID	MSC_LOCAL_ID_SETUP	BOM
ROUTING_SEQUENCE_ID	MSC_LOCAL_ID_SETUP	Routing
OPERATION_SEQUENCE_ID	MSC_LOCAL_ID_SETUP	Routing
RESOURCE_SEQ_NUM	MSC_LOCAL_ID_SETUP	Routing
DEPARTMENT_ID	MSC_LOCAL_ID_SETUP	Department/Resources
LINE_ID	MSC_LOCAL_ID_SETUP	Department/Resources
RESOURCE_ID	MSC_LOCAL_ID_SETUP	Department/Resources
PROJECT_ID	MSC_LOCAL_ID_MISC	Project/Tasks
TASK_ID	MSC_LOCAL_ID_MISC	Project/Tasks
COSTING_GROUP_ID	MSC_LOCAL_ID_MISC	Project/Tasks
SR_CATEGORY_ID	MSC_LOCAL_ID_MISC	Categories
DISPOSITION_ID_FCT	MSC_LOCAL_ID_DEMAND	Demand (Forecast)
DISPOSITION_ID_MDS	MSC_LOCAL_ID_DEMAND	Demand (MDS)

Entity Name	LID Table Name	Business Object
SALES_ORDER_ID	MSC_LOCAL_ID_DEMAND	Demand (Sales Order)
DEMAND_ID	MSC_LOCAL_ID_DEMAND	Demand (Sales Order)
DISPOSITION_ID	MSC_LOCAL_ID_SUPPLY	Supplies
PO_LINE_ID	MSC_LOCAL_ID_SUPPLY	Supplies (PO/Req)
SCHEDULE_GROUP_ID	MSC_LOCAL_ID_SUPPLY	Supplies (MPS)
DISPOSITION_ID_MPS	MSC_LOCAL_ID_SUPPLY	Supplies (MPS)
SR_MTL_SUPPLY_ID	MSC_LOCAL_ID_SUPPLY	Supplies (On Hand)
WIP_ENTITY_ID	MSC_LOCAL_ID_SUPPLY	Supplies (WIP)

The Purge program does not delete records related to following business objects from ODS or LID tables.

- Trading partners (organization, supplier, customer)
- Calendars
- Category sets
- Sourcing rules
- UOM

Note: For the items in the _LID tables, a complete refresh for ALL organizations deletes the data. When the profile option MSC: Purge Staging and Entity Key Translation Tables is set to Y, then the ID_LID tables are truncated during the key transformation in the ODS load phase.

Example: Complete Refresh = No, Delete Supplies and Demands

If you do not request a complete refresh, only supply/demand business object records and those records whose creation date is less than the user-entered date get deleted from the ODS and LID tables.

To purge staging tables

1. Use the Advanced Supply Chain Planner responsibility.
2. From the Navigator, choose Collections > Legacy System > Purge Interface Tables. The Submit a New Request window appears.
3. Select Single Request and click OK. The Submit Request window appears.
4. Select Purge Interface Tables from the Reports list. The Parameters window appears.
5. Set the parameters and click Submit.

The following table shows the values for this screen.

Field	Value
Instance	Legacy source instance against which the purge program is to be run.
Delete Rejected Records Only	Specify whether to delete only the rejected records or all records.

SRP Streams - Self Service

Legacy collections are supported for usage history for Demantra. Legacy collections of Install Base under Contracts are not currently supported. For more information about the EBS Service Parts Planning to Demantra Demand Management Integration, see "Demantra Demand Management to EBS Service Parts Planning Integration" in the *Oracle Demantra Integration Guide*.

To load usage history

1. From the Advanced Supply Chain Planner responsibility, navigate to Collections > Legacy > SRP Streams - Self Service. The Load SRP Streams form appears.
2. File Name: Specify the legacy data file. In the case of multiple files, the planner can unload a zip file with the individual .dat files.

Note: You can download templates for individual .dat files from this page.

3. Launch Download: Check this option if you would like to begin the download and launch the Demantra SPF Full Download workflow.
4. Instance: Choose from the available source instances.
5. Click Start Load Now to load the legacy data file(s) now.

Shipment History - Self Service

Legacy collections are supported for shipment history for Demantra. For more information, see:

- "Demantra Demand Management to EBS Service Parts Planning Integration" in the *Oracle Demantra Integration Guide*
- "Demantra Predictive Trade Planning to Siebel Integration" in the *Oracle Demantra Integration Guide*

.

To load usage history

1. From the Advanced Supply Chain Planner responsibility, navigate to Collections > Legacy > Shipment History - Self Service. The Load Shipment History form appears.
2. File Name: Specify the legacy data file. In the case of multiple files, the planner can unload a zip file with the individual .dat files.

Note: You can download templates for individual .dat files from this page.

3. Launch Download: Check this option if you would like to begin the download and launch the Demantra SPF GL Data Download workflow.
4. Instance: Choose from the available instances.
5. Click Start Load Now to load the legacy data file(s) now.

Loading Data from Flat Files

This chapter covers the following topics:

- Overview
- Preparing Transactional Data for Upload
- Loading Data from Flat Files to a VCP Destination Instance Using the Form-Based Application
- Collecting into a VCP Destination Instance Using the Self-Service Application
- Customizations

Overview

Using either an Oracle Applications form or the self-service application, you can upload transactional data (supply and demand) from flat files to the VCP destination instance. In the case of a centralized deployment, you can upload transactional data from flat files to the joint EBS/Planning instance.

With flat file loading, you identify the location of the data and control files for loading and select entities for pre-processing. With collecting via Self-Service, a set of .dat files are packaged into a single zip file before the collection is begun. The application requires a fixed naming convention for the .dat files that are loaded. These names are used to automatically identify the entity it contains and implies a corresponding .ctl file that controls how the file is loaded into the staging tables.

Ensure that the transactional data is uploaded to the planning server using either legacy systems directly or an Oracle EBS application. To avoid double counting, do not upload the same transaction data to both Legacy and EBS instances. For example, a sales order should not be uploaded using both EBS and Legacy instances.

Preparing Transactional Data for Upload

Before you can upload transactional data to an VCP destination instance, you must

format your data. An Excel template is provided to help you prototype or inspect the .dat files. Usually, implementations loading ERP data via flat (.dat) files have some sort of extraction and transformation process that extracts the ERP data from the legacy system and formats it to match the file and column organization required by the VCP destination instance .ctl files.

To download the Excel template from EBS

Using the Advanced Supply Chain Planner responsibility, navigate to the Load Data Files form (Collections > Legacy Systems > Collect Flat File Data -- Self Service) and select the Download link to download the Excel templates. Use the ExcelLoad.xlt file to insert and modify your .dat files.

To import .dat files the first time

If you are importing .dat files for the first time, then Excel prompts you to enter these values:

- Date format: The format of all the dates in the .dat files
- Delimiter: Used internally to separate different columns in the .dat file; make sure that it matches the value of profile option MSC: Self Service Loads Delimiter. Change these values in APS Menu > User Options. It opens a window to modify the values and select an action:
 - Apply: Applies the changes to the open sheet
 - Apply & Save: Applies the changes to the open sheet and saves the settings

Note: Once you enter these values, you do not need to enter them again.

Before uploading CSData.dat, set the date format in ExcelLoader to YYYY/MM/DD.

Loading Data from Flat Files to a VCP Destination Instance Using the Form-Based Application

To collect into a VCP destination instance using the form-based application

1. Navigate to the Planning Data Collection form (Collections > Oracle Systems > Load Transaction Data using Flat Files).

A Request Set form appears showing three programs: Load Transaction Data, Pre-Process Transaction Data, and Planning ODS Load. The Load Transaction Data program loads the transaction data through flat files into staging tables. Load Transaction Data accepts parameter values including the path for control and data

files.

The Pre-Process Transaction Data program preprocesses the transaction data and generates ids. Pre-Process Transaction Data enables you to specify the instance in which you want to load the transaction data.

Planning ODS Load program moves the data from the staging tables to the ODS.

2. Click in the Parameters field for Load Transaction Data.

The Parameters window appears.

3. Enter the required information and the file names for all the data files that you want to upload. Specify the maximum amount of time you would like to allocate to the concurrent program in the Time Out Duration field. You can either enter the directory path in the Data File's Directory field and then enter the file names for each entity to be uploaded in the File Name fields, or you can leave the Data File's Directory field blank and enter the complete path and file name of each entity in the File Name fields. The second option is useful if all the data files are not kept in the same directory.
4. When you finish entering information in the fields, click OK.

5. Click in the Parameters field for Pre-Process Transaction Data.

The Parameters window appears.

6. Select the instance from a list of values.

7. After specifying the instance in which you want to load the transaction data, specify the maximum time allowed for the process in the Time Out Duration field (in minutes).

The Processing Batch Size field determines the size of batches while preprocessing the records in the interface tables. An RDBMS COMMIT is performed for each batch. A larger batch size is faster but requires more system resources like UNDO tablespace. Each batch requires some overhead (like processing the commit and releasing the undoable data), so a larger number of smaller batches can increase the processing time. The current default batch size is 1000.

The Total Number of Workers field specifies the number of concurrent processes to be launched to process the data in parallel.

8. Specify the entities that you want to be preprocessed for the VCP destination instance. Yes indicates the entities that need to be preprocessed.

9. When you finish entering information in the fields, click OK.

10. Click in the Parameters field for Planning ODS Load.

The Parameters window appears.

The Planning ODS Load parameters required for data collection in the VCP destination instance is similar to the parameters required for legacy collections.

11. Specify the values for the parameters and click OK.
12. Click Submit in the Planning Data Collection window.
13. From the toolbar, choose View > Requests to view the status of the collections process.

When the request is complete, you can view the data in Collection Workbench.

Collecting into a VCP Destination Instance Using the Self-Service Application

To collect into a VCP destination instance using the self-service application

1. Double-click Collections > Oracle Systems > Load Transaction Data using Self Service Loads.
The Oracle Collaborative Planning page appears.
2. Click the Download link to download the Oracle Applications (OA) template.
All zipped .dat files, for example, bills of material and Calendar appear. A readme providing information on how to use the templates is also provided in this zip file.
3. Open the ExcelLoad.xlt file and from the APS menu, import your data files to view and modify. After making the changes, export the data file. Finally, zip all data files that need to be uploaded. For more information about the ExcelLoad.xlt file, see To download the Excel template from EBS, page 10-2.
4. Click Browse to navigate to the data files location.
5. Select the zip file containing data files to be uploaded.
6. Click Start Load Now.

A concurrent request is triggered.

After the completion of this request, navigate to Collections Workbench to view the collected data.

Customizations

System integrators may want to add custom validations for enabling preprocessing to filter out unwanted incoming data. The preprocessing engine provides hooks for each

entity, which can be used to plug-in custom validations.

Managing MLOG Tables

This chapter covers the following topics:

- Overview
- General Guidelines
- MLOG Growth
- MLOG Terminology
- How Rows are Cleared from MLOG\$ Tables
- How MLOGs are Used During Data Collection
- Managing the MLOG Growth for VCP Data Collections
- Most Common Performance Problems
- Loading Large Amounts of Data for Implementation

Overview

When the Planning Data Collection is run, a number of concurrent programs are launched to collect data from the EBS source tables (instance) to the VCP destination tables (instance). The Refresh Collection Snapshots program is responsible for taking 'snapshots' of the current data in the EBS source system. Many times, it is this program that causes the biggest performance impact for the entire Planning Data Collection process.

In a distributed installation where VCP applications run on separate instance from the EBS source transaction instance, the Refresh Collection Snapshots program is run on the EBS source instance. When the time between running Data Collections is such that very high volumes are recorded in the MLOG tables used to refresh snapshots, there are strategies to help improve performance.

General Guidelines

There are certain entities that change frequently throughout the day. Many customers perform a Complete Refresh Data Collection on a nightly basis. When this is done, a fast refresh of the snapshots is performed by the Refresh Collection Snapshots program for all entities except MTL_MATERIAL_TRANSACTIONS_TEMP. For this table/snapshot, a complete refresh of the snapshot is performed. If the volume of the data in the MLOG\$ table is too large, a fast refresh is not as effective as a complete refresh of the snapshots.

Note: If you run Refresh Collection Snapshots as a standalone program and use complete refresh or automatic refresh, then you must launch Data Collections with a complete refresh or data will be lost.

To improve the performance of the Complete Refresh Data Collection it is advisable to schedule the Refresh Collection Snapshots program to run periodically throughout the day to keep the volume of data in the MLOG\$ tables from growing too large during the day. This does not need to be done for all snapshots, just those that have a high volume of inserts, updates, and deletes throughout the day. When running the Refresh Collection Snapshots program, use the following parameters:

- Refresh Mode = Fast
- Snapshot Name = <desired snapshot>
- Threshold for Truncating Logs = NULL
- Degree = 0

To determine which entities within the system have large growth in the MLOG\$ tables, the following scripts can be used to monitor the changes in the snapshot logs:

1. Run SQL #26 -- Create Temporary Tables to Analyze Changes in Snapshot Logs, page D-20 to create temporary tables used for analyzing the changes in the snapshot logs.
2. Run SQL #25 -- Lock Statistics for All MLOGs Used for Data Collections, page D-18 to check the various MLOG\$ tables to get row counts for respective applications release.
3. Schedule SQL #25 -- Lock Statistics for All MLOGs Used for Data Collections, page D-18 in a cron job to run every 30-60 minutes for a week to monitor the row changes in the MLOG\$ tables.
4. Analyze the data in the temp tables that were created by SQL #26 -- Create Temporary Tables to Analyze Changes in Snapshot Logs, page D-20 by exporting

the output into an Excel spreadsheet using the following query:

```
select * from rcs_analysis_r12_temp;
```

When you analyze the output of each run, you should see the record count of the MLOG\$_tables increasing throughout the day up to the point where the complete refresh data collection is run. At that time, the record count should return to 0 and then start to grow again. It's the MLOG\$_tables with the largest amount of growth during the day that should be analyzed.

Example of data from rcs_analysis_r12_temp for the MLOG\$_OE_ORDER_LINES_ALL table:

Between 21:00 and 23:30, the volume of data is a 0. Then the data grows to between 2.7 and 3.5 million rows each day. If the Refresh Collection Snapshots program is scheduled to run every two hours starting at 06:30, the amount of data that needs to be processed during the complete refresh Data Collection is reduced to about 300,000. This can cause dramatic improvements to the overall data collection process. You may experience a 50-60% reduction in run times of the RCS program for the OE_ODR_LINES_SN snapshot after scheduling the program to run every two hours between runs of the Data Collection process.

Note: It is important that the Refresh Collection Snapshots program not be scheduled to run during the Data Collection process. Be sure to allow enough time for the last scheduled RCS program to complete prior to the scheduled data collection process.

MLOG Growth

MLOG management is the key to performance of Refresh Collection Snapshots. It is important to manage MLOGs to keep the program tuned to perform as fast as possible. If you are actively using data collections, performance problems occur for two primary reasons:

- MLOG tables are shared between VCP applications and other applications, usually daily Business Intelligence applications or custom reporting.
- Large source data changes occur via data loads and/or large transactional data volumes.

Once the initial collection has been performed and most data collected, then use the steps in Advanced Strategy for Handling MLOGs, page 11-8.

When you execute these steps, you will be setting the MLOGs to provide best performance.

You must also be aware if you have any MLOGs that are shared with other applications. In this case, the DBA will need to actively manage those other snapshots, refreshing them manually. A cron job may be the best method.

MLOG Terminology

MLOG\$ Table

An MLOG\$ table is the Snapshot Log table created against a base table. A row is inserted into the table each time a change (insert/update/delete) is made to the base table.

When a snapshot is refreshed in fast (incremental) mode, it reads the rows in the MLOG\$ table and uses this information to get the updated information from the base table into the snapshot for further processing.

Standard RDBMS functionality manages rows inserted into the MLOG tables. When a snapshot is refreshed in either fast or complete mode, then the RDBMS will delete rows from the MLOG table. However, if multiple snapshots reference the same MLOG, then all the snapshots must be refreshed before the RDBMS can remove any rows from the MLOG table.

The RDBMS creates snapshot logs on the base table and automatically names them using MLOG\$_[20 characters of the table name]. For example:

MRP_FORECAST_DATES (18 char) creates MLOG\$_MRP_FORECAST_DATES
BOM_DEPARTMENT_RESOURCES (24 char) creates
MLOG\$_BOM_DEPARTMENT_RESOU

In case of a tie, the RDBMS adds a number to the MLOG name. For example:

WSM_COPY_OP_RESOURCES creates MLOG\$_WSM_COPY_OP_RESOURCE
WSM_COPY_OP_RESOURCE_INSTANCES creates
MLOG\$_WSM_COPY_OP_RESOURCE1

Snapshot or Materialized View

Tables that reference the MLOG\$ log tables when a snapshot is refreshed using fast (or incremental) refresh. For a complete refresh, the snapshot pulls data directly from the base table and the RDBMS handles the MLOG in both cases to remove unneeded rows.

ALL_SNAPSHOTS.QUERY will show the query for a particular snapshot. See SQL #16
-- Show All Information on a Snapshot in DBA_SNAPSHOTS, page D-9 for more information.

Simple Snapshot: The Snapshot that queries rows from a single table. See How MLOGs Are Used During Data Collections, page 11-6 for examples.

Complex Snapshot: Snapshot query has multiple tables and therefore references multiple MLOG\$ tables to gather rows into the snapshot. See How MLOGs Are Used During Data Collections, page 11-6 for examples.

There are three refresh options:

- **Fast (or Incremental Refresh):** Default method of refreshing data into a snapshot for

changed rows in the base table. The snapshot refreshes the data based on changed rows read from the MLOG table. The RDBMS will then remove these rows from the MLOG table. If multiple snapshots use this MLOG table, then all must be refreshed before rows are deleted.

- Complete Refresh: Pulls all rows in the snapshot query into the snapshot directly from the base table, bypassing the MLOG table. The RDBMS will be aware of this action and delete rows from the MLOG table. If multiple snapshots use this MLOG table, then all must be refreshed before rows are deleted.
- Automatic Refresh: Specialized process exclusive to the VCP Refresh Collection Snapshots process. Designed to truncate the MLOG\$ tables and perform a complete refresh of the snapshots used for VCP Data Collections. The truncation of the MLOG helps to recover space and reset the high water mark for the table, which helps with performance and space allocation for the RDBMS.

It is only applicable for the list of snapshots used for VCP applications. See Appendix D -- Snapshots, page C-1 for a list of the snapshots.

Important: To maintain data integrity, after running an automatic refresh of a snapshot, then Data Collections must be run using complete refresh for all entities (or run a targeted refresh for the entity that uses that snapshot data).

How Rows are Cleared from MLOG\$ Tables

It is standard RDBMS functionality to manage the rows in the MLOG tables. When a snapshot is refreshed, then the rows from the MLOG\$ table are deleted. However, if multiple snapshots reference the same MLOG\$ table, then all the snapshots must be refreshed before any rows can be deleted from the MLOG\$ table. This is a primary cause of service requests that are logged with Oracle Support for issues with MLOGs. When there are many rows being inserted in the MLOG on a regular basis, then the delete performed by the RDBMS does not recover space and the size of the MLOG can grow. Therefore, Data Collections designed the automatic refresh into Refresh Collection Snapshots, so that the MLOG is truncated followed by a complete refresh of the snapshot. This is exclusive to Data Collections and can only be used for our list of snapshots. The basic steps are:

```
truncate table [owner].[MLOG$_name];  
exec DBMS_MVIEW.REFRESH ('[owner].[snapshot_name]', 'C');
```

Note: Since Data Collections also uses a sequence, you cannot use DBMS_MVIEW.REFRESH with Data Collection snapshots unless you first initiate the sequence.

How MLOGs are Used During Data Collection

MLOGs are created when the setup requests for Data Collections are run. The MLOG is used when refreshing the snapshots in the Refresh Collection Snapshots process to gather information on the changed rows that need to be collected by the Planning Data Pull process into the MSC_ST staging tables.

The following is an example of a simple snapshot and the data progression during Data Collections that moves the data from the EBS source tables to the MSC tables for use in VCP applications:

1. MTL_SYSTEM_ITEMS_B: The Inventory Items table.
2. MLOG\$_MTL_SYSTEM_ITEMS_B: Snapshot Log on the table.
3. MTL_SYS_ITEMS_SN: Data Collections snapshot for this table.
4. MRP_AP_SYS_ITEMS_V: View that references the snapshot using a synonym MRP_SN_SYS_ITEMS used by Planning Data Pull to populate.
5. MSC_ST_SYSTEM_ITEMS: The MSC staging table populated by the Planning Data Pull process.
6. MSC_SYSTEM_ITEMS: The MSC base table loaded during ODS Load process.

An example of a complex snapshot:

- BOM_OPR_SEQS_SN
- Tables seen in the ALL_SNAPSHOTS.QUERY output
- BOM_OPERATION_SEQUENCES
- BOM_OPERATIONAL_ROUTINGS

Note: The MASTER Column of ALL_SNAPSHOTS can only show a single table, so you must check the query to be certain that snapshot does not involve multiple tables OR use SQL #18 -- Use Snapshots to Find All Base Tables and MLOGs Used by a Snapshot, page D-10.

- MLOGs used by this snapshot to check for changed rows
- MLOG\$_BOM_OPERATION_SEQUEN
- MLOG\$_BOM_OPERATIONAL_ROUT

- MRP_AP_OP_RESOURCE_SEQS_V: View that references the snapshot (and other snapshots) using a synonym MRP_SN_OPR_SEQS. It is used by Planning Data Pull to populate.
- MSC_ST_OPERATION_RESOURCE_SEQS: The MSC staging table populated by Planning Data Pull process.
- MSC_OPERATION_RESOURCE_SEQS: The MSC base table which is loaded during ODS Load process.

Managing the MLOG Growth for VCP Data Collections

Primary Strategy -- Using Automatic Refresh

Keeping MLOG size manageable is the first requirement for performance. Do the following:

- Run Refresh Collections Snapshots using the automatic refresh mode to truncate an MLOG and then perform a complete refresh of the snapshot(s). Oracle recommends that you perform this for any MLOGs showing many rows. If you run for all snapshots, it can take a long time (several hours) and uses a lot of resources and UNDO space. If this is fine with the business and does not tax the system too heavily, then use all snapshots and get all snapshots truncated in one step. This procedure is not standard RDBMS functionality and is exclusive to the Refresh Collections Snapshot code. It can only be used for VCP snapshots.
- Run Data Collections with complete refresh for all entities you normally collect to ensure data integrity. Alternatively, you can run Data Collections with targeted refresh for the entities that use that snapshot.

Note: Sales Orders is special, and for a complete refresh, Oracle defaults the parameter to 'No' and collects sales orders in net change mode unless you explicitly set this parameter to 'Yes'.

- Oracle suggests that an automatic refresh be run on a periodic basis, usually weekly or monthly to recover space for MLOGs and keep Data Collections performance tuned. However, Oracle has found that this strategy is not good enough for customers who have:
 - Other applications with snapshots that reference the same MLOGs, and the other application is not being used or is run infrequently. In particular:
 - ENI: Product Intelligence snapshots are the primary cause of BOM MLOGs not being deleted and growing uncontrolled.

- OZF: Trade Management may have created a snapshot.
- High volumes and find that after executing automatic refresh, the performance of Refresh Snapshots is not improved or degenerates quickly.

Therefore, if the automatic refresh does not resolve your performance issues with Refresh Collection Snapshots, we recommend the advanced strategy for handling MLOGs.

Advanced Strategy for Handling MLOGs

This strategy works for any of the MLOG entities used for Data Collections. For specific problems, see Most Common Performance Problems, page 11-9.

When users are not updating the affected base tables, execute the following steps:

1. Get count(*) for the MLOGs affected.

```
select count(*) from [MLOG_name];
```

2. Truncate the MLOG(s) for the snapshot(s).

```
truncate table [owner].[MLOG$_name];
```

For BOM, WIP, WSM, MTL, OR WSH you need to do this for all MLOGs used for Data Collections.

3. Gather Statistics on the MLOG(s) while the MLOG has ZERO rows. From the System Administrator responsibility, run Gather Table Statistics with the following parameters:

- Table Owner: [OWNER]

- Table Name: [MLOG\$_name]

- All other parameters: Use default or in SQL*Plus, execute the following command:

```
begin FND_STATS.GATHER_TABLE_STATS ( '[owner]', '[MLOG$_name]', 10
); end;
```

4. Confirm that MLOGs have zero rows by checking DBA_TABLES information for those MLOGs. Run SQL #21 -- Size, NUM_ROWS and LAST_ANALYZED for MLOGs Used in Data Collections, page D-15.

The MLOGs only have more than zero rows if some user or process is updating the tables. If this happens, then you must find the user or process that is updating the tables and suspend that process.

5. Lock Statistics on the MLOG(s) while the MLOG has zero rows:

```
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ( '[owner]', '[MLOG_name]' );
```

6. Set up the database parameters to not use statisticss on the MLOG. As APPS user in SQL*Plus, set the database parameter _mv_refresh_use_stats = FALSE:

```
alter system set "_mv_refresh_use_stats"=FALSE;
```

The DBA can set in the init.ora to be used at all times.

7. Run Refresh Collection Snapshots against the snapshot(s) using complete refresh. You must do this for all snapshots used for Data Collections: BOM, WIP, WSM, MTL, or WSH. For others, you may use individual snapshots.

8. Gather table statistics for the snapshots. Use the same steps as #2 using the Snapshot_Name instead of the MLOG\$_.name for Table. In the latest code, the snapshots are owned by APPS and not the application. Confirm using SQL below:

```
Select * from dba_snapshots where name like '&snapshot_name' ;
```

9. Run Data Collections with targeted refresh with only Planners = Yes. This is a setup step for certain code improvements that help overall data collections performance. This step must be run for all the organizations being collected. Do not use a Collection Group for this run. To run in targeted mode, set the Planning Data Pull parameter Purge Previously Collected Data to 'No' and then the targeted collection method becomes available.

10. Run Data Collections with a complete refresh. This step should show the performance improvements.

11. If any other snapshots for other products also reference the MLOG being truncated, then refresh those snapshots after every run of Data Collections to keep the MLOGs from growing again.

Use SQL #15 -- Use Base Table to Find All Related Snapshots, page D-9 to find all the snapshots that are using the MLOG table for a single base table. Review Appendix D -- Snapshots, page C-1 to note any snapshot not seen in the list .

Note: If the production instance does not allow downtime to execute these steps, and if the MLOG\$ has zero rows in TEST and PROD, then Steps 1 - 3 may be run on a test instance and then the stats could be imported to the production instance. Import the stats to production during a slow period when you can complete the rest of these steps, truncate the MLOGs in production, and then start at Step #3 and continue the steps to completion.

Most Common Performance Problems

The most common performance problem scenarios are:

- OE_ODR_LINES_SN snapshot taking too much time.

- WIP snapshots taking too much time.
- BOM snapshots taking too much time.

If you don't know which snapshots are causing the problem, you can fix OE, WIP and BOM all at one time, or you can run a trace of the process Refresh Collection Snapshots (short name: MSRFWOR). The best practice is to use a Level 8 trace per setups in My Oracle Support Note #245974.1 steps #7 or #11 or you can set up a simple level 1 trace by checking the Enable Trace checkbox: From the System Administrator responsibility, navigate to Concurrent, then Program, and then Define AND the DBA must also run:

```
alter system set max_dump_file_size = 'unlimited';
```

Then use the Oracle DBA utility TKPROF to format the trace file into a more readable format for performance analysis:

```
tkprof trace_file_name.trc tkprof_file_name.txt sys=no explain=apps/[apps_pw] sort='(prsela,exeela,fchela)' print=20;
```

Ignore the first statement in the TKPROF: it is for the total refresh process that refreshed all the snapshots.

```
BEGIN MRP_CL_REFRESH_SNAPSHOT.REFRESH_SNAPSHOT(:errbuf,:rc,:A0,:A1,:A2,:A3);
END;
```

Check the next statement(s) in the TKPROF output to determine which snapshots are involved. For example, the first line of the second statement in this TKPROF example shows that WIP snapshots are involved:

```
INSERT INTO "WIP"."WIP_WOPR_RESS_SN" SELECT /*+ NO_MERGE("JV$") */ /*+
*/
```

When you have determined the problem snapshot(s) or MLOG(s) involved, the steps below can be executed to resolve the issue.

