



System Administration Guide

May 2016

Ex Libris Confidential

CONFIDENTIAL INFORMATION

The information herein is the property of Ex Libris Ltd. or its affiliates and any misuse or abuse will result in economic loss. DO NOT COPY UNLESS YOU HAVE BEEN GIVEN SPECIFIC WRITTEN AUTHORIZATION FROM EX LIBRIS LTD.

This document is provided for limited and restricted purposes in accordance with a binding contract with Ex Libris Ltd. or an affiliate. The information herein includes trade secrets and is confidential.

DISCLAIMER

The information in this document will be subject to periodic change and updating. Please confirm that you have the most current documentation. There are no warranties of any kind, express or implied, provided in this documentation, other than those expressly agreed upon in the applicable Ex Libris contract. This information is provided AS IS. Unless otherwise agreed, Ex Libris shall not be liable for any damages for use of this document, including, without limitation, consequential, punitive, indirect or direct damages.

Any references in this document to third-party material (including third-party Web sites) are provided for convenience only and do not in any manner serve as an endorsement of that third-party material or those Web sites. The third-party materials are not part of the materials for this Ex Libris product and Ex Libris has no liability for such materials.

TRADEMARKS

"Ex Libris," the Ex Libris bridge, Primo, Aleph, Alephino, Voyager, SFX, MetaLib, Verde, DigiTool, Preservation, Rosetta, URM, ENCompass, Endeavor eZConnect, WebVoyage, Citation Server, LinkFinder and LinkFinder Plus, and other marks are trademarks or registered trademarks of Ex Libris Ltd. or its affiliates.

The absence of a name or logo in this list does not constitute a waiver of any and all intellectual property rights that Ex Libris Ltd. or its affiliates have established in any of its products, features, or service names or logos.

Trademarks of various third-party products, which may include the following, are referenced in this documentation. Ex Libris does not claim any rights in these trademarks. Use of these marks does not imply endorsement by Ex Libris of these third-party products, or endorsement by these third parties of Ex Libris products.

Oracle is a registered trademark of Oracle Corporation.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Ltd.

Microsoft, the Microsoft logo, MS, MS-DOS, Microsoft PowerPoint, Visual Basic, Visual C++, Win32,

Microsoft Windows, the Windows logo, Microsoft Notepad, Microsoft Windows Explorer, Microsoft Internet Explorer, and Windows NT are registered trademarks and ActiveX is a trademark of the Microsoft Corporation in the United States and/or other countries.

Unicode and the Unicode logo are registered trademarks of Unicode, Inc.

Google is a registered trademark of Google Inc.

iPhone is a registered trademark of Apple Inc.

Copyright Ex Libris Limited, 2016. All rights reserved.

Document released: June 7, 2016

Web address: <http://www.exlibrisgroup.com>

System Architecture

This section includes:

- [Introduction](#)
- [Components](#)
- [Physical View of the System](#)
- [Overview of the Publishing Process](#)
- [Overview of the Index and Search Process](#)
- [Secure Socket Layer \(SSL\)](#)
- [Accessibility](#)



Introduction

Primo provides users with a universal solution for the discovery and delivery of print and digital information sources, regardless of format and location.

Primo offers advanced, high quality search results based on existing metadata. After finding the desired materials, Primo can facilitate the delivery of the physical item from the library or provide immediate access to digital copies.

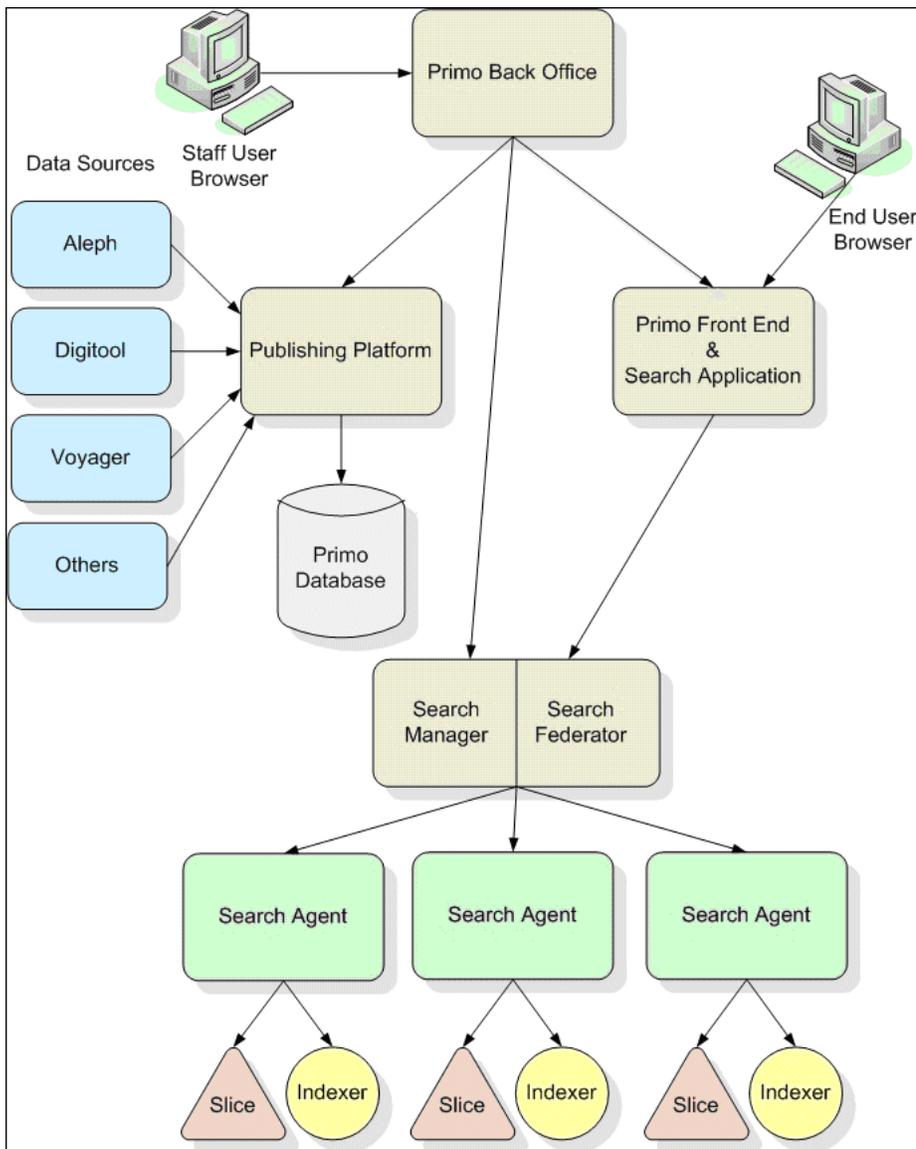
Primo is a Web-based solution that runs under UNIX. It comes with a built-in publishing platform that is used to process raw data (MARC, XML, MAB, and Dublin Core) and turn it into high-quality, indexed information that is saved in Primo Normalized XML (PNX) format. The processing of the data source information is performed by pipes that understand the various library source formats. The PNX data can be quickly and efficiently searched by the Search Engine.

Components

Primo consists of the following components:

- Publishing Platform
- Indexer
- Back Office
- Search Federator
- Search Agents
- Search Engine
- Front End
- Database

The following figure illustrates the logical view of Primo, including its various components. Each of its components is described in the following sections.



For image detail, press Ctrl++ Logical View of Primo System

Publishing Platform

The Publishing Platform enables the institution to consolidate the full range of institutional resources, including print collections, digital repositories, and electronic resources. The Publishing Platform manages the harvesting of this raw data from various data sources and its transformation into high quality, indexed information that can be quickly and efficiently searched by the Primo Search Engine. This data is stored in the Primo Normalized XML (PNX) format. Primo can harvest and normalize any data in standard XML format. Standard formats—for example, MARC, MAB and Dublin Core—have built-in template mappings. Templates can be customized during the implementation process. The processing of each data source is performed by pipes that recognize various source formats.

The Publishing Platform performs:

- Intelligent harvesting of data via FTP, file copy, or OAI
- Normalization of the data to the PNX format, which is stored in the Primo database
- Enrichment based on algorithms and external information

- De-duplication based on algorithms
- FRBRization

The Normalization Mappings and Enrichments are configurable using the Primo Back Office. For additional information about the Primo Back Office, refer to the Primo Back Office Guide.

The Publishing Platform supports scheduled and unattended harvesting and processing of different data formats, while enabling interactive monitoring and control over the entire set of activities.

Indexer

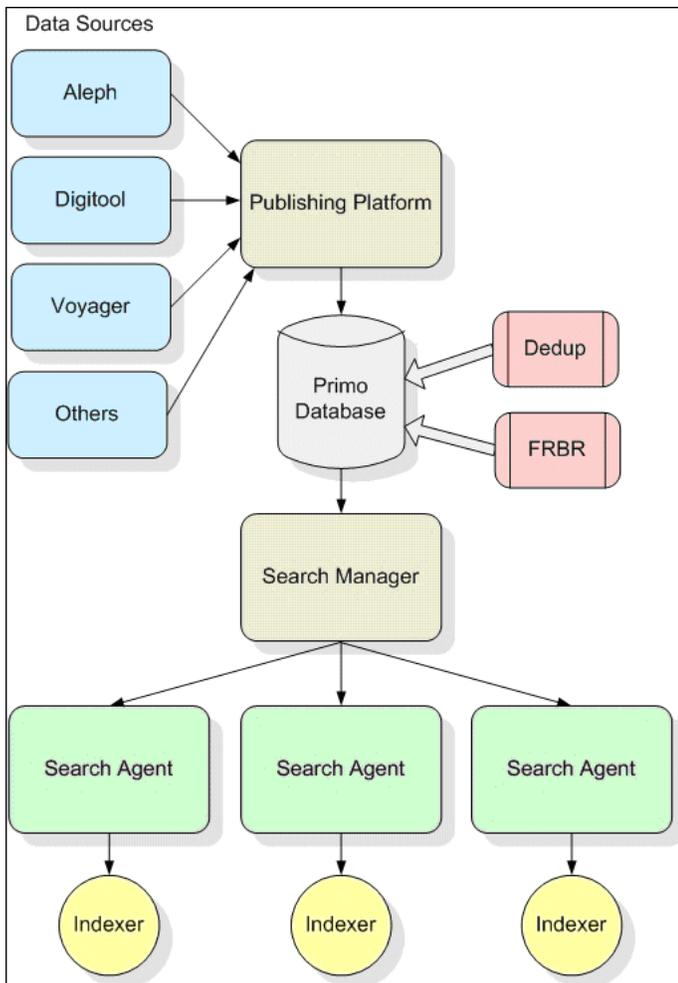
The Indexer is part of the Search Engine and is used to create slices. The Search Engine supports multiple slices of search data. Slices are an efficient way to allocate groups of equal-sized chunks of memory.

The Indexer automatically swaps itself with search machines so that slices can be made available without downtime.

The Indexer splits the actual data into manageable slices, which are loaded into memory. Each slice is searched by a dedicated thread; however, multiple slices can reside on the same memory machine so that all CPUs are utilized. Multiple machines may be used so that the system is not limited to the memory of one machine.

The following figure illustrates the indexing process in the Primo system.

Figure 1:



For image detail, press Ctrl++ Logical View of Primo System Indexing

For more information about the indexing process and swapping, refer to [Overview of the Index and Search Process](#).

Back Office

The Primo Back Office enables configuration and monitoring of all Primo components in an easy-to-use graphical interface.

The configuration of Primo in the Back Office is organized by the lifecycle of Primo, and includes:

- Initial configuration
- Ongoing maintenance

In addition, you can configure elements of the system individually by accessing them through the appropriate wizard. A site map is provided for direct access to the individual tasks for the advanced user. For additional information about the Primo Back Office, refer to the Primo Back Office Guide.

Search Federator

The Search Federator coordinates the search, utilizing all slices, and combines the search results into a unified result set.

Search Agents

Search agents are located on remote machines. These agents stop and start the search instance and the indexing process. When an agent starts, it sends a registration request to the Federator. The agent and the Federator communicate using the Java RMI protocol.

Search Engine

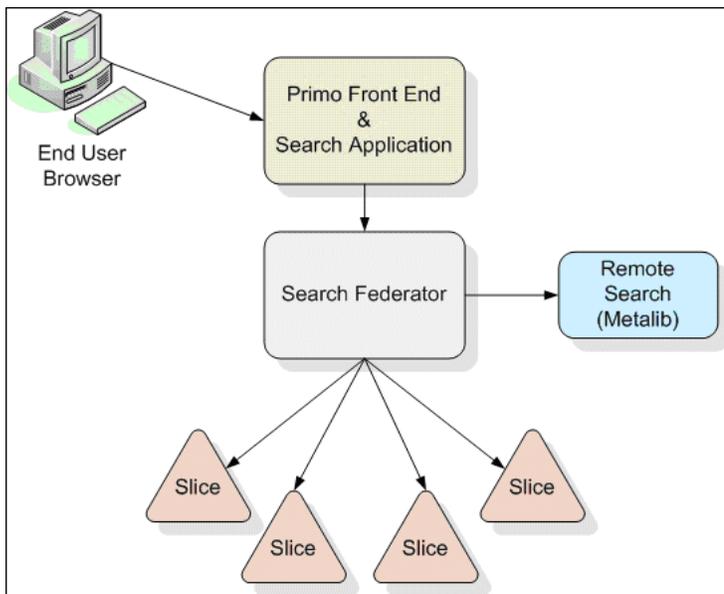
The Primo Search Engine retrieves library metadata from the local PNX database table and transforms it into useful information. The Search Engine extends Lucene functionality and supports multiple slices for very large data sets. These slices are prepared by the Indexer. Each slice is searched by a dedicated thread; however, multiple slices can reside on the same machine, so that all CPUs are utilized. Multiple machines may be used so that you are not limited to the memory of one machine.

Search functionality includes faceted navigation, “did you mean” suggestions, paging, and sorting.

Front End

The Primo Front End user interface is responsible for all interactions with the end user. It is a search tool that is both powerful and easy to use. Each institution can have its own fully customized view. Every view can have one or more tabs. Tabs enable a site to divide the Primo repository and records from remote resources into resource groups or types. Within a tab, several search scopes can be defined. Search scopes group records so they can be searched together. Using the Front End user interface, the end user searches the PNX database table for relevant items. After discovery, Primo indicates the availability of the resource in the source system and interacts with the source system to provide more information about the resource or delivers the resource to the end user. The Front End generates the actual HTML pages viewed by the end user.

The following figure illustrates the querying process in the Primo system.



For image detail, press Ctrl++ Logical View of Primo System Queries

Primo Database

The Primo database is based on Oracle 11g R2 RDBMS. The Oracle database contains the following primary types of content:

- Primo PNX records and user-contributed information (such as reviews and tags).
- Monitoring information, including statistics, detailed information on searches, and so forth.
- Primo configuration information.

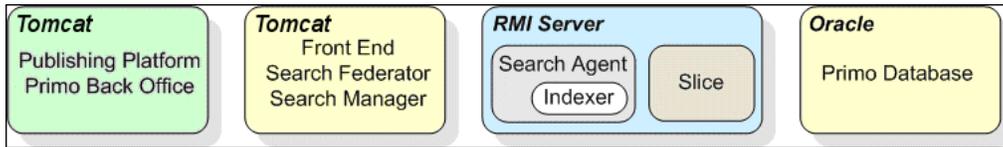
Physical View of the System

The system includes the following software components:

- Publishing Platform and Primo Back Office on a Tomcat servlet container.
- Front End, Search Federator, and Search Manager on a Tomcat servlet container.
- Search Agents (including Indexers) on an RMI server.
- Slices on an RMI server.
- Primo database on an Oracle database server.

The Primo software contains four different processes as shown in the following figure.

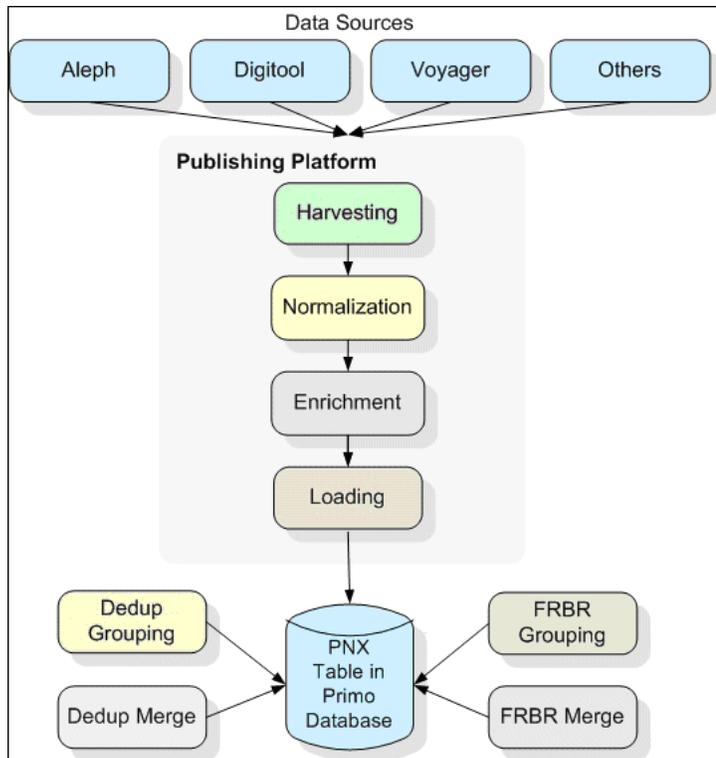
The processes can run on different machines, the same machine, or some on the same machine and the rest on different machines. Usually, it is recommended to have the Oracle and the Tomcat - Publishing Platform Primo Back Office on one machine and the Tomcat - Front End and the RMI Search Server on another machine.



For image detail, press Ctrl++ Physical View of Primo System

Overview of the Publishing Process

The Publishing Process is the process that Primo uses to retrieve and process data from external library systems. This process involves a number of steps, which are outlined in the following figure.



For image detail, press Ctrl++ Publishing Process

Each of the steps in the publishing process is discussed in detail in the following sections.

Preparing the Source Data

Before Primo begins harvesting data, the source system must prepare the following:

- The complete data source for the initial harvesting of the data.
- A delta of the data, including only changed records (new, updated, and deleted), for the ongoing update of the database.

All the information to be contained in the PNX database table should be included in a single record that can be identified by a unique and persistent ID. This means that any information that is related to the main record (for example, holdings information related to the bibliographic record) must be added to the extracted record.

The structure of the record must conform to one of the following formats:

- MARCXML (<http://www.loc.gov/standards/marcxml/>).
- Dublin Core XML (<http://dublincore.org/documents/dc-xml-guidelines/>).

NOTE: The record can include non-standard MARC or Dublin Core fields, including non-numeric codes for MARC.

Harvesting Data

A single source system can contain several data sources, such as the following systems:

- For SFX, Primo may harvest a single SFX instance, but a single SFX installation may include several data sources (that is, instances).
 - For Aleph and Voyager, every bibliographic database is a separate data source, but a single installation of Aleph or Voyager may include several bibliographic databases that are harvested by Primo. The harvesting stage is the first step in the publishing pipe. Primo supports several harvesting methods, including:
 - FTP/SFTP Harvesting - Primo can harvest files from a remote server. To perform FTP harvesting, Primo must be able to access the server using the server IP, the directory name, user name, and password. For ongoing harvesting, Primo retains the date and time of the last harvest. The Publishing Platform harvests all files with a server timestamp greater than the last harvesting date. File names must be unique. Optionally, the file can be deleted once it was successfully harvested.
 - Copy Harvesting—Primo can harvest files by copying files from any mounted drive. To perform copy harvesting, Primo must have READ permission for the directory.
 - OAI Harvesting—Primo can harvest records from an OAI server by sending an OAI-PMH request and processing the records that are sent in response to the request. To perform OAI harvesting, Primo needs the OAI server IP address and port number, as well as the OAI set to harvest. For ongoing harvesting, Primo retains the data and time of the last request.
-

Normalizing the Harvested Data

Before beginning the normalization stage, records are classified as follows:

- Normal records include new and updated records from the data source that need to be normalized and enriched before being loaded to the persistence layer.
 - Deleted records are records that were deleted in the data source and need to be deleted from the persistence layer. These records are deleted directly from the persistence layer and do not go through the normalization and enrichment stages. Records are divided into groups called bulks. The default bulk size is up to 1,000 records. Bulks are zipped in order to reduce the file count and the written data sizes, as well as improve performance. In the normalization process, the source records are converted to the PNX format using the normalization mapping set of the pipe. For more information about the PNX format, refer to the Primo Technical Guide.
-

Enriching the Data

Once the records are normalized, they may be enriched with additional data. Every publishing pipe can be assigned an enrichment set, which includes one or more enrichment routines.

Loading Data into the Primo Database

Once records have been normalized and enriched, they are loaded into the PNX table in the Primo database. The PNX table is an Oracle database table in which the PNX records are stored before they are retrieved and loaded to the Search Engine. The duplicate record detection process is also handled in the Primo database. The Primo database has a number of tables, which store various types of data used in Primo.

Processing Duplicate Records (Dedup)

During the duplicate record detection process (Dedup), the publishing platform locates duplicate records and assigns them the same matchID. This is performed by using the Dedup vector of the PNX record. For more information about the Dedup process and the matching algorithms, refer to the Duplicate Detection Process section in the Primo Technical Guide.

Dedup Process

The persistence layer duplication detection database stores the Dedup vector. Once the vector is stored in the database, it is removed from the normalized record. Part of the vector is indexed in order to locate candidates for a match.

When a record is matched against the database, the system first attempts to find a match based on the RecordID. If a match is not found, the record is new and the system tries to find a matching record based on the vector. If a match is found, the record is assigned the MatchID of the record with which it is matched. Once a match is found, the matching process stops. If a match is not found, the record is assigned a new MatchID.