Performance Issue with OE_ODR_LINES_SN

Perform these steps when other processes are not updating the table OE_ORDER_LINES_ALL:

1. Suspend workflow processes. Some customers have workflow processes that update order lines at a rate of several thousand lines per hour.

2. Get Count on MLOG\$_OE_ORDER_LINES_ALL:

```
select count(*) from MLOG$_OE_ORDER_LINES_ALL;
```

3. Truncate the MLOG\$ table:

```
truncate table ONT.MLOG$_OE_ORDER_LINES_ALL;
```

4. Confirm ZERO rows in the MLOG:

```
select count(*) from MLOG$_OE_ORDER_LINES_ALL;
```

5. Gather table stats on the MLOG. From the System Administrator responsibility, run

the single request Gather Table Statistics with the following parameters:

- OWNER: ONT
- TABLE: MLOG\$_OE_ORDER_LINES_ALL

Use the default for all the other parameters or in SQL*Plus, execute:

```
begin FND_STATS.GATHER_TABLE_STATS (
  'ONT', 'MLOG$_OE_ORDER_LINES_ALL', 10 ); end;
```

6. Confirm that num_rows in dba_tables for the MLOG table = 0:

```
select table_name, num_rows, last_analyzed from dba_tables
where table_name in ('MLOG$_OE_ORDER_LINES_ALL');
```

7. Lock the Statistics using the following command:

```
EXECUTE DBMS_STATS.LOCK_TABLE_STATS
( 'ONT', 'MLOG$_OE_ORDER_LINES_ALL' );
```

8. As APPS User in SQL*Plus, set the database parameter _mv_refresh_use_stats = FALSE:

```
alter system set "_mv_refresh_use_stats"=FALSE;
```

The DBA can set in the init.ora to be used at all times.

9. Run Refresh Collection Snapshots as a standalone request with the following parameters:

- Refresh Mode: Complete
- Snapshot Name: OE_ODR_LINES_SN
- Other parameters: Use default

10. Gather table statistics on the snapshot. From the System Administrator responsibility, run the single request Gather Table Statistics with the following parameters:

- Table Owner: ONT [or APPS in the latest code].
- Table Name: OE_ODR_LINES_SN. You can check for the latest code using:

```
Select * from dba_snapshots where name like '&snapshot_name' ;
```
- All other parameters: Use the default.

Alternatively, in SQL*Plus, execute the following:

```
begin FND_STATS.GATHER_TABLE_STATS ( 'ONT [or
APPS]', 'OE_ODR_LINES_SN', 10 ); end;
```

11. Run Data Collections with Targeted Refresh with only Planners = Yes. This is a

setup step for certain code improvements that help with overall data collections performance. This step must be run for all the organizations being collected. Do not use an Collection Group for this run. To run in targeted mode, set the Planning Data Pull parameter Purge Previously Collected Data to 'No' and then choose the targeted collection mode.

12. Run Data Collections with either a complete or targeted refresh with only the Sales Orders parameter set to 'Yes'.

Note: The Sales Order entity is unique. When complete refresh is specified for Data Collections, we default the Sales Order parameter to 'No' and run the net change refresh of sales orders automatically for performance reasons.

13. Observe that during the run of Data Collections, the performance is improved.

Performance Issue with WIP Snapshots

Use the following steps when other processes are not updating the WIP tables:

1. Get Count on MLOG\$_WIP tables:

```
select count(*) from MLOG$_WIP_DISCRETE_JOBS;
select count(*) from MLOG$_WIP_FLOW_SCHEDULES;
select count(*) from MLOG$_WIP_LINES;
select count(*) from MLOG$_WIP_OP_RESOURCE_INST;
select count(*) from MLOG$_WIP_OPERATION_NETWOR;
select count(*) from MLOG$_WIP_OPERATION_RESOUR;
select count(*) from MLOG$_WIP_OPERATIONS;
select count(*) from MLOG$_WIP_REPETITIVE_ITEMS;
select count(*) from MLOG$_WIP_REPETITIVE_SCHED;
select count(*) from MLOG$_WIP_REQUIREMENT_OPER;
select count(*) from MLOG$_WIP_SUB_OPERATION_RE;
```

2. Truncate the MLOG\$ tables:

```
truncate table WIP.MLOG$_WIP_DISCRETE_JOBS;
truncate table WIP.MLOG$_WIP_FLOW_SCHEDULES;
truncate table WIP.MLOG$_WIP_LINES;
truncate table WIP.MLOG$_WIP_OP_RESOURCE_INST;
truncate table WIP.MLOG$_WIP_OPERATION_NETWOR;
truncate table WIP.MLOG$_WIP_OPERATION_RESOUR;
truncate table WIP.MLOG$_WIP_OPERATIONS;
truncate table WIP.MLOG$_WIP_REPETITIVE_ITEMS;
truncate table WIP.MLOG$_WIP_REPETITIVE_SCHED;
truncate table WIP.MLOG$_WIP_REQUIREMENT_OPER;
truncate table WIP.MLOG$_WIP_SUB_OPERATION_RE;
```

3. Confirm zero rows in the MLOG\$:

```

select count(*) from MLOG$_WIP_DISCRETE_JOBS;
select count(*) from MLOG$_WIP_FLOW_SCHEDULES;
select count(*) from MLOG$_WIP_LINES;
select count(*) from MLOG$_WIP_OP_RESOURCE_INST;
select count(*) from MLOG$_WIP_OPERATION_NETWOR;
select count(*) from MLOG$_WIP_OPERATION_RESOUR;
select count(*) from MLOG$_WIP_OPERATIONS;
select count(*) from MLOG$_WIP_REPEATITIVE_ITEMS;
select count(*) from MLOG$_WIP_REPEATITIVE_SCHED;
select count(*) from MLOG$_WIP_REQUIREMENT_OPER;
select count(*) from MLOG$_WIP_SUB_OPERATION_RE;

```

4. Gather table stats on the MLOGs. From the System Administrator responsibility, run the single request Gather Table Statistics with the following parameters:
 - Table Owner: WIP
 - Table Name: MLOG\$_WIP_DISCRETE_JOBS
 - Other parameters: Use defaults.

Alternatively, in SQL*Plus, run:

```

begin FND_STATS.GATHER_TABLE_STATS ( 'WIP', ' MLOG$_WIP_DISCRETE_JOBS
', 10 ); end;

```

5. Repeat Step 4 for the complete list of MLOG tables:

```

MLOG$_WIP_FLOW_SCHEDULES
MLOG$_WIP_LINES
MLOG$_WIP_OP_RESOURCE_INST
MLOG$_WIP_OPERATION_NETWOR
MLOG$_WIP_OPERATION_RESOUR
MLOG$_WIP_OPERATIONS
MLOG$_WIP_REPEATITIVE_ITEMS
MLOG$_WIP_REPEATITIVE_SCHED
MLOG$_WIP_REQUIREMENT_OPER
MLOG$_WIP_SUB_OPERATION_RE

```

6. Confirm that num_rows in dba_tables for the MLOG table = 0:

```

select table_name, num_rows, last_analyzed from dba_tables
where table_name like 'MLOG$_WIP%';

```

7. Lock the Statistics.

```

EXECUTE DBMS_STATS.LOCK_TABLE_STATS
('WIP','MLOG$_WIP_DISCRETE_JOBS');
EXECUTE DBMS_STATS.LOCK_TABLE_STATS
('WIP','MLOG$_WIP_FLOW_SCHEDULES');
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('WIP','MLOG$_WIP_LINES');
EXECUTE DBMS_STATS.LOCK_TABLE_STATS
('WIP','MLOG$_WIP_OP_RESOURCE_INST');
EXECUTE DBMS_STATS.LOCK_TABLE_STATS
('WIP','MLOG$_WIP_OPERATION_NETWORK');
EXECUTE DBMS_STATS.LOCK_TABLE_STATS
('WIP','MLOG$_WIP_OPERATION_RESOURCE');
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('WIP','MLOG$_WIP_OPERATIONS');
EXECUTE DBMS_STATS.LOCK_TABLE_STATS
('WIP','MLOG$_WIP_REPEATITIVE_ITEMS');
EXECUTE DBMS_STATS.LOCK_TABLE_STATS
('WIP','MLOG$_WIP_REPEATITIVE_SCHEDULE');
EXECUTE DBMS_STATS.LOCK_TABLE_STATS
('WIP','MLOG$_WIP_REQUIREMENT_OPER');
EXECUTE DBMS_STATS.LOCK_TABLE_STATS
('WIP','MLOG$_WIP_SUB_OPERATION_RE');

```

8. As APPS User in SQL*Plus, set the database parameter `_mv_refresh_use_stats = FALSE`:

```
alter system set "_mv_refresh_use_stats"=FALSE;
```

The DBA can set in the init.ora to be used at all times.

9. Run Refresh Collection Snapshots as a standalone request with parameters:

- Refresh Mode: Complete
- Snapshot Name: WIP_DSCR_JOBS_SN
- All other parameters: Use default.

10. Run Refresh Collection Snapshots for the rest of the WIP snapshots:

```

WIP_FLOW_SCHDS_SN
WIP_WLINES_SN
WIP_OPR_RES_INSTS_SN
WIP_WOPR_NETWORKS_SN
WIP_WOPR_RESS_SN
WIP_WOPRS_SN
WIP_REPT_ITEMS_SN
WIP_REPT_SCHDS_SN
WIP_WREQ_OPRS_SN
WIP_WOPR_SUB_RESS_SN

```

11. Gather table stats on the snapshot. From the System Administrator responsibility, run the standalone request Gather Table Statistics with the following parameters:

- Table Owner: WIP [or APPS in the latest code].
- Table Name: WIP_DSCR_JOBS_SN

You can check the name using:

```
Select * from dba_snapshots where name like '&snapshot_name';
```

- All other parameters: Use default.

Alternatively, in SQL*Plus, execute:

```
begin FND_STATS.GATHER_TABLE_STATS ( 'WIP [or
APPS]', 'WIP_DSCR_JOBS_SN', 10 ); end;
```

12. Repeat Step #11 for the entire list of WIP snapshots shown in Step #10.
13. Run Data Collections with a targeted refresh with only Planners = Yes. This is a setup step for certain code improvements that help overall data collections performance. This must be run for all the organizations being collected; do not use a Collection Group for this run. To run targeted mode, set Planning Data Pull parameter Purge Previously Collected Data to 'No' and then the targeted collection mode becomes available.
14. Run a Data Collections with a targeted refresh with only Work in Process parameter set to 'Yes' or run Data Collections with a complete refresh and other parameters as default (or your normal Collections setup for a complete refresh).

Observe that during the run of Data Collections, the performance is improved.

Performance Issue with BOM Snapshots

Warning: BOM snapshots are used by multiple products (MSC - Data Collections and ENI - Product Intelligence), so further steps are involved in clean up of the MLOGs.

Perform these steps when other processes are not updating the BOM tables:

1. Determine which snapshots are involved. The BOM MLOGs may have multiple snapshots from different products accessing the same MLOG. For example, ENI - Product Intelligence installs snapshots like ENI_DBI_BOM_COMPONENTS_MV1 ENI_DBI_BOM_COMPONENTS_MV2 ENI_DBI_MFG_STEPS_JOIN_MV ENI_RES_1_MV You may observe that two or more of these snapshots exist on the system, or that other snapshots exist as well. Therefore, after fixing the MLOGs manually, in the future you must run Product Intelligence or manually refresh the ENI snapshots after running Data Collections to prevent the MLOGs from growing out of control again.

Run the following to check which snapshots exist on the system:

```

SELECT DISTINCT
amdr.MVIEW_NAME "Snapshot",
amdr.OWNER "Snapshot Owner",
amdr.DETAILOBJ_NAME "Base Table Name",
amdr.DETAILOBJ_OWNER "Base Table Owner",
log_table mlog$_name
FROM ALL_MVIEW_DETAIL_RELATIONS amdr,
dba_snapshot_logs dsl
where DETAILOBJ_NAME like 'BOM%'
and DETAILOBJ_TYPE = 'TABLE'
and (detailobj_owner, detailobj_name) not in
('MSC','MSC_COLLECTED_ORGS')
AND amdr.DETAILOBJ_NAME = dsl.MASTER
order by mview_name;

```

All BOM%SN objects are used for Data Collections. Multiple rows returned for a snapshot shows that snapshot is a complex snapshot that references multiple tables. This is normal. In this case, the Base Table Name and MLOG\$_NAME show the different tables being used for a snapshot.

If you want more information on the snapshots, you can review the snapshot query by running the following SQL:

```
select * from all_snapshots where name like '&snapshot_name';
```

and then review the column 'Query'.

2. Get Count on MLOG\$:

```

select count(*) from MLOG$_BOM_COMPONENTS_B;
select count(*) from MLOG$_BOM_CTO_ORDER_DEMAND;
select count(*) from MLOG$_BOM_DEPARTMENT_RESOU;
select count(*) from MLOG$_BOM_DEPARTMENTS;
select count(*) from MLOG$_BOM_OPERATION_NETWOR;
select count(*) from MLOG$_BOM_OPERATION_RESOUR;
select count(*) from MLOG$_BOM_OPERATION_SEQUEN;
select count(*) from MLOG$_BOM_OPERATIONAL_ROUT;
select count(*) from MLOG$_BOM_RES_INSTANCE_CHA;
select count(*) from MLOG$_BOM_RESOURCE_CHANGES;
select count(*) from MLOG$_BOM_RESOURCES;
select count(*) from MLOG$_BOM_STRUCTURES_B;
select count(*) from MLOG$_BOM_SUB_OPERATION_RE;
select count(*) from MLOG$_BOM_SUBSTITUTE_COMPO;

```

3. Truncate the MLOG\$ tables:

```

truncate table BOM.MLOG$_BOM_COMPONENTS_B;
truncate table BOM.MLOG$_BOM_CTO_ORDER_DEMAND;
truncate table BOM.MLOG$_BOM_DEPARTMENT_RESOU;
truncate table BOM.MLOG$_BOM_DEPARTMENTS;
truncate table BOM.MLOG$_BOM_OPERATION_NETWOR;
truncate table BOM.MLOG$_BOM_OPERATION_RESOUR;
truncate table BOM.MLOG$_BOM_OPERATION_SEQUEN;
truncate table BOM.MLOG$_BOM_OPERATIONAL_ROUT;
truncate table BOM.MLOG$_BOM_RES_INSTANCE_CHA;
truncate table BOM.MLOG$_BOM_RESOURCE_CHANGES;
truncate table BOM.MLOG$_BOM_RESOURCES;
truncate table BOM.MLOG$_BOM_STRUCTURES_B;
truncate table BOM.MLOG$_BOM_SUB_OPERATION_RE;
truncate table BOM.MLOG$_BOM_SUBSTITUTE_COMPO;

```

4. Confirm zero rows in the MLOG\$:

```

select count(*) from MLOG$_BOM_COMPONENTS_B;
select count(*) from MLOG$_BOM_CTO_ORDER_DEMAND;
select count(*) from MLOG$_BOM_DEPARTMENT_RESOU;
select count(*) from MLOG$_BOM_DEPARTMENTS;
select count(*) from MLOG$_BOM_OPERATION_NETWOR;
select count(*) from MLOG$_BOM_OPERATION_RESOUR;
select count(*) from MLOG$_BOM_OPERATION_SEQUEN;
select count(*) from MLOG$_BOM_OPERATIONAL_ROUT;
select count(*) from MLOG$_BOM_RES_INSTANCE_CHA;
select count(*) from MLOG$_BOM_RESOURCE_CHANGES;
select count(*) from MLOG$_BOM_RESOURCES;
select count(*) from MLOG$_BOM_STRUCTURES_B;
select count(*) from MLOG$_BOM_SUB_OPERATION_RE;
select count(*) from MLOG$_BOM_SUBSTITUTE_COMPO;

```

5. Gather table statistics on the MLOG. From the System Administrator responsibility, run the single request Gather Table Statistics with the following parameters:

- Table Owner: BOM
- Table Name: MLOG\$_BOM_COMPONENTS_B
- Other parameters: Use default.

Alternatively, in SQL*Plus run:

```

begin FND_STATS.GATHER_TABLE_STATS ( 'BOM' ,
MLOG$_BOM_COMPONENTS_B ', 10 ); end;

```

6. Repeat Step #5 for the complete list of MLOG tables:

```

MLOG$_BOM_CTO_ORDER_DEMAND
MLOG$_BOM_DEPARTMENT_RESOU
MLOG$_BOM_DEPARTMENTS
MLOG$_BOM_OPERATION_NETWOR
MLOG$_BOM_OPERATION_RESOUR
MLOG$_BOM_OPERATION_SEQUEN
MLOG$_BOM_OPERATIONAL_ROUT
MLOG$_BOM_RES_INSTANCE_CHA
MLOG$_BOM_RESOURCE_CHANGES
MLOG$_BOM_RESOURCES
MLOG$_BOM_STRUCTURES_B
MLOG$_BOM_SUB_OPERATION_RE
MLOG$_BOM_SUBSTITUTE_COMPO

```

7. Confirm that num_rows in dba_tables for the MLOG table = 0:

```

select table_name, num_rows, last_analyzed from dba_tables
where table_name like 'MLOG$_BOM%';

```

8. Lock the Statistics.

```

EXECUTE DBMS_STATS.LOCK_TABLE_STATS
('BOM', 'MLOG$_BOM_COMPONENTS_B');
EXECUTE DBMS_STATS.LOCK_TABLE_STATS
('BOM', 'MLOG$_BOM_CTO_ORDER_DEMAND');
EXECUTE DBMS_STATS.LOCK_TABLE_STATS
('BOM', 'MLOG$_BOM_DEPARTMENT_RESOU');
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('BOM', 'MLOG$_BOM_DEPARTMENTS');
EXECUTE DBMS_STATS.LOCK_TABLE_STATS
('BOM', 'MLOG$_BOM_OPERATION_NETWORK');
EXECUTE DBMS_STATS.LOCK_TABLE_STATS
('BOM', 'MLOG$_BOM_OPERATION_RESOUR');
EXECUTE DBMS_STATS.LOCK_TABLE_STATS
('BOM', 'MLOG$_BOM_OPERATION_SEQUEN');
EXECUTE DBMS_STATS.LOCK_TABLE_STATS
('BOM', 'MLOG$_BOM_OPERATIONAL_ROUT');
EXECUTE DBMS_STATS.LOCK_TABLE_STATS
('BOM', 'MLOG$_BOM_RES_INSTANCE_CHA');
EXECUTE DBMS_STATS.LOCK_TABLE_STATS
('BOM', 'MLOG$_BOM_RESOURCE_CHANGES');
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('BOM', 'MLOG$_BOM_RESOURCES');
EXECUTE DBMS_STATS.LOCK_TABLE_STATS
('BOM', 'MLOG$_BOM_STRUCTURES_B');
EXECUTE DBMS_STATS.LOCK_TABLE_STATS
('BOM', 'MLOG$_BOM_SUB_OPERATION_RE');
EXECUTE DBMS_STATS.LOCK_TABLE_STATS
('BOM', 'MLOG$_BOM_SUBSTITUTE_COMPO');

```

9. As APPS user in SQL*Plus, set the database parameter `_mv_refresh_use_stats = FALSE`:

```
alter system set "_mv_refresh_use_stats"=FALSE;
```

The DBA should set in the init.ora to be used at all times.

10. Run Refresh Collection Snapshots as a standalone request with the following parameters:

- Refresh Mode: Complete
- Snapshot Name: BOM_BOM_SN
- Other parameters: Use defaults.

11. Repeat Step #10 for all BOM snapshots:

```

BOM_CTO_ORDER_DMD_SN
BOM_INV_COMPS_SN
BOM_OPR_NETWORKS_SN
BOM_OPR_RESS_SN
BOM_OPR_RTNS_SN
BOM_OPR_SEQS_SN
BOM_RES_CHNGS_SN
BOM_RES_INST_CHNGS_SN
BOM_SUB_COMPS_SN
BOM_SUB_OPR_RESS_SN

```

12. Gather table statistics on the snapshot. From the System Administrator, run the single request Gather Table Statistics with the following parameters:

- Table Owner: BOM [or APPS in the latest code]. Check using:

```
select * from all_snapshots where name like '&snapshot_name';
```

- Table Name: BOM_BOMS_SN
- Other parameters: Use default

Alternatively, in SQL*Plus execute:

```
begin FND_STATS.GATHER_TABLE_STATS ( 'BOM [or APPS]' , 'BOM_BOMS_SN' ,  
10 ); end;
```

13. Repeat for entire list of BOM snapshots shown in Step #10.
14. Run Data Collections with a targeted refresh with only Planners = Yes. This is a setup step for certain code improvements that help overall Data Collections performance. This must be run for all the organizations being collected; do not use a Collection Group for this run. To run in targeted mode, set Planning Data Pull parameter Purge Previously Collected Data to 'No' and then Collection Method 'Targeted' becomes available.
15. Run Data Collections with a targeted refresh with only Bills of Material, Routing, Resources parameter set to 'Yes'. Alternatively, run Data Collections with a complete refresh and all other parameters set to default (or your normal Collections setup for complete refresh).

Observe that during the run of Data Collections, the performance is improved.

Important: To ensure the problem does not occur again, do the following:

Assuming that you are not currently using Product Intelligence (application short name ENI), you need to refresh the snapshots NOT used by Data Collections after each collections run. You may or may not have all the ENI snapshots listed above in Step #1, but each snapshot you have must be refreshed. You can use the following SQL to run a complete refresh of the snapshots:

```
exec DBMS_MVIEW.REFRESH ('[Snapshot_owner].  
[snapshot_name]' , 'C');
```

Examples:

```
exec DBMS_MVIEW.REFRESH ('APPS.ENI_RES_1_MV' , 'C');  
exec DBMS_MVIEW.REFRESH ('APPS.  
ENI_DBI_MFG_STEPS_JOIN_MV' , 'C');  
exec DBMS_MVIEW.REFRESH ('APPS.  
ENI_DBI_BOM_COMPONENTS_MV1' , 'C');  
exec DBMS_MVIEW.REFRESH ('APPS.  
ENI_DBI_BOM_COMPONENTS_MV2' , 'C');
```

This can be setup by the DBA to run as a cron job. Alternately, create a file and save as a .SQL file. Then create a custom concurrent request to run the file. This would be the

same type of setup seen for request Create BOM Snapshots. This process runs the file MSCBOMSN.sql and is stored under \$MSC_TOP/sql. This could be used as a model for the new concurrent request defined by your System Administrator.

MLOG\$_ENI_OLTP_ITEM_STAR is Growing Too Large, Can it Be Truncated?

The snapshot log MLOG\$_ENI_OLTP_ITEM_STAR has a large number of records and is growing daily. Is it possible to purge this log?

First, check to see if you have Daily Business Intelligence (DBI) by running the following query on the database:

```
SQL> SELECT patch_level
  FROM fnd_product_installations
 WHERE patch_level LIKE %BIS%
 /
```

If no rows are returned, no BIS products (which include DBI) are installed. If the query returns 11i.BIS.K with e-Business Suite 11.5.10, it is likely that you are not using DBI, since this is the base level for 11.5.10. DBI users should be using a higher patchset level. You could check with your users in case they are using a lower patchset level than Oracle recommends, and then upgrade DBI to the latest patchset level at the earliest opportunity.

If you are not using DBI (Oracle Product Intelligence) you can truncate the table and drop the materialized view log on it by running the following commands:

```
DROP SNAPSHOT LOG ON ENI.ENI_OLTP_ITEM_STAR;
TRUNCATE TABLE ENI.ENI_OLTP_ITEM_STAR;
```

The ENI_OLTP_ITEM_STAR table is still maintained but the MLOG\$ is not be populated. If you prefer, you can truncate the snapshot log instead of dropping it (if you want to maintain the snapshot log object for whatever reason) but you need to execute the truncate command on a regular basis so the log does not grow too large.

The MLOG\$ is only there to support fast refresh for the materialized views based on this table. In the absence of this, the materialized views can always be fully refreshed. By dropping the MLOG\$, you are not losing any information. Additionally, if you are not using DBI, then you can truncate the two associated materialized views below:

- MLOG\$_ISC_DBI_CFM_002_MV
- MLOG\$_ISC_DBI_CFM_000_MV

Loading Large Amounts of Data for Implementation

Option 1 -- Periodic Large Data Loads

1. Import the data.
2. Truncate the MLOG table using the automatic refresh of the snapshot(s).

3. Run Data Collections in targeted mode for that entity or a normal run of complete refresh of Data Collections.

Option 2 -- Large Data Load During Implementation

Run Data Collections in targeted mode for that entity or a normal run of complete refresh of Data Collections.

1. Ensure that you have a lot of space for TEMP and UNDO and for your application tablespaces.
2. Run Data Collections for all the orgs you are going to use in Value Chain Planning. Do NOT use Org Groups and try to load partial data. This actually takes more TEMP space than loading all the data at one time due to the technology used for handling Org Groups and TEMP tables.
3. Prepare to have the initial Data Collections set up with a very large timeout parameter. Between 900 - 1600 is not unusual for large amounts of data when loading empty MSC tables.
4. Once the first run of Data Collections is complete, run Gather Schema Stats on the MSC schema. The follow up runs of Data Collections should run fine and performance should be much better.
5. If loading heavy BOM or WIP data for the initial load, see Advanced Strategy for Handling MLOGs, page 11-8 and implement immediately if you encounter any issues in Refresh Collection Snapshots during initial collections.

Troubleshooting

This chapter covers the following topics:

- Overview
- First Steps
- Basic Performance Tuning for MSC Partitions
- Diagnosing Data Collection Problems
- Refresh Collection Snapshots or Planning Data Pull Errors
- ODS Load Errors
- Refresh Snapshot Process Doesn't Start Because of Concurrent Manager Unavailability
- Data Collection Fails with Several Messages
- Missing Data Collection Entities
- Tracing
- Other Troubleshooting Resources
- Data Field Notes

Overview

For a list of data collection errors, organized by error message, please review My Support Note 1227255.1 -- "Troubleshooting Errors with ATP/Planning Data Collections".

First Steps

You may encounter a failure for many reasons:

- After the latest cumulative patch has been applied.

- Original issue is still occurring after the latest patch is applied.
- Clone of an instance that is working fine.

The first step for troubleshooting is to review the following:

- Initial Setup, page 4-1 and Configuring Profile Options, page 7-1 for information about configuring the data collection. In particular, review the profile MSC: Source Setup Required section.
- Running Collections, page 8-1 for information about different collection options.
- MLOG Growth, page 11-3 for information about the strategies for managing MLOG growth.
- Appendix B -- Parameters, page A-1 for questions about parameters.
- Managing Tablespaces, page 4-4 for sizing and space issues.

Basic Performance Tuning for MSC Partitions

- MSC: Share Plan Partition = No is a required setting. Using MSC: Share Plan Partitions = Yes can cause performance issues, and requires that you perform steps to convert to No before you perform any volume testing or go live.
- Do not create extra plan partitions. Unused plan partitions consume space in the Oracle RDBMS SGA data dictionary and can be very bad for performance. There are eighteen or more tables in the MSC schema in which partitioning is used depending on the release. If ten partitions are created, this results in 180+ database objects, not including indexes. See Appendix F -- List of APS Partitioned Tables, page E-1 for more information.
- NEVER DELETE THE _0 PARTITION!!!! Deleting this original partition will cause the instance to stop functioning!! Since each partition in a Range partition only specifies the HIGH_VALUE limit of the range, this _0 partition with a HIGH_VALUE of 0 or -1 defines the lower limit of the PLAN_ID range.
- Determine if you have any free partitions. Use SQL #2 -- Show Created Instance Partitions, page D-2 to check for free_flag = 1 for instance partitions Use SQL #5 -- Show Plans and Associated Plan Partitions, page D-3 to check for free_flag = 1 for plan partitions If any free_flag partitions exist, then remove them using the steps described in Using Drop Partition or Drop Plan Partition, page 5-10.

Note: A plan name can be assigned, but not active and the free_flag = 1. This usually indicates that the plan has been purged and not

yet replaced by a new plan name. If performance problems are occurring, then delete the partition and create a new partition when needed.

- For any performance problems with Planning Data Collection, the planning process, or the Workbench in ASCP do the following:
 1. Determine if you have any free partitions and eliminate any partitions not being used. See [Setting Up Partitions, page 5-1](#) for more information.
 2. Run Gather Schema Statistics for MSC schema at 50%.
 - Source: BOM, INV, WIP, PO, ONT, MRP, WSH, WSM, APPS, AHL
 - Destination: MSC
 3. Run Analyze Plan partition program for the plan name.
- For more information about maintaining sufficient rollback for the VCP application, please see [My Oracle Support Note #266797.1 - "Rollback Segment Assignment and Automatic Undo Management Mode for MRP and APS Applications ORA-01555"](#).

Diagnosing Data Collection Problems

To check on specific requests that are running and show the SQL that is running on the RDBMS, see [My Oracle Support Note #186472.1 -- "Diagnostic Scripts: 11i - Hanging SQL - Find the Statement Causing Process to Hang"](#). Note:

- When you run this SQL, you provide the Request ID and it checks to find the SQL being run for this session.
- If you have several Planning Data Pull workers (or ODS Load Workers) running, then you may need to check several different requests.
- Run this SQL every ten to fifteen minutes and check if the output changes.

To see details on all the requests in a set, see [My Oracle Support Note #280295.1 -- "REQUESTS.sql Script for Parent/Child Request IDs and Trace File IDs"](#). Enter the request id for the request that launched the set and it will provide details of all the requests. The default request for the set is 'Planning Data Collection (Request Set Planning Data Collection)'.

To set up traces for requests if you need to get details on a performance issue, see [My Oracle Support Note #245974.1-- "FAQ - How to Use Debug Tools and Scripts for the VCP \(aka APS\) and EBS Applications"](#), steps. #7 and #11. #11 is very useful for setting up a trace for Refresh Collection Snapshots on the EBS source when it is launched by

data collections from the VCP destination. You can download guided navigation for both sections to better understand the steps involved.

Refresh Collection Snapshots or Planning Data Pull Errors

1. For errors in Refresh Collection Snapshots or Planning Data Pull, check that all setup requests have completed successfully.
2. Review Appendix C.3: Requests, page B-7 and check the Request Collection Snapshots log file to confirm that the setup requests were actually launched and that no errors occurred. There have been situations where the errors are not reported back to the main request.
3. Check the profile MSC: Source Setup Required.
 - If it is set to 'Yes', then the setup requests did not complete successfully when Refresh Snapshots was launched.
 - If it is set to 'No', then change to 'Yes' and run Refresh Collection Snapshots as a standalone request one more time. If there were problems setting up the objects the first time, then this will likely resolve the issue.

ODS Load Errors

1. If a timeout error is present in the ODS Loader worker log file, then run the request Planning Data Collection -- Purge Staging Tables with Validation parameter set to 'No'.
2. Run Gather Schema Statistics on MSC Schema. In a distributed installation, run on the VCP Destination.
3. Launch Standard Data Collections with increased timeout of 600 or higher. You may need as high as 1400 for the first run with lots of new data.
4. Set the Planning Data Pull parameter Analyze Staging Tables to 'Yes' to help with new data loads into empty tables.
5. After the successful completion, run Gather Schema Statistics on the MSC schema one more time.

Refresh Snapshot Process Doesn't Start Because of Concurrent Manager Unavailability

When launching Collections, the Planning Data Pull fails with the following error:

"Refresh Snapshot process could not get started after waiting for X minutes because of Concurrent Manager Unavailability". This can happen in a centralized instance when dblink information has been populated in msc_apps_instances inadvertently. When dblink information is populated, the Planning Data Pull tries to launch the refresh collection snapshot using the dblink information available, but it fails to find a connected instance, and finally times out.

To resolve, run the following sql:

```
SELECT instance_id,instance_code,
nvl(a2m_dblink,'Null'),nvl(m2a_dblink,'Null')
FROM msc_apps_instances ;
```

If the above sql returns a value for a2mdbling and m2adblink other than the value 'Null', then the setup is wrong, and you need to do the following:

1. Delete the records from the table MRP_AP_APPS_INSTANCES_ALL as follows:

```
DELETE
FROM mrp_ap_apps_instances_all
WHERE instance_code = '&instance_code';
Commit;
```

2. Go into the Advanced Supply Chain Planning Administrator responsibility.
3. Navigate to Admin > Instances.
4. Remove the dblink information. This will change the record in msc_apps_instances, mrp_ap_apps_instances_all and mrp_ap_apps_instance.
5. Confirm whether the dblink information has correctly populated as Null in the msc_apps_instances using the following sql:

```
SELECT instance_id,instance_code,
nvl (a2m_dblink,'Null'),nvl (m2a_dblink,'Null')
FROM msc_apps_instances;
```

For more information about dblinks, see Creating Links, page 4-6.

Data Collection Fails with Several Messages

If the Planning Data Pull concurrent program fails, review the logs produced by the planning data pull process. If the error message is among the list below, follow the correct action.

- Another Data Pull process is running.
- Staging tables data is not collected. Please submit the request 'Planning ODS Load'.
- Another Planning ODS Load process is running.
- Staging tables data is being purged.

On the destination instance, using SQL*Plus, make sure that column ST_STATUS

within MSC_APPS_INSTANCES equals zero for the instance being collected. At the destination instance, issue the following from within SQL*Plus:

```
select instance_code, st_status
from msc_apps_instances;

INS      ST_STATUS
-----  -----
crm          0
vis          3
```

MSC_APPS_INSTANCES contains the ST_STATUS column. This column is used by the collection process. The value of ST_STATUS is used to determine the stage of collection for the desired instance. The possible values are:

- (0) Staging tables are empty for the particular instance. No instance data exists. This is the normal state of the instance after a successful data collection. This column needs to be 0 for a data collection to execute for the instance.
- (1) Planning Data is being pulled from the source snapshots (complete refresh or net change) by the Planning Data Pull process. If set to 1, the following error will be in the Planning Data Pull Log file, 'Another Data Pull process is running'.
- (2) Data has already been pulled from the source snapshots and is ready to be collected by the Planning ODS load process. If set to 2, the following error will be in the Planning Data Pull Log file, 'Staging tables' data is not collected'. Please submit the request Planning ODS Load.
- (3) Data is being validated in the staging tables and then transformed from the staging tables and loaded to the Operational Data Store by the Planning ODS Load process. If set to 3, the following error will be in the Planning Data Pull Log file, 'Another Planning ODS Load process is running'.
- (4) Data is being purged from the staging tables. If set to 4, the following error will be in the Planning Data Pull Log file, 'Staging tables' data is being purged'.
- (5) There are processes running that will interfere with your request. Please investigate using the query:

```
select instance_code, st_status
from msc_apps_instances
```

If this has happened after an abnormal termination of the collection process, use the following SQL to reset and then rerun:

```
update msc_apps_instances set st_status=0 where instance_id=<>;
commit;
```

After all the collection tasks are successfully complete, the Planning Data Pull program updates ST_STATUS to 0. This indicates that the instance is ready for another collection. In the SQL example above, the crm ST_STATUS is set to 0 and is ready for another collection. The vis instance ST_STATUS is set to 3 and is currently executing the planning data pull. The crm instance is ready for collection. If the collection process is not running and the ST_STATUS column is > 0, reset the ST_STATUS columns, purge the staging tables and resubmit the collection.

Solution:

1. Navigate to Advanced Supply Chain Planner > Other > Request > Submit A New Request (even if ASCP is not installed).
2. Submit the concurrent request 'Purge Staging Tables' at the destination instance.
3. Uncheck validation to ensure this process completes without a warning.
4. Run the data collection.
 - Set Recalculate Sourcing History = No.
 - Set Recalculate NRA = No unless required.

Missing Data Collection Entities

Many times customers face issues where certain entities are not collected into the VCP destination instance. Refer to My Oracle Support Note 558477.1 -- Troubleshooting Missing Data Collection Entities for diagnostic information to track the various data collection entities from the EBS source tables to the VCP destination tables. This note contains scripts that provide information from the related EBS table, snapshot, synonym, planning view, staging table, and planning table. If data is not collected into the planning table, you can trace back through the output to see where the data went missing. For example, the problem could be that the data was in the snapshot, but not in the planning view. Investigation would then be needed to understand why the view did not pick up the data. Or maybe the issue is that the data from the EBS source table is not in the snapshot. In this case, investigation would be needed to understand why the Refresh Collection Snapshot program did not update the corresponding snapshot.

Tracing

Data collection errors should be level 4 traces with binds; performance issues should be level 8 traces with waits. For information about how to set up trace requests to get details on a performance issue, see My Oracle Support Note 245974.1 -- "FAQ -- How to Use Debug Tools and Scripts for the APS Suite".

Other Troubleshooting Resources

For additional assistance with troubleshooting, please see the following My Oracle Support notes:

Note Number	Title
415135.1	How To Create APS Related Snapshots And mLogs
211121.1	How to Run MSRFWOR - Refresh Collections Snapshots Concurrent Request from the Application
550005.1	How to Improve the Performance of the Refresh Collection Snapshots When Running Very High Volumes
1400555.2	Information Center: Data Collections Value Chain Planning Advanced Supply Chain Planning

See also:

- https://communities.oracle.com/portal/server.pt/community/value_chain_planning/321, the My Oracle Support Community for Value Chain Planning. This support board provides you with access to product and support experts, popular documents and past discussions.

Data Field Notes

Calendar: Oracle recommends that you load calendars separate from all other entities and before all other entities. When you collect all other entities, set Calendar selection to No..

Currency: Currencies collected from Oracle Order Management and Oracle Purchasing are posted in functional currency even if their source is in transactional currency.

Currency Conversion Rates: Collect currency and currency conversion rates for a past, current and future period from the source in order to support displaying the cost in reporting currency. The default value is No. Currency conversion rates are collected only:

- If profile option MSC: Planning Hub Currency Code is Yes
- For standard collections

See profile options:

- MSC: Currency Conversion Type

- MSC: Planning Hub Currency Code
- MSC: History Days for Currency Conversion
- MSC: Future Days for Currency Conversion

Discrete Jobs: A discrete job that is complete and not closed appears in the collected data with quantity zero. A discrete job that is complete and closed does not appear in the collected data. You can control this with profile option MSC: Collect Completed Jobs

Drop ship purchase orders: The collections process does not collect supplies against drop ship purchase orders because the planning engine does not plan drop ship sales orders.

End-Item Substitutes: You can see end-item substitutes in Collections Workbench as long as you have not defined a substitute set. If you have defined a substitute set, use Planner Workbench Items window to see the substitute set and end-item substitute.

Global Forecasts: You can review global forecast entries in Collections Workbench horizontal plan using rows Original and Cumulative Original.

Inactive forecasts: The collections process does not collect the demands of inactive forecasts if run in Full Refresh or Targeted Refresh collections mode

Intransit, Purchase Order, and Requisition: Oracle recommends that you always collect purchase order, requisition and intransit together. To do so, set Planning Data Pull parameter Purchase Orders/Purchase Requisitions to Yes.

Routings: The collections process does not collect routing operation resources with Schedule = No and does not display them in their routings. It also does not collect alternate resources of primary resources that are Schedule = No, even if the alternate resources themselves are Schedule = Yes.

Trading Partners: When loading trading partners, access file Parameters.txt and set Calendar_Overwrite_Flag to N.

Legacy Collections Data Notes

OWNING_PARTNER_SITE_CODE: Organization code

OWNING_TP_TYPE: Valid values are:

- 1: Supplier
- 2: Inventory Organization

PLANNING_PARTNER_SITE_CODE: Organization code

PLANNING_TP_TYPE: Valid values are:

- 1: Supplier
- 2: Inventory Organization

Resource Balance Flag: Indicates whether a resource is load balanced. Valid values are:

- 1: Yes
- 2: No

This flag is only for Oracle Process Manufacturing. Since you cannot use legacy collections with Oracle Process Manufacturing, always leave this field null.

Unit of Measure: Load all base unit of measure conversions without an item name. This creates rows in MSC_UOM_CONVERSIONS with inventory_item_id = 0, for example:

- Base UOM: LB
- Standard conversions: KG - LB, Gram - LB, Kiloton - LB, mg - LB, Oz - LB, Ton - LB,

For specific conversions across UOM Class or specific intra-class unit of measure conversions for some items, load them using the item name

Resetting Item Attribute Values

To reset item attribute values:

- Do not use a NULL string
- Instead, use form Item Attributes Mass Maintenance, section Item attributes available for update, condition Set Original Value

A

Parameters

This appendix covers the following topics:

- Overview
- Planning Data Pull Parameters
- Planning ODS Load Parameters
- Legacy Collections

Overview

There are three types of parameters that determine the collection of data. They include:

- Planning Data Pull Parameters
- Planning ODS Load Parameters
- Legacy Collection Parameters

These parameters are set when running collections, depending on the type of collection (Standard, Continuous or Legacy).

For more details about parameters, please see My Oracle Support Note #179522.1 - "Data Collection Parameters".

For more information about running collections, please see Running Collections, page 8-1.

Note: If you are running Standard Collections in Complete Refresh mode, then setting No will remove all data for the entity. Use Targeted Collection if you require to collect only certain data entities and are not going to use default values populated for Complete Refresh.

Planning Data Pull Parameters

Parameter	Values	Available for Net Change Refresh?
Instance	Source instance code from list of values.	
Collection Group	Collection group, set up in the Admin/Instances form in the Organization screen.	
	<p>Important: If you have very large BOM tables and plan to use collection groups to collect information for only certain orgs, be aware that this can use very significant TEMP space during the ODS Load. Oracle has seen reports where all orgs takes only 6-12 Gb of TEMP space, but when using a collection group, the TEMP space requirements exceed 30 Gb.</p>	
Number of Workers	One or greater. Increase this number to increase the amount of computational resources to devoted to the Planning Data Pull process. This allows you to specify the number of workers for the Data Pull, which can now be different from the number of workers specified for the ODS load process.	
	<p>It is recommended that you increase to 4-8 workers to improve performance. The default is 2.</p>	
Timeout (Minutes)	The maximum amount of time you would like to allocate to the Planning Data Pull process. If the Planning Data Pull process has not completed within this amount of time, it will be terminated with an error.	
	<p>For the initial run on a new instance, this may need to be set to 600-900. The default is 180.</p>	
Language	Default language.	

Parameter	Values	Available for Net Change Refresh?
Purge Previously Collected Data	Yes (default) or No. Setting this to Yes wipes out all data in the APS planning server operational data store associated with the selected source instance as the first step in the collections process. If you set this to Yes, the only allowable collection method is Complete Refresh. If you set this to No, the allowable collection methods are Targeted Replacement and Net Change.	
Collection Method	Complete Refresh/Targeted Refresh/Net Change Refresh.	
Analyze Staging Tables	Yes or No (default). Set this to Yes periodically to recompute database access statistics on the APS staging tables. This speeds up the subsequent Planning ODS Load process. Should only set to Yes if directed by Development.	
Approved Supplier Lists	Set to Yes if you want to collect Approved Supplier Lists.	Y
ATP Rules	Set to Yes if you want to collect ATP Rules.	
Bills of Materials/Routings/Resources	Set to Yes if you want to collect Bills of Materials/Routings/Resources.	Y
Bills of Resources	Set to Yes if you want to collect Bills of Resources.	Y
Calendars	Set to Yes if you want to collect Calendars and Calendar Assignments.	
Demand Class	Set to Yes if you want to collect the Demand Class.	
End Item Substitutions	Set to Yes if you want to collect the End Item Substitutions.	
Forecasts	Set to Yes if you want to collect Forecasts.	Y

Parameter	Values	Available for Net Change Refresh?
Items	Set to Yes if you want to collect Items.	Y
Key Performance Indicator Targets	Set to Yes if you want to collect Key Performance Indicator Targets.	
Master Demand Schedules	Set to Yes if you want to collect Master Demand Schedules.	Y
Master Production Schedules	Set to Yes if you want to collect Master Production Schedules. The default is No.	Y
On Hand	Set to Yes if you want to collect On Hand amounts.	Y
Planning Parameters	Set to Yes if you want to collect Planning Parameters.	
Planners	Set to Yes if you want to collect Planners.	
PO Receipts	Set to Yes if you want to collect PO Receipts.	Y
Projects/Tasks	Set to Yes to collect Projects/Tasks.	
Purchase Orders/Purchase Requisitions	Set to Yes to collect Purchase Orders/Purchase Requisitions.	Y
Reservations	Set to Yes to collect Reservations.	Y

Parameter	Values	Available for Net Change Refresh?
Resources Availability	<p>Valid Values:</p> <ul style="list-style-type: none"> • Regenerate and Collect Data. In this case, Regenerate and Collect Data will collect OPM data and discrete data. Default. • Do Not Collect Data. <p>Note: Profile option MRP: Cutoff Data Offset Months is used for Discrete Orgs only. It determines how many months of resource availability should be computed for resources and simulation sets. No resource availability is calculated beyond this horizon. For constrained plans beyond this horizon, resource capacity is viewed as infinite. Set this value to your plan horizon. A large setting (greater than 12 months) could have performance implications during ODS Load when the resource availability is calculated for each resource for each day.</p>	<p>Net change - discrete organizations.</p> <p>OPM organizations require Complete Refresh.</p>
Safety Stock	Set to Yes to collect Safety Stock. Only for User Defined Safety Stock from the Inventory application.	

Parameter	Values	Available for Net Change Refresh?
Sales Orders	<p>Set to Yes to collect Sales Orders.</p> <p>Note: Sales Orders are collected in the Net Change Mode for performance reasons and to reduce the ATP downtime during Complete Refresh.</p>	Y
Sourcing History	<p>Not used anymore. See ODS Load parameter: Recalculate Sourcing History.</p>	
Sourcing Rules	<p>Set to Yes to collect sourcing rules. BODs, Assignment Sets, Region to Site Mapping are also triggered by this parameter.</p>	
Sub Inventories	<p>Set to Yes to collect Sub Inventories.</p>	
Supplier Responses	<p>Set to Yes to collect Supplier Responses. For Collaborative Planning, Net Change not supported; only Complete or Targeted supported.</p>	

Parameter	Values	Available for Net Change Refresh?
Suppliers/Customers/Orgs	<p>Set to Yes to collect Suppliers/Customers/Orgs. Collects 4 partner types to MSC_TRADING_PARTNERS and related tables. Partner_Type:</p> <ul style="list-style-type: none"> 1 - Supplier 2 - Customer 3 - Organization 4 - Buyer 	
Transportation Details	<p>Set to Yes to collect Transportation Details. Shipping Details from WSH tables.</p>	Y
Unit Numbers	<p>Set to Yes to collect Unit Numbers. These are used for Project MFG for unit effectivity (serial number) in the bill of material.</p>	
Unit of Measures (UOM)	<p>Set to Yes to collect Unit of Measure.</p>	

Parameter	Values	Available for Net Change Refresh?
User Company Association	<p>This parameter is used for Collaborative Planning Only.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • No: During data collections, the destination instance does not accept the following: New users' company association and existing users with changes to their company association. • Create users and enable user company association: A user created on one of the source instances can be automatically created on the planning server or destination instance. You can use this option when you need to work with multiple source instances or systems. Default. • Create a new user, specify the user's contact information during setup, and initiate a data collection. The user is created on the destination instance or planning server and is assigned the Supply Chain Collaboration Planner responsibility. The new user can log onto Oracle Collaborative Planning and view data that is visible to the associated company. • Create a new user, specify the user's contact information during setup, and initiate a data collection. The user is created on the destination instance or planning server and is assigned the Supply Chain Collaboration Planner responsibility. The new user can log onto Oracle Collaborative Planning and view data that is visible to the associated company. <p>Create a new user on the source instance and specify the user's contact information during the setup. Create the</p>	

Parameter	Values	Available for Net Change Refresh?
	user on the destination instance or planning system and assign the user responsibilities. Start a data collection to complete the user's company association. This ensures that the user has all the assigned responsibilities.	
User Supplies and Demand	Set to Yes if you want to collect User Supplies and Demand. Used when manually populating MTL_USER_SUPPLY and/or MTL_USER_DEMAND.	Y
Work in Process	Set to Yes if you want to collect Work in Process. Jobs that are in status Unreleased, Released, Complete and On Hold (wip_discrete_jobs.status_type) collected.	Y
Sales Channel	Set to Yes if you want to collect Sales Channel. Used for Demantra only.	
Fiscal Calendar	Set to Yes if you want to collect the Fiscal Calendar. Used for Demantra only.	
Internal Repair Orders	Set to Yes if you want to collect Internal Repair Orders. Used for Depot Repair integration - SRP/SPP.	Y
External Repair Orders	Set to Yes if you want to collect External Repair Orders. Used for Depot Repair integration - SRP/SPP.	Y
Payback Demand/Supply	Set to Yes if you want to collect Payback Demand/Supply.	
Currency Conversion	Set to Yes if you want to collect Currency Conversion. Used for Demantra only.	
Delivery Details	Set to Yes if you want to collect Delivery Details. Used for OTM integration (Oracle Transportation Management).	

Planning ODS Load Parameters

Parameter	Values
Instance	Source instance code from list of values.
Timeout (Minutes)	Number of minutes before the concurrent program will end. The default is 60. For new instances or when using Recalculate options below, increase to a higher value.
Number of Workers	One or greater. Increase this number to increase the amount of computational resources to devoted to the Planning ODS Load process. The default is 3. Increase to 4-8 workers to improve performance.

Parameter	Values
Recalculate Sourcing History	<p>The value that you set here is the one that actually determines whether sourcing history is collected or not. If you select Yes, then VCP will collect all new sourcing history not already on the planning server in the time range [(today - x months) through (today)] from the source transaction system. The number x is given by the value that you set for the profile option MSC: Sourcing History Start Date Offset (in months). During planning, ASCP will use the total cumulative sourcing history on the planning server in addition to the planned sourcing in the plan to determine whether sourcing percentages in sourcing rules are being respected or not. Used in conjunction with the Purge Sourcing History parameter below, which deletes existing sourcing history before calculating sourcing history again.</p>
Purge Sourcing History	<p>For distributed installations, set this profile option on the VCP Destination. Offsets the starting date of sourcing history calculations and is used in conjunction with the Purge Sourcing History parameter below.</p>
	<p>Defaults to Null which Collects All Data. Set to prevent performance problems when the parameter is set to Yes. System performance can be degraded when there is a high value for this profile option and a high volume of source transaction data.</p>
	<p>Valid values are Yes and No (default). If you select Yes, then all sourcing history present on the planning server will be deleted before the collection process commences. All previous history is removed in the base table.</p>
	<p>If Sourcing History needs to be removed for business reasons, then you can set this to Yes, and Recalculate Sourcing History to No. This effectively removes all Sourcing History on the system.</p>

Parameter	Values
Refresh Collected Data in APCC Repository	<p>This parameter is used when you want to refresh the collected data in the APCC repository.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • Yes • No (default)

Legacy Collections

Parameter	Values
Purge Legacy Data	Runs MSCPPURB.pls - MSC_PURGE_LID
Purge Interface Tables	The MSC_SRS_INSTANCE_CODE_ODS is used to get the instance code. Runs MSCCLPSB.pls - MSC_CL_PURGE_STAGING

Concurrent Programs

This appendix covers the following topics:

- Overview
- Collection Requests
- Setup Requests
- Running Refresh Collection Snapshots as a Standalone Concurrent Request
- Performance or Errors with the Refresh Collection Snapshots Program

Overview

The Collections process uses a number of concurrent programs (requests) to refresh snapshots, pull planning data into staging tables on the planning server, and load data into the ODS. The parameters set will determine the requests that will be called during the collection process, and their order. You can view the process of the concurrent requests from the System Administration responsibility by choosing Requests > Monitor. Requests can also be scheduled from Requests > Schedule.

If your data needs considerable modification and transformation, you can run requests individually instead of running the full Collections process so that you can adjust the data at various stages. Requests can be run individually from the System Administrator responsibility by choosing Requests > Run.

Collection Requests

The following requests are involved in the collection process. For more information about collection requests, please see the listing of Advanced Supply Chain Planning Concurrent Programs at <http://oracleappscommunity.com/oracle/blog/1121/advanced-supply-chain-planning-concurrent-programs>.

Request	Description
Collections Synonyms	This concurrent process drops and creates synonyms during the collections run.
Collections Triggers	This concurrent process drops and creates triggers during the collections run.
Collections Views	This concurrent process drops and creates views during the collections run.
Create AHL Snapshots	This concurrent process drops and recreates AHL snapshots during the collection process. Typically, it runs as part of concurrent process Refresh Collection Snapshots when profile option MSC: Source set up required is Yes.
Create BOM Snapshots	This concurrent process drops and recreates BOM snapshots during the collection process. Typically, it runs as part of concurrent process Refresh Collection Snapshots when profile option MSC: Source set up required is Yes.
Create Collections BOM Snapshot Triggers	This concurrent process drops and creates BOM snapshot triggers during the collections run.
Create Collections BOM Views	This concurrent process drops and creates BOM views during the collections run.
Create Collections Demand Snapshot Triggers	This concurrent process drops and creates demand snapshot triggers during the collections run.
Create Collections Demand Views	This concurrent process drops and creates demand views during the collections run.
Create Collections Item Snapshot Triggers	This concurrent process drops and creates item snapshot triggers during the collections run.
Create Collections Item Views	This concurrent process drops and creates item views during the collections run.

Request	Description
Create Collections Other Snapshot Triggers	This concurrent process drops and creates other snapshot triggers during the collections run.
Create Collections Other Views	This concurrent process drops and creates other views during the collections run.
Create Collections Repair Order Snapshot Triggers	This concurrent process drops and creates repair order snapshot triggers during the collections run.
Create Collections Repair Order Views	This concurrent process drops and creates repair order views during the collections run.
Create Collections Routing Snapshot Triggers	This concurrent process drops and creates routing snapshot triggers during the collections run.
Create Collections Routing Views	This concurrent process drops and creates routing views during the collections run.
Create Collections Setup Views	This concurrent process drops and creates setup views during the collections run.
Create Collections Supply Snapshot Triggers	This concurrent process drops and creates supply snapshot triggers during the collections run.
Create Collections Supply Views	This concurrent process drops and creates supply views during the collections run.
Create Collections WIP Snapshot Triggers	This concurrent process drops and creates WIP snapshot triggers during the collections run.
Create Collections WIP Views	This concurrent process drops and creates WIP views during the collections run.
Create CSP Snapshots	This concurrent process drops and recreates CSP snapshots during the collection process. Typically, it runs as part of concurrent process Refresh Collection Snapshots when profile option MSC: Source set up required is Yes.

Request	Description
Create EAM Snapshots	<p>This concurrent process drops and recreates EAM snapshots during the collection process. Typically, it runs as part of concurrent process Refresh Collection Snapshots when profile option MSC: Source set up required is Yes.</p>
Create INV Snapshots	<p>This concurrent process creates new segment definitions in existing descriptive flexfields to hold data that may be required for inventory planning. It also populates profile values with the value corresponding to the descriptive flexfield attribute number for each attribute created.</p>
	<p>Parameters:</p> <ul style="list-style-type: none"> • Service Level (Item): The service level value that you want to maintain for the items. • Service Level (Org): The service level value that you want to maintain for the organizations. • Service Level (Customers): The service level value that you want to maintain for the customers.
Create MRP Snapshots	<p>This concurrent process drops and recreates MRP snapshots during the collection process. Typically, it runs as part of concurrent process Refresh Collection Snapshots when profile option MSC: Source set up required is Yes.</p>
Create OE Snapshots	<p>This concurrent process drops and recreates OE snapshots during the collection process. Typically, it runs as part of concurrent process Refresh Collection Snapshots when profile option MSC: Source set up required is Yes.</p>
Create PO Snapshots	<p>This concurrent process drops and recreates PO snapshots during the collection process. Typically, it runs as part of concurrent process Refresh Collection Snapshots when profile option MSC: Source set up required is Yes.</p>

Request	Description
Create WIP Snapshots	This concurrent process drops and recreates WIP snapshots during the collection process. Typically, it runs as part of concurrent process Refresh Collection Snapshots when profile option MSC: Source set up required is Yes.
Create WSH Snapshots	This concurrent process drops and recreates WSH snapshots during the collection process. Typically, it runs as part of concurrent process Refresh Collection Snapshots when profile option MSC: Source set up required is Yes.
Create WSM Snapshots	This concurrent process drops and recreates WSM snapshots during the collection process. Typically, it runs as part of concurrent process Refresh Collection Snapshots when profile option MSC: Source set up required is Yes.
Drop Changed Snapshots	This concurrent process drops the source snapshot definitions. Use it when there is an error or corruption in a particular snapshot or materialized view. The next collection recreates the snapshots. Parameter: Snapshot Name: The name of the snapshot to drop.
Planning Data Collection - Purge Staging Tables	<p>This concurrent process deletes the data from the staging tables. Use it either before regenerating and launching a new data pull or after the collections data pull fails.</p> <p>Parameters:</p> <ul style="list-style-type: none"> Instance: the instance-organization against which you want to run the concurrent process. Validation: Specify whether the concurrent process needs to verify if a planning data collection is running.