If there is a match on the RecordID, the system compares the vector in the incoming record with its vector in the database. If the vector is the same, the record is assigned its present MatchID. If it is not the same, the record is treated as a new record (the MatchID is removed) and the system tries to find it a matching record.

It is important to distinguish between the initial and ongoing Dedup process. During the initial publishing stage, all data sources are first loaded to the persistence layer, and only when all records have been loaded, will the duplicate detection process start. In the ongoing publishing stage, the duplicate detection process is run as part of every pipe. The difference is due to the fact that in the initial publishing stage, the Dedup is run in multi-processing mode to save time, while in ongoing publishing stage, every record is Deduped sequentially.

Merging Duplicate Records (Dedup)

Once all matching records are located in the Dedup process, the system creates a merged record.

Processing Records (FRBR)

The grouping process (FRBR) is based on creating a vector for every record. The vector includes one or more keys that identify the work it represents. Records that have a matching key (Primo attempts to match all keys in the record) are added to a group and are assigned the ID of the group (frbrID). Each record can belong to only one group. In other words, once a record is matched with an existing group, Primo terminates the grouping process for that record. Primo creates a merged record for the FRBR group. Unlike the Deduped merged record, the Search Engine retrieves and indexes both the merged FRBR record and the individual records in the FRBR group.

Overview of the Index and Search Process

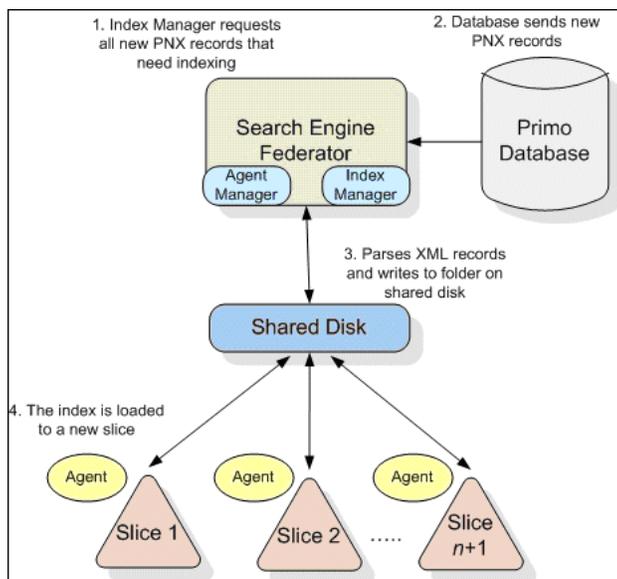
All new data must be indexed once it has been processed by the Publishing Process. This indexed data is accessed when the system performs a search.

Each Search Engine machine consists of one or more slices, which is the atomic entity of the Search Engine. Primo implements the following types of slices:

- RAM-based slices match relatively small document collections (backward compatible with PrimoVersion 1.x).
 - File system-based slices match large document collections and have lower memory consumption.
- The type of slice defines the search, the indexing technique, and the hot swap behavior.

Indexing in Primo

The following figure is an example of how Primo indexes data.



For image detail, press Ctrl++ Indexing in Primo

The indexing process in Primo consists of the following main phases:

- Indexing—The Index Manager takes the XML records and parses them. It then writes them to a temporary folder (<path>.mir) and indexes them. During the indexing, the PNX tokens are normalized (conversion to lowercase characters, removal of punctuation, and so forth).
- Optimization—The system optimizes the records into a single file for performance purposes.
- Mirroring—The indexed records are copied to a directory from which the slice loads.
- Swap—This phase replaces the old indexes with the new indexes with no downtime. Hot swap is done on the redundant machine for the N+1 Agent topology and the slice machine for the N+1 Slice topology.

Indexing on RAM Slice

The following indexing phases run sequentially: Indexing, Optimization, Mirroring, and Swap. After indexing completes, a redundant slice is loaded to RAM and facet cache is prepared. Once the new slice is ready, search functionality is moved to the new slice and the old slice is shut down. If the process fails during the Indexing or Optimization phases, the mirror directory is restored from the main directory.

Indexing on File System Slice

This indexing method is intended for large document collections that create large indexing files. The indexing phases operate as follows: The index directories are placed in an NFS file. The indexer places a copy of the mirror directory in the local path (direct-attached storage). After indexing is performed in a mirror directory, the new index is copied to the main directory, where optimization is performed on both the main and mirror folders. When new indexes are ready, a redundant slice is warmed up, using high frequency terms from the index to provide better search performance. Once the new slice is ready, search functionality is moved to the new slice.

The topology type of the Search Engine determines where the indexing and hot swap operations will take place.

Method 1: All In One

When Primo needs to create a new index, the Front End (FE) activates the Agent manager to prepare the index. Once the index is ready, the FE activates the agent to start up a new slice—that is, to load a new index. Once the new slice is running, the FE will use the new slice and the Search engine will shut down the old slice.

Method 2: 2-Tier/3-Tier

When Primo needs to create a new index, the FE activates the Agent manager on the Back Office to prepare the index. Once index is ready, the Front End will activate an additional Agent residing on the Search engine and instructs it to start up the new slice, that is, to load a new index. Once the new slice is running, the FE will start to use the new slice and the Search engine will shutdown the old slice.

Searching in Primo

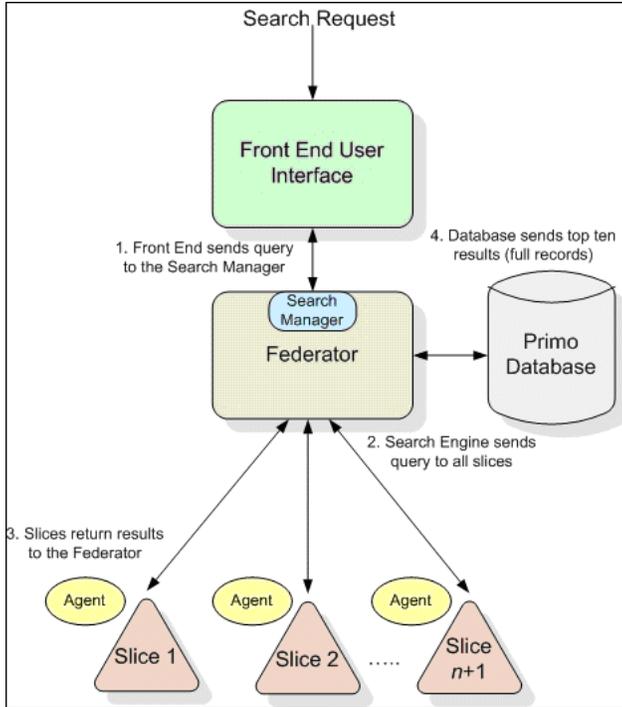
Primo uses the following types of searches to enhance search performance:

- RAM slice searches use the Lucene RAMDirectory, which loads the index into RAM, so that search operations are performed directly from RAM. This provides higher search performance; however, operation of this mode requires the available RAM to be at least 2.4 times the index size.
- File system slice searches utilizes OS in memory disk cache to achieve performance levels similar to that of the RAM slice search. This depends on the underlying implementation of buffering in the operating system. Because of NFS cache limitations, it is highly recommended to use direct attached storage for the Search engine servers. The servers will copy the slice from the NFS to their local disk.

When a user performs a search in the Primo Front End, the Search Engine sends a query to all registered slices (not via the agent on the slice machine). The slice searches its index in RAM and returns the top 200 IDs, their scores, and statuses to the Federator. At the same time, the slice begins calculating facets, using another thread. The Search Manager merges the results from all of the slices and does the following:

- Adds synonyms of each word being searched for to the query. The system searches for the synonyms as well as the words specified in the query by the user.
- Takes the top ten values (according to scores) and requests the full records from the PNX. It receives the records and stores these records to be displayed to the user in the Front End user interface.
- When the search results in either no or very few records, Primo searches again using stemming in order to retrieve additional records.

- Merges all records and retrieves the top 200 merged results. It then sends the facets of these records to the appropriate slices. The slices calculate how many times these records appear in the entire result set and return this information to the Search Manager.
- Searches for specific static facets in addition to the 200 facets searched for based on the search query.
- Retrieves an accurate count of facet values. It then sends the full PNX records and the facet information for the top ten values to the Front End user interface.



For image detail, press Ctrl++ Search Process

The Federator uses the host machine and port information in [search_schema.xml](#) when it performs a search (refer to [search_schema.xml](#)).

Agents are run separately. When an agent begins running, it registers to the Search Engine. Each slice is assigned a unique ID within the system. The path is the location of the directory from which the slice is loading.

The Primo N+1 topology provides a failover capability in search servers to ensure continuous availability of search operations. When a system failure is detected on one of the search servers, the indexer will automatically copy the failed slice and start up a replacement Search engine, which may take a few minutes. Until the new Search server starts, the existing servers will continue to process Primo searches, but the results will not be 100 percent complete. Once the replacement indexer is up and running, the searches will resume full accuracy.

NOTES:

For N+1 topologies that have a single slice (N=1), Primo searches will not be processed until the indexer starts up the failed slice.

Until a replacement of the failed server is up and running, the system will not be able to run indexing or swapping. In all other topologies, the monitor automatically tries to restart the slice and sends an e-mail notification to the System Administrator when a search engine failure occurs.

Filters are used as constraints that are added to the search and are used for quick matches (this is the same as using the Deploy button in the Back Office interface).

Linguistic Issues

- Primo linguistically supports English, German, French, and Danish languages by creating dictionaries per language (if available) for the following: recognition, stop word lists, special stemming algorithm (English and German only), misspelled words (English, Hebrew, and German only), synonyms (not French), pluralization algorithm, and phonetic algorithm for “did you mean” suggestions.
- Primo creates indexes for stop words to allow end users to search phrases that contain stop words.
- Primo defines a synonym collection per language. Administrators can extend this collection by updating the user_synonym file.

Did You Mean

If the configured result threshold is not met, the “did you mean” functionality is activated for the search. The “did you mean” is made up of the following:

- Metaphone and double metaphone algorithms take words and encode them.
- N-gram encoding (for example, people - peple, pople, and so forth). Using N gram encoding, the system checks how close the word being searched for is to the candidate.
- Text files of commonly misspelled words and homonyms (currently available in English only).
- The metaphone and N-Gram repositories are based on the regular index and are extended with phrases/typographical/grammatical mistakes that are learned from the search statistics.

NOTE: If the word exists in the dictionary, the Did you mean link is not displayed in the Front End.

In addition to searching in the local PNX database, Primo searches on remote repositories using MetaLib.

The actual data to be searched is split into manageable slices. These slices are prepared by the Indexer. Each slice is searched by a dedicated thread, but multiple slices can reside on the same machine so that all CPUs are utilized. In addition, multiple machines may be used so that the search process is not limited to the memory of one machine.

The Federator manages the dispersal of the search to all slices and the federation of the results into a unified result set, which is returned to the Front End and displayed to the user.

search_schema.xml

The search_schema.xml defines the number and location of each slice.

To view the file:

Enter the following commands to view the search_schema.xml file:

```
se_conf
```

```
vi search_schema.xml
```

The following is a sample search_schema.xml file:

```
<?xml version="1.0" encoding="UTF-8"?>
<search_schema version="1.0" xmlns="http://www.exlibrisgroup.com/xsd/jaguar/
search_schema">
<federator>
<facet_count>250</facet_count>
<min_res_for_stemming>25</min_res_for_stemming>
<max_facets_to_sumup>50000</max_facets_to_sumup>
<max_results_stemming>25</max_results_stemming>
<enable_warmup>true</enable_warmup>
<warmup_queries_number>500</warmup_queries_number>
<cache_results>true</cache_results>
<didymean_threshold>
```

```

<res_threshold>50</res_threshold>
<score_threshold>0.75</score_threshold>
</didymean_threshold>
<synonyms>
<levels>
<level desc="very high">0.8</level>
<level desc="high">0.1</level>
<level desc="normal">0.01</level>
<level desc="low">0.005</level>
<level desc="very low">0</level>
</levels>
<threshold>normal</threshold>
</synonyms>
</federator>
<didymean>
<preferred_result>sx&lt;/preferred_result> &lt;!-- sx/ngram --
></preferred_result>

```

```

<!-- sx/ngram -->
<sx_threshold>0.975</sx_threshold>
<ngram_threshold>0.65</ngram_threshold>
<freq_threshold>5</freq_threshold>
<!-- minimum occurrences a suggestion must appear in data -->
<candidates_to_examine>20</candidates_to_examine>
<field>title</field>
<dictionary_languages>eng</dictionary_languages>
<advanced_merge_factor>500</advanced_merge_factor>
<advanced_merge_docs>500</advanced_merge_docs>
<validate_didymean>true</validate_didymean>
<add_search_statistics>true</add_search_statistics>
<search_statistics_fetch_param>all</search_statistics_fetch_param>
<!-- search_statistics_fetch_param can be one of 3 things:
1) all = all the words will be fetched
2) a positive number x = x last searched words will be fetched
3) a date in the format dd/mm/yyyy = all the searched words from the date
inclusively will be fetched -->
</didymean>
<filters>
<filter>scope:(north)</filter>
<filter>scope:("kings")</filter>
<filter>facet_frbrtype:(7 OR 6)</filter>
</filters>
<repository active="true">
<path>/exlibris/primoindexes</path>
</repository>
<agents desc="collections of all search instances to be searched by jaguar"
auto_deploy="active" swapping_type="agents">
<agent port="9501" host="il-primoindexes06.corp.exlibrisgroup.com"
connection_type="remote" active="true">

```

```

<slices slice_count="2" name="main" active="true"
xmlns="http://www.exlibrisgroup.com/xsd/jaguar/search_schema">
<slice id="1">
<path>/exlibris_primo/inst_1_index</path>
<load2ram>>false</load2ram>
</slice>
<slice id="2">
<path>/exlibris_primo/inst_2_index</path>
<load2ram>>false</load2ram>
</slice>
</slices>
</agent>
<agent port="9501" host="il-primo04.corp.exlibrisgroup.com"
connection_type="local" active="true">
<slices slice_count="0" name="main" active="true"
xmlns="http://www.exlibrisgroup.com/xsd/jaguar/search_schema"/>
</agent>
</agents>
<multiple_front_ends>
<mfe_master>il-primo04.corp.exlibrisgroup.com:2701</mfe_master>
<mfe_slaves>il-primo04.corp.exlibrisgroup.com:2701</mfe_slaves>
<mfe_config>none</mfe_config>
</multiple_front_ends>
</search_schema>

```

Metasearch Functionality

Primo uses MetaLib as a metaSearch Engine, via the MetaLib X-Server. Primo sends MetaLib a list of resources and a query and receives a list of results as MARC records. Each result includes an OpenURL. Primo then converts the metasearch MARC records to PNX records using the on-the-fly MetaLib pipe. When relevant, the metasearch results are added to the local results and the merged list is then deduplicated and ranked, and facets are calculated.

Third Node Functionality

In addition to local and remote search, Primo supports plug-in capabilities of third-party search functions within the Primo application. This feature allows end users to utilize the strength of the Primo tool set (look and feel, caching, faceting highlighting, statistics, e-shelf, tagging, and reviews) with their search methods or special resources.



Secure Socket Layer (SSL)

SSL is a Transport layer protocol that provides secure communications over the Internet. For more information on implementing SSL, refer to the Patron Directory Service Guide.

Accessibility

Primo complies with the following HTML standards and guidelines to support, for example, screen readers and key navigation software:

- XHTML 1.0 Transitional
- WCAG 2.0 Guidelines Priority 2 (with exceptions)
- Section 508 (with exceptions)
- CSS level 3
- The W3C Web Content Accessibility Guidelines 1.0, Level A

In addition, the following tool was used to validate Primo's Web pages:

Total Validator – <http://www.totalvalidator.com/>

Starting and Stopping the System

This section includes the following topics, which describe how to start and stop the system servers:

- [System Startup and Shutdown Commands](#)
- [Shutting Down the Primo Servers](#)
- [Starting Up the Primo Servers](#)
- [Central Start/Stop Utility](#)
- [Verifying System Operation](#)

System Startup and Shutdown Commands

The following table lists the aliases and scripts that are used to start up or shut down the various servers (such as the Back Office and Front End servers) used with Primo:

Primo Startup and Shutdown Commands

Alias/Script	Description
be_start	Starts the Back Office (BO) server.
be_stop	Stops the BO.
be_restart	Stops and starts the BO server.
fe_start	Starts the Front End (FE) server. Note that the FE servers must be up and running before you can start the SE servers.
fe_stop	Stops the FE server.
fe_restart	Stops and starts the FE server. NOTE: Because this command may restart your server before it has finished shutting down, Ex Libris recommends using the following commands to restart the FE: fe_stop fe_start
se_start	Starts the Search Engine (SE) server.
se_stop	Stops the SE server.
se_restart	Stops and starts the SE server. NOTE: The FE servers must be up and running before you can start the SE servers.
start_apache	Starts the PDS Apache server.
stop_apache	Stops the PDS Apache server.
wd_start	Starts the Watchdog monitor.
wd_stop	Stops the Watchdog monitor.

<code>\$primoe_root/ primo_startup_all</code>	Starts the general Primo startup script.
<code>\$primoe_root/ primo_shutdown_all</code>	Starts the general Primo shutdown script.

Shutting Down the Primo Servers

IMPORTANT: The order in which the servers are shut down is important and should be applied to single server topologies as well.

To shut down the Primo servers:

1. Enter the following command to stop the Primo Apache server:

```
stop_apache
```

If you are receiving Primo requests on HTTP port 80, log in as the root user and enter the following commands to stop the Apache server:

```
apcb
```

```
./apachectl stop
```

NOTE:

If the Apache server is not running, you will see and can ignore the following error message:

```
./apachectl stop: httpd (no pid file) not running
```

2. Shut down the Back Office (BO) servers:
 - a. Enter the following command to stop the BO:

```
be_stop
```
 - b. Repeat these steps for each BO in your Primo topology.
3. Shut down the Front End (FE) servers, starting with the master FE:
 - a. Enter the following command to stop the FE:

```
fe_stop
```
 - b. Repeat these steps for each FE in your Primo topology.
4. Shut down the Search Engine (SE) servers:
 - a. Enter the following command to stop the SE:

se_stop

- b. Repeat these steps for each SE in your Primo topology.

Starting Up the Primo Servers

This procedure describes the steps used to start up Primo for single and multiple server topologies.

IMPORTANT: The order in which the servers are started up is important and should be applied to single server topologies as well.

To start up the Primo servers:

1. Start up the Back Office (BO) servers:
 - a. Enter the following command to start up the BO:
`be_start`
 - b. Repeat these steps for each BO in your Primo topology.
2. Start up the Front End (FE) servers, starting with the master FE:
 - a. Enter the following command to start up the FE:
`fe_start`
 - b. Repeat these steps for each FE in your Primo topology.
3. Start up the Search Engine (SE) servers:
 - a. Enter the following command to start up the SE:
`se_start`
 - b. Repeat these steps for each SE in your Primo topology.

NOTE: The FE servers must be up and running before you start the SEs.

4. Enter the following command to start the Primo Apache server:

`start_apache`

If you are receiving Primo requests on HTTP port 80, log in as the root user and enter the following commands to start the Apache server:

```
apcb
```

```
./apachectl_auto
```

Central Start/Stop Utility

This utility (see [Central Start/Stop Utility](#)) allows you to start up, shut down, and restart the Primo application from a master Primo server in multi-server topologies.

This mechanism only affects the Primo application. If you need to start/stop the Apache and database servers, refer to those sections in this guide.

```
Welcome to the Central Start/Stop Utility

This utility allows you start/stop all the Primo servers on this installation.

Please choose the action you want to run:

1. primo central start
2. primo central stop
3. primo central restart
```

Central Start/Stop Utility

Each time an action is executed, the logs are stored in the following directory on the master server:

```
/exlibris/primo/p3_1/primoe/central_start/log
```

NOTE: This utility is usually installed on the Back Office server, which must have SSH access to all other Primo servers in multi-server topologies. To provide automatic access, SSH keys must be implemented for each Primo server in your topology. Otherwise, the system will prompt you to enter a password each time another server is accessed by the utility.

Starting Up Primo

To start up Primo:

1. Log on to the master Primo server.
2. Enter the following commands to execute the Central Start/Stop utility:

```
primoe
cd central_start
./run_central
```

3. Enter **1**.

The utility displays the progress of the startup.

Stopping Primo

To shut down Primo:

1. Log on to the master Primo server.
2. Enter the following commands to execute the Central Start/Stop utility:

```
primoe  
cd central_start  
./run_central
```

3. Enter **2**.

The utility displays the progress of the shutdown.

Restarting Primo

To restart Primo:

1. Log on to the master Primo server.
2. Enter the following commands to execute the Central Start/Stop utility:

```
primoe  
cd central_start  
./run_central
```

3. Enter **3**.

The utility displays the progress of the restart.

Verifying System Operation

This procedure verifies the operation of the BO, FE, and SE servers after they have been restarted.

To verify that the system is working correctly:

1. Open the search user interface.
2. Perform a search.
3. Open the Back Office user interface.
4. Check the dashboard (refer to Dashboard Monitoring in the *Primo Back Office Guide*).
5. Check the system reports (refer to [Accessing Reports and Log Files](#)).

Overview of the Primo Directory Structure

This section includes:

- [Introduction](#)
- [The Primo Root Directory](#)
- [The Production Directory](#)
- [The Pipe Run Directory](#)
- [The Jaguar Root Directory](#)
- [Tomcat Directory Structure](#)
- [Primo Directory Aliases](#)



Introduction

The root directory of the Primo system is `/exlibris/primo/p<r>_<x>/`. You can install several instances of the Primo system on a single server. For example, you may initially install an instance of Primo for testing purposes, as well as a production instance. The `<r>` indicates the release in which Primo was initially installed and `<x>` indicates the instance number of the installation.