Request	Description
Planning Data Pull	This concurrent process pulls Oracle Applications data from the source (transaction) instance to the destination (planning) instance interface tables. See Planning Data Pull Parameters, page A-2 for more information,
Planning ODS Load	This concurrent process moves data from the destination (planning) instance interface tables to the destination (planning) instance planning tables. See Planning ODS Load Parameters, page A-10 for more information.
Refresh Collected Data in APCC Repository	This concurrent program refreshes the APCC repository. Parameters include instance (required) and Complete (Yes/No). The data range of the collected data available to APCC depends on the following profile options: MSC: Horizon for APCC Collected data - Forward days and MSC: Horizon for APCC Collected data - Backward days.

Request	Description
Refresh Collect Snapshots	<p>This concurrent request refreshes the snapshot in the source. After the refresh, the snapshot reflects the most current master data. If profile option MSC: Source set up required is Yes, this process runs the following concurrent processes:</p> <ul style="list-style-type: none"> • Create AHL Snapshots • Create BOM Snapshots • Create INV Snapshots • Create MRP Snapshots • Create OE Snapshots • Create PO Snapshots • Create WIP Snapshots • Create WSH Snapshots • Create WSM Snapshots

Setup Requests

The Request ID and Request Name are listed in the Refresh Collection Snapshots log file when these requests are launched as part of Data Collections.

Shortname	Request Name
MSCDROPS	Drop Changed Snapshots
MSCWSMSN	Create WSM Snapshots
MSCBOMSN	Create BOM Snapshots
MSCINVSN	Create INV Snapshots

Shortname	Request Name
MSCCSPSN	Create CSP Snapshots
MSCMRPSN	Create MRP Snapshots
MSCPOXSN	Create PO Snapshots
MSCONTSN	Create OE Snapshots
MSCWSHSN	Create WSH Snapshots
MSCAHLN	Create AHL Snapshots
MSCEAMSN	Create EAM Snapshots
MSCWIPSN	Create WIP Snapshots
MSCSYNMS	Collections Synonyms
MSCVIEWS	Collections Views
MSCTRIGS	Collections Triggers
MSCTRITM	Create Collections Item Snapshot Triggers
MSCTRBO	Create Collections BOM Snapshot Triggers
MSCTRRTG	Create Collections Routing Snapshot Triggers
MSCTRWIP	Create Collections WIP Snapshot Triggers
MSCTRDEM	Create Collections Demand Snapshot Triggers
MSCTRSUP	Create Collections Supply Snapshot Triggers
MSCTROTH	Create Collections Other Snapshot Triggers
MSCTRRPO	Create Collections Repair Order Snapshot Triggers
MSCVWSTP	Create Collections Setup Views

Shortname	Request Name
MSCVWITM	Create Collections Item Views
MSCVWBOM	Create Collections BOM Views
MSCVWRTG	Create Collections Routing Views
MSCVWWIP	Create Collections WIP Views
MSCVWDEM	Create Collections Demand Views
MSCVWSUP	Create Collections Supply Views
MSCVWOTH	Create Collections Other Views
MSCVWRPO	Create Collections Repair Order Views

Running Refresh Collection Snapshots as a Standalone Concurrent Request

Running a complete collection will launch the refresh collection snapshots concurrent request automatically, there are occasions where you might want to run the request on its own. For example, you can run the refresh collection snapshots concurrent request prior to running a complete collection. This reduces the time involved with running a complete data collection because all the snapshots are already up-to-date.

The Refresh Collection Snapshots request is available under a number of responsibilities:

- Material Planner
- Advanced Supply Chain Planner

In addition, using the System Administrator responsibility, it can be added to the Request Group OM Concurrent Programs so that it is visible in the Order Management responsibility where customers may run ATP data collections.

For distributed installation with separate instances for transactions (EBS source) and planning (VCP destination), the Refresh Collection Snapshots is always run on the EBS source instance.

To run the Refresh Collection Snapshots request:

1. Using the Advanced Supply Chain Planner responsibility, navigate to Setup, then

Run Requests.

2. Select the relevant organization and click OK.
3. In the Submit a New Request form, choose Single Request and click OK.
4. In the Submit Request form, Name field, enter Refresh Collection Snapshots.
5. In the Parameters form, fill in the following parameters:

Parameter	Description
Refresh Mode	<p>Valid values:</p> <ul style="list-style-type: none">• Fast Refresh: only processes changes since the last run of the Refresh Snapshots or data collections.• Complete Refresh: processes all the snapshots and completely refreshes the snapshots.• Automatic Refresh: explicitly truncates the snapshot log when you set the parameter Threshold = 0 and perform a complete refresh of the snapshots. The automatic refresh does not check the profile MSC: Source Setup Required, so if you are running for the first time after a patch or in a newly cloned instance, then use Complete Refresh.

Note: When running Data Collections request set, Refresh Snapshots always uses a Fast Refresh for performance reasons. This is usually not a problem, unless the snapshot logs have grown very large. Then corrective steps are required. See Managing the MLOG Growth for VCP Data Collections, page 11-7 for more information.

Parameter	Description
Snapshot Name	<p>Use default "All Snapshots" or choose a snapshot from the list.</p> <p>Important: When running as a standalone request, Oracle recommends that you run this for each snapshot <i>as required</i>. When launched for All Snapshots, then this can require a lot of UNDO (or Rollback) space since all snapshots and transactions are performed in one atomic transaction.</p> <p>For list of snapshots for your release see Appendix D -- Snapshots, page C-1.</p>
Threshold for Truncating Logs	<p>Used only in Automatic Refresh mode.</p> <p>Use Threshold = 0. This truncates the MLOG\$ Snapshot log table, which can help resolve performance issues.</p> <p>Oracle has no reason to recommend any other setting than zero at this time.</p> <p>For example, a value of 50 truncates the log if 50 or more lines are in the MLOG\$ snapshot log table and then performs a complete refresh of the snapshot.</p>

Parameter	Description
Degree	Refers to the degree of parallelism to be used by the database to process the request. Maximum value is 10. The higher the degree of parallelism, the faster the process will complete. Currently Oracle recommends the default value of 0.
	Ensure that you have followed all the steps for setting up the latest RDBMS patches. If the RDBMS patches are applied, then setting this value higher should not cause a problem.
	Note: Entering a number higher than 10 will result in the use of 10 for this value. Oracle's Support has seen RDBMS issues when this is higher than 1.

Note: See Running Collections for the First Time in a New Instance, page 8-9 for more information about how the MSC: Source Setup Required = Yes parameter affects the Refresh Collect Snapshots concurrent program.

Performance or Errors with the Refresh Collection Snapshots Program

The following performance issues or errors may be experienced with the Refresh Collection Snapshots program run either as standalone or part of the Data Collection request set.

- Review MLOG Growth, page 11-3 for comprehensive steps to fix and/or prevent performance issues in Refresh Collection Snapshots.
- Errors with tables or tablespace (ORA-1654, ORA-1653, ORA-1652 ORA-1655, ORA-1651), then you will need to add space to the system. See Managing Tablespaces, page 4-4 for information about managing space requirements for our applications.
- Snapshot errors (ORA-12008, ORA-12057): run Automatic refresh for the individual snapshot listed in the log file to attempt to correct the problem. If the error occurs for BOM, WIP, MTL, or WSH, then run for each of those snapshots. This should

help prevent the error from occurring on the next run.

ORA-12008 or ORA-12057 may be followed by a undo/rollback error (ORA-1555, ORA-30036). A lot of rollback is required for large data sets. Try adding more space to the rollback. Or if the rollback (undo) space is already very large (≥ 20 GB), then run the Refresh Snapshots for each snapshot individually using Automatic Refresh. My Oracle Support Note 266797.1 -- "Rollback Segment Assignment and Automatic Undo Management Mode for MRP and APS Applications ORA-01562 ORA-01555" may also be helpful with information about how to overcome these errors and how to set up rollback.

C

Snapshots

This appendix covers the following topics:

- Overview
- List of Snapshots
- Invalid Snapshots and Manually Refreshing the Snapshot

Overview

During the Refresh Collection Snapshots request, run during the Collections process, the snapshots for all the selected data objects are updated before the data is pulled to the VCP Planning Server.

The following SQL reveals the list of snapshots that are available in the release on the system.

```
select owner, name, master from all_snapshots
where name like '%SN'
and name not in ('MSC_ATP_PLAN_SN', 'WSH_DELIVERY_DATES_SN')
order by name;
```

This list changes with each release or may change depending on the level of code in the release. See My Oracle Support Note 179522.1 -- "Data Collection Use and Parameters" for the list of snapshots being maintained for each release.

List of Snapshots

List of snapshots for Value Chain Planning Applications. This list can be generated by SQL #22 -- Show MLOGs and Related Snapshots and Base Tables for Data Collections, page D-16.

In addition, the list includes snapshots that share the same MLOG as Data Collections snapshots. These other snapshots cannot be truncated using automatic refresh and must be truncated and refreshed manually. Marked by an asterisk (*) in the table, these include:

- ENI - Product Intelligence
- OZF - Trade Management

Multiple rows for a snapshot mean a complex snapshot where the query is for the multiple base tables and MLOGs shown in the table.

The snapshots are as follows:

Snapshot	Snapshot Owner	Base Table Name	Base Table Owner	MLOG\$
AHL_SCH_MTL_S_SN	APPS	AHL_SCHEDULER_E_MATERIALS	AHL	MLOG\$_AHL_SCHEDULE_MATERIALS
BOM_BOMS_SN	APPS	BOM_STRUCTURES_B	BOM	MLOG\$_BOM_STRUCTURES_B
BOM_CTO_ORDER_DMD_SN	BOM	BOM_CTO_ORDER_DEMAND	BOM	MLOG\$_BOM_CTO_ORDER_DEMAND
BOM_INV_CO_MPS_SN	APPS	BOM_COMPONENTS_B	BOM	MLOG\$_BOM_COMPONENTS_B
		BOM_STRUCTURES_B		MLOG\$_BOM_STRUCTURES_B
BOM_OPR_NET_WORKS_SN	APPS	BOM_OPERATION_SEQUENCE	BOM	MLOG\$_BOM_OPERATION_SEQUENCE
		BOM_OPERATIONAL_ROUTINES		MLOG\$_BOM_OPERATIONAL_ROUTINES
		BOM_OPERATION_NETWORKS		MLOG\$_BOM_OPERATION_NETWORKS

Snapshot	Snapshot Owner	Base Table Name	Base Table Owner	MLOG\$
BOM_OPR_RES_S_SN		BOM_OPERATI ON_SEQUENCE S		MLOG\$_BOM_ OPERATION_SE QUEN
		BOM_OPERATI ON_RESOURCE S		MLOG\$_BOM_ OPERATION_R ESOUR
		BOM_OPERATI ONAL_ROUTIN GS		MLOG\$_BOM_ OPERATIONAL _ROUT
BOM_OPR_RTN_S_SN	APPS	BOM_OPERATI ONAL_ROUTIN GS	BOM	MLOG\$_BOM_ OPERATIONAL _ROUT
BOM_OPR_SEQ_S_SN	APPS	BOM_OPERATI ON_SEQUENCE S	BOM	MLOG\$_BOM_ OPERATION_SE QUEN
		BOM_OPERATI ONAL_ROUTIN GS		MLOG\$_BOM_ OPERATIONAL _ROUT
BOM_RES_CHN_GS_SN	APPS	BOM_RESOURC E_CHANGES BOM_DEPART MENTS	BOM	MLOG\$_BOM_R ESOURCE_CHA NGES
				MLOG\$_BOM_ DEPARTMENTS
BOM_RES_INST_CHNGS_SN	APPS	BOM_DEPART MENTS	BOM	MLOG\$_BOM_ DEPARTMENTS
		BOM_RES_INST ANCE_CHANG ES		MLOG\$_BOM_R ES_INSTANCE_ CHA

Snapshot	Snapshot Owner	Base Table Name	Base Table Owner	MLOG\$
BOM_SUB_CO MPS_SN	APPS	BOM_STRUCTU RES_B BOM_COMPON ENTS_B BOM_SUBSTITU TE_COMPONE NTS	BOM BOM BOM	MLOG\$_BOM_S TRUCTURES_B MLOG\$_BOM_C OMPONENTS_B MLOG\$_BOM_S UBSTITUTE_CO MPO
BOM_SUB_OPR _RESS_SN	APPS	BOM_OPERATI ON_SEQUENCE BOM_SUB_OPE RATION_RESO URCES BOM_OPERATI ONAL_ROUTIN GS	BOM BOM BOM	MLOG\$_BOM_ OPERATION_SE QUEN MLOG\$_BOM_S UB_OPERATIO N_RE MLOG\$_BOM_ OPERATIONAL _ROUT
CSP_REPAIR_P O_HEADERS_S N	APPS	CSP_REPAIR_P O_HEADERS	CSP	MLOG\$_CSP_R EPAIR_PO_HEA DER
EAM_WO_REL ATIONSHIPS_S N	APPS	EAM_WO_REL ATIONSHIPS	EAM	MLOG\$_EAM_ WO_RELATION SHIPS
ENI_DBI_BOM_ COMPONENTS _MV1*	APPS	BOM_COMPON ENTS_B	BOM	MLOG\$_BOM_C OMPONENTS_B
ENI_DBI_BOM_ COMPONENTS _MV2*	APPS	BOM_COMPON ENTS_B	BOM	MLOG\$_BOM_C OMPONENTS_B
ENI_DBI_MFG_ STEPS_JOIN_M V*	APPS	BOM_OPERATI ONAL_ROUTIN GS	BOM	MLOG\$_BOM_ OPERATIONAL _ROUT

Snapshot	Snapshot Owner	Base Table Name	Base Table Owner	MLOG\$
ENI_RES_1_MV*	APPS	BOM_DEPARTMENTS	BOM	MLOG\$_BOM_DEPARTMENTS
MRP_FORECAST_DATES_SN	APPS	MRP_FORECAST_DATES	MRP	MLOG\$_MRP_FORECAST_DATES
MRP_FORECAST_DSGN_SN	APPS	MRP_FORECAST_DESIGNATORS	MRP	MLOG\$_MRP_FORECAST_DESIGNATORS
MRP_FORECAST_ITEMS_SN	APPS	MRP_FORECAST_ITEMS	MRP	MLOG\$_MRP_FORECAST_ITEMS
MRP_SCHEDULE_DATES_SN	APPS	MRP_SCHEDULE_DATES	MRP	MLOG\$_MRP_SCHEDULE_DATES
MTL_DEMAND_SN	APPS	MTL_DEMAND	INV	MLOG\$_MTL_DEMAND
MTL_ITEM_CATEGORIES_SN	APPS	MTL_ITEM_CATEGORIES	INV	MLOG\$_MTL_ITEM_CATEGORIES
MTL_ITEM_RELATIONSHIPS_SN		MTL RELATED ITEMS		MLOG\$_MTL RELATED ITEMS
MTL_MTRX_TM_P_SN	APPS	MTL_MATERIAL_TRANSACTONS_TEMP	INV	MLOG\$_MTL_MATERIAL_TRANSAC
MTL_OH_QTYS_SN	APPS	MTL_ONHAND_QUANTITIES_DETAIL	INV	MLOG\$_MTL_ONHAND_QUANTITIES
MTL_RESERVATIONS_SN	APPS	MTL_RESERVATIONS	INV	MLOG\$_MTL_RESERVATIONS

Snapshot	Snapshot Owner	Base Table Name	Base Table Owner	MLOG\$
MTL_SUPPLY_SN	APPS	MTL_SUPPLY	INV	MLOG\$_MTL_SUPPLY
MTL_SYS_ITEM_SN	APPS	MTL_SYSTEM_ITEMS_B	INV	MLOG\$_MTL_SYSTEM_ITEMS_B
MTL_TXN_REQ_SN	APPS	MTL_TXN_REQ_LINES	INV	MLOG\$_MTL_TXN_REQUEST_LINES
MTL_U_DEMAND_SN	APPS	MTL_USER_DEMAND	INV	MLOG\$_MTL_USER_DEMAND
MTL_U_SUPPLY_SN	APPS	MTL_USER_SUPPLY	INV	MLOG\$_MTL_USER_SUPPLY
OE_ODR_LINES_SN	APPS	OE_ORDER_LINES_ALL	ONT	MLOG\$_OE_ORDER_LINES_ALL
OZF_EARNING_SUMMARY_MV*	APPS	MTL_ITEM_CATEGORIES	INV	MLOG\$_MTL_ITEM_CATEGORIES
PO_ACCEPTANCES_SN	APPS	PO_ACCEPTANCES	PO	MLOG\$_PO_ACCEPTANCES
PO_CHANGE_REQUESTS_SN	APPS	PO_CHANGE_REQUESTS	PO	MLOG\$_PO_CHANGE_REQUESTS
PO_SI_CAPA_SN	APPS	PO_SUPPLIER_ITEM_CAPACITY	PO	MLOG\$_PO_SUPPLIER_ITEM_CAPACITY
WIP_DSCR_JOB_SN	APPS	WIP_DISCRETE_JOBS	WIP	MLOG\$_WIP_DISCRETE_JOBS
WIP_FLOW_SCHEDULES_SN	APPS	WIP_FLOW_SCHEDULES	WIP	MLOG\$_WIP_FLOW_SCHEDULES

Snapshot	Snapshot Owner	Base Table Name	Base Table Owner	MLOG\$
WIP_OPR_RES_I NSTS_SN	APPS	WIP_OP_RESO URCE_INSTAN CES	WIP	MLOG\$_WIP_O P_RESOURCE_I NST
WIP_REPT_ITE MS_SN	APPS	WIP_REPEATIV E_ITEMS	WIP	MLOG\$_WIP_R EPETITIVE_ITE MS
WIP_REPT_SCH DS_SN	APPS	WIP_REPEATIV E_SCHEDULES	WIP	MLOG\$_WIP_R EPETITIVE_SCH ED
WIP_WLINES_S N	APPS	WIP_LINES	WIP	MLOG\$_WIP_LI NES
WIP_WOPRS_S N	APPS	WIP_MSC_OPE N_JOB_STATUS ES	WIP	MLOG\$_WIP_M SC_OPEN_JOB_ STA
		WIP_DISCRETE _JOBS		MLOG\$_WIP_DI SCRETE_JOBS
		WIP_OPERATIO NS		MLOG\$_WIP_O PERATIONS
WIP_WOPR_NE TWORKS_SN	APPS	WIP_OPERATIO N_NETWORKS	WIP	MLOG\$_WIP_O PERATION_NE TWOR
WIP_WOPR_RE SS_SN	APPS	WIP_DISCRETE _JOBS	WIP	MLOG\$_WIP_DI SCRETE_JOBS
		WIP_MSC_OPE N_JOB_STATUS ES		MLOG\$_WIP_M SC_OPEN_JOB_ STA
		WIP_OPERATIO N_RESOURCES		MLOG\$_WIP_O PERATION_RES OUR"

Snapshot	Snapshot Owner	Base Table Name	Base Table Owner	MLOG\$
WIP_WOPR_SU_B_RESS_SN	APPS	WIP_SUB_OPERACTION_RESOURCES	WIP	MLOG\$_WIP_SUB_OPERATION_RESOURCES
WIP_WREQ_OP_RS_SN	APPS	WIP_MSC_OPERATION_JOB_STATUS	WIP	MLOG\$_WIP_MSC_OPEN_JOB_STATUS
		WIP_DISCRETE_JOBS		MLOG\$_WIP_DISCRETE_JOBS
		WIP_REQUIREMENT_OPERATIONS		MLOG\$_WIP_REQUIREMENT_OPERATIONS
WSH_DELIVERY_DETAILS_SN	WSH	WSH_DELIVERY_DETAILS	WSH	MLOG\$_WSH_DELIVERY_DETAILS
WSH_TRIP_SN	APPS	WSH_TRIPS	WSH	MLOG\$_WSH_TRIPS
WSH_TRIP_STOP_SN	APPS	WSH_TRIP_STOPS	WSH	MLOG\$_WSH_TRIP_STOPS
WSM_LJ_OPRS_SN	APPS	WSM_COPY_OPERATIONS	WSH	MLOG\$_WSM_COPY_OPERATIONS
WSM_LJ_OPR_NETWORK_SN	APPS	WSM_COPY_OP_NETWORKS	WSM	MLOG\$_WSM_COPY_OP_NETWORKS
WSM_LJ_OPR_RESOURCE_INSTANCE_SN	APPS	WSM_COPY_OP_RESOURCE_INSTANCES	WSM	MLOG\$_WSM_COPY_OP_RESOURCE_INSTANCE
WSM_LJ_OPR_RESOURCE_SN	APPS	WSM_COPY_OP_RESOURCES	WSM	MLOG\$_WSM_COPY_OP_RESOURCES

Snapshot	Snapshot Owner	Base Table Name	Base Table Owner	MLOG\$
WSM_LJ_REQ_ OPRS_SN	APPS	WSM_COPY_RE QUIREMENT_O PS	WSM	MLOG\$_WSM_ COPY_REQUIR EMENT

Invalid Snapshots and Manually Refreshing the Snapshot

Note: This information relates to Global ATP data collection as well as Advanced Supply Chain Planning data collection.

Do not be concerned if you run the following SQL query and find that %SN has a status = INVALID using the following SQL script:

```
select distinct object_name, status from all_objects
where object_name like '%SN'
and object_name not in ('MSC_ATP_PLAN_SN', 'WSH_DELIVERY_DATES_SN')
and object_type in ('MATERIALIZED VIEW')
order by object_name;
```

Seeing an invalid snapshot is normal and of no concern. The object will be rebuilt and validated when Collections is run. Source snapshots may have become invalid in either their status or in the data contained therein. This could occur for many reasons. If you encounter the following error:

```
ORA-12034: snapshot log on "<schema>". "<snapshot_name>" younger than last
refresh
```

Oracle recommends that you run refresh of that snapshot via the application. Alternatively, the following code can be run as the APPS user on the source instance to manually execute the code in the request:

```
SELECT MRP_AP_REFRESH_S.NEXTVAL FROM DUAL;

exec dbms_snapshot.refresh('<owner>.
<snapshot_name>', 'COMPLETE');

-- note you can refresh one or more snapshots with one NEXTVAL

exec dbms_snapshot.refresh('<owner>. <snapshot_name>', 'COMPLETE');

exit;
```

Note: The select statement for NEXTVAL is only for Collection Snapshots.

D

SQL Commands

This appendix covers the following topics:

- Overview
- Partition Table SQL Commands
- Snapshot SQL Commands
- MLOG SQL Commands
- Tablespace SQL Commands

Overview

Many of the VCP collection commands can be run directly using SQL commands within SQL*Plus. They may not work correctly in a SQL Browser. Always use a test instance to check these scripts before using on a live instance. You may have to reformat some of these statements due to the way the browser is displaying them.

Partition Table SQL Commands

SQL #1 -- List Partitioned Tables in the System

This SQL lists the partitioned tables that are in the system.

```

-- Run this from SQL*Plus  the column HIGH_VALUE is column type LONG and
-- will not format correctly in a spreadsheet
set lines 200
set colsep |
spool SQL-1_137293.1.txt

SELECT
  table_name,
  partition_name,
  num_rows,
  high_value,
  sample_size,
  last_analyzed,
  global_stats
FROM
  all_tab_partitions
WHERE
  table_name like 'MSC%'
  order by substr(partition_name,instr
  (partition_name,'_',-1,1)+1) -- To check the partition count of each
table use order by table_name, partition_name
spool off

```

SQL #2 -- Show Created Instance Partitions

This SQL shows the instance partitions that have been created.

Free Flag valid values:

1 = Yes

2 = No

```

select      instance_id,
            free_flag,
            creation_date,
            last_update_date
from       msc_inst_partitions;

```

SQL #3 -- Show Instance Partitions Created and Assigned to an EBS Source Instance

This SQL shows the instance partitions that have been created and assigned to an EBS source instance, as well as their current status.

```

select      instance_code,
instance_id,
apps_ver,
a2m_dblink,
m2a_dblink,
st_status
from msc_apps_instances

```

SQL #4 -- Show Plans Created in the System

This SQL shows the plans that have been created in the system.

Plan_type valid values are:

1 = Manufacturing Plan

```

2 = Production Plan
3 = Distribution Plan
4 = Inventory Plan
5 = Distribution Plan (DRP)
6 = SNO Schedule
More plan types
7 = Production Schedule (PS)
8 = Service Parts Plan (SPP)
9 = Service Parts IO Plan (SPIO)
101 = Rapid Plan MRP
102 = Rapid Plan MPS
103 = Rapid Plan MPP

select
  mp.COMPILE_DESIGNATOR "Plan Name",
  mp.PLAN_ID "Plan ID",
  mp.SR_INSTANCE_ID "Instance ID",
  mtp.ORGANIZATION_CODE "Owning Org",
  mp.PLAN_COMPLETION_DATE "Last Run Date",
  decode (mp.PLAN_TYPE, 1, 'Manufacturing MRP', 2, 'Production MPS',
         3, 'Master MPP', 4, 'IO Plan', 5, 'Distribution DRP',
         7, 'PS Production Schedule', 6, 'SNO Schedule',
         8, 'Service Parts SPP', 9, 'Service IO Plan',
         101, 'Rapid Plan MRP', 102, 'Rapid Plan MPS',
         103, 'Rapid Plan MPP', plan_type)"Plan Type",
  decode (md.PRODUCTION, 1, 'Yes', 2, 'No', NULL, 'No') "Production Flag",
  decode (md.LAUNCH_WORKFLOW_FLAG, 1, 'Yes', 2, 'No', NULL, 'No') "Launch Workflow",
  decode (md.INVENTORY_ATP_FLAG, 1, 'Yes', 2, 'No', NULL, 'No') "ATP Plan",
  mp.CURR_START_DATE "Start Date",
  mp.CUTOFF_DATE "End Date"
  from
  msc_designators md,
  msc_plans mp,
  msc_trading_partners mtp
  where
  md.designator=mp.compile_designator and
  md.sr_instance_id=mp.sr_instance_id and
  mtp.sr_instance_id=mp.sr_instance_id and
  md.sr_instance_id=mtp.sr_instance_id and
  mtp.sr_tp_id=md.organization_id and
  mtp.sr_tp_id=mp.organization_id and
  mp.organization_id=md.organization_id
  and mtp.partner_type = 3
  ORDER BY "Plan Name";

```

SQL #5 -- Show Plans and Associated Plan Partitions

This SQL shows plans and their associated plan partitions.