The Primo system directories and subdirectories are described in the following sections.

The Primo Root Directory

The following table lists the directory structure of the Primo root directory (/exlibris/primo/publish/_ng/primo/).

NOTE: The publish directory is the default directory used by the Primo Back End and the search directory is the default directory used by the Primo Front End.

Primo Root Directory Structure

Path	Description
home/system/publish/bin home/system/search/bin	Contains the Tomcat startup/shutdown scripts and other console application scripts. Alias: be_bin, fe_bin.
home/system/publish/conf home/system/search/conf	Contains the configuration information for the system's Front End and Back Office. Alias: be_conf, fe_conf.
home/profile/publish/publish/templates	Contains publishing pipe configuration files and normalization rule templates.
home/profile/publish/publish/production	Contains the configuration files, normalization mapping set, and data from the publishing pipes in the production environment. For additional information about the structure of the production directory, refer to The Production Directory .

The Production Directory

The production directory includes configuration and data files that play a role in the production environment. This directory is not overwritten when the system is upgraded. Since the production directory can be very large, it can be moved to another location. The location of the production directory is defined in the installation configuration files. The following table lists the subdirectory structure of the production directory (home/profile/publish/publish/production).

Production Directory Structure

Path (relative to the production directory)	Description
conf	Contains the production configuration files used by all pipes, including the PNX XSD and deduplication matching rules. These configuration files are copied from another directory during the installation.
conf/normalizationExport	When a normalization set is deployed, the XML files used by the normalization process are created in this directory. There is a subdirectory for every normalization set.
log	Contains the Tomcat and general Primo log files.
pipes	Contains all the directories created per pipe. The directory name is the name of the pipe. Alias: be_pipes
pipes/<pipe-name>	When a new pipe is created, a directory with the pipe name is created with the following subdirectories: conf and <Source_Code>.
pipes/<pipe-name>/conf	Contains two configuration files: <ul style="list-style-type: none"> •rules.xml—the normalization mapping set used by the pipe. •source_rules_config.xml—information about the data source that is harvested by the pipe.

pipes/<job-name>/<Source_Code>	<p>Contains subdirectories for every run of the pipe. The directory name includes the date and time of the pipe run in the YYYY-MM-DD@HH:MM:SS format. For example: 2006-07-25@20.37.07</p> <p>For additional information about the structure of the pipe run directory, refer to The Pipe Run Directory.</p>
processes	<p>Contains subdirectories for each process. Within each process there are subdirectories, in the date/time format, for every run. The log files for each run are stored in these subdirectories.</p>

The Pipe Run Directory

The pipe run directory is created at the time the pipe is run. The directory name includes the date and time of the pipe run in the YYYY-MM-DD@HH:MM:SS format.

The following table lists the directory structure of the Pipe Run directory.

Pipe Run Directory Structure

Path (relative to the pipe run directory)	Description
harvest	Contains the original files harvested from the source.
log	<p>Contains the following log files:</p> <ul style="list-style-type: none"> •Pipe log—The format of the file is as follows: work_<job-name>.log •Harvest log—The format of the file is as follows: work_<job-name>_harvest.log •DedupGrouper.log—Contains log information pertaining to the Dedup stage. •FrbrGrouper.log—Contains log information pertaining to the FRBR stage.

The Jaguar Root Directory

The following table lists the directory structure of the Jaguar Search Agent root directory (/exlibris/primop<_>_<x>/ng/jaguar/).

Jaguar Directory Structure

Path (relative to the pipe run directory)	Description
home/profile/analysis	Search Engine analysis configuration .
home/profile/indexes/ last_fetched_date_ index.date	Contains the last indexing date.
home/profile/indexes/ last_fetched_date_ mir.date	Contains the last indexing date.
home/system/conf/ search_schema.xml	Search schema configuration file.
home/system/conf/ indexing_schema.xml	Indexing schema configuration file.

Tomcat Directory Structure

The following table lists the directory structure of the Tomcat-related directories.

Tomcat Directory Structure	
Path (relative to the home directory)	Description
profile	Contains your local settings and data. This document does not discuss the structure of this directory.
profile/publish/overwrites profile/search/overwrites	Contain templates of configuration files.
system	This is the active software directory. It is not recommended to change any file in this directory directly. Make all changes in the profile/publish/overwrites and profile/search/overwrites directories and use set_globals to implement the changes.
system/publish/bin system/search/bin system/publish/conf	Contains the Tomcat startup/shutdown and other console application scripts.
system/search/conf	Contains configuration files such as JBPM process definition and menu definition files.
system/publish/conf/i18n system/search/conf/i18n	Contains configuration files for i18n (internationalization) support in the Web interfaces.
system/tomcat/publish system/tomcat/search	The Tomcat home directories.
profile/publish/publish/ production/log profile/search/log	Contains the logs of the Tomcat-based modules.
system/tomcat/publish/ webapps system/tomcat/search/ webapps/	This directory is a package of all other modules (Enterprise Archive - the .ear file) and is the interaction directory between Tomcat and Primo. This directory contains the following files: <ul style="list-style-type: none"> •primo_publishing#admin •primo_library#libweb

	<ul style="list-style-type: none"> •jaguar#web—Contains software for the Search Federator. •PrimoWebServices/primo_publishing#webservices This directory contains a set of .war directories. Each directory is a standalone Web application.
system/tomcat/publish/birt/Report	This directory contains Primo report definition files.

Primo Directory Aliases

The following table lists the available aliases in Primo for quick navigation within the system directories.

NOTE: The <r> indicates the release in which Primo was initially installed and <x> indicates the Primo copy.

Primo System Aliases

Alias	Path
Primo:	
primo_home	/exlibris/primo/p<r>_<x>/ng/primo/home
Front End:	
fe_bin	/exlibris/primo/p<r>_<x>/ng/primo/home/system/search/bin
fe_conf	/exlibris/primo/p<r>_<x>/ng/primo/home/system/search/conf
fe_log	/exlibris/primo/p<r>_<x>/ng/primo/home/profile/search/log
fe_deploy	/exlibris/primo/p<r>_<x>/ng/primo/home/system/tomcat/search
fe_profile	/exlibris/primo/p<r>_<x>/ng/primo/home/profile/search
fe_web	/exlibris/primo/p<r>_<x>/ng/primo/home/system/tomcat/search/webapps/primo_library#libweb
Back Office:	
be_bin	/exlibris/primo/p<r>_<x>/ng/primo/home/system/publish/bin
be_conf	/exlibris/primo/p<r>_<x>/ng/primo/home/system/publish/conf
be_log	/exlibris/primo/p<r>_<x>/ng/primo/home/profile/publish/publish/production/log
be_deploy	/exlibris/primo/p<r>_<x>/ng/primo/home/system/tomcat/publish
be_profile	/exlibris/primo/p<r>_<x>/ng/primo/home/profile/publish

be_production	/exlibris/primop<_>ng/primohome/profile/publish/publish/production
be_pipes	/exlibris/primop<_>ng/primohome/profile/publish/publish/production/pipes
Search Engine:	
se_bin	/exlibris/primop<_>ng/jaguar/home/system/bin
se_conf	/exlibris/primop<_>ng/jaguar/home/system/conf
se_profile	/exlibris/primop<_>ng/jaguar/home/profile
se_log	/exlibris/primop<_>ng/jaguar/home/profile/log

Performing System Cleanup

This section describes the scripts and procedures that you can use to remove unwanted data, indexes, and log files.

CAUTION: The procedures in this section can cause unrecoverable loss of data or unplanned downtime. Contact Ex Libris Support if you need assistance with any of these procedures.

This section includes:

- [Clean Database Scripts](#)
- [Re-Indexing the Database](#)
- [Deleting and Re-Indexing the Database](#)
- [Removing Unwanted Data](#)
- [Cleaning Primo Log Directories](#)
- [Cleaning Apache Log Directories](#)
- [Cleaning PDS Log Files](#)
- [Disk Space Commands](#)

Clean Database Scripts

The following scripts are used to clean data from Primo sites:

- `clean_indexes.sh`—This script deletes the indexes, which is the first step of re-indexing from scratch. For more information, see [Re-Indexing the Database](#).
- `clean_data_before_reload.sh`—This script deletes the PNX database and indexes, but does not delete dedup IDs and user-generated content (including tags, reviews, e-Shelf, and statistics). This script is intended for sites that are in production and need to re-create the PNX database and re-index from scratch. For more information, see [Deleting and Re-Indexing the Database](#).

CAUTION: Following the execution of the `clean_data_before_reload.sh` command, pipes can take longer to run.

- `clean_data_all_before_golive.sh`—This script deletes the entire database, including all user-generated data, indexes, and dedup IDs. It is intended for Primo sites that are switching to production, so that they can start with a clean database.

All of the scripts are located and executed from the `be_bin` directory. A log file is created for each script under the same directory, using the following naming convention:

`<script_name>.log`

To run a script, enter the following commands:

`be_bin`

`<script_name>`

Re-Indexing the Database

This procedure allows you to re-index from scratch without causing downtime. The Front End uses existing indexes for search retrieval, and hotswapping replaces the old indexes with new ones.

To re-index the database without deleting the current indexes in memory:

1 Log on to the FE server and enter the following commands to view the search_schema.xml file:

```
se_conf
```

```
vi search_schema.xml
```

2 Search for the <agent> tag to get the path to the <indexes> directory.

In the following example, /exlibris/primo/p1_1/ng/jaguar/home/profile/indexes is the path of the indexes directory:

```
<agents desc="collections of all search instances to be searched by jaguar"
auto_deploy="active" swapping_type="slices">
<agent port="9501" connection_type="local" host="primo1.corp.exlibrisgroup.com"
active="true">
<slices slice_count="1" name="main" active="true">
<slice id="1">
<path>/exlibris/primo/p1_1/ng/jaguar/home/profile/indexes/
inst_1_index</path>
<load2ram>true</load2ram>
</slice>
</slices>
</agent>
```

3 On each SE server, move to the directory that contains the indexes directory and enter the following command to back it up, where <date> is the current date:

```
cp -Rp indexes indexes.<date>
```

4 Log on to the BE server as the primo user.

5 Enter the following commands to delete the index files:

```
be_bin
```

```
./clean_indexes.sh
```

6 From a Web browser, enter the URL for the Back Office server. For example:

<http://primo.exlibris.com:1602/primo...acegilogin.jsp>

7 From the Back Office home page, click Primo Home > Monitor Primo Status > Process Monitoring.

8 To re-index the database, click Execute for the Indexing_and_Didumean_and_Hotswapping process.

9 Log on to the FE server as the primo user.

10 Enter the following commands to monitor the job execution on the FE:

```
fe_log
```

```
tail -f library_server.log
```

NOTE: Do not restart the FE and SE servers so that you will be able to keep searching with the existing indexes.

Deleting and Re-Indexing the Database

This procedure allows the customer to perform a full re-load of all data sources and re-indexing from scratch, since the database and indexes are first deleted. Until the re-load and re-indexing has completed, the system will not be available. This procedure is mainly used during the implementation phase to test normalization rule sets to see how data displays in the FE. If you want to re-index from scratch only, see [Re-Indexing the Database](#).

NOTES:

You can re-load an entire data source without first deleting the database by running an "update" pipe. This means that all of the records in the database for the relevant data source are updated and re-indexed. You can run an "update" pipe by using the following Pipe Type option on the Define Pipe page in the Primo Back Office:

☒ No Harvesting - Update Data Source – This pipe uses the data from the already harvested records for the data source.

To delete and re-index the database:

1 Log on to the BE server as the primo user and enter the following command to stop the Back Office:

```
be_stop
```

2 Enter the following commands to clean the database and indexes. If you want to clean only the indexes, use the `clean_indexes.sh` command instead of the `clean_data_before_reload.sh` command.

```
be_bin
```

```
./clean_data_before_reload.sh
```

CAUTION: Following the execution of the `clean_data_before_reload.sh` command, pipes can take longer to run.

If you want to clean only the indexes, use the `clean_indexes.sh` command.

3 Enter the following command to restart the Back Office:

```
be_restart
```

4 For the FE server and the SE servers, log on as the primo user and enter the relevant commands to restart the Primo modules:

```
afe_stop
```

```
bfe_start
```

```
cse_stop
```

```
dse_start
```

5 From a Web browser, enter the URL for the Back Office server. For example:

<http://primo.exlibris.com:1602/primo...acegilogin.jsp>

6 From the Back Office home page, click Primo Home > Monitor Primo Status > Pipe Monitoring.

7 For each relevant pipe, click Execute.

NOTE: Since only one pipe can run at a time, execute the next relevant pipe after the previous pipe completes.

8 After all relevant pipes have completed, log on to the BE server as the primo user and enter the following command to connect old tags with PNX records:

```
./restore_ext_tags.sh
```

9 From the Back Office home page, click Primo Home > Monitor Primo Status > Process Monitoring.

10 Click Execute for the Indexing_and_Didumean_and_Hotswapping process.

11 Log on to the FE server as the primo user.

12 Enter the following commands to monitor the job execution on the FE:

```
fe_log
```

```
tail -f library_server.log
```

Removing Unwanted Data

After an extended period of use, it may become necessary to remove unwanted user data from the Primo directory structure.

To remove unwanted data:

1Verify that there are no pipes running.

2Enter the `be_pipes` alias to move to the pipes directory. For example:

```
be_pipes
```

3Enter the `pwd` command to display the PATH of the current directory:

```
/exlibris/primo/p1_1/ng/primo/home/profile/publish/publish/production/pipes
```

4Enter the `ls` command to display the contents of the directory. For example:

```
ls
Blue_Bay_Aleph/ Primo_Demo_Marc_Pipe/
Blue_Bay_Aleph_No_Harvest/ Primo_Demo_Metalib_Pipe/
Blue_Bay_Digitol/ Primo_Demo_Metalib_Update_DS/
```

5Enter the following commands to move to a pipe directory:

```
cd <pipe name>/<source code>
```

6Enter the `ls` command to display the contents of the directory. For example:

```
ls
2008-11-20@19.00.56/ 2008-11-20@19.17.18/ 2008-11-20@19.25.04/
2008-11-20@19.14.32/ 2008-11-20@19.17.28/ 2008-11-20@19.37.32/
```

7Enter the following command to remove all directories and subdirectories from the current directory:

```
rm -r *
```

8Enter the `<fe|be|se>_log` alias to move to the logs directory. For example:

```
be_log
```

9Delete the contents of each log directory.

Cleaning Primo Log Directories

The following scripts allow you to clean the Primo log directories:

- `clean_all_logs.sh`—deletes BE, FE, SE logs in an all-in-one environment or a mixed environment (for example, FE and SE roles are assigned to the same machine). This command is stored under the `be_bin` directory.
- `clean_be_logs.sh`—deletes BE logs on servers that are assigned the BE role. This command is stored under the `be_bin` directory.
- `clean_fe_logs.sh`—deletes FE logs on servers that are assigned the FE role. This command is stored under the `fe_bin` directory.
- `clean_se_logs.sh`—deletes SE logs on servers that are assigned the SE role. This command is stored under the `se_bin` directory.

To execute a delete Primo log script:

1Log on to the server as the primo user.

2Enter an alias to move to the directory that contains the log script that you want to run. For example, to run the `clean_all_logs.sh` script, enter the following command:

```
be_bin
```

3Enter the delete log script. For example, to clear all log directories, enter the following command:

```
./clean_all_logs.sh [number of days old]
```

NOTE: If you do not specify a value for `[number of days old]`, the default value of 30 days is used.

4At the following prompt, enter `y` to delete the log files that are 30 days old:

```
Are you certain you want to delete the BE, FE and SE logs that are  
30 days old? Continue? y/n:
```

Cleaning Apache Log Directories

The `clean_apache_logs.sh` script allows you to clean the Apache log directory by backing up and compressing the following log files:

- `access_log`
- `access_log_ssl`
- `error_log`
- `error_log_ssl`

This script is located under the following directory (`be_bin`), where `<r>` indicates the release in which Primo was initially installed and `<c>` indicates the Primo copy:

```
/exlibris/primo/p<r><c>/ng/primo/home/system/publish/bin
```

To execute the delete Apache log script:

1 Log on to the server as the primo user.

NOTE: If your environment redirects Apache to port 80, you will need to log on as the root user to run the script. To log on as the root user, enter `su` and then enter the password of the root user when prompted.

2 Enter the following command to move to the directory that contains the log script:

```
be_bin
```

3 Enter the following command to run the script:

```
./clean_apache_logs.sh
```

Cleaning PDS Log Files

The `clean_pds_logs.sh` backs up the active PDS log file to a compressed file, adding a time stamp to the file name (for example, `pds_server.log.24Mar09.gz`) and creates an empty `pds_server.log` file in the source directory (`$LOGDIR`). Ex Libris Support recommends that you run the script once a month.

This script is located under the following directory (`be_bin`), where `<x>` is the Primo version in which Primo was first installed and `<r>` is the Primo instance:

```
/exlibris/primo/p<r>_<x>/ng/primo/home/system/publish/bin
```

To execute the delete PDS log script:

1Log on to the server as the primo user.

2Enter the following command to move to the directory that contains the log script:

```
be_bin
```

3Enter the following command to run the script:

```
./clean_pds_logs.sh
```

Disk Space Commands

The amount of available disk space in your file systems determines the amount of data you can store in your database and make available to your users. After setting up the initial amount of disk space, your system might need more space— for example, if you add new libraries to your data sources.

Disk space in the following two locations can become a critical factor:

- Pipes directories
- Index directory

Checking Disk Space

To check disk space in the pipes directory:

1Enter the following command to move to the production directory:

```
be_production
```

2Enter the following command to display the total amount of disk space of the directory, the amount of used disk space, and the available disk space:

```
df -kh
```

The following is an example of the output:

```
primo.corp.exlibrisgroup.com-p1(1) >>be_production
primo.corp.exlibrisgroup.com-p1(1) >>df -kh
Filesystem Size Used Avail Use% Mounted on
filer01:/vol/data4/primo
400G 359G 42G 90% /exlibris
primo.corp.exlibrisgroup.com-p1(1) >>
```

To check disk space in the index directory:

1Go to the index directory.

2Enter the following command to display disk information:

```
df -kh
```

The following is an example of the output:

```
ll-primo05.corp.exlibrisgroup.com-p1(1)>>df -kh
Filesystem Size UsedAvailUse%Mounted on
Filer01:/vol/data1/primoshare 15G 14G91% /exlibris_primo
```

Freeing Disk Space

You can free disk space by cleaning up the old pipes and process execution directories, as well as the log directories. For more information on cleaning log directories, refer to [Cleaning Primo Log Directories](#).

Primo Port Configurations

This section includes:

- [Introduction](#)
- [Redirecting Ports](#)
- [IP Tables](#)
- [Preventing Unwanted Re-Direction](#)

Introduction

The default HTTP ports for Primo's Back Office (BO), Front End (FE), and PDS are 160y, 170y, and 899y, respectively (where y is the copy number, starting with 1). The default HTTPS port (Secure HTTP) for PDS is 1443.

The non-standard nature of these defaults can present a barrier to use. To overcome this barrier, these ports should not be changed and access to them should not be blocked. Instead, the standard ports (such as 80 for HTTP and 443 for HTTPS) should be redirected or translated to Primo's default ports.

IMPORTANT: **IMPORTANT:** The use of mod_jk to redirect the port to port 80 is not recommended or supported by Ex Libris Support.

Redirecting Ports

To perform port redirection on your Primo server, you can choose from the following methods:

- A web server can either be a proxy or rewrite the URL.
- A network-based or host-based firewall can perform Port Address Translations (PAT) and Network Address Port Translations (NAPT).
- A network-based load balancer can perform PAT and NAPT.

Given the large variety of environments, there is no single solution that will work in all cases. A solution should be selected based on current and future need, along with available resources and technical skill. The ability of the vendors to assist with both initial setup and ongoing issues should also be considered.

To redirect ports, Ex Libris recommends the use of IP tables, which is a host-based firewall found on most modern Linux systems. It is often already being used, is well understood, and performs well. An example implementation can be found below.

NOTE: Because port redirection is provided by load balancers and firewalls, Ex Libris does not provide support for the methods described in this section.

Overlapping Services

A request needs to be uniquely identifiable in order for it to be redirected correctly. In a default configuration, using a different port is sufficient. However, there can be a problem if two services (such as the Primo FE and PDS) run on the same server and need to be accessed from the same port (such as port 80). In this case, the only uniquely identifiable element is the URL. To use the URL, the solution must be Layer 7 aware – that is understands HTTP/HTTPS. This can both limit possible solutions and increase complexity.

One alternative is to setup PDS to use SSL-encrypted sessions to move the default PDS port to 1443 and the redirect port to 443. Since the services no longer need the same port, there is no longer a conflict.

Another alternative is to use distinctive IP addresses and/or DNS names to change the destination port based on the destination IP address and DNS name. For example, assume that the Primo FE and PDS both run on server1, which has IP address = 10.20.30.40 and DNS server1=10.20.30.40. If IP address 10.20.30.41 and DNS server1-pds = 10.20.30.41 are added to server1 to change the access for Primo and PDS to server1-pds, port redirection would be set up as follows:

- server1:80 redirects to server1:1701
- server1-pds:80 redirects to server1-pds:8991

IP Tables

Linux 2.4.x and 2.6.x kernels provide the `ip_tables` packet filter that allows for stateful and stateless packet filtering, NAT, PAT, and other packet manipulations. Specific rules are grouped into chains, which are also grouped into tables. The command-line program `iptables` allows you to set up, maintain, and inspect the tables, chains, and rules. In the examples below, each solution is presented as a set of `iptables` commands. For more information, refer to the `iptables` documentation for your operating system.

Since these commands must usually be run as the root user, they should be reviewed carefully before execution. In addition, the commands will need to be run every time the server is rebooted. On many systems this can be automated by saving the output of `iptables-save` in `/etc/sysconfig/iptables`.

IP Table Examples

In the following examples, Primo and the PDS are both installed on a single server called `server1` (All-in-One topology). It has an IP address of `10.10.5.5` in which the DNS maps to `primo.library.edu`. The BO, FE, and PDS use the default ports of `1601`, `1701`, and `8991`, respectively.

Example 1: Port 80 to Primo FE

This is the simplest port redirection that consists of the following parts:

- Redirecting requests on port 80 to port 1701.
- Ensuring access to ports 80 and 1701.
- Allowing established connections to continue.

NOTE: This solution can also be used to redirect to the BO and PDS.

```
# Port Redirection
iptables -t nat -A OUTPUT -d localhost -p tcp --dport 80 -j REDIRECT --to-ports 1701
iptables -t nat -A OUTPUT -d 10.10.5.5 -p tcp --dport 80 -j REDIRECT --to-ports 1701
iptables -t nat -A PREROUTING -d 10.10.5.5 -p tcp --dport 80 -j REDIRECT --to-ports 1701

# Allow access to ports 80 and 1701
iptables -A INPUT -p tcp -i eth0 --dport 80 -j ACCEPT
iptables -A INPUT -p tcp -i eth0 --dport 1701 -j ACCEPT

# Allow established connections to continue
iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
```

Example 2: Port 80 to Primo FE, Port 443 to PDS

This port redirection deals with the issue of overlapping services by using PDS with SSL. See the Patron Directory Services Guide for details on setting up PDS to run on port 1443 with SSL.

In addition, the URLs for the PDS_URL, PDS_INTERNAL_URL, and PDS_Configuration_URL fields on the Primo Home > Advanced Configuration > General Configuration > Installation subsystem page needs to be updated (for example, <http://primo.library.edu:8991/pds> to <https://primo.library.edu/pds>).

```
# Port Redirection - FE
iptables -t nat -A OUTPUT -d localhost -p tcp --dport 80 -j REDIRECT --to-ports 1701
iptables -t nat -A OUTPUT -d 10.10.5.5 -p tcp --dport 80 -j REDIRECT --to-ports 1701
iptables -t nat -A PREROUTING -d 10.10.5.5 -p tcp --dport 80 -j REDIRECT --to-ports 1701

# Port Redirection - PDS
iptables -t nat -A OUTPUT -d localhost -p tcp --dport 443 -j REDIRECT --to-ports 1443
iptables -t nat -A OUTPUT -d 10.10.5.5 -p tcp --dport 443 -j REDIRECT --to-ports 1443
iptables -t nat -A PREROUTING -d 10.10.5.5 -p tcp --dport 443 -j REDIRECT --to-ports 1443

# Allow access to ports 80, 1701, 443, & 1443
iptables -A INPUT -p tcp -i eth0 --dport 80 -j ACCEPT
iptables -A INPUT -p tcp -i eth0 --dport 1701 -j ACCEPT
iptables -A INPUT -p tcp -i eth0 --dport 443 -j ACCEPT
iptables -A INPUT -p tcp -i eth0 --dport 1443 -j ACCEPT

# Allow established connections to continue
iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
```

Example 3: Port 80 to Primo FE and PDS

This port redirection deals with the issue of overlapping services by adding the IP address 10.12.6.6 to server1. A DNS entry is created that maps pds.library.edu to 10.12.6.6.

In addition, the URLs for the PDS_URL, PDS_INTERNAL_URL, and PDS_Configuration_URL fields on the Primo Home > Advanced Configuration > General Configuration > Installation subsystem page needs to be updated (for example, <http://primo.library.edu:8991/pds> to <http://pds.library.edu/pds>).

```
# Port Redirection - FE (10.10.5.5)
iptables -t nat -A OUTPUT -d localhost -p tcp --dport 80 -j REDIRECT --to-ports 1701
iptables -t nat -A OUTPUT -d 10.10.5.5 -p tcp --dport 80 -j REDIRECT --to-ports 1701
```

```
iptables -t nat -A PREROUTING -d 10.10.5.5 -p tcp --dport 80 -j REDIRECT --to-ports 1701
```

```
# Port Redirection – PDS (10.12.6.6)
```

```
iptables -t nat -A OUTPUT -d 10.12.6.6 -p tcp --dport 80 -j REDIRECT --to-ports 8991
```

```
iptables -t nat -A PREROUTING -d 10.12.6.6 -p tcp --dport 80 -j REDIRECT --to-ports 8991
```

```
# Allow access to ports 80, 1701, & 8991
```

```
iptables -A INPUT -p tcp -i eth0 --dport 80 -j ACCEPT
```

```
iptables -A INPUT -p tcp -i eth0 --dport 1701 -j ACCEPT
```

```
iptables -A INPUT -p tcp -i eth0 --dport 8991 -j ACCEPT
```

```
# Allow established connections to continue
```

```
iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
```

```
iptables -A OUTPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
```

Preventing Unwanted Re-Direction

Primo allows you to prevent unwanted re-direction from the PDS sign-in page by defining a list of trusted sites in the following file:

`/p4_X/ng/primo/home/system/search/conf/trusted-sites.txt`

This list should include all Primo Front End host names, one per row. If any host names are defined, the system will prevent the re-direction to any other servers and display an error page.

After a change to the file, the Front End servers must be restarted.



Accessing Reports and Log Files

You can access reports and log files to obtain information about your system.

This section includes:

- [Primo Reports](#)
- [Accessing Primo Logs](#)

Primo Reports

The following table lists the reports available in Primo. The Primo reports are created in printable PDF format. For information on how to access reports, refer to the Primo Back Office Guide.

Primo Reports	
Report Name	Description
List Reports:	
Code tables	Displays all code tables per subsystem. Includes information on the level of the table and if it is enabled.
Data Sources	Displays the sources of data in the system.
General Parameters	Displays the system parameters.
IPs	Displays the on campus IP addresses associated with institutions.
Institutions	Displays the system institutions.
Mapping Tables	Displays the mapping tables by subsystem.
Pipes	Displays the pipes used to publish system data. Includes the harvesting and normalizer types and when the pipe was last run.
Processes	Displays the processes and when they were last run.
Restrictions	Displays the search restrictions. Includes the user group, and if the restrictions apply to users that are on or off campus.
Scopes	Displays the search scopes and includes search and delivery restrictions.
Staff Users	Displays the registered system users, their roles, and contact information.
Top Searches Based on Processing Time	Displays the search string, number of searches, average number of results, and average response time of the top searches, based on processing time.

Top Searches with No Results	Displays the search string, number of searches, and average response time of the top searches with no results.
Views	Displays the preconfigured system screen views and whether they are currently deployed.
Statistical Reports:	
Browse String Statistics	Statistics concerning the search strings used for Browse searches performed in the system.
Click Through Statistics	Lists statistics concerning the number of clicks registered for different records in the system.
Hourly Remote Statistics	Lists the remote search statistics, broken down by the hour and length of the search in seconds.
Hourly Search Statistics	Lists the local search statistics, broken down by the hour and length of the search in seconds.
PNX Click Statistics	Graphs the local and remote search statistics, which includes the total searches, number of results, and response time.
PNX Statistics	Lists PNX statistics.
Search Statistics	Statistics concerning the searches performed in the system.
Search Strings Statistics	Statistics concerning the search strings used for searches performed in the system.
System Monitoring	The I/O, memory, and CPU usage by time.
Users Personal Sets Report	Statistics concerning personal sets created with the Find Database function (MetaLib).
Annual Event Statistics (new)	Displays annual statistics for a selected institution, view, and event.
Annual Search Statistics (new)	Displays annual search statistics for a selected institution and view.
Click Events Statistics (new)	Displays click event statistics for a selected institution, view, and period of time.
Daily Search Statistics (new)	Displays hourly statistics for a selected institution, view, and day.
Facet Clicks Statistics (new)	Displays facet click statistics for a selected institution, view, and period of time.

Search Statistics (new)	Displays search statistics for a selected institution, view, and period of time.
Search Terms by Letter Statistics (new)	Displays the top search terms by letter for a selected institution, view, and period of time.
Top Search Terms Statistics (new)	Displays the top search terms for a selected institution, view, and period of time.

Accessing Primo Logs

Pipe Logs

Errors can be viewed in the Back Office user interface. If there are no errors in the Back Office user interface, but the pipe stopped in the middle of a pipe run, you need to check the log file for the pipe on the server.

To view a pipe's logs on the server:

1 Enter the following commands to move to the pipes directory:

```
be_pipes
```

```
cd <pipe_name>/<source_name>/<pipe_date>/log/
```

Some common errors found in the pipe logs are:

- No valid files found in source directory—This can occur when Date to harvest from is later than the date of the file.
- Failed while trying to split records—Indicates a problem with your source files.

The log file includes the following information:

- Timestamp
- Type of message, which can be one of the following:
 - FATAL—The pipe failed.
 - ERROR—A record failed during processing.
 - WARN—A non-critical error occurred.
 - INFO—Information messages.
 - DEBUG—Debug messages generated only during debugging of the application.
- Thread name—Indicates the pipe task and bulk being processed.
- Log message

The following is a sample section of a pipe log file:

```
08-08-2006 12:45:23,504 INFO [DirectoryTreeBuilder] () - Building directory tree.
The production path is /exlibris/primo/p1_1/ng/primo/primo_...08-08@12.45.17
08-08-2006 12:45:23,530 INFO [DirectoryTreeBuilder] () - Loading bulk data into
the database.
08-08-2006 12:45:23,574 INFO [DirectoryTreeBuilder] () - Execution of
com.exlibris.publish.engine.harvest.DirectoryTreeBuilder took 89
08-08-2006 12:45:23,584 INFO [DirectoryTreeBuilder] () - Token messaged
consumed.
08-08-2006 12:45:23,585 INFO [DirectoryTreeBuilder] () - Loading token to resume
execution.
08-08-2006 12:45:23,591 INFO [DirectoryTreeBuilder] () - Resuming token
08-08-2006 12:45:23,616 INFO [DirectoryTreeBuilder] () - Finished processing
thread
```

```
08-08-2006 12:45:28,114 INFO [OAISplitter] () - Saving bulk. bulk Id=0
deleted=false
08-08-2006 12:45:29,382 INFO [OAISplitter] () - Loading bulk data into the
database.
08-08-2006 12:45:33,150 INFO [OAISplitter] () - Saving bulk. bulk Id=1
deleted=false
08-08-2006 12:45:34,241 INFO [OAISplitter] () - Loading bulk data into the
database.
```

Tomcat-Related Logs

The Tomcat-related log files can be accessed by entering the following aliases on the server:

- **be_log** – The /exlibris/primop<v>_<c>/ng/primop/home/profile/publish/publish/production/log directory holds the library_server.log and publish_server.log files.
- **fe_log** – The /exlibris/primop<v>_<c>/ng/primop/home/profile/search/log directory holds the localhost_access_log.log files.
- **se_log** – The /exlibris/primop<v>_<c>/ng/jaguar/home/profile/log directory holds the agent_9501.log files.

localhost_access_log.log

The localhost_access_log.<date>.log is an HTTP access log. It lists all HTTP requests with a GET/POST listing. This log file is similar to the Apache HTTPD server access log.

The following is an example of the contents of a log file:

```
10.1.116.135 - - [22/Mar/2015:14:59:46 +0200] "GET /primop_library/libweb/
webservices/rest/v1/pnxs/PC/TN_gale_ofa324979896?inst=PC HTTP/1.1" 200 53
22905 - - 10.1.116.135
10.1.116.135 - - [22/Mar/2015:14:59:46 +0200] "GET /favicon.ico HTTP/1.1" 200
748 83 - - -
```

To access the search logs on the server:

1 Enter the following command to move to the search/log directory:

```
fe_log
```

library_server.log, publish_server.log, and agent_9501.log

The library_server.log, publish_server.log, and agent_9501.log files are the main log files for Tomcat activity. Primo modules that run under Tomcat write information to this log file.

Each time the server is restarted, a timestamp is added. Your log directory might appear as follows:

```
-rw-rw-r-- 1 prm exlibris 174985 Dec 29 23:57 library_server.log.2005-12-29
-rw-rw-r-- 1 prm exlibris 17546674 Dec 30 23:57 library_server.log.2005-12-30
-rw-rw-r-- 1 prm exlibris 281074 Dec 31 23:57 library_server.log.2005-12-31
```

```
-rw-rw-r-- 1 prm exlibris 5698992 Jan 1 11:09 library_server.log.0101.1050
-rw-rw-r-- 1 prm exlibris 36822 Jan 1 10:51 library_server.log.0101.1100
-rw-rw-r-- 1 prm exlibris 36822 Jan 1 11:01 library_server.log.0101.1108
-rw-rw-r-- 1 prm exlibris 142407 Jan 1 14:25 library_server.log
```

From this example, it is clear that the server was up from the 29/12 - 31/12 (rolled over). On 01/01, the server was restarted at 10:50, 11:00, and 11:08.

The library_server.log file is the current active log file. Using the following command can be useful when debugging the server.

```
tail -f library_server.log
```

For example:

```
2005-11-06 19:10:17,988 INFO [com.exlibris.Primo.delivery.DeliverySessionsMng]
Cleaning old delivery session - 1131296885128.
```

```
2005-11-06 19:15:32,941 INFO [com.exlibris.Primo.repository.jobs.
IndexFileSynchronizerJob] found 8 items to index
```

```
2005-11-06 19:15:33,425 INFO [com.exlibris.Primo.
repository.jobs.IndexFileSynchronizerJob] Indexing DIGITAL ENTITY:4530 at
484ms
```

```
2005-11-06 19:08:49,128 ERROR [com.exlibris.Primo
.delivery.DefaultAccessRightsChecker] Fail to parse access right metadata.
Exception: error: The element type "xb:access_right_md" must be terminated by
the matching end-tag "</xb:access_right_md>". Cause:
```

```
org.xml.sax.SAXParseException: The element type "xb:access_right_md" must be
terminated by the matching end-tag "</xb:access_right_md>".
```

Watchdog Logs

The watchdog.log file enables you to keep track of all watchdog activity and the status of all monitored components.

To view the log file:

1Enter the following command to move to the /exlibris/primo/p3_1/ng/jaguar/home/profile/log directory:

```
se_log
```

2Enter the following command to view the log file:

```
vi watchdog.log
```

For example:

```
2008-08-29 09:29:04,765 INFO [t-Thread-0] [c-Monitorer] - Loading of xml
completed
```

2008-08-29 09:29:07,761 INFO [t-Thread-0] [c-Monitorer] - Scheduler started, adding processes

2008-08-29 09:29:07,761 INFO [t-Thread-0] [c-Monitorer] - process 1, named Database has been scheduled to run every 10 seconds

2008-08-29 09:29:07,761 INFO [t-Thread-0] [c-Monitorer] - process 2, named Back End has been scheduled to run every 10 seconds

2008-08-29 09:29:07,761 INFO [t-Thread-0] [c-Monitorer] - process 3, named PDS has been scheduled to run every 60 seconds

2008-08-29 09:29:07,761 INFO [t-Thread-0] [c-Monitorer] - process 4, named Front End has been scheduled to run every 10 seconds

2008-08-29 09:29:07,761 INFO [t-Thread-0] [c-Monitorer] - reset watchdog configuration

2008-08-29 09:29:08,790 INFO [t-DefaultQuartzScheduler_Worker-1] [c-MonitorSchedulerJob] - be.Back End retruned status: 'OK'

2008-08-29 09:29:08,837 INFO [t-DefaultQuartzScheduler_Worker-0] [c-MonitorSchedulerJob] - be.Database retruned status: 'OK'

2008-08-29 09:29:09,773 INFO [t-DefaultQuartzScheduler_Worker-2] [c-MonitorSchedulerJob] - be.PDS retruned status: 'NOT OK'

2008-08-29 09:29:15,645 INFO [t-DefaultQuartzScheduler_Worker-0] [c-EmailNotifier] - Sending failure email notification to

Managing the Primo Database

The Primo database is based on Oracle 11g RDBMS.

This section includes:

- [Introduction to the Oracle Database](#)
- [Database Schema](#)
- [Database Utilities](#)

Introduction to the Oracle Database

A typical Primo installation includes a single database schema, P<r><x>_PRM00, where <r> indicates the release in which Primo was initially installed and <x> indicates the number of the installation (for example, P21_PRM00). This schema contains data and configuration information.

Oracle Users in Primo

The following are the Primo Oracle users:

- P<r><n>_PRM00—Connects to the Oracle database (default password: P<r><n>_PRM00). This user can select, insert, update, and delete tables from the P<r><n>_PRM00 schema.
- PRIMO_ADMIN—In addition to the privileges of the P<r><n>_PRM00 user, this user can create, delete, and alter Oracle tables, indexes, users, triggers, and so forth. The PRIMO_ADMIN user is used in all Primo procedures.
- PRIMO_DBA—Provides access to database administration operations on the Primo system, such as startup and shutdown.
- P<r><n>_RPT00: Provides access to the RPT00 views, which are configured from tables in PRM00. These views allow you to create your own BIRT reports, which can be viewed from the Back Office. For example, the PRIMO_BO_AUDIT_TABLE view provides access to the data regarding database activity in the Back Office. For more information on creating user-defined reports, see the Primo Technical Guide.
- P<r><n>_PRM00_SHARED: Used internally by the system to manage shared data for Primo Central and should not be removed.

The connection between these Oracle users and Primo servers and procedures is transparent to the end user using the Primo Front End and Back Office interfaces.

Tablespaces in Primo

An Oracle database consists of several logical units called tablespaces. Each tablespace consists of one or more physical data files that can be stored on one or more disks.

The following table shows the Primo tablespaces. Note that each tablespace may have multiple files.

Primo Tablespace Information

Tablespace Name	Usage	Physical File
LOG	Oracle DB system tables	/exlibris/oradata/prm3/log01.dbf
SYSAUX	Oracle DB system tables	/exlibris/oradata/prm3/ sysaux01.dbf
SYSTEM	Oracle system tables	/exlibris/oradata/prm3/ system01.dbf

TEMP	Temporary space for sorting, index creation, and so forth.	/exlibris/oradata/prm3/temp01.dbf
TS_P_DAT	Primo tables	/exlibris/oradata/prm3/prm3_ts_p_dat_01.dbf
TS_P_IDX	Primo indexes	/exlibris/oradata/prm3/prm3_ts_p_idx_01.dbf
TS_ENRICH_DATA	PNX enrichment related content	/exlibris/oradata/prm3/prm3_ts_enrich_data_01.dbf
TS_ENRICH_IDX	Enrichment indexes	/exlibris/oradata/prm3/prm3_ts_enrich_idx_01.dbf
TS_EXL_DEF	Ex Libris definitions	/exlibris/oradata/prm3/prm3_ts_exl_def_01.dbf
TS_PNX_ADD_DATA	PNX additional data	/exlibris/oradata/prm3/prm3_ts_pnx_add_data_01.dbf
TS_PNX_ADD_IDX	PNX additional data indexes	/exlibris/oradata/prm3/prm3_ts_pnx_add_idx_01.dbf
TS_PNX_DATA	PNX content	/exlibris/oradata/prm3/prm3_ts_pnx_data_01.dbf
TS_PNX_IDX	PNX indexex	/exlibris/oradata/prm3/prm3_ts_pnx_idx_01.dbf
TS_PNX_EXT_DATA	PNX extension content	/exlibris/oradata/prm3/prm3_ts_pnx_ext_data_01.dbf
TS_PNX_EXT_IDX	PNX extension indexes	/exlibris/oradata/prm3/prm3_ts_pnx_ext_idx_01.dbf
TS_SRC_REC_DATA	Harvested source record content	/exlibris/oradata/prm3/prm3_ts_src_rec_data_01.dbf
TS_SRC_REC_IDX	Harvested source record indexes	/exlibris/oradata/prm3/prm3_ts_src_rec_idx_01.dbf
TS_STATS_DATA	Primo application statistics content	/exlibris/oradata/prm3/prm3_ts_stats_data_01.dbf
TS_STATS_IDX	Primo application statistics indexes	/exlibris/oradata/prm3/prm3_ts_stats_idx_01.dbf
TSLOB	Oracle DB system tables	/exlibris/oradata/prm3/tslob01.dbf

UNDOTBS1	Oracle DB system tables	/exlibris/oradata/prm3/ undotbs101.dbf
USERS	Oracle DB system tables	/exlibris/oradata/prm3/users01.dbf

Database Schema

The Primo database consists of many tables used to store data, system information, process states, and more. The following is a list of the table types in the Primo database. The prefix of each table describes its major function.