MSC_PLAN_PARTITIONS shows how planning partitions are being used when the

profile MSC: Shared Plan Partition = No

When the profile MSC: Shared Plan Partition = Yes, then NO lines should appear in MSC_PLAN_PARTITIONS. **

Free Flag valid values:

1 = Yes

2 = No

```
select      plan_id,
            plan_name,
            free_flag,
            partition_number
  from      msc_plan_partitions
```

PLAN_ID	PLAN_NAME	FREE_FLAG	PARTITION_NUMBE R
347	DG-APS1	2	347
391	PRN SCP	1	391
411	411	1	411

The example above shows that:

- DG-APS1 is a plan that is being used (free_flag = 2).
- PRN SCP is a plan that has been purged, but a new plan has not been defined that uses that partition_number. The plan name is replaced when a new plan and the plan options have been defined.
- Plan name 411 is a new partition created by the Create APS Partitions request and has never been used.
- This means that two new plans could be defined in the Names form in ASCP. After the plan options are defined and saved, this table will be updated with the plan name and the free_flag = 2.
- If no rows with free_flag = 1 are returned, then you may still create the new plan in the Names form, but when attempting to define and save the plan options, an error will be received similar to 'No free partitions available, Contact the DBA to create partitions'. This is because the Names form only inserts into the table MSC_DESIGNATORS. When plan options are defined, then the process checks MSC_PLAN_PARTITIONS and updates the free_flag and plan_name. The tables inserted/updated are MSC_PLANS, MSC_PLAN_ORGANIZATIONS, MSC_DESGINATORS and MSC_SUBINVENTORIES.

- Use the concurrent request Create APS Partitions with parameters inst_partition_count = 0 and plan_partition_count = 1 to create a new plan partition.

SQL #6 -- Delete All Tables Where INSTANCE_ID Column Appears

This SQL deletes all the tables where the INSTANCE_ID column appears.

```
set heading off
set pagesize 500
spool delete_instance_id.sql

select distinct 'delete from ||TABLE_NAME|| where instance_id = nn;'
  -- replace nn with your instance_id
  from dba_tab_columns
  where column_name = 'INSTANCE_ID'
    and owner = 'MSC';
spool off
```

The script will have lines like the following and nn should have your INSTANCE_ID:

```
delete from MSC_APPS_INSTANCES where instance_id = nn;
delete from MSC_ATP_SOURCES_TEMP where instance_id = nn;
delete from MSC_COLL_PARAMETERS where instance_id = nn;
```

Edit the file to remove the SQL statement and the spool off command, and then submit the file in SQL*Plus:

```
SQL>@delete_instance_id.sql
SQL>commit;
```

SQL #7 -- Delete All Tables Where SR_INSTANCE_ID Column Appears

This SQL deletes all tables where the SR_INSTANCE_ID column appears.

```
set heading off
set pagesize 500
spool delete_sr_instance_id.sql

select
  distinct 'delete from ||TABLE_NAME||'
  where sr_instance_id = nn;
  -- replace nn with your instance_id
  from dba_tab_columns
  where
    column_name = 'SR_INSTANCE_ID'
    and owner = 'MSC'
    and table_name like 'MSC%';

spool off
-- The script will have lines like the following and nn should have your
instance_id
delete from MSC_ABC_CLASSES where sr_instance_id = nn;
delete from MSC_ALLOCATION_ASSIGNMENTS where sr_instance_id = nn;
delete from MSC_ALLOC_DEMANDS where sr_instance_id = nn;
```

Edit the file to remove the SQL statement and the spool off command, and then submit the file in SQL*Plus:

```
SQL>@delete_sr_instance_id.sql
SQL>commit;
```

Note: This script reports an error when attempting to delete from a %

MV or %SN table. This error can be ignored.

SQL #8 -- Drop Plan Partition

This SQL drops a plan partition.

```
declare
    errbuf varchar2(1000);
    retcode number ;
begin /* Dropping a plan partition */
    msc_manage_plan_partitions.drop_force_partition(errbuf,retcode,
&plan_partition_number,1);
end ;
/
```

SQL #9 -- Drop Instance Partition

This SQL drops an instance partition.

```
declare
    errbuf varchar2(1000);
    retcode number ;
begin /* Dropping an instance partition */
    msc_manage_plan_partitions.drop_force_partition(errbuf,retcode,
&instance_partition_number,2);
end ;
/
```

Important: In 12.0, you must delete the instance listed (Advanced Planning Adminstrator responsibility, navigation: Admin / Instances) before using this SQL to handle the new partitioned tables for the MSC_ST% staging tables.

SQL #10 -- Check MRP_AP_APPS_INSTANCES Table on the EBS Source Instance and Delete Line

This SQL script checks the MRP_AP_APPS_INSTANCES table on the EBS source instance and deletes the line from that table.

Note: For upgraded instances, check both tables.

SQL #11 -- Update All Tables Where SR_INSTANCE_ID Column Appears

This SQL script updates all tables where the SR_INSTANCE_ID column appears.

Important: Do not use for 11.5.10 and above.

```

set heading off
set linesize 200
set pagesize 500
spool update_sr_instance_id.sql

select distinct 'update ||TABLE_NAME|| set sr_instance_id = where
sr_instance_id = ;'
-- replace and with your instance_id's as directed
from dba_tab_columns
where column_name = 'SR_INSTANCE_ID'
and owner = 'MSC';
spool off

```

In this example, we used 41 and 21 for SR_INSTANCE_IDS. The script will have lines like the following:

```

update MSC_ABC_CLASSES set sr_instance_id = 41 where sr_instance_id =
21;
update MSC_ALLOCATION_ASSIGNMENTS set sr_instance_id = 41 where
sr_instance_id = 21;
update MSC_ALLOC_DEMANDS set sr_instance_id = 41 where sr_instance_id =
21;

```

Edit the file to remove the SQL statement and the spool off command and then submit the file in SQL*Plus:

```

SQL>@update_sr_instance_id.sql
SQL>commit;

```

Note: This script reports an error when attempting to update a %MV or %SN table. This error can be ignored.

SQL #12 -- Update All Tables Where INSTANCE_ID Column Appears

This SQL updates all tables where the INSTANCE_ID column appears.

Important: Do not use this script for 11.5.10 and above.

```

set heading off
set linesize 200
set pagesize 500
spool update_instance_id.sql

select distinct 'update ||TABLE_NAME|| set instance_id = where
instance_id = ;'
-- replace and with your instance_id's as directed
from dba_tab_columns
where column_name = 'INSTANCE_ID'
and owner = 'MSC';

spool off

```

In the following example, 41 and 21 were used as INSTANCE_IDs:

```

update MSC_APPS_INSTANCES set instance_id = 41 where instance_id = 21;
update MSC_ATP_SOURCES_TEMP set instance_id = 41 where instance_id = 21;
update MSC_COLL_PARAMETERS set instance_id = 41 where instance_id = 21;

```

Edit the file to remove the SQL statement, and the spool off command, and then submit

the file in SQL*Plus:

```
SQL>@update_instance_id.sql
SQL>commit;
```

SQL #13 -- Delete All Tables Where the INSTANCE_ID Column Appears for ODP Data

This SQL script deletes all tables where the INSTANCE_ID column appears. This is used for ODP data.

```
set heading off
set pagesize 500
spool delete_ODP_instance_id.sql

select distinct 'delete from ||TABLE_NAME||' where instance_id = nn;
-- replace nn with your instance_id
from dba_tab_columns
where table_name in
(select object_name from all_objects where object_name like 'MSD%'
and object_type like 'TABLE')
and column_name like 'INSTANCE_ID';
spool off
```

In this example, nn represents your INSTANCE_ID:

```
delete from MSD_APP_INSTANCE_ORGS where instance_id = nn;
delete from MSD_ITEM_RELATIONSHIPS where instance_id = nn;
delete from MSD_LOCAL_ID_SETUP where instance_id = nn;
```

Edit the file to remove the SQL statement, and the spool off command, and then submit the file in SQL*Plus:

```
SQL>@delete_odp_instance_id.sql
SQL>commit;
```

SQL #14 -- Delete All Tables Where SR_INSTANCE_ID Column Appears for ODP Data

This SQL deletes all tables where the SR_INSTANCE_ID column appears. This is used for ODP data.

```
set heading off
set pagesize 500
spool delete_ODP_sr_instance_id.sql

select distinct 'delete from ||TABLE_NAME||' where sr_instance_id = nn;
-- replace nn with your instance_id
from dba_tab_columns
where table_name in
(select object_name from all_objects where object_name like 'MSD%'
and object_type like 'TABLE')
and column_name like 'SR_INSTANCE_ID';

spool off
```

In this example, nn represents your SR_INSTANCE_ID:

```
delete from MSD_DEMAND_PLANS where sr_instance_id = nn;
delete from MSD_DP_PLANNING_PCT_DENORM where sr_instance_id = nn;
delete from MSD_DP_SCN_ENTRIES_DENORM where sr_instance_id = nn;
```

Edit the file to remove the SQL statement and the spool off command and then submit

the file in SQL*Plus:

```
SQL>@delete_odp_sr_instance_id.sql
SQL>commit;
```

SQL #14b -- Delete All Tables Where the Instance Column Appears for ODP Data

This SQL script deletes all tables where the instance column appears.

```
set heading off
set pagesize 500
spool delete_ODP_instance.sql

select distinct 'delete from '||TABLE_NAME||' where instance = nn;'
-- replace nn with your instance_id
from dba_tab_columns
where table_name in
(select object_name from all_objects where object_name like 'MSD%'
and object_type like 'TABLE')
and column_name like 'INSTANCE';

spool off
```

The script will have lines like the following and nn should have your INSTANCE_ID.

```
delete from MSD_BACKUP_LEVEL_VALUES where instance = nn;
delete from MSD_BOM_COMPONENTS where instance = nn;
delete from MSD_BOOKING_DATA where instance = nn;
```

Edit the file to remove the SQL statement and the spool off command and then submit the file in SQL*Plus:

```
SQL>@delete_odp_instance.sql
SQL>commit;
```

Snapshot SQL Commands

SQL #15 -- Use Base Table to Find All Related Snapshots

This SQL script finds all the snapshots that are using the MLOG table for a single base table. See SQL # 22 -- Show MLOGs and Related Snapshots and Base Tables for Data Collections, page D-16 and Appendix D -- Snapshots, page C-1 for data collection objects..

```
SELECT DISTINCT
amdr.MVIEW_NAME "Snapshot",
amdr.OWNER "Snapshot Owner",
amdr.DETAILOBJ_NAME "Base Table Name",
amdr.DETAILOBJ_OWNER "Base Table Owner",
log_table mlog$_name
FROM ALL_MVIEW_DETAIL_RELATIONS amdr,
dba_snapshot_logs dsl
WHERE amdr.DETAILOBJ_NAME LIKE '&Base_table_name'
AND amdr.DETAILOBJ_TYPE = 'TABLE'
AND amdr.DETAILOBJ_NAME = dsl.MASTER
order by detailobj_name, mview_name;
```

SQL #16 -- Show All Information on a Snapshot in DBA_SNAPSHOTS

Review the snapshot information in the database for one particular snapshot including the query used by the snapshot.

Note: The Master column can only show one MLOG\$ table, so check the Query column to see which tables are involved. Alternatively, use SQL #18 -- Use Snapshots to Find All Base Tables and MLOGs Used by a Snapshot, page D-10 to see all base tables and MLOGs.

```
select * from dba_snapshots where name like
'&snapshot_name';
```

SQL #17 -- Show List of Snapshots Used by VCP Data Collections

Different from output of SQL #22 -- Show MLOGs and Related Snapshots and Base Tables for Data Collections, page D-16 and Appendix D -- Snapshots, page C-1, this SQL script shows only VCP snapshots. Currently all snapshots with name like %SN, except MSC_ATP_PLAN_SN and WSH_DELIVERY_DETAILS_SN, are used for Data Collections. MSC_ATP_PLAN_SN is used by ATP Plans to store record of items in the ATP plan. WSH_DEVLIERY_DETAILS_SN is used by Oracle Transportation Management (OTM).

Note: Master column will show the base table. If it is a complex snapshot, then multiple tables can be used and aren't be shown here. Check the Query column to see if multiple tables are involved for the snapshot. Alternatively, run SQL #18 -- Use Snapshots to Find All Base Tables and MLOGs Used by a Snapshot, page D-10.

```
select * from dba_snapshots where name like '%SN'
and name not in
('MSC_ATP_PLAN_SN', 'WSH_DELIVERY_DETAILS_SN')
order by name;
```

SQL #18 -- Use Snapshots to Find All Base Tables and MLOGs Used by a Snapshot

Since SQL #17 -- Show List of Snapshots Used by VCP Data Collections, page D-10 output can only show one table in the Master column, you must either examine the Query column or run this SQL to get list of base tables/MLOGs being used for a snapshot.

Note: DERIVED_MLOG\$_NAME: If you have a rare 27 character MLOG name, then only 26 characters can be shown in this output and the name may not be correct. Run SQL #15 -- Use Base Table to Find All Related Snapshots, page D-9 using the base table if required.

```
select OWNER "Snapshot Owner",
MVIEW_NAME "Snapshot",
DETAILOBJ_OWNER "Base Table Owner",
DETAILOBJ_NAME "Base Table Name",
'MLOG$_' || SUBSTR(DETAILOBJ_NAME,1,20) derived_mlog$_name
FROM ALL_MVIEW_DETAIL_RELATIONS
where MVIEW_NAME like '&Snapshot'
and DETAILOBJ_TYPE='TABLE'
order by mvview_name, detailobj_name;
```

MLOG SQL Commands

SQL #19 -- Get COUNT(*) on MLOGs

Alternately, you can use SQL #20 -- Gather Table Statistics for All MLOGs Used in Data Collections, page D-12 to gather table statistics and then use SQL #21 -- Size, NUM_ROWS and LAST_ANALYZED for MLOGs Used in Data Collections, page D-15 to get more information. Single example:

```
select count(*) from &owner_&MLOG_name;
like select count(*) from BOM.MLOG$_.BOM_COMPONENTS_B;
```

When running the full script, ignore ORA-942 errors. The errors mean that your release does not have this MLOG.

```

set echo on
select count(*) from AHL.MLOG$_AHL_SCHEDULE_MATERIA;
select count(*) from BOM.MLOG$_BOM_COMPONENTS_B;
select count(*) from BOM.MLOG$_BOM_CTO_ORDER_DEMAND;
select count(*) from BOM.MLOG$_BOM_DEPARTMENTS;
select count(*) from BOM.MLOG$_BOM_OPERATIONAL_ROUT;
select count(*) from BOM.MLOG$_BOM_OPERATION_NETWOR;
select count(*) from BOM.MLOG$_BOM_OPERATION_RESOUR;
select count(*) from BOM.MLOG$_BOM_OPERATION_SEQUEN;
select count(*) from BOM.MLOG$_BOM_RESOURCE_CHANGES;
select count(*) from BOM.MLOG$_BOM_RES_INSTANCE_CHA;
select count(*) from BOM.MLOG$_BOM_STRUCTURES_B;
select count(*) from BOM.MLOG$_BOM_SUBSTITUTE_COMPO;
select count(*) from BOM.MLOG$_BOM_SUB_OPERATION_RE;
select count(*) from CSP.MLOG$_CSP_REPAIR_PO_HEADER;
select count(*) from EAM.MLOG$_EAM_WO_RELATIONSHIPS;
select count(*) from MRP.MLOG$_MRP_FORECAST_DATES;
select count(*) from MRP.MLOG$_MRP_FORECAST_DESIGNA;
select count(*) from MRP.MLOG$_MRP_FORECAST_ITEMS;
select count(*) from MRP.MLOG$_MRP_SCHEDULE_DATES;
select count(*) from INV.MLOG$_MTL_DEMAND;
select count(*) from INV.MLOG$_MTL_ITEM_CATEGORIES;
select count(*) from INV.MLOG$_MTL_MATERIAL_TRANSAC;
select count(*) from INV.MLOG$_MTL_ONHAND_QUANTITIE;
select count(*) from INV.MLOG$_MTL RELATED ITEMS;
select count(*) from INV.MLOG$_MTL_RESERVATIONS;
select count(*) from INV.MLOG$_MTL_SUPPLY;
select count(*) from INV.MLOG$_MTL_SYSTEM_ITEMS_B;
select count(*) from INV.MLOG$_MTL_TXN_REQUEST_LINE;
select count(*) from INV.MLOG$_MTL_USER_DEMAND;
select count(*) from INV.MLOG$_MTL_USER_SUPPLY;
select count(*) from OMT.MLOG$_OE_ORDER_LINES_ALL;
select count(*) from PO.MLOG$_PO_ACCEPTANCES;
select count(*) from PO.MLOG$_PO_CHANGE_REQUESTS;
select count(*) from PO.MLOG$_PO_SUPPLIER_ITEM_CAP;
select count(*) from WIP.MLOG$_WIP_DISCRETE_JOBS;
select count(*) from WIP.MLOG$_WIP_FLOW_SCHEDULES;
select count(*) from WIP.MLOG$_WIP_LINES;
select count(*) from WIP.MLOG$_WIP_MSC_OPEN_JOB_STA;
select count(*) from WIP.MLOG$_WIP_OPERATIONS;
select count(*) from WIP.MLOG$_WIP_OPERATION_NETWOR;
select count(*) from WIP.MLOG$_WIP_OPERATION_RESOUR;
select count(*) from WIP.MLOG$_WIP_OP_RESOURCE_INST;
select count(*) from WIP.MLOG$_WIP_REPEATITIVE_ITEMS;
select count(*) from WIP.MLOG$_WIP_REPEATITIVE_SCHED;
select count(*) from WIP.MLOG$_WIP_REQUIREMENT_OPER;
select count(*) from WIP.MLOG$_WIP_SUB_OPERATION_RE;
select count(*) from WSH.MLOG$_WSH_TRIPS;
select count(*) from WSH.MLOG$_WSH_TRIP_STOPS;
select count(*) from WSM.MLOG$_WSM_COPY_OPERATIONS;
select count(*) from WSM.MLOG$_WSM_COPY_OP_NETWORKS;
select count(*) from WSM.MLOG$_WSM_COPY_OP_RESOURCE;
select count(*) from WSM.MLOG$_WSM_COPY_OP_RESOURCE1;
select count(*) from WSM.MLOG$_WSM_COPY_REQUIREMENT;

```

SQL #20 -- Gather Table Statistics for All MLOGs Used in Data Collections

Gather table statistics on the table, and then check use SQL #21 -- Show Size, NUM_ROWS and LAST_ANALYZED for MLOGs Used in Data Collections, page D-15 to query dba_tables and dba_segments to show info on the MLOGs.

```
begin FND_STATS.GATHER_TABLE_STATS ('[owner]', '[MLOG_NAME]', 10); end;
```

The Gather Table Stats command may error if the object does not exist in your release.

You can ignore errors like the following:

```
ORA-20000: Unable to analyze TABLE "CSP"."MLOG$_CSP_REPAIR_PO_HEADER",  
insufficient privileges or does not exist  
ORA-06512: at "APPS.FND_STATS", [error continues]
```

Here is an example of the full command:

```

begin FND_STATS.GATHER_TABLE_STATS ( 'AHL' , 'MLOG$_AHL_SCHEDULE_MATERIA' ,
10);end;
begin FND_STATS.GATHER_TABLE_STATS ( 'BOM' , 'MLOG$_BOM_COMPONENTS_B' ,10);
end;
begin FND_STATS.GATHER_TABLE_STATS ( 'BOM' , 'MLOG$_BOM_CTO_ORDER_DEMAND' ,
10);end;
begin FND_STATS.GATHER_TABLE_STATS ( 'BOM' , 'MLOG$_BOM_DEPARTMENTS' ,10);
end;
begin FND_STATS.GATHER_TABLE_STATS ( 'BOM' , 'MLOG$_BOM_OPERATIONAL_ROUT' ,
10);end;
begin FND_STATS.GATHER_TABLE_STATS ( 'BOM' , 'MLOG$_BOM_OPERATION_NETWOR' ,
10);end;
begin FND_STATS.GATHER_TABLE_STATS ( 'BOM' , 'MLOG$_BOM_OPERATION_RESOUR' ,
10);end;
begin FND_STATS.GATHER_TABLE_STATS ( 'BOM' , 'MLOG$_BOM_OPERATION_SEQUEN' ,
10);end;
begin FND_STATS.GATHER_TABLE_STATS ( 'BOM' , 'MLOG$_BOM_RESOURCE_CHANGES' ,
10);end;
begin FND_STATS.GATHER_TABLE_STATS ( 'BOM' , 'MLOG$_BOM_RES_INSTANCE_CHA' ,
10);end;
begin FND_STATS.GATHER_TABLE_STATS ( 'BOM' , 'MLOG$_BOM_STRUCTURES_B' ,10);
end;
begin FND_STATS.GATHER_TABLE_STATS ( 'BOM' , 'MLOG$_BOM_SUBSTITUTE_COMPO' ,
10);end;
begin FND_STATS.GATHER_TABLE_STATS ( 'BOM' , 'MLOG$_BOM_SUB_OPERATION_RE' ,
10);end;
begin FND_STATS.GATHER_TABLE_STATS ( 'CSP' , 'MLOG$_CSP_REPAIR_PO_HEADER' ,
10);end;
begin FND_STATS.GATHER_TABLE_STATS ( 'EAM' , 'MLOG$_EAM_WO_RELATIONSHIPS' ,
10);end;
begin FND_STATS.GATHER_TABLE_STATS ( 'MRP' , 'MLOG$_MRP_FORECAST_DATES' ,
10);end;
begin FND_STATS.GATHER_TABLE_STATS ( 'MRP' , 'MLOG$_MRP_FORECAST_DESIGNA' ,
10);end;
begin FND_STATS.GATHER_TABLE_STATS ( 'MRP' , 'MLOG$_MRP_FORECAST_ITEMS' ,
10);end;
begin FND_STATS.GATHER_TABLE_STATS ( 'MRP' , 'MLOG$_MRP_SCHEDULE_DATES' ,
10);end;
begin FND_STATS.GATHER_TABLE_STATS ( 'INV' , 'MLOG$_MTL_DEMAND' ,10);end;
begin FND_STATS.GATHER_TABLE_STATS ( 'INV' , 'MLOG$_MTL_ITEM_CATEGORIES' ,
10);end;
begin FND_STATS.GATHER_TABLE_STATS ( 'INV' , 'MLOG$_MTL_MATERIAL_TRANSAC' ,
10);end;
begin FND_STATS.GATHER_TABLE_STATS ( 'INV' , 'MLOG$_MTL_ONHAND_QUANTITIE' ,
10);end;
begin FND_STATS.GATHER_TABLE_STATS ( 'INV' , 'MLOG$_MTL RELATED ITEMS' ,10);
end;
begin FND_STATS.GATHER_TABLE_STATS ( 'INV' , 'MLOG$_MTL_RESERVATIONS' ,10);
end;
begin FND_STATS.GATHER_TABLE_STATS ( 'INV' , 'MLOG$_MTL_SUPPLY' ,10);end;
begin FND_STATS.GATHER_TABLE_STATS ( 'INV' , 'MLOG$_MTL_SYSTEM_ITEMS_B' ,
10);end;
begin FND_STATS.GATHER_TABLE_STATS ( 'INV' , 'MLOG$_MTL_TXN_REQUEST_LINE' ,
10);end;
begin FND_STATS.GATHER_TABLE_STATS ( 'INV' , 'MLOG$_MTL_USER_DEMAND' ,10);
end;
begin FND_STATS.GATHER_TABLE_STATS ( 'INV' , 'MLOG$_MTL_USER_SUPPLY' ,10);
end;
begin FND_STATS.GATHER_TABLE_STATS ( 'ONT' , 'MLOG$_OE_ORDER_LINES_ALL' ,
10);end;
begin FND_STATS.GATHER_TABLE_STATS ( 'PO' , 'MLOG$_PO_ACCEPTANCES' ,10);end;
begin FND_STATS.GATHER_TABLE_STATS ( 'PO' , 'MLOG$_PO_CHANGE_REQUESTS' ,10);
end;
begin FND_STATS.GATHER_TABLE_STATS ( 'PO' , 'MLOG$_PO_SUPPLIER_ITEM_CAP' ,
10);end;
begin FND_STATS.GATHER_TABLE_STATS ( 'WIP' , 'MLOG$_WIP_DISCRETE_JOBS' ,10);

```

```

end;
begin FND_STATS.GATHER_TABLE_STATS ('WIP','MLOG$_WIP_FLOW_SCHEDULES',10);end;
begin FND_STATS.GATHER_TABLE_STATS ('WIP','MLOG$_WIP_LINES',10);end;
begin FND_STATS.GATHER_TABLE_STATS ('WIP','MLOG$_WIP_MSC_OPEN_JOB_STA',10);end;
begin FND_STATS.GATHER_TABLE_STATS ('WIP','MLOG$_WIP_OPERATIONS',10);
end;
begin FND_STATS.GATHER_TABLE_STATS ('WIP','MLOG$_WIP_OPERATION_NETWOR',10);end;
begin FND_STATS.GATHER_TABLE_STATS ('WIP','MLOG$_WIP_OPERATION_RESOUR',10);end;
begin FND_STATS.GATHER_TABLE_STATS ('WIP','MLOG$_WIP_OP_RESOURCE_INST',10);end;
begin FND_STATS.GATHER_TABLE_STATS ('WIP','MLOG$_WIP_REPEATITIVE_ITEMS',10);end;
begin FND_STATS.GATHER_TABLE_STATS ('WIP','MLOG$_WIP_REPEATITIVE_SCHED',10);end;
begin FND_STATS.GATHER_TABLE_STATS ('WIP','MLOG$_WIP_REQUIREMENT_OPER',10);end;
begin FND_STATS.GATHER_TABLE_STATS ('WIP','MLOG$_WIP_SUB_OPERATION_RE',10);end;
begin FND_STATS.GATHER_TABLE_STATS ('WSH','MLOG$_WSH_TRIPS',10);end;
begin FND_STATS.GATHER_TABLE_STATS ('WSH','MLOG$_WSH_TRIP_STOPS',10);
end;
begin FND_STATS.GATHER_TABLE_STATS ('WSM','MLOG$_WSM_COPY_OPERATIONS',10);end;
begin FND_STATS.GATHER_TABLE_STATS ('WSM','MLOG$_WSM_COPY_OP_NETWORKS',10);end;
begin FND_STATS.GATHER_TABLE_STATS ('WSM','MLOG$_WSM_COPY_OP_RESOURCE',10);end;
begin FND_STATS.GATHER_TABLE_STATS ('WSM','MLOG$_WSM_COPY_OP_RESOURCE1',10);end;
begin FND_STATS.GATHER_TABLE_STATS ('WSM','MLOG$_WSM_COPY_REQUIREMENT',10);end;

```

SQL #21 -- Show Size, NUM_ROWS and LAST_ANALYZED for MLOGs Used in Data Collections

NUM_ROWS shows count when last analyzed. You must run SQL #20 -- Gather Table Statistics for All MLOGs Used in Data Collections, page D-12 before you run this SQL for accurate results. Size from DBA_SEGMENTS and NUM_ROWS should have some reasonable correlation. If size in megabytes is large but number of rows is small, then this indicates that many deletes have been run and the high water mark is not realistic with respect to the actual size.

Choices to fix this are:

- Refresh the snapshot(s) for the MLOG with automatic refresh mode, then gather the table statistics again for this MLOG and check this SQL once more to make sure size is reduced.