Primo Database Tables

Schema	Description	Tables
C_C Tables	Common Configuration Tables. Tables with common fields are grouped together into a single table to improve processing speed. One of the fields of a common table is the name of the table to which each record belongs. This group of tables contains configuration data for these common tables and all the records of the tables.	<ul style="list-style-type: none"> • C_C_TABLE_OF_TABLES—contains metadata information for the C_C_MAPPING_TABLES and C_C_CODE_TABLES tables. • C_C_CODE_COLUMN_NAMES—contains metadata information on each column for the C_C_MAPPING_TABLES and C_C_CODE_TABLES tables. • C_C_MAPPING_TABLES—contains the records of the generic tables (each table has six columns). • C_C_CODE_TABLES—contains the records of code tables (for internalization support).
C_G Tables	Global Configuration Tables. Tables that contain global system configuration parameters.	<ul style="list-style-type: none"> • C_G_CONFIGURATION—contains global system configuration parameters.
C_I Tables	Institution Configuration Tables. Tables that contain configuration parameters concerning the institution providing bibliographical services.	<ul style="list-style-type: none"> •C_I_INSTITUTION •C_I_INSTITUTION_IP •C_I_LIBRARY •C_I_RESTRICTION
C_N Tables	Normalization Configuration Tables. Tables that contain global normalization rules for the publishing process.	<ul style="list-style-type: none"> •C_N_BASIC_ROUTINES •C_N_ENRICHMENT •C_N_ENRICHMENT_MAPPING

		<ul style="list-style-type: none"> •C_N_ENRICHMENT_SET •C_N_MAPPING_SET •C_N_MAPPING_SOURCE •C_N_MAPPING_TARGET • C_N_MAPPING_TRANSFORMATION •C_N_SCOPES •C_N_SOURCE_CONDITION • C_N_TRANSFORMATION_GROUP •C_N_DATA_SOURCE • C_N_PROCESS_CONFIGURATION
C_V Tables	Viewing Configuration Tables. Tables that contain end user viewing information.	<ul style="list-style-type: none"> •C_V_LAYOUT_SET •C_V_LAYOUT_SET_PAGES •C_V_PAGES •C_V_PAGE_TILES •C_V_SCOPES •C_V_SCOPE_VALUES •C_V_TABS •C_V_TAB_SCOPE •C_V_TILES •C_V_VIEWS •C_V_VIEW_LAYOUT_SET •C_V_VIEW_TILE_CONFIG
M_P Tables	Pipe Monitoring Configuration Tables. Tables that contain pipe monitoring information.	<ul style="list-style-type: none"> •M_P_BULK_CONTEXT •M_P_FAILED_RECORDS •M_P_PROCESS_CONTEXT •M_P_WORK •M_P_PROC_EXE
R Tables	Parameters Configuration Tables. Tables that contain user parameters.	<ul style="list-style-type: none"> •R_ROLE •R_STAFF •R_STAFF_ROLE_MAPPING
S Tables	Statistics Configuration Tables. Tables that contain accumulated system statistics. The Primo reporting system uses these tables as its data source.	<ul style="list-style-type: none"> •S_CLICK_SUMMARIES •S_SEARCH_SUMMARIES

U Tables	User Configuration Tables. Tables that contain end user customization and personal parameters.	<ul style="list-style-type: none"> ●U_BASKET—contains the electronic basket. ●U_PREFERENCE—contains user preferences. ●U_RSS—contains RSS information. ●U_SAVED_SEARCH—contains user-saved searches.
P Tables	PNX Configuration Tables. PNX-related tables.	<ul style="list-style-type: none"> ●P_PNX—contains all the PNX records. For a full description of this table, refer to P_PNX Table. ●P_PNX_EXTENSION—contains additional information on PNX records, where required. For example, tags, reviews, and syndetics (an external knowledge base that holds extra information about the record, for example, the TOC of a book). ●P_SOURCE_RECORD—contains original source records for renormalization, if required. ●P_DEDUP_VECTOR—contains information to support the dedup process. ●P_FRBR_KEYS—contains information to support the FRBR process. ●P_FRBR_MATCHES—temporary table for use during FRBR processing. ●P_DEDUP_COMPONENT ●P_PNX_CLICK
		<ul style="list-style-type: none"> ●P_PNX_XREF ●P_RPNX

NOTE: All default data created by the Primo system is noted as updated by Primo.

P_PNX Table

The P_PNX table contains the Primo PNX records. The following table lists and describes the fields in the P_PNX table.

P_PNX Table Fields

Field Name	Description
ID	The internal identification number of the record.
RECORDID	The source system record identification number of the record.
PART_1, PART_2,PART_3, PART_4, PART_5, STRING_CLOB	The PNX xml code of the record.
CREATION_DATE	The date on which the record was created in Primo.
UPDATED_DATE	The date on which the record was updated in Primo.
UPDATED_BY	The name of the user who ran the pipe that created the record.
MATCHID	The dedup match identification number of the record.
GROUPID	The FRBR match identification number of the record.
PERSISTCONTROL	The current processing stage of the record. Possible values: 1 - NEW_STATUS 2 - UPDATED_STATUS 3 - AFTER_DEDUPED_STATUS 4 - AFTER_DEDUPED_MERGE_STATUS 5 - AFTER_FRBR_STATUS 6 - AFTER_FRBR_MERGE_STATUS 7 - DELETED_STATUS
SECONTROL	The current indexing status. Possible values: 1 - SE_CONTROL_NEW_STATUS 2 - SE_CONTROL_UPDATED_STATUS 3 - SE_CONTROL_DELETED_STATUS
PIPE_NAME	The name of the pipe from which the record was loaded.
PREVIOUS_TYPE	For dedup and dedup merge purposes.
SOURCE_ID	The name of the source from which the record was loaded.
DATA_SOURCE_CODE	The code name of the data source associated with the pipe.
BULK_ID	The name of the bulk from which the record was loaded.

TYPE	The type of the record. Possible values: 1 - REGULAR_TYPE 2 - DEDUP_MEMBER_TYPE 3 - DEDUP_SINGLE_TYPE 4 - DEDUP_MERGE_TYPE 5 - FRBR_MEMBER_TYPE 6 - FRBR_SINGLE_TYPE 7 - FRBR_MERGE_TYPE
EXTENSION_EXISTS	Indicates whether an extension exists.
PUBLISHING_WORK_ID	Pipe ID.

Database Utilities

The following table lists the database utilities that are supported in Primo.

Supported Database Utilities

Number	Utility Name	Refer to
O/1	Oracle Server	Oracle Server (Util O/1)
O/1/1	Activate Oracle Server	Activate Oracle Server (Util O/1/1)
O/1/2	Close Oracle Server	Close Oracle Server (Util O/1/2)
O/1/3	Show Running Oracle Server	Show Running Oracle Server (Util O/1/3)
O/1/4	Show Oracle Server Status	Show Oracle Server Status (Util O/1/4)
O/2	Oracle Listener	Oracle Listener (Util O/2)
O/2/1	Activate Oracle Listener	Activate Oracle Listener (Util O/2/1)
O/2/2	Close Oracle Listener	Close Oracle Listener (Util O/2/2)
O/2/3	Show Running Oracle Listener	Show Running Oracle Listener (Util O/2/3)
O/2/4	Show Listener Status	Show Listener Status (Util O/2/4)
O/2/5	Show Listener Services	Show Listener Services (Util O/2/5)
O/3	Oracle Logs	Oracle Logs (Util O/3)
O/3/1	View Oracle ALERT LOG	View Oracle ALERT LOG (Util O/3/1)
O/6	NLS	NLS (Util O/6)
O/6/1	Show NLS Parameters	Show NLS Parameters (Util O/6/1)
O/7	Archiving	Archiving (Util O/7)

O/7/1	Turning Archiving On	Turning Archiving On (Util O/7/1)
O/7/2	Turning Archiving Off	Turning Archiving Off (Util O/7/2)
O/7/3	Show Archiving Status	Show Archiving Status (Util O/7/3)
O/9	Database Users	Database Users (Util O/9)
O/9/1	List Database Users	List Database Users (Util O/9/1)
O/9/2	Create a New User	Create a New User (Util O/9/2)
O/11	Oracle user password utility	Oracle User Password Utility (Util O/11)
O/13	Database Files	Database Files (Util O/13)
O/13/1	List of Database Files	List of Database Files (Util O/13/1)
O/13/2	Resize Oracle Datafile	Resize Oracle Datafile (Util O/13/2)
O/13/3	Add File to Tablespace	Add File to Tablespace (Util O/13/3)
O/13/4	Show Datafile Free Blocks by Kbytes	Show Datafile Free Blocks by Kbytes (Util O/13/4)
O/13/5	Show Datafile Free Blocks by BlockID	Show Datafile Free Blocks by BlockID (Util O/13/5)
O/14	Database Free/Used Space	Database Free/Used Space (Util O/14)
O/14/1	All Tablespaces Free Space Summary	All Tablespaces Free Space Summary (Util O/14/1)
O/14/2	Number of Free Extents by Size in a Tablespace	Number of Free Extents by Size in a Tablespace (Util O/14/2)
O/14/3	All Free Extents of Min Size in a Tablespace	All Free Extents of Min Size in a Tablespace (Util O/14/3)
O/14/4	Space Used by a Repository/ Repositories in Each Tablespace	Space Used by a Repository/ Repositories in Each Tablespace (Util O/14/4)
O/14/5	Space Used by a Group of Repositories in Each Tablespace	Space Used by a Group of Repositories in Each Tablespace (Util O/14/5)

O/14/6	Coalesce Contiguous Free Extents	Coalesce Contiguous Free Extents (Util O/14/6)
O/14/8	Primo Tablespaces Total/Free/Used Space Report	Primo Tablespaces Total/Free/Used Space Report (Util O/14/8)
O/14/9	Clean Temporary Tablespace Free Storage	Clean Temporary Tablespace Free Storage (Util O/14/9)
O/14/10	Space Used by PRM00 Schema	Space Used by a PRM00 Schema (Util O/14/10)
O/17	Database Tablespaces	Database Tablespaces (Util O/17)
O/17/1	Create a Tablespace	Create a Tablespace (Util O/17/1)
O/17/2	List Tablespace Files	List Tablespace Files (Util O/17/2)
O/17/4	Show Tablespaces Definition	Show Tablespaces Definition (Util O/17/4)
O/17/5	Show Tablespace Allocated/Free/Used Space	Show Tablespace Allocated/Free/Used Space (Util O/17/5)
O/18	Oracle Statistics	Oracle Statistics (Util O/18)
O/18/1	Performance Statistics	Performance Statistics (Util O/18/1)
O/18/2	Rollback Segments Definitions	Rollback Segments Definitions (Util O/18/2)
O/18/3	Rollback Segments Dynamic Allocation	Rollback Segments Dynamic Allocation (Util O/18/3)
O/18/4	View Long Operations	View Long Operations (Util O/18/4)
O/18/5	I/O Statistics	IO Statistics (Util O/18/5)
O/18/6	Sort Operations	Sort Operations (Util O/18/6)
O/19	Shared Pool	Shared Pool (Util O/19)
O/19/1	Show SGA Buffers	Show SGA Buffers (Util O/19/1)
O/19/2	Flush Shared Pool	Flush Shared Pool (Util O/19/2)
O/20	Multi Threaded Server	Multi Threaded Server (Util O/20)

O/20/1	Show MTS Parameters	Show MTS Parameters (Util O/20/1)
O/20/2	Show Listener Services	Show Listener Services (Util O/20/2)

Oracle Server (Util O/1)

The Oracle Server menu allows you to manage the Oracle server.

Figure 3:

O. Managing ORACLE

- 0. Exit Procedure
 - 1. Oracle Server
 - 2. Oracle Listener
 - 3. Oracle Logs
 - 4. Resumable Space Allocation
 - 6. Nls
 - 7. Archiving
 - 9. Database Users
 - 10. SQL*Plus Session
 - 11. Oracle user password utility
 - 12. Database Verification Utility
 - 13. Database Files
 - 14. Database Free/Used Space
 - 16. Database Links
 - 17. Database Tablespaces
 - 18. Oracle Statistics
 - 19. Shared Pool
 - 20. Multi Threaded Server
 - 21. Create/Recreate reports(RPT00) schema
- Please select [exit]:

For image detail, press Ctrl++ Managing Oracle Menu

To access the Oracle Server menu:

1Log on to a Primo server (BE, FE, or SE) with the primo user.

NOTE: If you are using a dedicated Oracle server, you must log on to a server that has a Primo component (BE, FE, or SE) installed in order to access Util O.

2Enter the following commands to display the Managing Oracle menu.

```
dlib prm00
```

```
util o
```

3Enter option 1 to display the Oracle Server menu.

Figure 4:

O.1 Oracle Server

- 0. Exit Procedure

1. Activate Oracle Server
2. Close Oracle Server
3. Show Running Oracle Server
4. Show Oracle Server Status

For image detail, press Ctrl++ Oracle Server Menu

Activate Oracle Server (Util O/1/1)

In order for Primo to interact with Oracle, the Oracle server must be running. When Primo is installed on the same server as the Primo database, it does not require the Listener. However, the Listener must run on the server if a third-party product is used to connect Primo and the database or if there is a remote server connected to the database. For example, when Primo is installed on one server and the database is on a different server, the Listener must be running on the database server in order for Primo to operate correctly.

Primo and Oracle may start automatically at boot time (if this option is set during installation).

NOTES:

- ☒ This operation requires the PRIMO_DBA user name and password.
- ☒ This utility will not be available if the Oracle database and the Primo application are installed on separate servers.

To start up the Oracle server:

1 Enter option 1 from the Oracle Server menu.

The following prompt displays:

```
To continue you will need to enter PRIMO DBA username/password.  
Username/password: primo_dba/<primo_dba password>
```

2 Enter the PRIMO_DBA user name and password to activate the Oracle server.

NOTE: The PRIMO_DBA user, which is created during installation, is provided with database administration privileges that allow you to start up or shut down the database.

Close Oracle Server (Util O/1/2)

NOTE: This operation requires the PRIMO_DBA user name and password.

This utility shuts down the Oracle server immediately by activating the Oracle shutdown immediate option. All the clients connected to the server are immediately logged out.

To shut down the Oracle server:

1 Enter option 2 from the Oracle Server menu. To access this menu, see [Oracle Server \(Util O/1\)](#).

The following prompt displays:

```
Do you want to restart Oracle server after closing? yes/[no]
```

2 Enter `yes` to shut down and the restart the server automatically.

The following prompt displays:

```
To close Oracle server enter PRIMO DBA username/password.  
username/password:primo_dba/<primo_dba password>
```

NOTE: To restart the server later, see [Activate Oracle Server \(Util O/1/1\)](#).

3Enter the PRIMO_DBA user name and password to activate the Oracle server.

NOTE: The PRIMO_DBA user, which is created during installation, is provided with database administration privileges that allow you to start up or shut down the database.

Show Running Oracle Server (Util O/1/3)

This utility displays the background processes and the dispatchers and shared servers that are used by your Oracle instance (database).

NOTE: If the utility fails to generate any output, activate the Oracle server (see [Activate Oracle Server \(Util O/1/1\)](#)). To display Oracle processes, enter option 3 from the Oracle Server menu.

The running processes for your Oracle instance displays. For example:

```
oracle 22017 1 0 Jan01 ? 00:00:34 ora_pmon_prm1
oracle 22019 1 0 Jan01 ? 00:00:01 ora_osp0_prm1
oracle 22021 1 0 Jan01 ? 00:00:05 ora_mman_prm1
oracle 22024 1 0 Jan01 ? 00:36:37 ora_dbw0_prm1
oracle 22026 1 0 Jan01 ? 00:13:56 ora_lgwr_prm1
oracle 22028 1 0 Jan01 ? 00:02:23 ora_ckpt_prm1
oracle 22030 1 0 Jan01 ? 00:01:18 ora_smon_prm1
oracle 22032 1 0 Jan01 ? 00:00:01 ora_reco_prm1
oracle 22034 1 0 Jan01 ? 00:00:45 ora_cjq0_prm1
oracle 22044 1 0 Jan01 ? 00:00:48 ora_mmon_prm1
oracle 22069 1 0 Jan01 ? 00:00:41 ora_mmln_prm1
oracle 22071 1 0 Jan01 ? 00:34:52 ora_d000_prm1
oracle 22073 1 0 Jan01 ? 00:43:00 ora_d001_prm1
oracle 22075 1 0 Jan01 ? 00:44:03 ora_d002_prm1
oracle 22077 1 0 Jan01 ? 00:42:52 ora_d003_prm1
oracle 22079 1 0 Jan01 ? 00:46:16 ora_d004_prm1
oracle 22081 1 0 Jan01 ? 00:42:26 ora_d005_prm1
oracle 22099 1 0 Jan01 ? 00:43:38 ora_d006_prm1
oracle 22101 1 0 Jan01 ? 00:43:22 ora_d007_prm1
oracle 22422 1 0 Jan01 ? 00:00:00 ora_qmnc_prm1
oracle 22662 1 0 Jan01 ? 00:00:00 ora_q000_prm1
oracle 22697 1 0 Jan01 ? 00:00:05 ora_q001_prm1
oracle 25373 1 28 Jan05 ? 13:23:52 ora_s000_prm1
oracle 29099 1 31 Jan05 ? 13:23:30 ora_s001_prm1
```

NOTE: This utility is relevant only if you are running the Oracle server on the same node as the Primo server.

Show Oracle Server Status (Util O/1/4)

This utility displays the status of the Oracle server.

To display the status of the Oracle server, enter option 4 from the Oracle Server menu.

The following is an example of the output of the Util O/1/4 Show Oracle Server Status utility:

Figure 5:

```
INSTANCE_N HOST_NAME VERSION STARTUP_TI STATUS LOGINS
-----
```

```
prm1 il-primo05.corp 10.2.0.1.0 01-JAN-07 OPEN ALLOWED
.exlibrisgroup.
com
```

BANNER

```
-----
Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 - Prod
PL/SQL Release 10.2.0.1.0 - Production
CORE 10.2.0.1.0 Production
TNS for Linux: Version 10.2.0.1.0 - Production
NLSRTL Version 10.2.0.1.0 - Production
```

Oracle Listener (Util O/2)

The Oracle Listener menu allows you to manage the Oracle Listener.
Figure 6:

O.2 Oracle Listener

- ```

0. Exit Procedure
1. Activate Oracle Listener
2. Close Oracle Listener
3. Show Running Oracle Listener
4. Show Listener Status
5. Show Listener Services
```

For image detail, press Ctrl++ Oracle Listener Menu

**NOTE:** This utility will not be available if the Oracle database and the Primo application are installed on separate servers.

### To access the Oracle Listener menu:

- 1Log on to the server with the primo user.
- 2Enter the following commands to display the Managing Oracle menu.  
dlib prm00  
util o
- 3Enter option 2 to display the Oracle Listener menu.

## Activate Oracle Listener (Util O/2/1)

**NOTE:** This utility requires the Oracle software owner password.

When a user process makes a connection request using a connect string, the Oracle Listener process examines the request and connects the user process to a server process. If Oracle and Primo are installed on the same server and no third-party products are used to connect to the database and no connections are made from a remote server, Primo can work without the Listener. In any other case, both the Oracle server and the Oracle Listener must be running. They may be started automatically at boot time (this is determined during installation) and also controlled by the Primo Oracle Management utilities.

### To start up the Oracle Listener:

1Enter option 1 from the Oracle Listener menu.

The following prompt displays:

```
To continue you will need to enter Oracle's password.
Password:
```

2Enter the Oracle password to start up the Oracle Listener.

### ***Close Oracle Listener (Util O/2/2)***

**NOTE:** This utility requires the Oracle software owner password.

This utility shuts down the Oracle Listener immediately. You have an option to restart the listener automatically.

### To shut down the Oracle Listener:

1Enter option 2 from the Oracle Listener menu.

The following prompt displays:

```
Do you want to restart Oracle Listener after closing? yes/[no]
```

2Enter `yes` to shut down and then restart the Listener automatically.

The following prompt displays:

```
To restart Oracle Listener enter oracle's password.
Password:
```

**NOTE:** To start the Listener later, see [Activate Oracle Listener \(Util O/2/1\)](#).

3Enter the Oracle password to restart the Oracle Listener.

### ***Show Running Oracle Listener (Util O/2/3)***

This utility displays the active Oracle Listener. To display the active Oracle Listener, enter option 3 from the Oracle Listener menu.

The following example shows the results of this utility:

```
oracle 5127 1 0 Jan01 ? 00:08:09 /exlibris/app/oracle/product/102/bin/tnslsnr
LISTENER -inherit
```

### ***Show Listener Status (Util O/2/4)***

This utility displays the status of the Oracle Listener.

To display the status of the Oracle Listener, enter option 4 from the Oracle Listener menu.

The following example shows the results of this utility:

```
LSNRCTL for Linux: Version 10.2.0.1.0 - Production on 07-JAN-2007 14:06:58

Copyright (c) 1991, 2005, Oracle. All rights reserved.
```

```
Connecting to (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1521))
STATUS of the LISTENER
```

```

Alias LISTENER
Version TNSLSNR for Linux: Version 10.2.0.1.0 - Production
Start Date 01-JAN-2007 07:14:47
Uptime 6 days 6 hr. 52 min. 11 sec
Trace Level off
Security ON: Local OS Authentication
SNMP ON
Listener Log File /exlibris/app/oracle/product/102/network/log/listener.log
Listening Endpoints Summary...
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=il-
primo05.corp.exlibrisgroup.com)(PORT=1521)))
Services Summary...
Service "prm1.il-primo05.corp.exlibrisgroup.com" has 1 instance(s).
Instance "prm1", status READY, has 9 handler(s) for this service...
Service "prm1_XPT.il-primo05.corp.exlibrisgroup.com" has 1 instance(s).
Instance "prm1", status READY, has 9 handler(s) for this service...
The command completed successfully
```

### **Show Listener Services (Util O/2/5)**

This utility displays the Oracle Listener services. To display the Oracle Listener services, enter option 5 from the Oracle Listener menu.

The following example shows the results of this utility:

```
LSNRCTL for Linux: Version 10.2.0.1.0 - Production on 07-JAN-2007 14:07:33
```

```
Copyright (c) 1991, 2005, Oracle. All rights reserved.
```

```
Connecting to (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1521))
Services Summary...
Service "prm1.il-primo05.corp.exlibrisgroup.com" has 1 instance(s).
Instance "prm1", status READY, has 9 handler(s) for this service...
Handler(s):
"DEDICATED" established:3 refused:0 state:ready
LOCAL SERVER
"D007" established:72296 refused:0 current:5 max:972 state:ready
DISPATCHER <machine: il-primo05.corp.exlibrisgroup.com, pid: 22101>
(ADDRESS=(PROTOCOL=ipc)(KEY=#22101.1))
"D006" established:71245 refused:0 current:5 max:972 state:ready
DISPATCHER <machine: il-primo05.corp.exlibrisgroup.com, pid: 22099>
(ADDRESS=(PROTOCOL=ipc)(KEY=#22099.1))
"D005" established:71779 refused:0 current:3 max:972 state:ready
DISPATCHER <machine: il-primo05.corp.exlibrisgroup.com, pid: 22081>
(ADDRESS=(PROTOCOL=ipc)(KEY=#22081.1))
```

"D004" established:72279 refused:0 current:2 max:972 state:ready  
DISPATCHER <machine: il-primo05.corp.exlibrisgroup.com, pid: 22079>  
(ADDRESS=(PROTOCOL=ipc)(KEY=#22079.1))  
"D003" established:71468 refused:0 current:1 max:972 state:ready  
DISPATCHER <machine: il-primo05.corp.exlibrisgroup.com, pid: 22077>  
(ADDRESS=(PROTOCOL=tcp)(HOST=il-  
primo05.corp.exlibrisgroup.com)(PORT=38790))  
"D002" established:72525 refused:0 current:0 max:972 state:ready  
DISPATCHER <machine: il-primo05.corp.exlibrisgroup.com, pid: 22075>

(ADDRESS=(PROTOCOL=tcp)(HOST=il-  
primo05.corp.exlibrisgroup.com)(PORT=38789))  
"D001" established:71704 refused:0 current:4 max:972 state:ready  
DISPATCHER <machine: il-primo05.corp.exlibrisgroup.com, pid: 22073>  
(ADDRESS=(PROTOCOL=tcp)(HOST=il-  
primo05.corp.exlibrisgroup.com)(PORT=38788))  
"D000" established:60036 refused:0 current:1 max:972 state:ready  
DISPATCHER <machine: il-primo05.corp.exlibrisgroup.com, pid: 22071>  
(ADDRESS=(PROTOCOL=tcp)(HOST=il-  
primo05.corp.exlibrisgroup.com)(PORT=38787))  
Service "prm1\_XPT.il-primo05.corp.exlibrisgroup.com" has 1 instance(s).  
Instance "prm1", status READY, has 9 handler(s) for this service...  
Handler(s):  
"DEDICATED" established:3 refused:0 state:ready  
LOCAL SERVER  
"D007" established:72296 refused:0 current:5 max:972 state:ready  
DISPATCHER <machine: il-primo05.corp.exlibrisgroup.com, pid: 22101>  
(ADDRESS=(PROTOCOL=ipc)(KEY=#22101.1))  
"D006" established:71245 refused:0 current:5 max:972 state:ready  
DISPATCHER <machine: il-primo05.corp.exlibrisgroup.com, pid: 22099>  
(ADDRESS=(PROTOCOL=ipc)(KEY=#22099.1))  
"D005" established:71779 refused:0 current:3 max:972 state:ready  
DISPATCHER <machine: il-primo05.corp.exlibrisgroup.com, pid: 22081>  
(ADDRESS=(PROTOCOL=ipc)(KEY=#22081.1))  
"D004" established:72279 refused:0 current:2 max:972 state:ready  
DISPATCHER <machine: il-primo05.corp.exlibrisgroup.com, pid: 22079>  
(ADDRESS=(PROTOCOL=ipc)(KEY=#22079.1))  
"D003" established:71468 refused:0 current:1 max:972 state:ready  
DISPATCHER <machine: il-primo05.corp.exlibrisgroup.com, pid: 22077>  
(ADDRESS=(PROTOCOL=tcp)(HOST=il-  
primo05.corp.exlibrisgroup.com)(PORT=38790))  
"D002" established:72525 refused:0 current:0 max:972 state:ready  
DISPATCHER <machine: il-primo05.corp.exlibrisgroup.com, pid: 22075>  
(ADDRESS=(PROTOCOL=tcp)(HOST=il-  
primo05.corp.exlibrisgroup.com)(PORT=38789))

"D001" established:71704 refused:0 current:4 max:972 state:ready  
DISPATCHER <machine: il-primo05.corp.exlibrisgroup.com, pid: 22073>

```
(ADDRESS=(PROTOCOL=tcp)(HOST=il-
primo05.corp.exlibrisgroup.com)(PORT=38788))
"D000" established:60036 refused:0 current:1 max:972 state:ready
DISPATCHER <machine: il-primo05.corp.exlibrisgroup.com, pid: 22071>
(ADDRESS=(PROTOCOL=tcp)(HOST=il-
primo05.corp.exlibrisgroup.com)(PORT=38787))
The command completed successfully
```

---

## Oracle Logs (Util O/3)

The Oracle Logs menu allows you to view the Oracle log file.

Figure 7:

### O.3 Oracle Logs

- 
- 0. Exit Procedure
  - 1. View Oracle ALERT LOG

For image detail, press Ctrl++ Oracle Logs Menu

#### To access the Oracle Logs menu:

- 1 Log on to the server with the primo user.
- 2 Enter the following commands to display the Managing Oracle menu.  
dlib prm00  
util o
- 3 Enter option 3 to display the Oracle Logs menu.

#### **View Oracle ALERT LOG (Util O/3/1)**

This utility displays the latest entries to the Oracle alert log. By default, the last 100 lines are displayed.

#### To view the Oracle alert log:

- 1 Enter option 1 from the Oracle Logs menu.
- The following prompt displays:

```
The Database Contains the Following Files:
Enter number of lines to see from log : [100]
```

- 2 Enter the number of lines to display from the log.
- The following example displays the last 15 lines of the log:

```
Fri Jun 26 00:47:00 2009
SMON: enabling cache recovery
Fri Jun 26 00:47:03 2009
Successfully onlined Undo Tablespace 1.
Fri Jun 26 00:47:03 2009
SMON: enabling tx recovery
Fri Jun 26 00:47:04 2009
Database Characterset is UTF8
```

```

Opening with Resource Manager plan: SYSTEM_PLAN
where NUMA PG = 1, CPUs = 2
replication_dependency_tracking turned off (no async multimaster replication
found)
Starting background process QMNC
QMNC started with pid=14, OS id=15328
Fri Jun 26 00:47:12 2009
Completed: ALTER DATABASE OPEN

Enter CR to continue...

```

3Type **enter** to return to the Oracle Logs menu.

## NLS (Util O/6)

The NLS menu allows you to view the NLS (National Language Support) parameters.  
Figure 8:

```

O.6 NLS

0. Exit Procedure
1. Show NLS Parameters

```

For image detail, press Ctrl++ NLS Menu

### To access the NLS menu:

- 1Log on to the server with the primo user.
- 2Enter the following commands to display the Managing Oracle menu.  
dlib prm00  
util o
- 3Enter option 6 to display the NLS menu.

### Show NLS Parameters (Util O/6/1)

Primo uses the UTF8 character set. This utility shows the NLS (National Language Support) definition of the database.

### To view the NLS parameter settings:

- 1Enter option 1 from the NLS menu.
- For example, the utility displays the settings of the parameters:  
Figure 9:

```

PARAMETER VALUE
=====
NLS_LANGUAGE AMERICAN
NLS_TERRITORY AMERICA
NLS_CURRENCY $
NLS_ISO_CURRENCY AMERICA
NLS_NUMERIC_CHARACTERS .,
NLS_CALENDAR GREGORIAN
NLS_DATE_FORMAT DD-MON-RR
NLS_DATE_LANGUAGE AMERICAN

```

```
NLS_CHARACTERSET UTF8
NLS_SORT BINARY
NLS_TIME_FORMAT HH.MI.SSXFF AM
NLS_TIMESTAMP_FORMAT DD-MON-RR HH.MI.SSXFF AM
NLS_TIME_TZ_FORMAT HH.MI.SSXFF AM TZR
NLS_TIMESTAMP_TZ_FORMAT DD-MON-RR HH.MI.SSXFF AM TZR
NLS_DUAL_CURRENCY $
NLS_NCHAR_CHARACTERSET UTF8
NLS_COMP BINARY
NLS_LENGTH_SEMANTICS BYTE
NLS_NCHAR_CONV_EXCP FALSE
```

19 rows selected.

Enter CR to continue...

2Type **enter** to return to the NLS menu.

---

## Archiving (Util O/7)

Primo backup and recovery procedures are based on Oracle. In order to provide complete recovery of data up to the time of failure, the Oracle database mode must be archivelog mode. This ensures full recovery up to the time of failure. Hot backup cannot be performed without first enabling archivelog mode of the database.

If the database is in noarchivelog mode, only cold backups can be performed. A cold backup restores data to the time the last backup was performed. This means that all changes from the time of failure are lost.

Ex Libris highly recommends archivelog mode because in this mode, both cold and hot backups can be used to recover the database right up to the time of the failure. To ensure a complete recovery, all the archive files that were generated from the time of the backup (hot or cold) until the time of failure must be available.

Refer to the Oracle backup manual for more information on the preliminary actions that are required before using Util O/7.

### NOTES:

❗ Changing the archiving mode shuts down the database and restarts it in archivelog mode.

When running pipes (such as pipes set to No Harvesting - Update Data Source) that add or change a large amount of data, it is recommended that you stop Oracle archiving, as this slows down the process and fills up the disk. Immediately after the process is complete, perform a full cold backup and then turn archiving back on.

❗ Always perform a full cold backup immediately after switching to archivelog mode. If you do not do this, there will be a gap in the archivelog files, which will prevent a full recovery.

The Archiving menu allows you to manage archiving.

Figure 10:

### O.7 Archiving

-----

#### 0. Exit Procedure

1. Turning Archiving On
2. Turning Archiving Off
3. Show Archiving Status

For image detail, press Ctrl++ Archiving Menu

**To access the Archiving menu:**

- 1 Log on to the server with the primo user.
- 2 Enter the following commands to display the Managing Oracle menu.  
dlib prm00  
util o
- 3 Enter option 7 to display the Archiving menu.

### **Turning Archiving On (Util O/7/1)**

When you run your database in ARCHIVELOG mode, you enable the archiving of the redo log.

#### **NOTES:**

- ☒ Turning archiving on requires a Primo DBA user name and password.
- ☒ Changing the archiving mode shuts down the database and restarts it in archivelog mode.

#### **To turn on Oracle archiving mode:**

- 1 Stop Primo processes (servers and batch procedures) using the startup scripts described in [Starting and Stopping the System](#).
- 2 From the Archiving menu, enter option 1 to turn on archiving.
- 3 At the following prompt, enter the PRIMO\_DBA user name/password:

To continue you will need to enter PRIMO\_DBA username/password.  
username/password:

- 4 Restart Primo using the startup scripts described in [Starting and Stopping the System](#).

### **Turning Archiving Off (Util O/7/2)**

When you run your database in NOARCHIVELOG mode, you disable the archiving of the redo log.

#### **NOTES:**

- ☒ Requires Primo DBA user name and password.
- ☒ Changing the archiving mode shuts down the database and restarts it with archivelog mode off.

#### **To turn off Oracle archiving mode:**

- 1 Stop Primo processes (servers and batch procedures) using the startup scripts, as described in [Starting and Stopping the System](#).
- 2 From the Archiving menu, enter option 2 to turn off archiving.
- 3 At the following prompt, enter the PRIMO\_DBA user name/password:

To continue you will need to enter PRIMO\_DBA username/password.  
username/password:

- 4 Restart Primo using the startup scripts described in [Starting and Stopping the System](#).

### **Show Archiving Status (Util O/7/3)**

This utility displays the current archiving status of Oracle.

**NOTE:** You will need a Primo DBA user name and password to view the status.

To view the archiving status, enter option 3 from the Archiving menu to view the status. For example, the following output displays if archiving is off:

Figure 11:

```
SQL*Plus: Release 10.2.0.1.0 - Production on Sun Jan 7 14:09:12 2007
```

```
Copyright (c) 1982, 2005, Oracle. All rights reserved.
```

```
idle> Connected.
```

```
idle> idle> Database log mode No Archive Mode
```

```
Automatic archival Disabled
```

```
Archive destination /exlibris/oradata/prm1/arch/
```

```
Oldest online log sequence 6717
```

```
Current log sequence 6721
```

```
idle> Disconnected from Oracle Database 10g Enterprise Edition Release
10.2.0.1.0 - Production
```

```
With the Partitioning, OLAP and Data Mining Scoring Engine options
```

```
When archiving is on, the following two lines are displayed, as follows:
```

```
SQL> SQL> Database log mode Archive Mode
```

```
Automatic archival Enabled
```

For image detail, press Ctrl++ Archiving Status

In a production database, always set the Database log mode to Archive mode.

---

## Database Users (Util O/9)

The Database Users menu allows you to view and create database users.

Figure 12:

### O.9. Database Users

-----  
0. Exit Procedure

1. List Database Users

2. Create a New User

Please select [exit]:

For image detail, press Ctrl++ Database Users Menu

### To access the Database Users menu:

1Log on to the server with the primo user.

2Enter the following commands to display the Managing Oracle menu.

```
dlib prm00
```

```
util o
```

3Enter option 9 to display the Database Users menu.

### List Database Users (Util O/9/1)

This utility lists all of the users in the database. To list the database users, enter option 1 from the Database Users menu. For example:

Figure 13:

```
The Database prm1 Contains the Following Users:
```

```
=====
```

```
ANONYMOUS
CTXSYS
DBSNMP
DIP
DMSYS
EXFSYS
MDSYS
MGMT_VIEW
ORACLE_OCM
ORDPLUGINS
ORDSYS
OUTLN
P21_PRM00
P23_PRM00
PRIMO
PRIMO_ADMIN
PRIMO_BACKUP
PRIMO_DBA
SI_INFORMTN_SCHEMA
SYS
SYSMAN
SYSTEM
TSMSYS
WMSYS
XDB

Enter CR to continue...
```

For image detail, press Ctrl++ List of Database Users

**NOTE:** Some of the users are Primo collection users and others are administrative users.

**Create a New User (Util O/9/2)**

This utility creates a new user with a default password that is the same as the user name.

**NOTE:** If the name of the user already exists, all the tables and data of that user are deleted. The user is created with empty tables.

**To create a new database user:**

- 1Enter option 2 from the Database Users menu.
- 2At the following prompt, enter the new user name:

```
Enter User Name to Create New User:
```

- 3At the following prompt, enter `yes` to continue:

```
enter yes to create oracle user <new_user>
```

- 4At the following prompt, enter `no` to reconfirm:

```
default password is P21_<new_user>
if user <new_user> exists all data will be erased!!!
enter no to reconfirm
```

5At the following prompt, type **enter** to return to the Database Users menu:

```
source create_ora_user_b P21_<new_user>
create_ora_user_b P21_<new_user>

SQL*Plus: Release 10.2.0.4.0 - Production on Sat Jul 11 06:46:35 2009

Copyright (c) 1982, 2007, Oracle. All Rights Reserved.

Enter user-name:
Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.4.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> EXIT
Disconnected from Oracle Database 10g Enterprise Edition Release 10.2.0.4.0 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL*Plus: Release 10.2.0.4.0 - Production on Sat Jul 11 06:46:35 2009

Copyright (c) 1982, 2007, Oracle. All Rights Reserved.

Enter user-name:
Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.4.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> old 1: DROP USER &&1 CASCADE
new 1: DROP USER P21_<new_user> CASCADE
DROP USER P21_<new_user> CASCADE
*
ERROR at line 1:
ORA-01918: user 'P21_<new_user>' does not exist

User created.

Grant succeeded.

Disconnected from Oracle Database 10g Enterprise Edition Release 10.2.0.4.0 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

Enter CR to continue...
```

---

## Oracle User Password Utility (Util O/11)

This utility allows you to change the passwords for the following Oracle users:

- PRIMO\_ADMIN
- PRIMO\_DBA
- P<r><n>\_PRM00
- P<r><n>\_PRM00\_SHARED
- P<r><n>\_RPT00
- PRIMO\_BACKUP
- PRIMO

**NOTE:** The password for each Oracle user must be identical on all Primo servers (BE, FE, and SE).

### To change the password of an Oracle user:

1Shut down all Primo servers (BE, FE, and SE).

2From the BE server, enter the following commands to display the Managing Oracle menu:

```
dlib prm00
```

```
util o
```

3Enter option 11 to change an Oracle user's password.

4At the following prompt, enter the user name to update:

```
Change Oracle user password.
The script must be executed on all Primo servers.
The Primo software must be down before the script execution.
Enter user name (PRIMO_ADMIN,PRIMO_DBA,P21_PRM00):
```

5At the following prompt, enter the new password:

```
Password input must be identical in all Primo servers.
Please, enter new password. Password may contain letters, numbers and _
(underscore).
```

6At the following prompt, enter the PRIMO\_DBA user name/password. Otherwise, type **enter** to exit.

```
If you want to update this password in Oracle
Enter PRIMO_DBA user/password i.e. PRIMO_DBA/PRIMO_DBA , or press
[Enter] to exit :
```

**NOTE:** This step is omitted for the FE and SE servers because the password was changed previously on the BE server.

7At the following prompt, type **enter** to return to the Managing Oracle menu.

```
Change password in Oracle
Changing DB password in BE global.properties file...
Changing DB password in FE global.properties file...
Changing DB password in SE global.properties file...
Running set globals...Please wait...
Enter CR to continue...
```

8Repeat steps 2 through 7 for the FE and SE servers.

**NOTE:** When changing the password for the PRIMO\_ADMIN and PRIMO\_DBA users, the system does not apply changes to the global.properties files.

9Start all Primo servers (BE, FE, and SE).

---

## Database Files (Util O/13)

The Database Files menu contains utilities that allow you to manage database files.

Figure 14:

### O.13 Database Files

-----

#### 0. Exit Procedure

1. List of Database Files
2. Resize Oracle Datafile
3. Add File to Tablespace
4. Show Datafile Free Blocks by KBytes
5. Show Datafile Free Blocks by BlockID

Please select [exit]:

For image detail, press Ctrl++ Database Files Menu

### To access the Database Files menu:

1Log on to the server with the primo user.

2Enter the following commands to display the Managing Oracle menu.

```
dlib prm00
```

```
util o
```

3Enter option 13 to display the Database Files menu.

### List of Database Files (Util O/13/1)

This utility lists the Oracle data files and their sizes. To list the database files, enter option 1 from the Database Files menu. For example:

Figure 15:

```
The Database prm1 Contains the Following Files:
```

```
=====
```

```
T NAME SIZE K F
```

```

```

```
LOG /exlibris/oradata/prm1/prm1_log01.dbf 71680 6
SYSAUX /exlibris/oradata/prm1/prm1_sysaux01.dbf 1048576 3
SYSTEM /exlibris/oradata/prm1/prm1_system01.dbf 655360 1
TSLOB /exlibris/oradata/prm1/prm1_tslob01.dbf 512000 5
TS_P_DAT /exlibris/oradata/prm1/prm1_ts_p_dat_01.dbf 7340032 7
UNDOTBS1 /exlibris/oradata/prm1/prm1_undotbs01.dbf 2097152 2
USERS /exlibris/oradata/prm1/prm1_users01.dbf 204800 4
```

```
T NAME SIZE K F
```

```

```

```
TEMP /exlibris/oradata/prm1/prm1_temp01.dbf 4194304 1
```

Enter CR to continue...

For image detail, press Ctrl++ List of Database Files

### **Resize Oracle Datafile (Util O/13/2)**

This utility allows you to enlarge or reduce the size of an Oracle data file.

**NOTE:** Requires the PRIMO\_DBA user name and password.

#### **To change the size of an Oracle data file:**

1Enter option 2 from the Database Files menu.

2At the following prompt, enter the PRIMO\_DBA user name/password:

To resize a database file enter PRIMO\_DBA username/password.  
username/password:

3At the following prompt, enter the name of the tablespace:

Enter Tablespace name:

4At the following prompt, enter the file name (including the full path) that you want to resize:

Enter file name to resize:

5At the following prompt, enter the new file size:

Enter new file size (MB):

6At the following prompt, enter `y` to continue:

confirm (y/[n]):

7At the following prompt, type **enter** to continue:

```
resizing

SQL*Plus: Release 10.2.0.4.0 - Production on Sat Jul 11 00:19:27 2009

Copyright (c) 1982, 2007, Oracle. All Rights Reserved.

idle> Connected.
idle> idle>
Database altered.

idle> Disconnected from Oracle Database 10g Enterprise Edition Release
10.2.0.4.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
Enter CR to continue...
```

## Add File to Tablespace (Util O/13/3)

Tablespaces are composed of one or more data files. This utility allows you to add files to tablespaces. If a tablespace does not have enough free space for these files, it needs to be expanded. For information on expanding tablespaces, see [Database Tablespaces \(Util O/17\)](#).

**NOTE:** This utility requires the PRIMO\_DBA user name and password.

### To add a data file to a tablespace:

1Enter option 3 from the Database Files menu.

2At the following prompt, enter the PRIMO\_DBA user name/password:

```
To add a file to a tablespace enter PRIMO_DBA username/password.
username/password:
```

3At the following prompt, enter the name of the tablespace:

```
Enter Tablespace name:
```

4At the following prompt, enter the new file name (including the full path):

```
Enter new file name:
```

5At the following prompt, enter the file size:

```
Enter file size (MB):
```

6At the following prompt, enter `y` to continue:

```
confirm (y/[n]):
```

7At the following prompt, type **enter** to continue:

```
SQL*Plus: Release 10.2.0.4.0 - Production on Sat Jul 11 00:44:50 2009

Copyright (c) 1982, 2007, Oracle. All Rights Reserved.

idle> Connected.
idle> idle>
Tablespace altered.

idle> Disconnected from Oracle Database 10g Enterprise Edition Release
10.2.0.4.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

Enter CR to continue...
```

### **Show Datafile Free Blocks by Kbytes (Util O/13/4)**

This utility displays the free data file blocks. To display the free data file blocks, enter option 4 to display from the Database Files menu. For example:

Figure 16:

```
TABLES F BLOCK_ID KBYTES NAME

TS_P_D 7 162169 128 /exlibris/oradata/prm1/prm1_ts_p_da
AT t_01.dbf

SYSTEM 1 55681 64 /exlibris/oradata/prm1/prm1_system0
1.dbf

SYSAUX 3 24977 64 /exlibris/oradata/prm1/prm1_sysaux0
1.dbf

SYSAUX 3 24993 64 /exlibris/oradata/prm1/prm1_sysaux0
1.dbf

TABLES F BLOCK_ID KBYTES NAME

SYSAUX 3 25185 64 /exlibris/oradata/prm1/prm1_sysaux0
1.dbf

SYSAUX 3 32889 64 /exlibris/oradata/prm1/prm1_sysaux0
1.dbf

UNDOTB 2 81 64 /exlibris/oradata/prm1/prm1_undotbs
S1 01.dbf

Enter CR to continue...
```

For image detail, press Ctrl++ List Free Blocks of Data

### **Show Datafile Free Blocks by BlockID (Util O/13/5)**

This utility displays the free data file blocks per tablespace.