OR

- Use steps to truncate the MLOG and refresh the snapshot in complete mode, then gather table statisticss again for this MLOG and check this SQL once more to make sure size is reduced.

```

SELECT dt.owner, dt.table_name, dt.num_rows, dt.last_analyzed, ds.
bytes/1024/1024 "Size in MB", dt.tablespace_name, dt.initial_extent, dt.
next_extent, ds extents FROM dba_tables dt, dba_segments ds WHERE
table_name=segment_name and table_name IN
( 'MLOG$_AHL_SCHEDULE_MATERIA', 'MLOG$_BOM_COMPONENTS_B', 'MLOG$_BOM_CTO_OR
DER_DEMAND', 'MLOG$_BOM_DEPARTMENTS', 'MLOG$_BOM_OPERATIONAL_ROUT', 'MLOG$_
BOM_OPERATION_NETWOR',
' MLOG$_BOM_OPERATION_RESOUR', 'MLOG$_BOM_OPERATION_SEQUEN', 'MLOG$_BOM_RES
OURCE_CHANGES', 'MLOG$_BOM_RES_INSTANCE_CHA', 'MLOG$_BOM_STRUCTURES_B',
' MLOG$_BOM_SUBSTITUTE_COMPO', 'MLOG$_BOM_SUB_OPERATION_RE', 'MLOG$_CSP REP
AIR_PO_HEADER', 'MLOG$_EAM_WO_RELATIONSHIPS', 'MLOG$_MRP_FORECAST_DATES',
' MLOG$_MRP_FORECAST_DESIGNA', 'MLOG$_MRP_FORECAST_ITEMS', 'MLOG$_MRP_SCHED
ULE_DATES', 'MLOG$_MTL_DEMAND', 'MLOG$_MTL_ITEM_CATEGORIES', 'MLOG$_MTL_MAT
ERIAL_TRANSAC',
' MLOG$_MTL_ONHAND_QUANTITIE', 'MLOG$_MTL RELATED ITEMS', 'MLOG$_MTL_RESERV
ATIONS', 'MLOG$_MTL_SUPPLY', 'MLOG$_MTL_SYSTEM_ITEMS_B', 'MLOG$_MTL_TXN_REQ
UEST_LINE',
' MLOG$_MTL_USER_DEMAND', 'MLOG$_MTL_USER_SUPPLY', 'MLOG$_OE_ORDER_LINES_AL
L', 'MLOG$_PO_ACCEPTANCES', 'MLOG$_PO_CHANGE_REQUESTS', 'MLOG$_PO_SUPPLIER_
ITEM_CAP',
' MLOG$_WIP_DISCRETE_JOBS', 'MLOG$_WIP_FLOW_SCHEDULES', 'MLOG$_WIP_LINES',
' MLOG$_WIP_MSC_OPEN_JOB_STA', 'MLOG$_WIP_OPERATIONS', 'MLOG$_WIP_OPERATION_
NETWOR',
' MLOG$_WIP_OPERATION_RESOUR', 'MLOG$_WIP_OP_RESOURCE_INST', 'MLOG$_WIP REP
ETITIVE_ITEMS', 'MLOG$_WIP_REPEATITIVE_SCHED', 'MLOG$_WIP_REQUIREMENT_OPER
',
' MLOG$_WIP_SUB_OPERATION_RE', 'MLOG$_WSH_TRIPS', 'MLOG$_WSH_TRIP_STOPS', 'M
LOG$_WSM_COPY_OPERATIONS', 'MLOG$_WSM_COPY_OP_NETWORKS', 'MLOG$_WSM_COPY_O
P_RESOURCE',
' MLOG$_WSM_COPY_OP_RESOURCE1', 'MLOG$_WSM_COPY_REQUIREMENT' )
order by table_name;

```

SQL #22 -- Show MLOGs and Related Snapshots and Base Tables for Data Collections

Shows Data Collections snapshots AND also the snapshots for other products that share the MLOG tables for Data Collections. Used for the table in Appendix D -- Snapshots, page C-1.

SORT OPTIONS - default is by Snapshot

OPTION B - sort by MLOG Name, Snapshot

OPTION C - sort by Base Table Name, Snapshot

```

SELECT DISTINCT
amdr.MVIEW_NAME "Snapshot",
amdr.OWNER "Snapshot Owner",
amdr.DETAILOBJ_NAME "Base Table Name",
amdr.DETAILOBJ_OWNER "Base Table Owner",
log_table mlog$_name
FROM ALL_MVIEW_DETAIL_RELATIONS amdr,
dba_snapshot_logs dsl
where DETAILOBJ_TYPE = 'TABLE'
and (detailobj_owner, detailobj_name) not in
(( 'MSC', 'MSC_COLLECTED_ORGS'))
AND amdr.DETAILOBJ_NAME = dsl.MASTER
and DETAILOBJ_NAME in
('AHL_SCHEDULE_MATERIALS', 'BOM_COMPONENTS_B', 'BOM_CTO_ORDER_DEMAND', 'BOM
_DEPARTMENTS', 'BOM_OPERATIONAL_ROUTINGS', 'BOM_OPERATION_NETWORKS',
'BOM_OPERATION_RESOURCES', 'BOM_RESOURCE_CHANGES', 'BOM_RES_INSTANCE_CHANG
ES', 'BOM_STRUCTURES_B', 'BOM_SUBSTITUTE_COMPONENTS',
'BOM_SUB_OPERATION_RESOURCES', 'CSP_REPAIR_PO_HEADERS', 'EAM_WO_RELATIONSH
IPS', 'MRP_FORECAST_DATES', 'MRP_FORECAST_DESIGNATORS', 'MRP_FORECAST_ITEMS
',
'MRP_SCHEDULE_DATES', 'MTL_DEMAND', 'MTL_ITEM_CATEGORIES', 'MTL_MATERIAL_TR
ANSACTIONS_TEMP', 'MTL_ONHAND_QUANTITIES_DETAIL', 'MTL RELATED ITEMS',
'MTL_RESERVATIONS', 'MTL_SUPPLY', 'MTL_SYSTEM_ITEMS_B', 'MTL_TXN_REQUEST_LI
NES', 'MTL_USER_DEMAND', 'MTL_USER_SUPPLY', 'OE_ORDER_LINES_ALL', 'PO_ACCEPT
ANCES',
'PO_CHANGE_REQUESTS', 'PO_SUPPLIER_ITEM_CAPACITY', 'WIP_DISCRETE_JOBS', 'WI
P_FLOW_SCHEDULES', 'WIP_LINES', 'WIP_MSC_OPEN_JOB_STATUSES', 'WIP_OPERATION
S',
'WIP_OPERATION_NETWORKS', 'WIP_OPERATION_RESOURCES', 'WIP_OP_RESOURCE_INST
ANCES', 'WIP_REPEATITIVE_ITEMS', 'WIP_REPEATITIVE_SCHEDULES', 'WIP_REQUIREMEN
T_OPERATIONS',
'WIP_SUB_OPERATION_RESOURCES', 'WSH_DELIVERY_DETAILS', 'WSH_TRIPS', 'WSH_TR
IP_STOPS', 'WSM_COPY_OPERATIONS', 'WSM_COPY_OP_NETWORKS', 'WSM_COPY_OP_RESO
URCES',
'WSM_COPY_OP_RESOURCE_INSTANCES', 'WSM_COPY_REQUIREMENT_OPS')
order by MVIEW_NAME; -- replace this line to use OPTION B or C
-- order by log_table, mview_name -- OPTION B sort by MLOG Name
-- order by DETAILOBJ_NAME, mview_name -- OPTION C sort by Base Table
Name

```

SQL #23 -- Show All MLOGs with Size Larger Than 10 MB

Takes a baseline reading of the MLOGs on the system. This can be used for all MLOGs on the system larger than 10 MB, not just entities used for Data Collections. Oracle suggests that any MLOGs smaller than 10 MB can be ignored.

```

SELECT DBA.OWNER "Owner",
dba.segment_name "MLOG$ Name",
dba.bytes/1024/1024 "MLOG Size in MB",
log.master "Base Table Name",
fav.application_name "Application Name"
FROM all_snapshot_logs log,
dba_segments DBA,
fnd_application_vl fav
WHERE dba.segment_name LIKE 'MLOG$%'
AND dba.segment_name=log.log_table
AND dba.bytes>10000000
AND dba.owner=fav.application_short_name
ORDER BY bytes DESC;

```

SQL #24 -- Truncate all MLOGs Used for Data Collections

After this SQL is run, you must run Refresh Collection Snapshots as a standalone request using complete refresh for all snapshots. If the table does not exist in your release, then that statement will fail. This is not a problem and can be ignored

```
TRUNCATE TABLE AHL.MLOG$_AHL_SCHEDULE_MATERIA;
TRUNCATE TABLE BOM.MLOG$_BOM_COMPONENTS_B;
TRUNCATE TABLE BOM.MLOG$_BOM_CTO_ORDER_DEMAND;
TRUNCATE TABLE BOM.MLOG$_BOM_DEPARTMENTS;
TRUNCATE TABLE BOM.MLOG$_BOM_OPERATIONAL_ROUT;
TRUNCATE TABLE BOM.MLOG$_BOM_OPERATION_NETWOR;
TRUNCATE TABLE BOM.MLOG$_BOM_OPERATION_RESOUR;
TRUNCATE TABLE BOM.MLOG$_BOM_OPERATION_SEQUEN;
TRUNCATE TABLE BOM.MLOG$_BOM_RESOURCE_CHANGES;
TRUNCATE TABLE BOM.MLOG$_BOM_RES_INSTANCE_CHA;
TRUNCATE TABLE BOM.MLOG$_BOM_STRUCTURES_B;
TRUNCATE TABLE BOM.MLOG$_BOM_SUBSTITUTE_COMPO;
TRUNCATE TABLE BOM.MLOG$_BOM_SUB_OPERATION_RE;
TRUNCATE TABLE CSP.MLOG$_CSP_REPAIR_PO_HEADER;
TRUNCATE TABLE EAM.MLOG$_EAM_WO_RELATIONSHIPS;
TRUNCATE TABLE MRP.MLOG$_MRP_FORECAST_DATES;
TRUNCATE TABLE MRP.MLOG$_MRP_FORECAST_DESIGNA;
TRUNCATE TABLE MRP.MLOG$_MRP_FORECAST_ITEMS;
TRUNCATE TABLE MRP.MLOG$_MRP_SCHEDULE_DATES;
TRUNCATE TABLE INV.MLOG$_MTL_DEMAND;
TRUNCATE TABLE INV.MLOG$_MTL_ITEM_CATEGORIES;
TRUNCATE TABLE INV.MLOG$_MTL_MATERIAL_TRANSAC;
TRUNCATE TABLE INV.MLOG$_MTL_ONHAND_QUANTITIE;
TRUNCATE TABLE INV.MLOG$_MTL RELATED ITEMS;
TRUNCATE TABLE INV.MLOG$_MTL_RESERVATIONS;
TRUNCATE TABLE INV.MLOG$_MTL_SUPPLY;
TRUNCATE TABLE INV.MLOG$_MTL_SYSTEM_ITEMS_B;
TRUNCATE TABLE INV.MLOG$_MTL_TXN_REQUEST_LINE;
TRUNCATE TABLE INV.MLOG$_MTL_USER_DEMAND;
TRUNCATE TABLE INV.MLOG$_MTL_USER_SUPPLY;
TRUNCATE TABLE ONT.MLOG$_OE_ORDER_LINES_ALL;
TRUNCATE TABLE PO.MLOG$_PO_ACCEPTANCES;
TRUNCATE TABLE PO.MLOG$_PO_CHANGE_REQUESTS;
TRUNCATE TABLE PO.MLOG$_PO_SUPPLIER_ITEM_CAP;
TRUNCATE TABLE WIP.MLOG$_WIP_DISCRETE_JOBS;
TRUNCATE TABLE WIP.MLOG$_WIP_FLOW_SCHEDULES;
TRUNCATE TABLE WIP.MLOG$_WIP_LINES;
TRUNCATE TABLE WIP.MLOG$_WIP_MSC_OPEN_JOB_STA;
TRUNCATE TABLE WIP.MLOG$_WIP_OPERATIONS;
TRUNCATE TABLE WIP.MLOG$_WIP_OPERATION_NETWOR;
TRUNCATE TABLE WIP.MLOG$_WIP_OPERATION_RESOUR;
TRUNCATE TABLE WIP.MLOG$_WIP_OP_RESOURCE_INST;
TRUNCATE TABLE WIP.MLOG$_WIP_REPEATITIVE_ITEMS;
TRUNCATE TABLE WIP.MLOG$_WIP_REPEATITIVE_SCHED;
TRUNCATE TABLE WIP.MLOG$_WIP_REQUIREMENT_OPER;
TRUNCATE TABLE WIP.MLOG$_WIP_SUB_OPERATION_RE;
TRUNCATE TABLE WSH.MLOG$_WSH_TRIPS;
TRUNCATE TABLE WSH.MLOG$_WSH_TRIP_STOPS;
TRUNCATE TABLE WSM.MLOG$_WSM_COPY_OPERATIONS;
TRUNCATE TABLE WSM.MLOG$_WSM_COPY_OP_NETWORKS;
TRUNCATE TABLE WSM.MLOG$_WSM_COPY_OP_RESOURCE;
TRUNCATE TABLE WSM.MLOG$_WSM_COPY_OP_RESOURCE1;
TRUNCATE TABLE WSM.MLOG$_WSM_COPY_REQUIREMENT;
```

SQL #25 -- Lock Statistics for All MLOGs Used for Data Collections

If you run SQL #24 -- Truncate all MLOGs Used for Data Collections, page D-17 to truncate, then run SQL #20 -- Gather Table Statistics for All MLOGs Used in Data Collections, page D-12 to gather the table statistics. Then use this SQL to lock the stats.

The required complete refresh of the snapshots can be run before or after these steps are completed, but if automated processes or users are working on the system, then MLOGs may start to accumulate rows.

```

EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('AHL', 'MLOG$_AHL_SCHEDULE_MATERIA';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('BOM', 'MLOG$_BOM_COMPONENTS_B';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('BOM', 'MLOG$_BOM_CTO_ORDER_DEMAND';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('BOM', 'MLOG$_BOM_ORDER_ROUT';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('BOM', 'MLOG$_BOM_OPERATIONAL_ROUT';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('BOM', 'MLOG$_BOM_OPERATION_NETWORK';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('BOM', 'MLOG$_BOM_OPERATION_RESOURCE';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('BOM', 'MLOG$_BOM_OPERATION_SEQUEN';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('BOM', 'MLOG$_BOM_RESOURCE_CHANGES';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('BOM', 'MLOG$_BOM_RES_INSTANCE_CHA';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('BOM', 'MLOG$_BOM_STRUCTURES_B';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('BOM', 'MLOG$_BOM_SUBSTITUTE_COMPO';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('BOM', 'MLOG$_BOM_SUB_OPERATION_RE';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('CSP', 'MLOG$_CSP_REPAIR_PO_HEADER';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('EAM', 'MLOG$_EAM_WO_RELATIONSHIPS';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('MRP', 'MLOG$_MRP_FORECAST_DATES';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('MRP', 'MLOG$_MRP_FORECAST_DESIGNA';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('MRP', 'MLOG$_MRP_FORECAST_ITEMS';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('MRP', 'MLOG$_MRP_SCHEDULE_DATES';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('INV', 'MLOG$_MTL_DEMAND';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('INV', 'MLOG$_MTL_ITEM_CATEGORIES';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('INV', 'MLOG$_MTL_MATERIAL_TRANSAC';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('INV', 'MLOG$_MTL_ONHAND_QUANTITIE';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('INV', 'MLOG$_MTL RELATED ITEMS';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('INV', 'MLOG$_MTL_RESERVATIONS';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('INV', 'MLOG$_MTL_SUPPLY';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('INV', 'MLOG$_MTL_SYSTEM_ITEMS_B';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('INV', 'MLOG$_MTL_TXN_REQUEST_LINE';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('INV', 'MLOG$_MTL_USER_DEMAND';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('INV', 'MLOG$_MTL_USER_SUPPLY';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('ONT', 'MLOG$_OE_ORDER_LINES_ALL';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('PO', 'MLOG$_PO_ACCEPTANCES';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('PO', 'MLOG$_PO_CHANGE_REQUESTS';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('PO', 'MLOG$_PO_SUPPLIER_ITEM_CAP';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('WIP', 'MLOG$_WIP_DISCRETE_JOBS';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('WIP', 'MLOG$_WIP_FLOW_SCHEDULES';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('WIP', 'MLOG$_WIP_LINES';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('WIP', 'MLOG$_WIP_MSC_OPEN_JOB_STA';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('WIP', 'MLOG$_WIP_OPERATIONS';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('WIP', 'MLOG$_WIP_OPERATION_NETWORK';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('WIP', 'MLOG$_WIP_OPERATION_RESOURCE';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('WIP', 'MLOG$_WIP_OP_RESOURCE_INST';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('WIP', 'MLOG$_WIP_REPEATITIVE_ITEMS';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('WIP', 'MLOG$_WIP_REPEATITIVE_SCHED';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('WIP', 'MLOG$_WIP_REQUIREMENT_OPER';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('WIP', 'MLOG$_WIP_SUB_OPERATION_RE';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('WSH', 'MLOG$_WSH_TRIPS';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('WSH', 'MLOG$_WSH_TRIP_STOPS';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('WSM', 'MLOG$_WSM_COPY_OPERATIONS';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('WSM', 'MLOG$_WSM_COPY_OP_NETWORKS';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('WSM', 'MLOG$_WSM_COPY_OP_RESOURCE';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('WSM', 'MLOG$_WSM_COPY_OP_RESOURCE1';
EXECUTE DBMS_STATS.LOCK_TABLE_STATS ('WSM', 'MLOG$_WSM_COPY_REQUIREMENT';

```

SQL#26 -- Create Temporary Tables to Analyze Changes in Snapshot Logs

```
drop table rcs_analysis_11i_temp;

drop table rcs_analysis_r12_temp;

create table rcs_analysis_11i_temp
(script_date          DATE,
 bom_structures_b   NUMBER,
 bom_cto_order_demand NUMBER,
 bom_components_b   NUMBER,
 bom_operational_rout NUMBER,
 bom_operation_networ NUMBER,
 bom_operation_resour NUMBER,
 bom_operation_sequen NUMBER,
 bom_resource_changes NUMBER,
 bom_substitute_compo NUMBER,
 bom_sub_operation_re NUMBER,
 mtl_demand          NUMBER,
 mtl_material_transac NUMBER,
 mtl_onhand_quantitie NUMBER,
 mtl_reservations   NUMBER,
 mtl_supply          NUMBER,
 mtl_system_items_b  NUMBER,
 mtl_txn_request_line NUMBER,
 mtl_user_demand     NUMBER,
 mtl_user_supply     NUMBER,
 mrp_forecast_dates  NUMBER,
 mrp_forecast_designa NUMBER,
 mrp_forecast_items   NUMBER,
 mrp_schedule_dates   NUMBER,
 oe_order_lines_all  NUMBER,
 po_acceptances      NUMBER,
 po_change_requests  NUMBER,
 po_supplier_item_cap NUMBER,
 wip_discrete_jobs   NUMBER,
 wip_flow_schedules  NUMBER,
 wip_lines            NUMBER,
 wip_operations       NUMBER,
 wip_operation_resour NUMBER,
 wip_repetitive_items NUMBER,
 wip_repetitive_sched NUMBER,
 wip_requirement_oper NUMBER);

create table rcs_analysis_r12_temp
(script_date          DATE,
 ahl_schedule_materia NUMBER,
 bom_operation_sequen NUMBER,
 wip_operation_resour NUMBER,
 wsm_copy_op_networks NUMBER,
 wsm_copy_op_resource  NUMBER,
 wsm_copy_op_resource1 NUMBER,
 po_change_requests   NUMBER,
 wip_operations       NUMBER,
 wip_op_resource_inst NUMBER,
 wip_repetitive_sched NUMBER,
 bom_structures_b     NUMBER,
 mtl_material_transac NUMBER,
 mtl_system_items_b   NUMBER,
 mtl_user_demand      NUMBER,
 mtl_user_supply      NUMBER,
 mrp_forecast_items   NUMBER,
 po_acceptances       NUMBER,
```

```
wip_requirement_oper NUMBER,  
wip_sub_operation_re NUMBER,  
bom_operation_resour NUMBER,  
bom_resource_changes NUMBER,  
bom_sub_operation_re NUMBER,  
mtl_item_categories NUMBER,  
mrp_forecast_designa NUMBER,  
wip_flow_schedules NUMBER,  
wip_repetitive_items NUMBER,  
wsm_copy_operations NUMBER,  
bom_components_b NUMBER,  
mrp_forecast_dates NUMBER,  
mrp_schedule_dates NUMBER,  
oe_order_lines_all NUMBER,  
wip_lines NUMBER,  
bom_operation_networ NUMBER,  
mtl_demand NUMBER,  
mtl_onhand_quantitie NUMBER,  
po_supplier_item_cap NUMBER,  
wip_discrete_jobs NUMBER,  
bom_operational_rout NUMBER,  
bom_substitute_compo NUMBER,  
eam_wo_relationships NUMBER,  
mtl_reservations NUMBER,  
mtl_supply NUMBER,  
bom_res_instance_cha NUMBER,  
mtl_txn_request_line NUMBER,  
wip_operation_networ NUMBER,  
wsh_trips NUMBER,  
wsh_trip_stops NUMBER,  
wsm_copy_requirement NUMBER);
```

SQL #27 -- Check MLOG\$ Tables to Get Row Counts

```
DECLARE

    -- R12 Count Variables
    v_r12_ahl_schedule_materia      NUMBER;
    v_r12_bom_operation_sequen     NUMBER;
    v_r12_wip_operation_resour     NUMBER;
    v_r12_wsm_copy_op_networks    NUMBER;
    v_r12_wsm_copy_op_resource    NUMBER;
    v_r12_wsm_copy_op_resourcel   NUMBER;
    v_r12_po_change_requests     NUMBER;
    v_r12_wip_operations          NUMBER;
    v_r12_wip_op_resource_inst    NUMBER;
    v_r12_wip_repetitive_sched    NUMBER;
    v_r12_bom_structures_b        NUMBER;
    v_r12_mtl_material_transac   NUMBER;
    v_r12_mtl_system_items_b      NUMBER;
    v_r12_mtl_user_demand         NUMBER;
    v_r12_mtl_user_supply         NUMBER;
    v_r12_mrp_forecast_items     NUMBER;
    v_r12_po_acceptances         NUMBER;
    v_r12_wip_requirement_oper    NUMBER;
    v_r12_wip_sub_operation_re    NUMBER;
    v_r12_bom_operation_resour   NUMBER;
    v_r12_bom_resource_changes   NUMBER;
    v_r12_bom_sub_operation_re    NUMBER;
    v_r12_mtl_item_categories    NUMBER;
    v_r12_mrp_forecast_designa   NUMBER;
    v_r12_wip_flow_schedules     NUMBER;
    v_r12_wip_repetitive_items   NUMBER;
    v_r12_wsm_copy_operations    NUMBER;
    v_r12_bom_components_b        NUMBER;
    v_r12_mrp_forecast_dates     NUMBER;
    v_r12_mrp_schedule_dates     NUMBER;
    v_r12_oe_order_lines_all     NUMBER;
    v_r12_wip_lines               NUMBER;
    v_r12_bom_operation_networ   NUMBER;
    v_r12_mtl_demand              NUMBER;
    v_r12_mtl_onhand_quantitie   NUMBER;
    v_r12_po_supplier_item_cap    NUMBER;
    v_r12_wip_discrete_jobs       NUMBER;
    v_r12_bom_operational_rout   NUMBER;
    v_r12_bom_substitute_compo   NUMBER;
    v_r12_eam_wo_relationships   NUMBER;
    v_r12_mtl_reservations        NUMBER;
    v_r12_mtl_supply              NUMBER;
    v_r12_bom_res_instance_cha   NUMBER;
    v_r12_mtl_txn_request_line   NUMBER;
    v_r12_wip_operation_networ   NUMBER;
    v_r12_wsh_trips               NUMBER;
    v_r12_wsh_trip_stops          NUMBER;
    v_r12_wsm_copy_requirement    NUMBER;

    -- Other Variables
    v_script_date                  DATE;

BEGIN

    SELECT sysdate  INTO v_script_date from dual;
    SELECT count(*) INTO v_r12_ahl_schedule_materia  FROM ahl.
mlog$_ahl_schedule_materia;
    SELECT count(*) INTO v_r12_bom_operation_sequen  FROM bom.
mlog$_bom_operation_sequen;
```

```

SELECT count(*) INTO v_r12_wip_operation_resour  FROM wip.
mlog$_wip_operation_resour;
  SELECT count(*) INTO v_r12_wsm_copy_op_networks  FROM wsm.
mlog$_wsm_copy_op_networks;
  SELECT count(*) INTO v_r12_wsm_copy_op_resource  FROM wsm.
mlog$_wsm_copy_op_resource;
  SELECT count(*) INTO v_r12_wsm_copy_op_resource1 FROM wsm.
mlog$_wsm_copy_op_resource1;
  SELECT count(*) INTO v_r12_po_change_requests  FROM po.
mlog$_po_change_requests;
  SELECT count(*) INTO v_r12_wip_operations  FROM wip.
mlog$_wip_operations;
  SELECT count(*) INTO v_r12_wip_op_resource_inst  FROM wip.
mlog$_wip_op_resource_inst;
  SELECT count(*) INTO v_r12_wip_repetitive_sched  FROM wip.
mlog$_wip_repetitive_sched;
  SELECT count(*) INTO v_r12_bom_structures_b  FROM bom.
mlog$_bom_structures_b;
  SELECT count(*) INTO v_r12_mtl_material_transac  FROM inv.
mlog$_mtl_material_transac;
  SELECT count(*) INTO v_r12_mtl_system_items_b  FROM inv.
mlog$_mtl_system_items_b;
  SELECT count(*) INTO v_r12_mtl_user_demand  FROM inv.
mlog$_mtl_user_demand;
  SELECT count(*) INTO v_r12_mtl_user_supply  FROM inv.
mlog$_mtl_user_supply;
  SELECT count(*) INTO v_r12_mrp_forecast_items  FROM mrp.
mlog$_mrp_forecast_items;
  SELECT count(*) INTO v_r12_po_acceptances  FROM po.
mlog$_po_acceptances;
  SELECT count(*) INTO v_r12_wip_requirement_oper  FROM wip.
mlog$_wip_requirement_oper;
  SELECT count(*) INTO v_r12_wip_sub_operation_re  FROM wip.
mlog$_wip_sub_operation_re;
  SELECT count(*) INTO v_r12_bom_operation_resour  FROM bom.
mlog$_bom_operation_resour;
  SELECT count(*) INTO v_r12_bom_resource_changes  FROM bom.
mlog$_bom_resource_changes;
  SELECT count(*) INTO v_r12_bom_sub_operation_re  FROM bom.
mlog$_bom_sub_operation_re;
  SELECT count(*) INTO v_r12_mtl_item_categories  FROM inv.
mlog$_mtl_item_categories;
  SELECT count(*) INTO v_r12_mrp_forecast_designa  FROM mrp.
mlog$_mrp_forecast_designa;
  SELECT count(*) INTO v_r12_wip_flow_schedules  FROM wip.
mlog$_wip_flow_schedules;
  SELECT count(*) INTO v_r12_wip_repetitive_items  FROM wip.
mlog$_wip_repetitive_items;
  SELECT count(*) INTO v_r12_wsm_copy_operations  FROM wsm.
mlog$_wsm_copy_operations;
  SELECT count(*) INTO v_r12_bom_components_b  FROM bom.
mlog$_bom_components_b;
  SELECT count(*) INTO v_r12_mrp_forecast_dates  FROM mrp.
mlog$_mrp_forecast_dates;
  SELECT count(*) INTO v_r12_mrp_schedule_dates  FROM mrp.
mlog$_mrp_schedule_dates;
  SELECT count(*) INTO v_r12_oe_order_lines_all  FROM ont.
mlog$_oe_order_lines_all;
  SELECT count(*) INTO v_r12_wip_lines  FROM wip.
mlog$_wip_lines;
  SELECT count(*) INTO v_r12_bom_operation_networ  FROM bom.
mlog$_bom_operation_networ;
  SELECT count(*) INTO v_r12_mtl_demand  FROM inv.
mlog$_mtl_demand;
  SELECT count(*) INTO v_r12_mtl_onhand_quantitie  FROM inv.
mlog$_mtl_onhand_quantitie;