#### **To display the free data file blocks:**

1Enter option 5 from the Database Files menu.

2At the following prompt, enter the name of the tablespace:

Tablespace Name:

3At the following prompt, specify the datafile number of the tablespace:

Datafile Number:

**NOTE:** To determine the datafile number, see [List of Database Files \(Util O/13/1\)](#).

4At the following prompt, type **enter** to return to the Database Files menu:

Figure 17:

```
Enter value for ts: old 4: where TABLESPACE_NAME = '&&ts'
new 4: where TABLESPACE_NAME = 'TS_P_DAT'
Enter value for fl_no: old 5: and FILE_ID = &&fl_no
new 5: and FILE_ID = 9

BLOCK_ID BYTES

33149737 93782016
33068329 665845760
33013801 445644800
32946601 550502400
32847913 807403520
32786729 500170752
32761001 208666624
32760361 3145728
32758697 2097152
32758441 1048576
32758185 1048576

Enter CR to continue...
```

For image detail, press Ctrl++ List of Free Data Blocks by Block ID

---

## Database Free/Used Space (Util O/14)

The Database Free/Used Space menu allows you to display information about the free and used space in the tablespace.

Figure 18:

### O.14. Database Free/Used Space

- ```
-----
0. Exit Procedure
1. All Tablespaces Free Space Summary
2. Number of Free Extents by Size in a Tablespace
3. All Free Extents of Min Size in a Tablespace
4. Space Used by a Repository/Repositories in Each Tablespace
5. Space Used by a Group of Repositories in Each Tablespace
6. Coalesce Contiguous Free Extents
7.
8. Primo Tablespaces Total/Free/Used Space Report
9. Clean Temporary Tablespace Free Storage
```

For image detail, press Ctrl++ Database Free/Used Space Menu

To access the Database Free/Used Space menu:

- 1 Log on to the server with the primo user.
- 2 Enter the following commands to display the Managing Oracle menu.
dlib prm00
util o
- 3 Enter option 14 to display the Database Free/Used Space menu.

All Tablespaces Free Space Summary (Util O/14/1)

This utility provides details about free space in the Oracle DBA_FREE_SPACE table. It generates a report with the following four columns:

- **TABLESPACE_NAME**: The name of the tablespace.
 - **TOTAL_FREE_SPACE**: The total amount of free space in the tablespace (in megabytes).
 - **MAX_EXTENT**: The size of the largest contiguous extent of the tablespace (in megabytes).
 - **NUM_FREE_EXTENTS**: The number of free extents in the tablespace.
- If a tablespace has no free space left, it will not appear in the report.

NOTE: It is important to review this report from time to time to prepare additional resources for the database. To display a summary of the free space for tablespaces, enter option 1 from the Database Free/Used Space menu. For example:

Figure 19:

```

TABLESPACE_NAME TOTAL_FREE_SPACE MAX_EXTENT
NUM_FREE_EXTENTS
-----
TS0 9.94296183 9.94296183 1
TS_P_IDX 1024.50027 1024.50027 1
UNDOTBS1 2861.94711 729.338137 33
SYSAUX 69.9759389 21.6994198 17
TSLOB 71.9770382 71.9770382 1
USERS 200.031756 200.031756 1
LOG 63.9726412 63.9726412 1
SYSTEM 159.96287 159.96287 1
TS1 9.94296183 9.94296183 1
TS_P_DAT 229289.89 3970.18089 438

Temporary Tablespace Space Usage

TABLESPACE_NAME SIZE M
-----
TEMP 0

Sort Segments Usage (in Temporary Tablespace)

TABLESPACE_NAME Total M Used M Free M
-----
TEMP 16381 0 16381

Enter CR to continue...

```

For image detail, press Ctrl++ Display of Free Space

Number of Free Extents by Size in a Tablespace (Util O/14/2)

This utility lists the number of extents of a certain size (truncated in megabytes) in the tablespace.

To display the free extents for a tablespace, enter

1Enter option 2 from the Database Free/Used Space menu

2At the following prompt, enter the tablespace name:

Enter Tablespace name:

3At the following prompt, type **enter** to return to the Database Free/Used Space menu:

Figure 20:

```
SIZE IN MB NUM OF EXTENTS
```

```
-----  
3970 28  
3944 1  
3737 1  
3496 1  
3445 1  
3388 1  
3257 1  
3170 1  
3098 1  
3084 1  
3063 1
```

```
Enter CR to continue...
```

For image detail, press Ctrl++ List of Free Extents

All Free Extents of Min Size in a Tablespace (Util O/14/3)

This utility lists the exact size (in megabytes) of all free extents that are larger than a given size. You are prompted for the tablespace name and the minimum size (in megabytes) that you wish to investigate.

To display the free extents for a tablespace:

4Enter option 3 from the Database Free/Used Space menu.

5At the following prompt, enter the tablespace name:

Enter Tablespace name:

6At the following prompt, enter the minimum size of the extents to display. Enter to 0 to display all of the extents.

Enter Min size (MB) of free extent [0=ALL]:

7At the following prompt, type **enter** to return to the Database Free/Used Space menu:

```
EXTENT_SIZE
-----
3970.18089
3970.18089
3970.18089
3970.18089
3970.18089
3970.18089

Enter CR to continue...
```

**Space Used by a Repository/Repositories in Each Tablespace
(Util O/14/4)**

This utility lists the amount of space that each repository occupies in each tablespace. If a truncated name is used, all of the repositories starting with the given characters are listed, and the occupied space is listed for each one of them.

To display the size of each repository per tablespace:

- 1Enter option 4 from the Database Free/Used Space menu.
- 2At the following prompt, enter the unit name (such as prm or prm00):

```
Enter unit name (full or truncated, e.g. primo):
```

- 3At the following prompt, type **enter** to return to the Database Free/Used Space menu:

```
OWNER TABLESPACE_NAME SIZE_MB
-----
P21_PRM00 TS_P_DAT 8008.27408

Enter CR to continue...
```

**Space Used by a Group of Repositories in Each Tablespace
(Util O/14/5)**

This utility lists the total amount of space that all of the repositories whose names start with the given characters occupy in each tablespace.

To display the total amount of space for all of the repositories per tablespace:

- 1Enter option 5 from the Database Free/Used Space menu.
- 2At the following prompt, enter the unit name (such as prm):

```
Enter first 3 characters of unit code (e.g. primo):
```

- 3At the following prompt, type **enter** to return to the Database Free/Used Space menu:

```
TABLESPACE_NAME SIZE_MB
-----
TS_P_DAT 8008.27408
```

Enter CR to continue...

Coalesce Contiguous Free Extents (Util O/14/6)

This utility is not required when using a locally managed tablespace. It is retained for backward compatibility (and will be removed in future versions).

NOTE: This utility requires the PRIMO_DBA user name and password.

Free space in a database may be composed of extents of various sizes. It is worthwhile to use this procedure to coalesce the contiguous free extents to create larger free extents. Perform this procedure periodically.

To coalesce free extents:

1Enter option 6 from the Database Free/Used Space menu.

2At the following prompt, enter the PRIMO_DBA user name and password:

To Coalesce Tablespaces enter PRIMO_DBA username/password.
username/password:

3At the following prompt, type **enter** to return to the Database Free/Used Space menu:

```
SQL*Plus: Release 10.2.0.4.0 - Production on Sat Jul 11 20:37:29 2009
```

```
Copyright (c) 1982, 2007, Oracle. All Rights Reserved.
```

```
Connected to:
```

```
Oracle Database 10g Enterprise Edition Release 10.2.0.4.0 - 64bit Production  
With the Partitioning, OLAP, Data Mining and Real Application Testing options
```

```
SQL> alter tablespace LOG coalesce;  
alter tablespace SYSAUX coalesce;  
alter tablespace SYSTEM coalesce;  
alter tablespace TS0 coalesce;  
alter tablespace TS1 coalesce;  
alter tablespace TSLOB coalesce;  
alter tablespace TS_P_DAT coalesce;  
alter tablespace TS_P_IDX coalesce;  
alter tablespace USERS coalesce;
```

```
Disconnected from Oracle Database 10g Enterprise Edition Release 10.2.0.4.0 -  
64bit Production  
With the Partitioning, OLAP, Data Mining and Real Application Testing options
```

```
SQL*Plus: Release 10.2.0.4.0 - Production on Sat Jul 11 20:37:29 2009
```

```
Copyright (c) 1982, 2007, Oracle. All Rights Reserved.
```

```
SQL> Connected.
```

```
SQL> SQL>
```

```
'ALERTTABLESPACE'||TABLESPACE_NAME||'COALESCE;'
```

```
-----  
alter tablespace LOG coalesce;  
alter tablespace SYSAUX coalesce;  
alter tablespace SYSTEM coalesce;  
alter tablespace TS0 coalesce;  
alter tablespace TS1 coalesce;  
alter tablespace TSLOB coalesce;  
alter tablespace TS_P_DAT coalesce;  
alter tablespace TS_P_IDX coalesce;  
alter tablespace USERS coalesce;
```

9 rows selected.

Tablespace altered.

Tablespace altered.

```
SQL> Disconnected from Oracle Database 10g Enterprise Edition Release  
10.2.0.4.0 - 64bit Production  
With the Partitioning, OLAP, Data Mining and Real Application Testing options  
Enter CR to continue...
```

Primo Tablespaces Total/Free/Used Space Report (Util 0/14/8)

This utility lists the total space that each repository occupies in each tablespace and the amount of free space in the table. To display the total space each repository occupies, enter option 8 from the Database Free/Used Space menu.

For example:

Figure 21:

```
NAME TOTAL SIZE M  
-----
```

```
TS_P_IDX 1024  
TS0 10  
TSLOB 72  
TS_P_DAT 259072  
TS1 10
```

```
NAME TOTAL FREE M  
-----
```

```
TS0 9  
TS_P_IDX 1023  
TSLOB 71  
TS1 9  
TS_P_DAT 229163
```

```
NAME TOTAL USED M
```

```
-----  
TS_P_DAT 29907
```

```
Enter CR to continue...
```

For image detail, press Ctrl++ Tablespace Report

Clean Temporary Tablespace Free Storage (Util O/14/9)

In some cases, the temporary tablespace does not free non-used pages quickly enough. This utility manually frees these pages.

To clean temporary tablespace:

1Enter option 9 from the Database Free/Used Space menu.

2At the following prompt, enter the name of the temporary tablespace:

```
To Free Temporary Segments Enter Temporary Tablespace:
```

3At the following prompt, enter the PRIMO_DBA user name and password:

```
Enter PRIMO_DBA username/password:
```

4At the following prompt, type **enter** to return to the Database Free/Used Space menu:

```
QL*Plus: Release 10.2.0.4.0 - Production on Sat Jul 11 12:52:41 2009
```

```
Copyright (c) 1982, 2007, Oracle. All Rights Reserved.
```

```
idle> Connected.
```

```
idle>
```

```
Tablespace altered.
```

```
idle> idle> Disconnected from Oracle Database 10g Enterprise Edition Release  
10.2.0.4.0 - 64bit Production  
With the Partitioning, OLAP, Data Mining and Real Application Testing options  
Enter CR to continue...
```

Space Used by a PRM00 Schema (Util O/14/10)

This utility displays the space used by the PRM00 schema. To display the space used by the PRM00 schema, enter option 10 from the Database Free/Used Space menu. For example:

Figure 22:

```
NAME TOTAL SIZE M  
-----
```

```
P21_PRM00 8003
```

Enter CR to continue...

Database Tablespaces (Util O/17)

The Database Tablespace menu allows you to manage tablespaces.

Figure 23:

O.17. Database Tablespaces

-
- 0. Exit Procedure
 - 1. Create a Tablespace
 - 2. List Tablespace Files
 - 4. Show Tablespace Definitions
 - 5. Show Tablespace Allocated/Free/Used Space
- Please select [exit]:

For image detail, press Ctrl++ Database Tablespace Menu

To access the Database Tablespaces menu:

- 1Log on to the server with the primo user.
- 2Enter the following commands to display the Managing Oracle menu.
dlib prm00
util o
- 3Enter option 17 to display the Database Tablespaces menu.

Create a Tablespace (Util O/17/1)

Typically, all the required tablespaces are created during system installation. This utility allows you to create additional tablespaces.

To create a tablespace:

- 1Enter option 1 from the Database Tablespaces menu.
- 2At the following prompt, enter the PRIMO_DBA user name and password:

To Create a new Tablespace, Enter PRIMO_DBA username/password.
username/password:

- 3At the following prompt, enter the name of the new tablespace:

Enter Tablespace name:

- 4At the following prompt, enter the new file name (include full path):

Enter new file name (full path) :

- 5At the following prompt, enter the size of the file:

Enter new file size (MB):

- 6At the following prompt, enter the allocation type (AUTO or UNIFORM):

```
=====
Tablespaces can be created with a UNIFORM size for all extents
or with allocation type AUTOALLOCATE which means
Oracle will decide how to define extents
Util o 17 4 can be used to see current definitions
for existing tablespaces
=====
```

```
Tablespace Allocation Type : [AUTO/UNIFORM]
```

7If you have selected UNIFORM, enter the uniform size of each extent:

```
UNIFORM SIZE : [128K/1M/4M/128M/1920M]
```

8At the following prompt, enter *y* to confirm:

```
Tablespace: TEST1
File: /exlibris/oradata/prm0/test1_01.dbf
File size: 100MB
Allocation : UNIFORM SIZE 4M
confirm (y/[n]):
```

9At the following prompt, enter *y* to continue:

```
CREATE TABLESPACE TEST1
DATAFILE '/exlibris/oradata/prm0/test1_01.dbf' SIZE 100M
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 4M
SEGMENT SPACE MANAGEMENT AUTO
ONLINE;
Enter CR to continue...
```

10At the following prompt, enter *y* to continue:

```
SQL*Plus: Release 10.2.0.4.0 - Production on Sat Jul 11 15:13:43 2009

Copyright (c) 1982, 2007, Oracle. All Rights Reserved.

idle> Connected.
idle> idle> 2 3 4 5
Tablespace created.

idle> idle> Disconnected from Oracle Database 10g Enterprise Edition Release
10.2.0.4.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
Enter CR to continue...
```

11At the following prompt, type **enter** to return to the Database Tablespaces menu:

```
SQL*Plus: Release 10.2.0.4.0 - Production on Sat Jul 11 15:13:43 2009

Copyright (c) 1982, 2007, Oracle. All Rights Reserved.
```

```

idle> Connected.
idle> idle> 2 3 4 5
Tablespace created.

idle> idle> Disconnected from Oracle Database 10g Enterprise Edition Release
10.2.0.4.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
Enter CR to continue...

```

List Tablespace Files (Util O/17/2)

This utility lists the tablespaces in the database.

To list a tablespace's files:

- 1Enter option 2 from the Database Tablespaces menu.
- 2At the following prompt, enter the name of the tablespace:

```

Enter Tablespace name:

```

- 3At the following prompt, type **enter** to return to the Database Tablespaces menu:

```

Tablespace TS_P_DAT consist of the following files:

NAME SIZE K F
-----
/exlibris/oradata/prm1/prm1_ts_p_dat_01.dbf 7340032 7
Enter CR to continue...

```

Show Tablespaces Definition (Util O/17/4)

You may need to increase the space that certain database tables use. All tablespaces in Primo are created as locally-managed tablespaces. When working with locally-managed tablespaces, the word `local` appears in the EXT-MGMT (extent management) column.

When using locally-managed tablespaces, the following types of extent allocation appear in the ALLOC_TYP column:

- SYSTEM (Auto Allocate)—In auto allocation, Oracle automatically assumes full control, allocating extents as needed and taking into account the initial allocation of the table/index as configured in the create table/index command. For example, if initial allocation of the table/index is 1GB, Oracle might split the 1GB into 50 extents, a single extent, or any other combination of extents.
- UNIFORM—In uniform allocation, the DBA determines a standard size for all the extents in the tablespace. All the extents in the tablespace are of that size, without regard to the extent definitions in the Create Table command. The DBA determines which table is assigned to which tablespace depending on the table size. Hence, in uniform allocation, there is no fragmentation, and space utilization is optimal. For example, when creating a tablespace with a uniform extent size of 10 MB, five extents will be used for a table of 50 MB.

To display the tablespace definitions, enter option 4 from the Database Tablespaces menu.

Figure 24:

```

TS_NAME EXT_MGMT ALLOC_TYP INIT_EXT NEXT_EXT TYPE STAT
-----

```

```
LOG LOCAL SYSTEM 65536 PERM ONL
SYSAUX LOCAL SYSTEM 65536 PERM ONL
SYSTEM LOCAL SYSTEM 65536 PERM ONL
TEMP LOCAL UNIFORM 1048576 1048576 TEMP ONL
TEST1 LOCAL UNIFORM 4194304 4194304 PERM ONL
TSLOB LOCAL SYSTEM 65536 PERM ONL
TS_P_DAT LOCAL SYSTEM 65536 PERM ONL
UNDOTBS1 LOCAL SYSTEM 65536 UNDO ONL
USERS LOCAL UNIFORM 40960 40960 PERM ONL
Enter CR to continue...
```

For image detail, press Ctrl++ Tablespace Definitions Example

In addition to the extent management types, this utility display the following information for each tablespace:

- Segment allocation
- Tablespace (for permanent or temporary objects or for undo segments)
- Tablespace status (online or offline)

Show Tablespace Allocated/Free/Used Space (Util O/17/5)

This utility displays the following parameters for a tablespace:

- Total tablespace size
- Amount of free space
- Amount of used space

To display allocated, free, and used space for a tablespace:

1Enter option 5 from the Database Tablespaces menu.

2At the following prompt, enter the name of the tablespace:

```
Enter Tablespace name :
```

3At the following prompt, type **enter** to return to the Database Tablespaces menu:

```
Tablespace TS_P_DAT :
```

```
-----
TOTAL SIZE M
```

```
7168
```

```
-----
TOTAL FREE M
```

```
5597
```

```
-----
TOTAL USED M
```

```
1570
```

```
Enter CR to continue...
```

Oracle Statistics (Util O/18)

The Oracle Statistics menu allows you to display Oracle statistics.
Figure 25:

```
O.18. Oracle Statistics
-----
0. Exit Procedure
1. Performance Statistics
2. Rollback Segments Definitions
3. Rollback Segments Dynamic Allocation
4. View Long Operations
5. IO Statistics
6. Sort Operations
Please select [exit]:
```

For image detail, press Ctrl++ Oracle Statistics Menu

To access the Oracle Statistics menu:

- 1 Log on to the server with the primo user.
- 2 Enter the following commands to display the Managing Oracle menu.
dlib prm00
util o
- 3 Enter option 18 to display the Oracle Statistics menu.

Performance Statistics (Util O/18/1)

This utility provides Oracle performance statistics for fine-tuning your database. To display the performance statistics, enter option 1 from the Oracle Statistics menu.

Figure 26:

```
SYSTEM GLOBAL AREA (sga)
BYTES
734003200
=====
BUFFER CACHE HIT RATIO (db_block_buffers)
GETS MISSES RATIO
4054336 225243 94.44%
=====
STATISTIC (db_block, DBWR, sort_area)
NAME VALUE
opened cursors current 32
db block gets 937066
consistent gets 3117272
physical reads 225243
physical writes 92082

Enter CR to continue...
```

Rollback Segments Definitions (Util O/18/2)

This utility displays the rollback segments definitions.

To display the rollback segments definitions:

1 Enter option 2 from the Oracle Statistics menu.

2 At the following prompt, type **enter** to continue:

```
SYSTEM GLOBAL AREA (sga)
BYTES
734003200
=====
BUFFER CACHE HIT RATIO (db_block_buffers)
GETS MISSES RATIO
4054336 225243 94.44%
=====
STATISTIC (db_block, DBWR, sort_area)
NAME VALUE
opened cursors current 32
db block gets 937066
consistent gets 3117272
physical reads 225243
physical writes 92082
DBWR checkpoints 37
redo log space requests 0
sorts (memory) 105305
sorts (disk) 1

Enter CR to continue...
```

3 At the following prompt, type **enter** to return to the Oracle Statistics menu:

```
=====
DATA DICTIONARY CACHE (shared_pool_size)
GETS MISSES RATIO
1442213 22892 98.41%
=====
LIBRARY CACHE (shared_pool_size)
EXECUTIONS MISSES LIBCACHEPROZ
552613 1160 99.79%

Enter CR to continue...
```

Rollback Segments Dynamic Allocation (Util O/18/3)

This utility displays the rollback segments dynamic allocation. To display the rollback segments dynamic allocation, enter option 3 from the Oracle Statistics menu. For example:

Figure 27:

```
NAME EXT RSSIZE WRITES SHRN AVGSHR WRAPS CUREXT WAITS
```

```
SYST 6 385024 5160 0 0 0 0 0
_SYS 69 121757696 4661252 2 2097152 4 2 0
_SYS 72 146923520 4952028 2 2097152 2 68 0
_SYS 68 142729216 11195842 3 2796202 6 0 1
_SYS 72 168943616 7438446 3 2446677 1 69 0
_SYS 64 101834752 2825568 3 2446677 2 2 0
_SYS 73 162652160 6144544 3 1398101 0 70 0
_SYS 72 154263552 11230104 3 1747626 3 3 0
_SYS 64 101834752 3382754 2 3145728 4 2 0
_SYS 64 175235072 6435860 3 3495253 5 2 1
_SYS 70 152166400 5199072 3 2097152 2 2 0
```

Enter CR to continue...