```

```

SELECT count(*) INTO v_r12_po_supplier_item_cap    FROM po.
mlog$_po_supplier_item_cap;
    SELECT count(*) INTO v_r12_wip_discrete_jobs      FROM wip.
mlog$_wip_discrete_jobs;
    SELECT count(*) INTO v_r12_bom_operational_rout   FROM bom.
mlog$_bom_operational_rout;
    SELECT count(*) INTO v_r12_bom_substitute_compo   FROM bom.
mlog$_bom_substitute_compo;
    SELECT count(*) INTO v_r12_eam_wo_relationships   FROM eam.
mlog$_eam_wo_relationships;
    SELECT count(*) INTO v_r12_mtl_reservations      FROM inv.
mlog$_mtl_reservations;
    SELECT count(*) INTO v_r12_mtl_supply            FROM inv.
mlog$_mtl_supply;
    SELECT count(*) INTO v_r12_bom_res_instance_cha  FROM bom.
mlog$_bom_res_instance_cha;
    SELECT count(*) INTO v_r12_mtl_txn_request_line  FROM inv.
mlog$_mtl_txn_request_line;
    SELECT count(*) INTO v_r12_wip_operation_networ  FROM wip.
mlog$_wip_operation_networ;
    SELECT count(*) INTO v_r12_wsh_trips             FROM wsh.
mlog$_wsh_trips;
    SELECT count(*) INTO v_r12_wsh_trip_stops        FROM wsh.
mlog$_wsh_trip_stops;
    SELECT count(*) INTO v_r12_wsm_copy_requirement  FROM wsm.
mlog$_wsm_copy_requirement;

    INSERT INTO rcs_analysis_r12_temp
        (script_date
         ,ahl_schedule_materia
         ,bom_operation_sequen
         ,wip_operation_resour
         ,wsm_copy_op_networks
         ,wsm_copy_op_resource
         ,wsm_copy_op_resourcel
         ,po_change_requests
         ,wip_operations
         ,wip_op_resource_inst
         ,wip_repetitive_sched
         ,bom_structures_b
         ,mtl_material_transac
         ,mtl_system_items_b
         ,mtl_user_demand
         ,mtl_user_supply
         ,mrp_forecast_items
         ,po_acceptances
         ,wip_requirement_oper
         ,wip_sub_operation_re
         ,bom_operation_resour
         ,bom_resource_changes
         ,bom_sub_operation_re
         ,mtl_item_categories
         ,mrp_forecast_designa
         ,wip_flow_schedules
         ,wip_repetitive_items
         ,wsm_copy_operations
         ,bom_components_b
         ,mrp_forecast_dates
         ,mrp_schedule_dates
         ,oe_order_lines_all
         ,wip_lines
         ,bom_operation_networ
         ,mtl_demand
         ,mtl_onhand_quantitie
         ,po_supplier_item_cap
         ,wip_discrete_jobs

```

```

,bom_operational_rout
,bom_substitute_compo
,eam_wo_relationships
,mtl_reservations
,mtl_supply
,bom_res_instance_cha
,mtl_txn_request_line
,wip_operation_networ
,wsh_trips
,wsh_trip_stops
,wsm_copy_requirement)
VALUES
  (v_script_date
  ,v_r12_ahl_schedule_materia
  ,v_r12_bom_operation_sequen
,v_r12_wip_operation_resour
,v_r12_wsm_copy_op_networks
,v_r12_wsm_copy_op_resource
,v_r12_wsm_copy_op_resourcel
,v_r12_po_change_requests
,v_r12_wip_operations
,v_r12_wip_op_resource_inst
,v_r12_wip_repetitive_sched
,v_r12_bom_structures_b
,v_r12_mtl_material_transac
,v_r12_mtl_system_items_b
,v_r12_mtl_user_demand
,v_r12_mtl_user_supply
,v_r12_mrp_forecast_items
,v_r12_po_acceptances
,v_r12_wip_requirement_oper
,v_r12_wip_sub_operation_re
,v_r12_bom_operation_resour
,v_r12_bom_resource_changes
,v_r12_bom_sub_operation_re
,v_r12_mtl_item_categories
,v_r12_mrp_forecast_designa
,v_r12_wip_flow_schedules
,v_r12_wip_repetitive_items
,v_r12_wsm_copy_operations
,v_r12_bom_components_b
,v_r12_mrp_forecast_dates
,v_r12_mrp_schedule_dates
,v_r12_oe_order_lines_all
,v_r12_wip_lines
,v_r12_bom_operation_networ
,v_r12_mtl_demand
,v_r12_mtl_onhand_quantitie
,v_r12_po_supplier_item_cap
,v_r12_wip_discrete_jobs
,v_r12_bom_operational_rout
,v_r12_bom_substitute_compo
,v_r12_eam_wo_relationships
,v_r12_mtl_reservations
,v_r12_mtl_supply
,v_r12_bom_res_instance_cha
,v_r12_mtl_txn_request_line
,v_r12_wip_operation_networ
,v_r12_wsh_trips
,v_r12_wsh_trip_stops
,v_r12_wsm_copy_requirement);

END;
/
commit;

```

Tablespace SQL Commands

SQL #28 -- Get Table Sizes for Tables Exceeding 10 MB

```
set pages 50
set linesize 200
col bytes format 999,999,999,999

SELECT
    owner,
    substr(segment_name,1,40) Table_or_Index_Name,
    bytes table_or_index_size,
    tablespace_name,
    partition_name,
    pct_increase,
    blocks,
    extents,
    initial_extent,
    next_extent,
    Min_extents,
    max_extents
FROM
    dba_segments
WHERE
    segment_name like 'MSC%' and
    bytes > 10000000
    order by tablespace_name, bytes desc;
-- When running data collections and needing to check sizes, the
segment_name should also be checked in the transaction instance for
'MRP%' 'WIP%' 'MTL%' 'BOM%' 'OE%' 'MSD%' 'MSD%'
```

SQL #29 -- Get Tablespace Information for Data Collection Schemas

```
SELECT
    fav.application_name app_name,
    fav.application_short_name app_s_name,
    fav.application_id app_id,
    fpi.tablespace,
    fpi.index_tablespace
FROM
    fnd_application_vl fav,
    fnd_product_installations fpi
WHERE
    fav.application_id = fpi.application_id and
    application_short_name in
    ('WIP', 'MSC', 'MSD', 'MSR', 'INV', 'ONT', 'OE', 'BOM', 'MRP', 'AHL', 'PO')
    order by 2;
```

SQL #30 -- Review Tables and Indices with Zero Percent Increase

This SQL checks that all tables and indices have PCT_INCREASE=0. If that tablename returned ends in MV or SN, then ignore the output.

```

SELECT
  owner,
  table_name,
  pct_increase
FROM
  all_tables
WHERE
  table_name like 'MSC%'  and
  PCT_INCREASE !=0;

SELECT
  owner,
  table_name,
  index_name,
  pct_increase
FROM
  all_indexes
WHERE
  index_name like 'MSC%'  and
  PCT_INCREASE !=0;

SELECT *
  table_owner,
  table_name,
  partition_name,
  pct_increase
FROM
  all_tab_partitions
WHERE
  table_name like 'MSC%'  AND
  PCT_INCREASE !=0;

```

SQL #31 -- Check the Tablespace Settings

This SQL checks the tablespace settings. PCT_INCREASE should be 0 here also. The tablespace_name might not be the same as below if it was changed during the installation.

```

select
  TABLESPACE_NAME,
  INITIAL_EXTENT,
  NEXT_EXTENT,
  MAX_EXTENTS,
  PCT_INCREASE
from
  DBA_TABLESPACES
where
  TABLESPACE_NAME in
  ('APPS_TS_TX_DATA',
  'APPS_TS_TX_IDX',
  'APPS_TS_SUMMARY',
  'APPS_TS_INTERFACE',
  'APPS_TS_NOLOGGING',
  'APPS_TS_ARCHIVE');
-- FOR OLD 11.5.9 and below tablespace mode being used
where
  TABLESPACE_NAME like ('MSC%');

```

SQL #32 -- Check the Total Space Available in the Tablespaces

This SQL checks the total space available in the tablespaces.

```

SELECT
  tablespace_name ,
  round(sum(BYTES)/power(2,20)) TOTAL_MB
FROM
  dba_data_files
where
  TABLESPACE_NAME in
  ('APPS_TS_TX_DATA',
  'APPS_TS_TX_IDX',
  'APPS_TS_SUMMARY',
  'APPS_TS_INTERFACE',
  'APPS_TS_NOLOGGING',
  'APPS_TS_ARCHIVE')
group by TABLESPACE_NAME;
-- FOR OLD 11.5.9 and below tablespace mode being used
where
  TABLESPACE_NAME like ('MSC%');

```

SQL #33 -- Check Total Free Space in MSC Tablespace

```

SELECT
  TABLESPACE_NAME ,
  round(sum(BYTES)/power(2,20)) FREE_MB
FROM
  (select    BYTES,
  TABLESPACE_NAME   from DBA_FREE_SPACE
where
  TABLESPACE_NAME in
  ('APPS_TS_TX_DATA',
  'APPS_TS_TX_IDX',
  'APPS_TS_SUMMARY',
  'APPS_TS_INTERFACE',
  'APPS_TS_NOLOGGING',
  'APPS_TS_ARCHIVE'))
group by TABLESPACE_NAME;
-- FOR OLD 11.5.9 and below tablespace mode being used
where
  TABLESPACE_NAME like ('MSC%');

```

SQL #34 -- List Individual Datafiles and Space Allocation

This SQL will list the individual datafiles and space allocated to each file.

```

SELECT
  substr(tablespace_name,1,8) TS_NAME,
  substr(file_name,1,40) FILE_NAME,
  bytes/1024/1024 MB,
  autoextensible,
  file_id
FROM
  dba_data_files
where
  TABLESPACE_NAME in
  ('APPS_TS_TX_DATA',
  'APPS_TS_TX_IDX',
  'APPS_TS_SUMMARY',
  'APPS_TS_INTERFACE',
  'APPS_TS_NOLOGGING',
  'APPS_TS_ARCHIVE')
order by tablespace_name, file_id;
-- FOR OLD 11.5.9 and below tablespace mode being used
where
  TABLESPACE_NAME like ('MSC%');

```

SQL #35 -- Check Individual Datafiles for Free Space Available

```
SELECT
  tablespace_name,
  sum(bytes/1024/1024)  MB,
  file_id
FROM
  dba_free_space
where
  TABLESPACE_NAME in
  ( 'APPS_TS_TX_DATA',
  'APPS_TS_TX_IDX',
  'APPS_TS_SUMMARY',
  'APPS_TS_INTERFACE',
  'APPS_TS_NOLOGGING',
  'APPS_TS_ARCHIVE')
group by tablespace_name, file_id
order by tablespace_name, file_id;
-- FOR OLD 11.5.9 and below tablespace mode being used
where
  TABLESPACE_NAME like ('MSC%');
group by tablespace_name, file_id
order by tablespace_name, file_id;
```

List of VCP Partitioned Tables

This appendix covers the following topics:

- Summary List of Partitioned Tables
- ODS and PDS Partitioned Table
- Partitioned Staging Tables for Data Collections
- MV Partitioned Tables
- Partitioned Tables for APCC

Summary List of Partitioned Tables

The list below shows the partitioned tables that are created in the MSC Schema for VCP. Understanding and managing the size of the VCP tablespaces (MSCD (Data) and MSCX (Indexes)) is discussed in Managing Tablespaces, page 4-4.

There are several reasons for these partitions: They allow for quick access to data striped by partition/plan_id, we can truncate a partition much quicker than deleting data for a specific plan_id, and we can use exchange partition function to swap partitions with TEMP tables.

We partition tables for different purposes in the applications:

- MSC base tables are used to store collected data (ODS instance partition) or for plan data (PDS plan partition) or can be used for BOTH ODS and PDS purposes.
- MSC_ST% staging tables are used for data collections (excluding MSC_ST%F).
- MSC%F tables are used for Advanced Planning Command Center (APCC).
- The %_MV tables (Materialized View) are special and have a separate partitioning scheme.

This list was compiled using the following SQL to help segregate the different types of tables.

Summary list of tables:

```
#1 - Get all tables
select table_name
from dba_part_tables
where owner = 'MSC'

#2 - Get the ODS and PDS tables
SELECT
table_name
from dba_part_tables
where owner = 'MSC'
and table_name not in
(select table_name from dba_part_tables
where owner = 'MSC' and table_name like 'MSC_ST%'
-- tables for Collections Staging tables
union
select table_name from dba_part_tables
where owner = 'MSC' and table_name like 'MSC%F' -- tables for APCC
)
order by table_name;

#3 ? Get Staging tables for Data Collections
select table_name
from dba_part_tables
where owner = 'MSC'
and table_name like 'MSC_ST%'
and table_name not in
(select table_name from dba_part_tables
where owner = 'MSC' and table_name like 'MSC%F' -- tables for APCC
) order by table_name;

#4 ? Get APCC tables for Archive Plan data and APCC staging tables used
to populate the APCC Plan Data
select table_name
from dba_part_tables
where owner = 'MSC'
and table_name like 'MSC%F'
order by table_name;
```

ODS and PDS Partitioned Table

These tables are used for either collected data ODS, planning data PDS, or both. Partitions are created accordingly.

TABLE NAMES FOR ODS AND PDS BASE TABLES	PDS/OPS/BOTH
--	---------------------

MSC_ALLOC_DEMANDS	Both
MSC_ALLOC_SUPPLIES	Both
MSC_ANALYSIS_AGGREGATE	PDS

TABLE NAMES FOR ODS AND PDS BASE TABLES	PDS/OPS/BOTH
--	---------------------

MSC_ATP_PEGGING	PDS
-----------------	-----

MSC_ATP_SUMMARY_RES	Both
---------------------	------

MSC_ATP_SUMMARY_SD	Both
--------------------	------

MSC_ATP_SUMMARY_SO	ODS
--------------------	-----

MSC_ATP_SUMMARY_SUP	Both
---------------------	------

MSC_BOMS	Both
----------	------

MSC_BOM_COMPONENTS	Both
--------------------	------

MSC_CRITICAL_PATHS	PDS
--------------------	-----

MSC_DELIVERY_DETAILS	ODS
----------------------	-----

MSC_DEMANDS	Both
-------------	------

MSC_DMD_SCN_METRICS	Both
---------------------	------

MSC_DOC_ATTACHMENTS	ODS
---------------------	-----

MSC_EAM_ACT_ASSOCIATIONS	ODS
--------------------------	-----

MSC_EAM_ASSET_EQUIP_DTLS	ODS
--------------------------	-----

MSC_EXCEPTION_DETAILS	PDS
-----------------------	-----

MSC_EXC_DETAILS_ALL	PDS
---------------------	-----

MSC_FULL_PEGGING	PDS
------------------	-----

MSC_ITEM_CATEGORIES	ODS
---------------------	-----

MSC_ITEM_EXCEPTIONS	PDS
---------------------	-----

MSC_ITEM_HIERARCHY_MV	Special **
-----------------------	------------

TABLE NAMES FOR ODS AND PDS BASE TABLES	PDS/OPS/BOTH
--	---------------------

MSC_ITEM_SUBSTITUTES	Both
MSC_JOB_OPERATIONS	Both
MSC_JOB_OPERATION_NETWORKS	Both
MSC_JOB_OP_RESOURCES	Both
MSC_JOB_OP_RES_INSTANCES	Both
MSC_JOB_REQUIREMENT_OPS	Both
MSC_LONG_TEXT	ODS
MSC_NET_RESOURCE_AVAIL	Both
MSC_NET_RES_INST_AVAIL	Both
MSC_OPERATION_RESOURCES	Both
MSC_OPERATION_RESOURCE_SEQS	Both
MSC_ORG_AGGR_IBUC	ODS
MSC_ORP_MATERIAL_PLANS	PDS
MSC_ORP_RESOURCE_PLANS	PDS
MSC_PART_DEMANDS	PDS
MSC_PART_PEGGING	PDS
MSC_PART_SUPPLIES	PDS
MSC_PLAN_CONSTRAINTS	PDS
MSC_PQ_RESULTS	PDS
MSC_REGIONS	ODS

TABLE NAMES FOR ODS AND PDS BASE TABLES	PDS/OPS/BOTH
--	---------------------

MSC_REGION_LOCATIONS	ODS
MSC_RESOURCE_HIERARCHY_MV	Special **
MSC_RESOURCE_INSTANCE_REQS	Both
MSC_RESOURCE_REQUIREMENTS	Both
MSC_ROUTINGS	Both
MSC_ROUTING_OPERATIONS	Both
MSC_RP_CTB_DONOR_COMPONENTS	PDS
MSC_RP_CTB_ORDER_COMPONENTS	PDS
MSC_RP_ITEM_UPDATES	PDS
MSC_RP_KPI	PDS
MSC_RP_RELEASED_ORDERS	PDS
MSC_SALES_ORDERS	ODS
MSC_SHORT_TEXT	ODS
MSC_SINGLE_LVL_PEG	PDS
MSC_SR_PITEM_EXCEPTIONS	PDS
MSC_SUPPLIER_REQUIREMENTS	PDS
MSC_SUPPLIES	Both
MSC_SYSTEM_ITEMS	Both
MSC_VISITS	ODS
MSC_WORK_BREAKDOWN_STRUCT	ODS

TABLE NAMES FOR ODS AND PDS BASE TABLES	PDS/OPS/BOTH
MSC_WO_ATTRIBUTES	ODS
MSC_WO_MILESTONES	ODS
MSC_WO_OPERATION_REL	ODS
MSC_WO_SUB_COMP	Both
MSC_WO_TASK_HIERARCHY	ODS
MSC_ZONE_REGIONS	ODS

Special ** see MV partitioned tables below for more information

Partitioned Staging Tables for Data Collections

There are partitions for the MSC_ST% staging tables that are controlled by the entries made in the Instances form from the Advanced Planning Administrator responsibility. The MSC_ST% staging tables are used in data collections and inserted when the Planning Data Pull process pulls data from the EBS source tables/views.

The partitions DEF and LEG are default partitions created at install. When you enter a line in the instances form, then triggers are used to create a specific partition for that ODS Instance partition

Example:

When a single ODS partition is created, then the list of partitions for a single table would appear as follows if the ODS instance partition is 21:

```
SELECT
  TABLE_NAME,
  PARTITION_NAME
FROM
  ALL_TAB_PARTITIONS
WHERE
  TABLE_NAME like 'MSC_ST_SYSTEM_ITEMS'
```

Table Name	Partition Name
MSC_ST_SYSTEM_ITEMS	ST_SYSTEM_ITEMS_LEG
MSC_ST_SYSTEM_ITEMS	ST_SYSTEM_ITEMS_DEF

Table Name	Partition Name
MSC_ST_SYSTEM_ITEMS	SYSTEM_ITEMS_21

Note: The instance partition 21 does not contain the ST prefix for the partition name. Also, these staging tables do not have __ (double underscore) like the base table for collected data and only use a _ (single underscore).

Complete listing:

TABLE NAMES FOR PARTITIONED STAGING TABLES

MSC_ST_BOMS

MSC_ST_BOM_COMPONENTS

MSC_ST_CALENDAR_DATES

MSC_ST_DELIVERY_DETAILS

MSC_ST_DEMANDS

MSC_ST_DEPARTMENT_RESOURCES

MSC_ST_DOC_ATTACHMENTS

MSC_ST_EAM_ACT_ASSOCIATIONS

MSC_ST_EAM_ASSET_EQUIP_DTLS

MSC_ST_ITEM_CATEGORIES

MSC_ST_ITEM_SUBSTITUTES

MSC_ST_JOB_OPERATIONS

MSC_ST_JOB_OP_RESOURCES

MSC_ST_JOB_REQUIREMENT_OPS

TABLE NAMES FOR PARTITIONED STAGING TABLES

MSC_ST_LONG_TEXT

MSC_ST_NET_RESOURCE_AVAIL

MSC_ST_NET_RES_INST_AVAIL

MSC_ST_OPERATION_COMPONENTS

MSC_ST_OPERATION_RESOURCES

MSC_ST_OPERATION_RESOURCE_SEQS

MSC_ST_PROCESS_EFFECTIVITY

MSC_ST_REGIONS

MSC_ST_REGION_LOCATIONS

MSC_ST_RESOURCE_INSTANCE_REQS

MSC_ST_RESOURCE_REQUIREMENTS

MSC_ST_ROUTING_OPERATIONS

MSC_ST_SALES_ORDERS

MSC_ST_SHIFT_DATES

MSC_ST_SHORT_TEXT

MSC_ST_SOURCING_HISTORY

MSC_ST_SOURCING_RULES

MSC_ST_SUPPLIES

MSC_ST_SYSTEM_ITEMS

MSC_ST_TRADING_PARTNERS

TABLE NAMES FOR PARTITIONED STAGING TABLES

MSC_ST_TRADING_PARTNER_SITES

MSC_ST_VISITS

MSC_ST_WORK_BREAKDOWN_STRUCT

MSC_ST_WO_ATTRIBUTES

MSC_ST_WO_MILESTONES

MSC_ST_WO_OPERATION_REL

MSC_ST_WO_SUB_COMP

MSC_ST_WO_TASK_HIERARCHY

MSC_ST_ZN_AGGR_IBUC

MSC_ST_ZONE_REGIONS

MV Partitioned Tables

Two tables are created as Materialized Views: MSC_RESOURCE_HIERARCHY_MV and MSC_ITEM_HIERARCHY_MV. They do not follow the standard partition creation rules.

Important: These tables should never be manipulated. They are not removed or changed by any process such as Drop Partition or Create APS Partition requests. SQL #8 -- Drop Plan Partitions, page D-6 drops plan partitions but does not drop these partitions. When using SQL #7 -- Delete All Tables Where the SR_INSTANCE_ID Column Appears, page D-5 and SQL #11 -- Update All Tables Where the SR_INSTANCE_ID Column Appears, page D-1, these tables error in the script. This is not a problem and can be ignored. When the views are reinitialized, the proper information is written to the tables. You can use the SQL below to view information about these tables.

```

Select table_name, partition_name
From all_tab_partitions
Where table_name in
( 'MSC_ITEM_HIERARCHY_MV' , 'MSC_RESOURCE_HIERARCHY_MV' )
order by table_name

```

The installation of VCP creates the following partitions for these tables. These are the only tables where 0 partition may have data. This is expected if the VCP Plan partition_number is >31; Partition_1 is populated for plan partitions >31 and <61; Partition_2 is used for >61.

Table Name	Partition Name
MSC_ITEM_HIERARCHY_MV	ITEM_HIERARCHY_0
MSC_ITEM_HIERARCHY_MV	ITEM_HIERARCHY_1
MSC_ITEM_HIERARCHY_MV	ITEM_HIERARCHY_2
MSC_RESOURCE_HIERARCHY_MV	RES_HIERARCHY_0
MSC_RESOURCE_HIERARCHY_MV	RES_HIERARCHY_1
MSC_RESOURCE_HIERARCHY_MV	RES_HIERARCHY_2

Partitioned Tables for APCC

Advanced Planning Command Center also uses partitioned tables. APCC handles its partitioning separately and automatically. These tables are NOT part of standard data collections and ASCP, IO, DRP or SPP plans. They are not affected by Drop Partition or Create APS Partitions requests. There are also APCC-specific MSC_ST%F staging tables. These are for APCC to use when moving data from PDS plan output to the APCC tables. They are not used in standard data collections.

TABLE NAMES FOR APCC PLAN TABLES AND APCC STAGING TABLES

MSC_BUDGETS_F

MSC_COSTS_F

MSC_DEMANDS_CUM_F

MSC_DEMANDS_F

MSC_DEMANTRA_F
MSC_EXCEPTIONS_F
MSC_ITEMS_F
MSC_ITEM_INVENTORY_F
MSC_ITEM_ORDERS_F
MSC_ITEM_WIPS_F
MSC_RESOURCES_CUM_F
MSC_RESOURCES_F
MSC_SUPPLIERS_F
MSC_SUPPLIES_F
MSC_ST_BUDGETS_F
MSC_ST_COSTS_F
MSC_ST_DEMANDS_CUM_F
MSC_ST_DEMANDS_F
MSC_ST_DEMANTRA_F
MSC_ST_DEMANTRA_ODS_F
MSC_ST_EXCEPTIONS_F
MSC_ST_ITEMS_F
MSC_ST_ITEM_INVENTORY_F
MSC_ST_ITEM_ORDERS_F
MSC_ST_ITEM_WIPS_F

MSC_ST_RESOURCES_CUM_F

MSC_ST_RESOURCES_F

MSC_ST_SUPPLIERS_F

MSC_ST_SUPPLIES_F

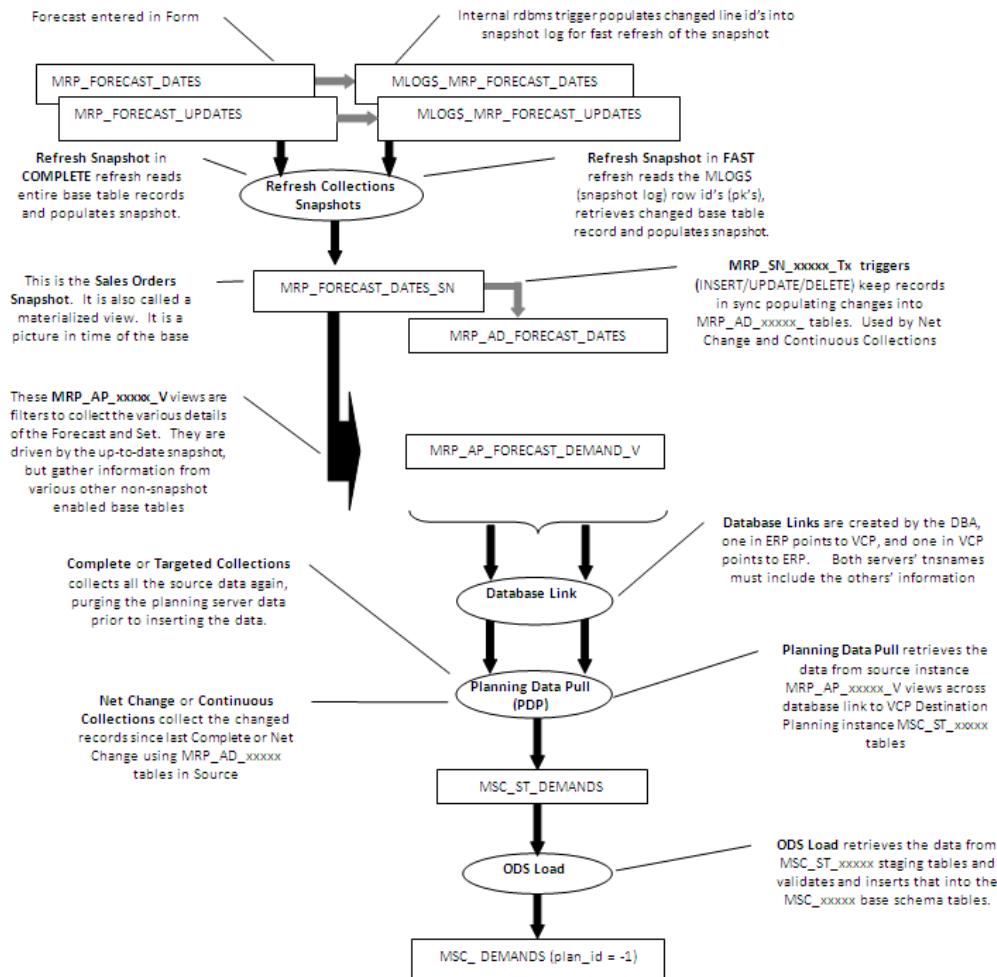
F

Detailed Integration Flows

This appendix covers the following topics:

- Forecast to ASCP Data Flow
- MDS to VCP Data Flow
- Purchasing Supply to ASCP Data Flow
- Sales Orders to ASCP Data Flow
- Sales Orders to MRP Data Flow
- WIP to ASCP Data Flow

Forecast to ASCP Data Flow



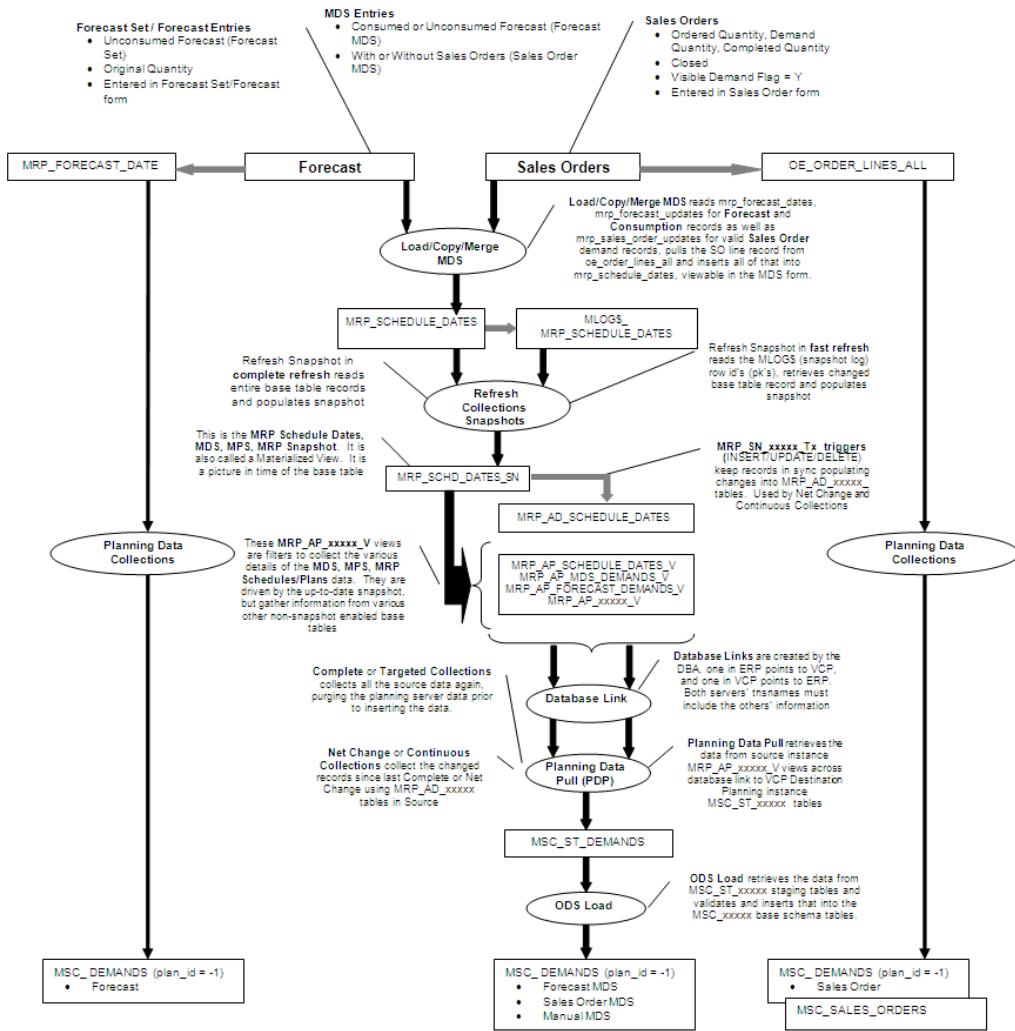
Data Flow Description:

1. New sales order entered with visible_demand_flag = Y and scheduled_ship_date not null.
2. Snapshot does not contain that new entered sales order.
3. The MRP_AP_xxxxx_V views do not contain that new sales order.
4. Internal rdbms trigger populates the newly entered sales order RowID into Snapshot Log (MLOG\$ objects).
5. Launch ASCP Data Collections.
6. Refresh Snapshot reads row id from MLOG\$.

7. Refresh Snapshot gets record from base table using row id.
8. Refresh Snapshot Populates/Updates the snapshot with current table record information.
9. As the snapshot is updated, the MRP_SN_xxxxx_T triggers fire.
10. The triggers update/insert into MRP_AD_SALES_ORDERS table.
11. The MRP_AP_xxxxx_V views now have the new sales order.
12. Planning Data Pull reads the MRP_AP_xxxxx_V views.
13. Planning Data Pull validates and calculates the sales order's data.
14. Planning Data Pull pulls data from the EBS Source instance over the database links.
 - ERP-based programs look in mrp_ap_apps_instances_all on source instance to get the link name to VCP.
 - VCP-based programs look in msc_apps_instances planning instances to get the link name to ERP.
15. Planning Data Pull inserts that pulled data into the MSC_ST_xxxxx Staging Tables.
16. ODS Load reads the MSC_ST_xxxxx Staging Tables.
17. ODS Load validates and calculates the sales order's data.
18. ODS Load purges existing ODS data and loads new/change records using either Complete or Net Change methods.
19. ODS Load data is hard coded with PLAN_ID = -1 for Collected Raw Source Data.
20. ODS Data is read by the MBP Memory Based Planner.
21. ODS Data is read by Production Scheduler.
22. ODS Data is read by Strategic Network Optimization.
23. ODS Data is read by Demand Planning.

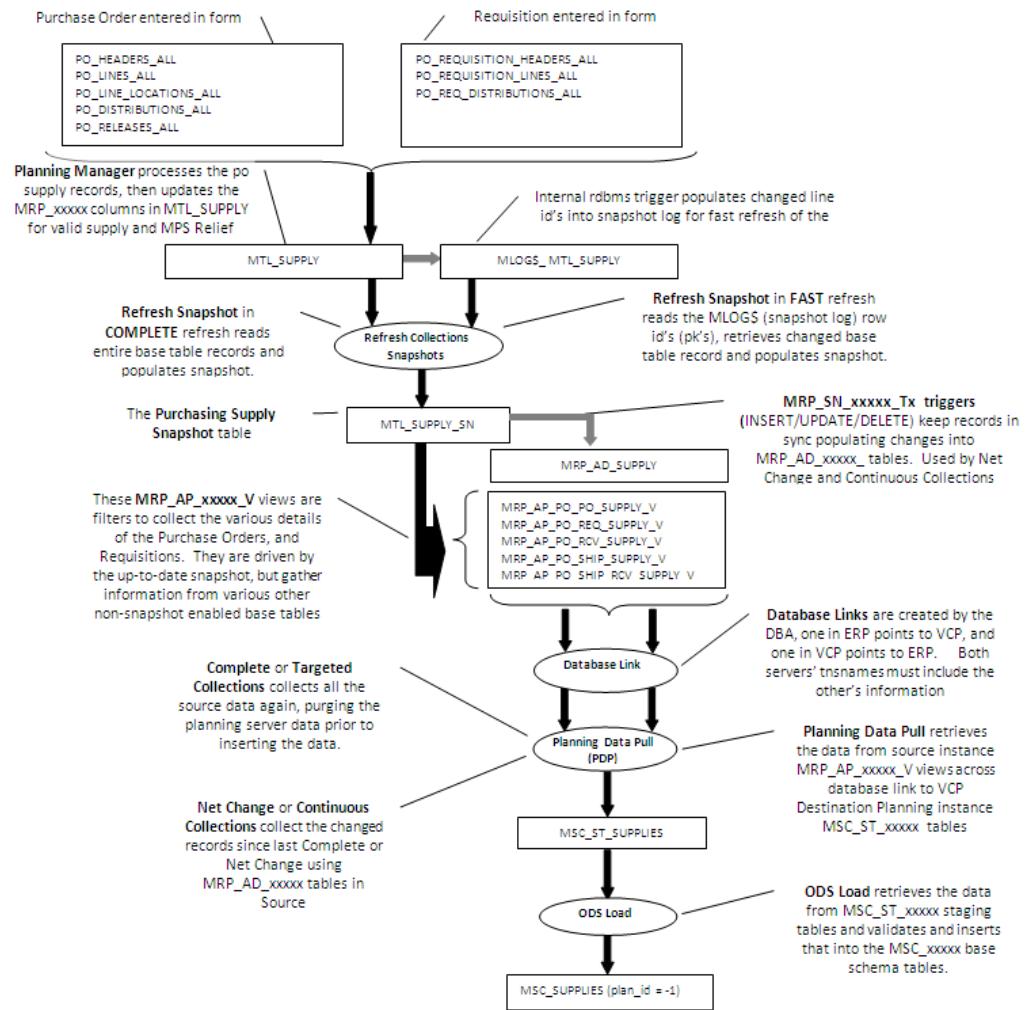
See My Oracle Support Note #421787.1 -- "MDS, Forecast, SO to ASCP Data Flow Diagnostics" for more information about troubleshooting and diagnostics.

MDS to VCP Data Flow



See My Oracle Support Note #421787.1 -- "MDS, Forecast, SO to ASCP Data Flow Diagnostics" for more information about troubleshooting and diagnostics.

Purchasing Supply to ASCP Data Flow

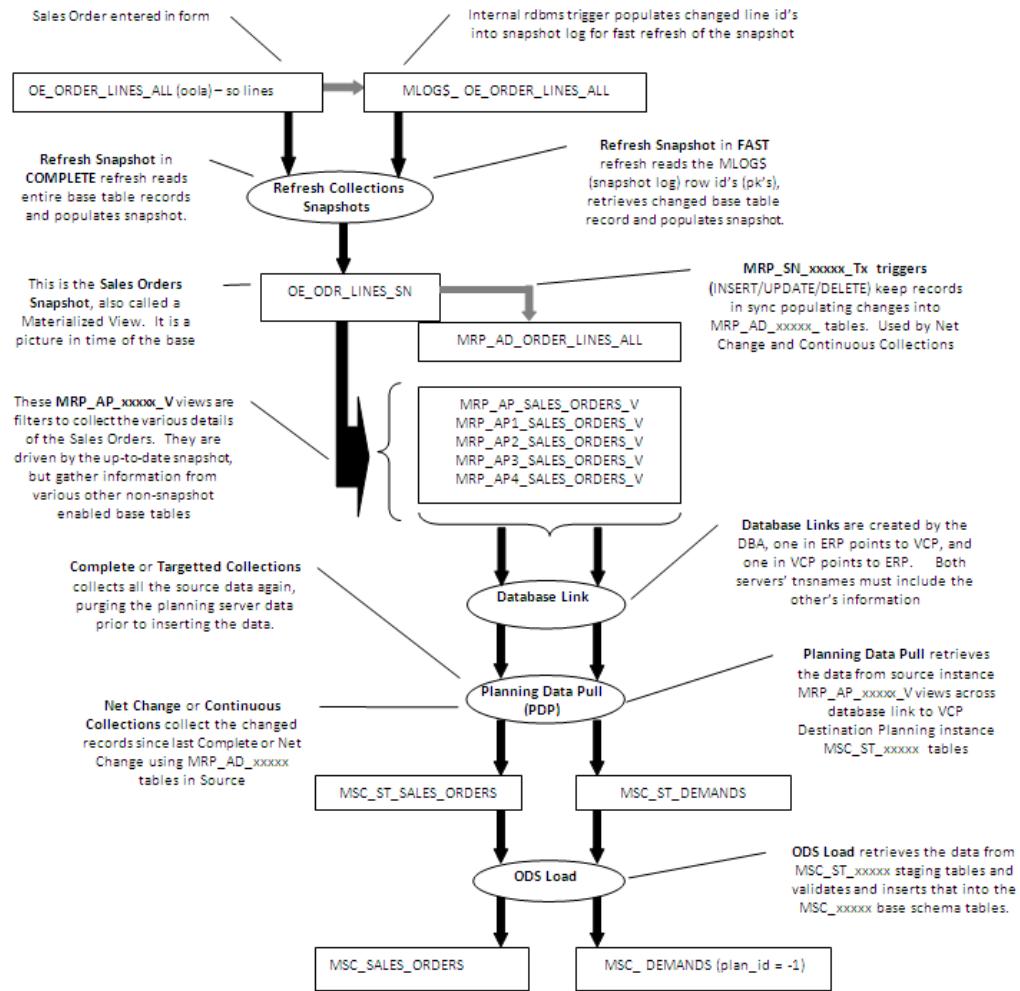


Description of Data Flow:

1. New purchase orders or requisitions in approved status.
2. Snapshot does not contain that newly entered purchase order or requisition.
3. The **MRP_AP_xxxxx_V** views do not contain the new purchase order or requisition.
4. Internal rdbms trigger populates the newly entered purchase order or requisition RowID into Snapshot Log (**MLOG\$** objects).
5. Launch ASCP Data Collections.
6. Refresh Snapshot reads row id from **MLOG\$**.

7. Refresh Snapshot gets record from base table using row id.
8. Refresh Snapshot populates/updates the snapshot with current table record information.
9. As the snapshot is updated, the MRP_SN_xxxxx_T triggers fire.
10. The triggers update/insert into the MRP_AD_SUPPLY table.
11. The MRP_AP_xxxxx_V views now have the new purchase order or requisition.
12. Planning Data Pull reads the MRP_AP_xxxxx_V views.
13. Planning Data Pull validates and calculates the purchase order or requisition data.
14. Planning Data Pull pulls data from EBS source instance over the database links:
 - ERP-based programs look in mrp_ap_apps_instances_all on the source instance to get the link name to VCP.
 - VCP-based programs look in msc_apps_instances planning instances to get the link name to ERP.
15. Planning Data Pull inserts that pulled data into the MSC_ST_xxxxx staging tables.
16. ODS Load reads the MSC_ST_xxxxx staging tables.
17. ODS Load validates and calculates the purchase order or requisition data.
18. ODS Load purges existing ODS data and loads new/change records using either the Complete or Net Change methods.
19. ODS Load data is hard coded with PLAN_ID = -1 for Collected Raw Source Data.
20. ODS data is read by the MBP Memory Based Planner.
21. ODS data is read by Production Scheduler.
22. ODS data is read by Strategic Network Optimization.

Sales Orders to ASCP Data Flow



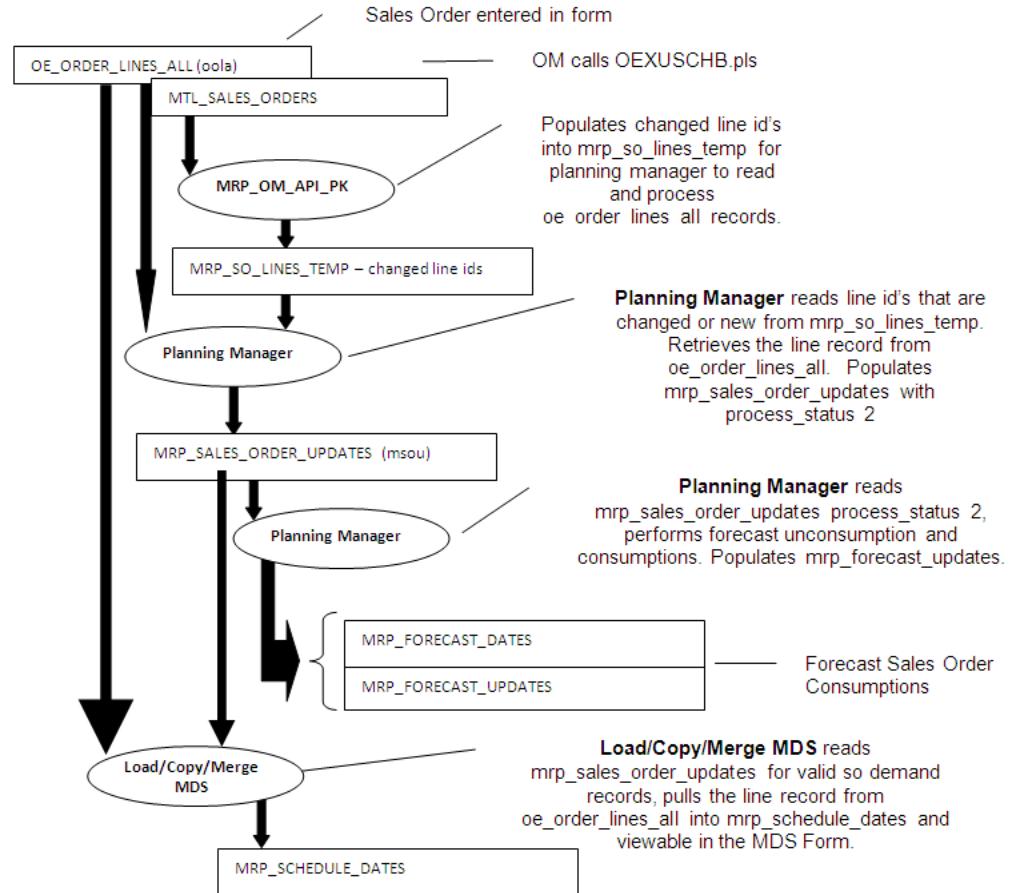
Description of Data Flow:

1. New sales order entered with visible_demand_flat = Y and scheduled_ship_date not null.
2. Snapshot does not contain that newly entered sales order.
3. The MRP_AP_xxxxx_V views do not contain the new sales order.
4. Internal rdbms trigger populates the newly entered sales order RowID into Snapshot Log (MLOG\$ objects).
5. Launch ASCP Data Collections.

6. Refresh Snapshot reads row id from MLOG\$.
7. Refresh Snapshot gets record from base table using row id.
8. Refresh Snapshot populates/updates the snapshot with current table record information.
9. As the snapshot is updated, the MRP_SN_xxxxx_T triggers fire.
10. The triggers update/insert into the MRP_AD_SALES_ORDERS table.
11. The MRP_AP_xxxxx_V views now have the new sales order.
12. Planning Data Pull reads the MRP_AP_xxxxx_V views.
13. Planning Data Pull validates and calculates the sales order's data.
14. Planning Data Pull pulls data from EBS source instance over the database links:
 - ERP-based programs look in mrp_ap_apps_instances_all on the source instance to get the link name to VCP.
 - VCP-based programs look in msc_apps_instances planning instances to get the link name to ERP.
15. Planning Data Pull inserts that pulled data into the MSC_ST_xxxxx staging tables.
16. ODS Load reads the MSC_ST_xxxxx staging tables.
17. ODS Load validates and calculates the sales order's data.
18. ODS Load purges existing ODS data and loads new/change records using either the Complete or Net Change methods.
19. ODS Load data is hard coded with PLAN_ID = -1 for Collected Raw Source Data.
20. ODS data is read by the MBP Memory Based Planner.
21. ODS data is read by Production Scheduler.
22. ODS data is read by Strategic Network Optimization.
23. ODS data is read by Demand Planning.

See My Oracle Support Note #412375.1 -- "Sales Orders to ASCP Data Flow Diagram" for diagnostics, major profiles, dependencies and diagnostic scripts.

Sales Orders to MRP Data Flow



Description of Data Flow:

1. OM code inserts header info into OE_ORDER_HEADERS_ALL.
2. OM code inserts line info into OE_ORDER_LINES_ALL with:
 - VISIBLE_DEMAND_FLAG = Y
 - SHIP_FROM_ORG_ID is NOT NULL
 - SHIP_TO_ORG_ID is NOT NULL
 - SOLD_TO_ORG_ID is NOT NULL
 - INVOICE_TO_ORG_ID is NOT NULL
3. OM inserts Sales Order data into MTL_SALES_ORDERS.

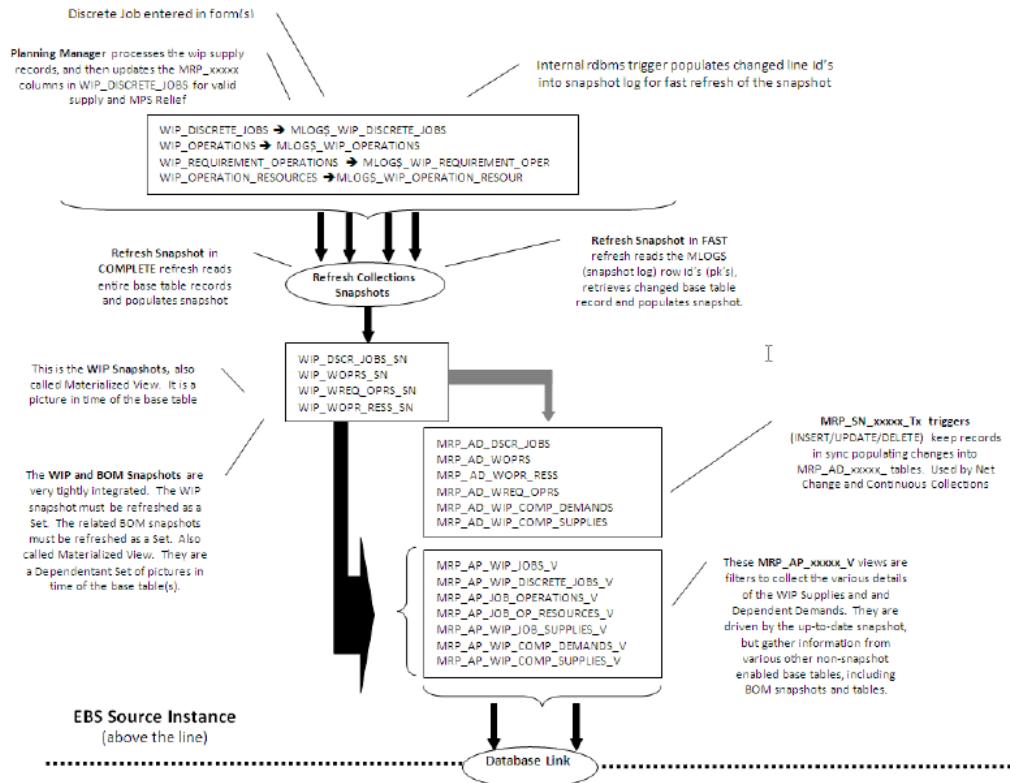
- OEXUSCHB.pls: Procedure Insert_Into_Mtl_Sales_Orders

4. Sales Order data is in MTL_DEMAND_OMOE view.
 - Used to link reservations between oe_order_lines_all and mtl_demand
5. MRP code Inserts Line Info into MRP_SO_LINES_TEMP .
 - MRPOAPIB.pls: MRP_OM_API_PK INSERT INTO mrp_so_lines_temp
6. Planning Manager processes records in MRP_SO_LINES_TEMP with process_status 2.
7. Planning Manager reads the Line_Ids from MRP_SO_LINES_TEMP, queries those lines from OE_ORDER_LINES_ALL, inserts into MRP_SALES_ORDER_UPDATES with process_status 2, and then sets the record in MRP_SO_LINES_TEMP to process_status 5 if successful.
8. Planning Manager processes the records in MRP_SALES_ORDER_UPDATES with status 2, set the status to 5 if successful, else errors the record with error messages.
9. Planning Manager then populates MRP_FORECAST_DATES and MRP_FORECAST_UPDATES.
10. Load/Copy/Merge MDS reads MRP_SALES_ORDER_UPDATES for valid demand records in MRP, then pulls the sales order line from OE_ORDER_LINES_ALL into MRP_SCHEDULE_DATES for that MDS name.

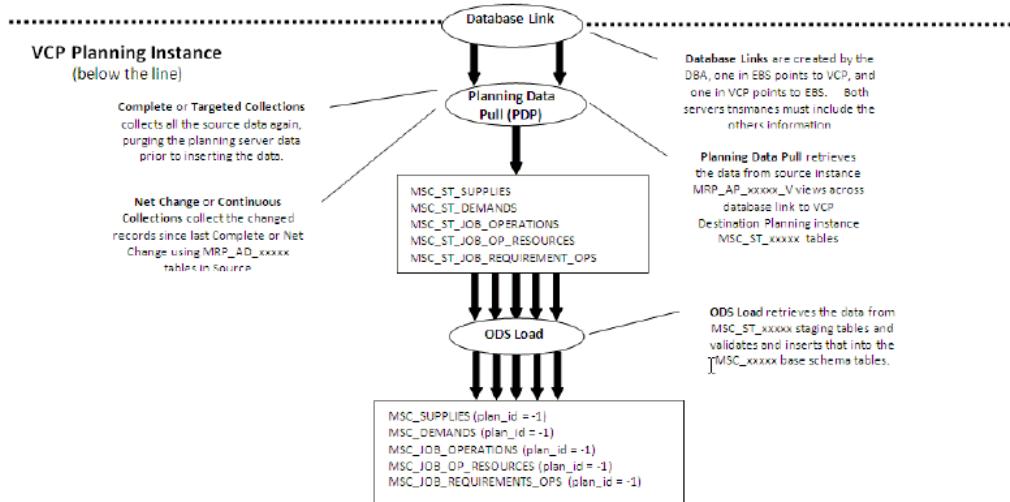
See My Oracle Support Note #414544.1 -- "Sales Orders to MRP Data Flow Diagram" for more information about profiles involved, major files, major related objects, diagnostic scripts, and common issues.

WIP to ASCP Data Flow

EBS side:



VCP side:



See My Oracle Support Note #413539.1 -- "WIP to ASCP Data Flow Diagram" for more information about troubleshooting and diagnostic scripts.

Index

A

APCC
partitioned tables, E-10
Application Instances window
setting up, 6-6

C

centralized planning, 3-3
collections
definitions, 2-6
net change, 8-2
Collection Workbench, 2-7
configurations
supported, 3-2
continuous collections
running, 8-15
continuous data collection, 2-7

D

data collection
architecture, 3-1
setting up, 8-11
simplified architecture, 3-3
decentralized planning, 3-3

E

entity, 2-7

F

field notes, 12-8

Find Requests, 8-16
form-based application, 9-15, 10-2

L

legacy collection
collect into legacy instance, 9-15
collect into planning instance, 10-2
loading transaction data using flat files into
Planning instance, 10-1
non-Oracle ERP systems, 9-2
overview, 9-1
preprocessing, 9-9
process, 9-3
running, 9-15
setting up batch uploads, 9-5
setting up legacy instance, 9-14
transaction data, 9-3
legacy system
data flow, 9-3

M

machine options
rapid planning, 3-7
machine options
general vcp products, 3-5

O

Operational Data Store (ODS), 2-6
Oracle Applications Data Store (ADS), 2-6
organization specific collections, 8-17
single source to multiple destinations, 8-22

Organization window
setting up, 6-9

P

page

Collaborative Planning Suite, 9-17
Oracle Collaborative Planning
 Load Data Files, 10-4

Parameters, 8-16

partitioned staging tables, E-6

partitioned table

 ODS and PDS, E-2

Planning Data Store (PDS), 2-6

profile options, 6-3, 6-3, 6-3, 6-4, 7-21, 7-21

 MRP: ATP Database Link, 6-3

 MSC: Aggregate Resource Name Flexfield
 Attribute, 7-22

 MSC: Collect Completed Jobs, 8-7

 MSC: Collect Completed Jobs, 12-9

 MSC: Cost of Using a BOM/ Routing Flexfield
 Attribute, 7-23

 MSC: Currency Conversion Type, 8-6

 MSC: Currency Conversion Type, 12-8

 MSC: Future Days for Currency Conversion,
 8-6

 MSC: Future Days for Currency Conversion,
 12-8

 MSC: History Days for Currency Conversion,
 8-6

 MSC: Planning Hub Currency Code, 8-6

 MSC: Priority for Substitute Items Flexfield
 Attribute, 7-23

 MSC: Resource Group for a Line Flexfield
 Attribute, 7-23

 MSC: Sourcing History Start Date Offset (in
 months), A-11

 MSO: Penalty Cost Factor for Exceeding Item
 Material Capacity Flexfield Attribute, 7-22

 MSO: Penalty cost factor for exceeding
 material capacity (Organization) flexfield, 7-22

 MSO: Penalty Cost Factor for Exceeding
 Material Capacity Flexfield Attribute, 7-23

 MSO: Penalty cost factor for exceeding
 resource capacity (Organization) flexfield, 7-22

 MSO: Penalty cost factor for exceeding
 resource capacity (Resource) flexfield, 7-22

MSO: Penalty cost factor for exceeding
 transportation capacity (Organization), 7-22

MSO: Penalty cost factor for exceeding
 transportation capacity flexfield attribute, 7-23

MSO: Penalty cost factor for late demands
 (Organization) flexfield attribute, 7-22

MSO: Penalty cost factor for late forecasts, 7-
 23

MSO: Penalty Cost Factor for Late Item
 Demands Flexfield Attribute, 7-21

MSO: Penalty cost factor for late sales orders,
 7-23

profile options

 MSC: Planning Hub Currency Code, 12-8

S

Schedule, 8-16

self-service application, 9-17

setting up

 Application Instances window, 6-6

 Organization window, 6-9

setup for collection of transaction data, 9-3

standard data collection, 2-6

 running, 8-11

 setting up, 8-12

supported configurations, 3-2

T

transaction data collection, 9-3

V

VCP applications, 1-2

VCP Integration, 1-3

W

window

 Load Transaction Data Parameters, 10-3

 Planning Data Collections, 10-3

 Planning ODS Load Parameters, 10-3

windows

 Application Instances, 9-14

 Collection Workbench, 8-13

 Continuous Collections, 8-16

 Requests, 8-16

 Data Pull Parameters, 8-11

Flat File Loader Parameters, 9-15
ODS Load Parameters, 8-12
Planning Data Collection, 8-11
Planning ODS Load Parameters, 9-16
Pre-Process Monitor Parameters, 9-16
Purge Parameters, 9-18
Schedule, 8-12