View Long Operations (Util O/18/4)

This utility displays Oracle long operations, if they occur in the system at the time the utility is run. The following information displays:

- SID: session identifier.
- OPNAME: operation name.
- TARGET: the object on which the operation is being performed.
- DONE SO FAR: percentage of work already done.

To display the Oracle long operations:

1Enter the following commands to display the Managing Oracle menu:

```
dlib prm00
```

```
util o
```

2Enter option 18 to display the Oracle Statistics menu.

3Enter option 4.

4At the following prompt, type **cntl-c** to exit the display:

```
Long Operation Currently running :
To stop do Ctrl C
```

IO Statistics (Util O/18/5)

This utility displays the following information:

- BLOCK_GETS—block gets for this session.
- CONSISTENT_GETS—consistent gets for this session.
- PHYSICAL_READS—physical reads for this session.
- BLOCK_CHANGES—block changes for this session.
- CONSISTENT_CHANGES—consistent changes for this session.

To display I/O statistics, enter option 5 from the Oracle Statistics menu. For example:

Figure 28:

```
TO STOP DO ctrl C
```

```
BLOCK_GETS CONSISTENT_GETS PHYSICAL_READS BLOCK_CHANGES
CONSISTENT_CHANGES
34881 418724 4141 34949 2
34881 418727 4141 34949 2
.
.
.
```

Type **ctrl-c** to exit the display.

Sort Operations (Util O/18/6)

This utility displays sort operations if they occur in the system when the utility is running. To display sort operations, enter option 6 from the Oracle Statistics menu. For example:

Figure 29:

```
Sort Operations in the system:
.
.
.
```

Type **ctrl-c** to exit the display

Shared Pool (Util O/19)

The Shared Pool menu allows you to display information associated with shared pools.

Figure 30:

```
O.19. Shared Pool
-----
0. Exit Procedure
1. Show SGA Buffers
2. Flush Shared Pool
Please select [exit]:
```

For image detail, press Ctrl++ Shared Pool Menu

To access the Shared Pool menu:

1Log on to the server with the primo user.

2Enter the following commands to display the Managing Oracle menu.

```
dlib prm00
```

```
util o
```

3Enter option 19 to display the Oracle Statistics menu.

Show SGA Buffers (Util O/19/1)

This utility lists the various SGA buffers. To display SGA buffers, enter option 1 from the Shared Pool menu. For example:

Figure 31:

```
.
```

.
.
NAME BYTES

fixed_sga 2043456
buffer_cache 461373440
log_buffer 6340608
ENQUEUE STATS 11928
VIRTUAL CIRCUITS 3736
transaction 417296
table definiti 896
KGSKI scheduler heap 2 de 232
KTCN: Obj Invalidation Se 2336
kgl lock hash table state 9800
ksunfy: nodes of hierarch 320

NAME BYTES

ASM file 19200
qmn tasks 4128
kwqmncal: allocate buffer 4048
kspd run-time context 16
kzekm heap descriptor 304
incr ckpt write count arr 168
kglsim main lru size 151040
FileOpenBlock 1447104
Core dump directory 520
log_checkpoint_timeout 12360
PX subheap 130616

NAME BYTES

partitioning d 352520
message pool freequeue 954768
sched job queue 3616
LGWR-network Server info 27648
Parameter Handle 1656
PARAMETER TABLE 2048
state objects 5680
pso tbs: ksunfy 78000
recov_kgqbtctx 4392
Cursor Stats 1290728
enqueue 355224

Enter CR to continue...

Flush Shared Pool (Util O/19/2)

This utility removes all of the Oracle objects from the shared pool.

To remove the Oracle objects from the shared pool:

1Enter option 2 from the Shared Pool menu.

2At the following prompt, enter the PRIMO_DBA user name and password:

To continue you will need to enter PRIMO_DBA username/password.
username/password:

Multi Threaded Server (Util O/20)

In a standard Oracle configuration, a separate server process is created on behalf of each user process. This is called a Dedicated Server Process (or Shadow process) because it acts only on behalf of the associated user process.

Oracle also supports the Shared Server Architecture (or Multi Threaded Server Architecture - MTS) in which there are several server processes, each serving several user processes.

In Primo, The MTS infrastructure exists in the database but is only implemented in special cases, and in coordination with Ex Libris. This group of utilities supports MTS.

The following menu allows you to manage these servers.

Figure 32:

O.20. Multi Threaded Server

0. Exit Procedure

1. Show MTS Parameters

2. Show Listener Services

Please select [exit]:

For image detail, press Ctrl++ Multi Threaded Server Menu

To access the Multi Threaded Server menu:

1Log on to the server with the primo user.

2Enter the following commands to display the Managing Oracle menu.

dlib prm00

util o

3Enter option 20 to display the Multi Threaded Server menu.

Show MTS Parameters (Util O/20/1)

This utility lists the MTS parameters. To display the MTS parameters, enter option 1 from Multi Threaded Server menu.

For example:

Figure 33:

PRIMO_DBA/PRIMO_DBA

SQL*Plus: Release 10.2.0.4.0 - Production on Sat Jul 11 16:45:09 2009

Copyright (c) 1982, 2007, Oracle. All Rights Reserved.

idle> Connected.

idle> idle> idle> Disconnected from Oracle Database 10g Enterprise Edition
Release 10.2.0.4.0 - 64bit Production

With the Partitioning, OLAP, Data Mining and Real Application Testing options
Enter CR to continue...

For image detail, press Ctrl++ Show MTS Parameters

Show Listener Services (Util O/20/2)

This utility lists the MTS dispatchers per instance. To display the MTS dispatchers, enter option 2 from the Multi Threaded Server menu. For example:

Figure 34:

```
MTS Dispatchers by Instance
=====

Service "ENC" has 1 instance(s).
Instance "ENC", status UNKNOWN, has 1 handler(s) for this service...
Handler(s):
Service "prm1" has 1 instance(s).
Instance "prm1", status UNKNOWN, has 1 handler(s) for this service...
Handler(s):
Service "prm1.rattlesnake.corp.exlibrisgroup.com" has 1 instance(s).
Instance "prm1", status READY, has 1 handler(s) for this service...
Handler(s):
Service "prm1_XPT.rattlesnake.corp.exlibrisgroup.com" has 1 instance(s).
Instance "prm1", status READY, has 1 handler(s) for this service...
Handler(s):
Enter CR to continue...
```

For image detail, press Ctrl++ Show Liistener Services

Guidelines for Fault-Tolerant Configurations

This section provides the guidelines needed to define fault-tolerant Primo configurations.

This section includes:

- [Multiple BO Configuration](#)
- [Multiple Front End Configurations](#)
- [Multiple PDS Configuration](#)
- [Load Balancer Requirements](#)

Multiple BO Configuration

A multiple BO configuration in Primo allows clients to configure one or more failover servers for the Primo BO. In this configuration, only one server is active, while the others are inactive.

NOTE: If a failure occurs in the active BO server, an inactive server must be started manually to provide BO service.

Guidelines

When configuring a multiple BO configuration, the following guidelines must be followed:

- Only one BO server can be active at a time; all others must be inactive.
- To avoid having to change a number of Back Office parameters manually, all BO servers must be directed to a single database using a load balancer. Alternatively, if a load balancer is not used, access the Installation subsystem of the General Configuration Wizard in the Primo Back Office and change the following parameters to use the host name of the new Back Office:
 - pds_url, pds_configuration_url, Reports URL, Help Base URL, reporting_base, primo_admin_base, primo_base, and Search Statistics Report URL.
- The BE_HOST parameter must be defined as a common host name on all BO servers.
- The BO servers must share the same ng/primo/home/profile/publish/publish/production directory using NFS.
- Add an entry to the be_stop script to remove the startup control file from the ng/primo/home/profile/publish/publish/production directory.
- To ensure that only one BO server is active and running, add the following checks to the be_start script:
 - Check to see that the other BO servers are accessible and run the be_stop script for BO servers.
 - Check to see that there are no connections to the other BO server's BO port.
 - Check that the start up control file does not exist in the shared ng/primo/home/profile/publish/publish/production directory.
- Add an entry to the be_start script to create a startup control file in the ng/primo/home/profile/publish/publish/production directory for the active BO server.

Multiple Front End Configurations

To improve workload balance, it is possible to define multiple Front End servers (FE). At this time, the following Multiple Front End (MFE) configurations are possible:

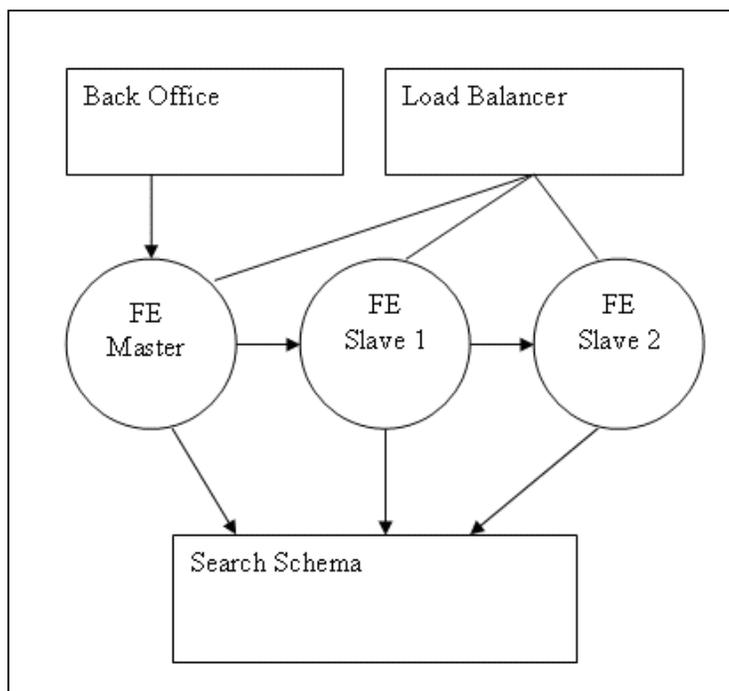
- Multiple FEs and a single Search Engine server (Many-to-One configuration)
- Multiple FEs and multiple Search Engine servers (Many-to-Many configuration)

NOTE: To benefit from these configurations, a load balancer must be integrated within your Primo installation. For more information, see [Load Balancer Requirements](#).

Many-to-One Configuration

In this configuration (as shown in the following figure), every FE has the same SE configuration. One of the FEs is defined as the "master," which receives commands from the Back Office (BO) and performs agent slicing monitoring. The other FEs are defined as "slaves," which receive commands that are forwarded from the master.

Figure 35:

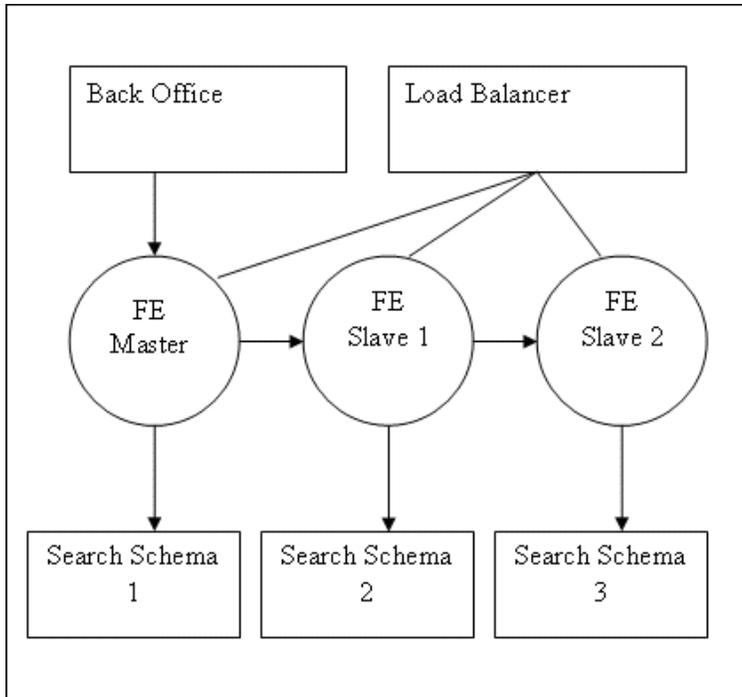


For image detail, press Ctrl++ Many-to-One Configuration

Many-to-Many Configuration

In addition to workload balancing across FEs, this configuration (as shown in the following figure) distributes the load of numerous searches over multiple Search Engines, preventing downtime if a Search Engine fails.

Figure 36:



For image detail, press Ctrl++ Many-to-Many Configuration

NOTES:

In a Many-to-Many configuration, the parent index directories of the master and slave FEs should be different. The following example shows a correct directory structure in which the parent directories (as shown in bold) are different:

Master FE:

/exlibris/primo/indexes/master_index/inst_1_index

Slave FE:

/exlibris/primo/indexes/slave_index/inst_1_index

Failover of an FE

When the BO is started, it checks to see if the master FE is running. If it is initially down, the following message may display:

The FrontEnd seems to be down. Continuing to check every 30 seconds

It will check the master every 30 seconds until it is up. Once the master is up, the BO will retrieve a list of the slave FEs in case the master fails. If the master fails, the BO randomly chooses one of the slaves to be the new master.

Configuring the FE Servers

From the BO interface, populate the MFE_MASTER, MFE_FRONTENDS, and MFE_CONFIG fields on the Primo Home > Advanced Configuration > General Configuration page to assign the roles of the FE servers in a multiple FE configuration.

NOTE: Contact Ex Libris Support before implementing this configuration.

Multiple PDS Configuration

A multiple PDS configuration in Primo allows sites to configure one or more failover servers for the Primo PDS. In this configuration, only one server is active, while the others are inactive.

NOTE: If a failure occurs in the active PDS server, an inactive server must be started manually to provide PDS service.

Guidelines

When configuring a multiple PDS configuration, the following guidelines must be followed:

- All PDS servers must be configured using the following page in the Back Office:
Primo Home > Ongoing Configuration Wizards > PDS Configuration Wizard
 - All PDS servers must use the host name of the currently configured PDS. Update the pds_configuration_url field on the following page in the Back Office:
Primo Home > Advanced Configuration > General Configuration
 - Only one PDS server can be active at a time. All other PDS servers must be inactive. Set the pds_url and pds_internal_url fields on the following page in the Back Office to the host name of the active PDS:
Primo Home > Advanced Configuration > General Configuration
 - All FE servers must be restarted after a change to the Primo Home > Advanced Configuration > General Configuration page.
- For information on load balancer configurations, see [Load Balancer Requirements](#).

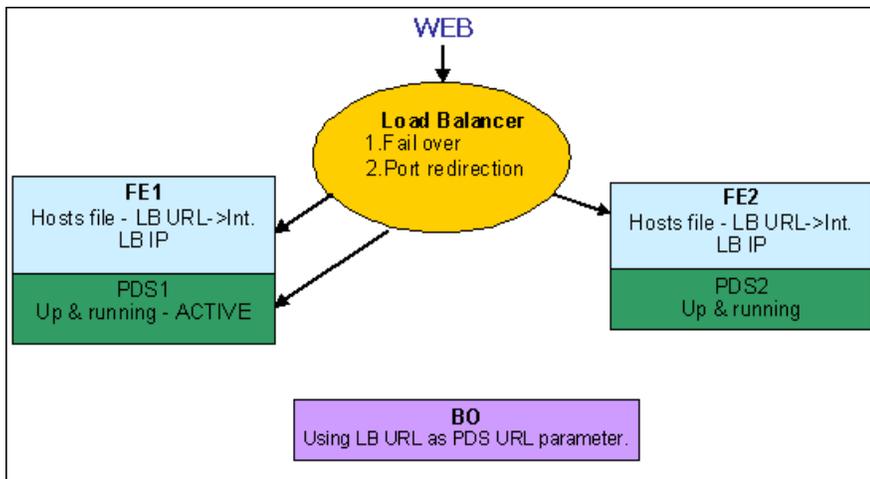
Load Balancer Requirements

To support fault tolerance, high availability, and better performance, Primo MFE (Multiple Front End) configurations allow load balancing. For more information on Primo topologies, contact Ex Libris Support. The load balancer (LB) distributes the work load between the active Front Ends (FE) and distributes the entire work load to the remaining FE's if an FE fails. LBs can be hardware-based (such as F5, Cisco, and so forth) or software-based (such as Apache). Primo does not require a specific type of LB, but it must support sticky sessions.

How Does It Work?

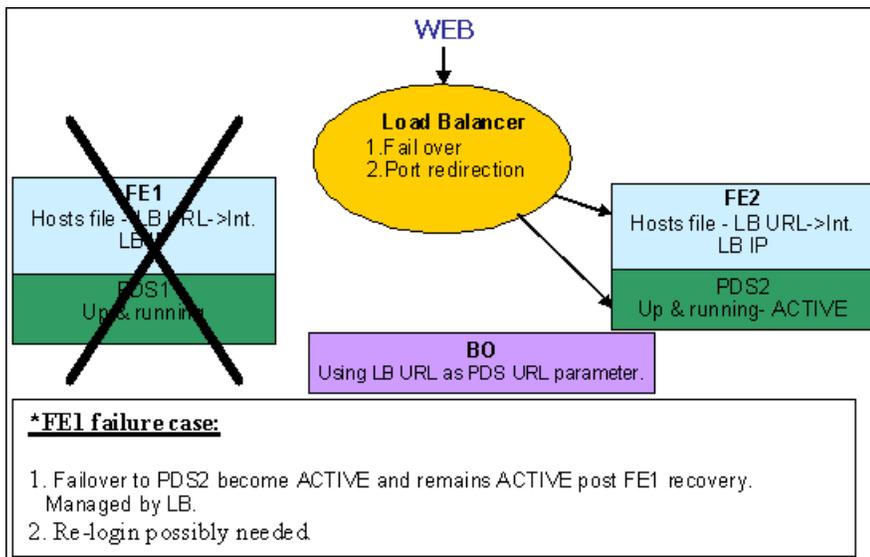
In all topology setups, all MFE's are active and respond to incoming requests. A typical Primo MFE configuration includes two or more Primo Front End (FE) servers running on a host-assigned internal IP, receiving HTTP request via port 1701. End users identified by source IPs must be balanced between existing Primo servers based on the Primo servers' computational power. For example, if your environment uses two Primo servers and both have an equal amount of computational power, the load balancer should assign half of the sessions to each server. Each FE must be running a PDS server, but only one of them is active at a time. The following figures illustrate the failover process when an LB is used in an MFE configuration. Note that PDS2 will remain active after the restoration of FE1/PDS1.

Figure 37:



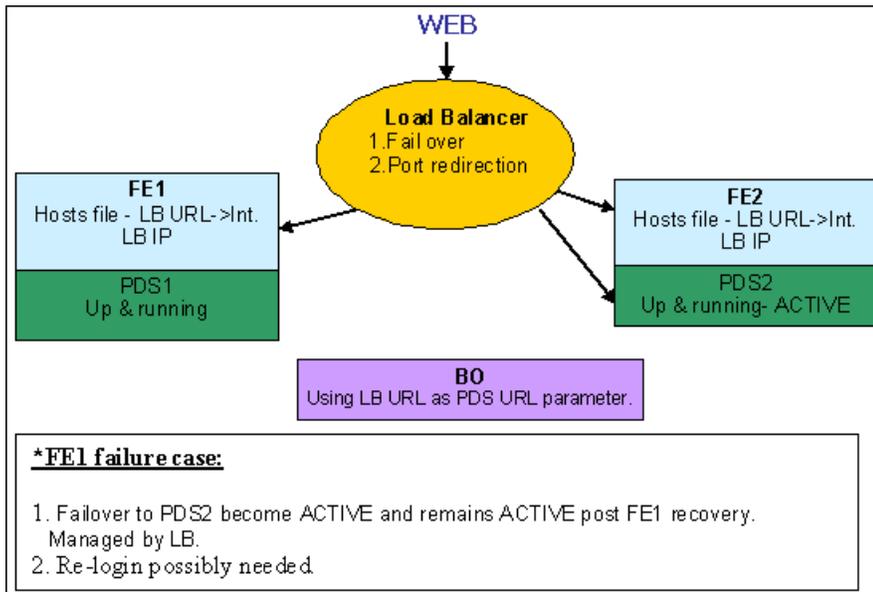
For image detail, press Ctrl++ Primo Load Balancing Diagram

Figure 38:



For image detail, press Ctrl++ Front End Failure - Load Balancer

Figure 39:



For image detail, press Ctrl++ Front End Restoration - Load Balancer

Load Balancer Setup

The setup of a LB varies between the various types of LBs. You need to make sure that all the MFEs are defined correctly in the LB and that the PDS is defined correctly in the LB. To support the failover of the PDS server, you need to configure the `pds_url` and `pds_internal_url` parameters in the Back Office to the LB address.

Sticky Sessions

The sticky session feature enables the load balancer to bind a user's session to a specific application instance so that all requests coming from the user during the session are sent to the same application instance.

Primo requires this feature to be activated on the LB so that all of the HTTP requests belonging to a user's session are routed to the same server.

Removing Servers for Maintenance Purposes

Any of the Primo FE servers can be removed from the load balancing group for maintenance purposes. The load balancing group must be reconfigured so that the load balancer does not continue to send requests to the removed server.

NOTE: Removing one or more FE servers decreases Primo performance.

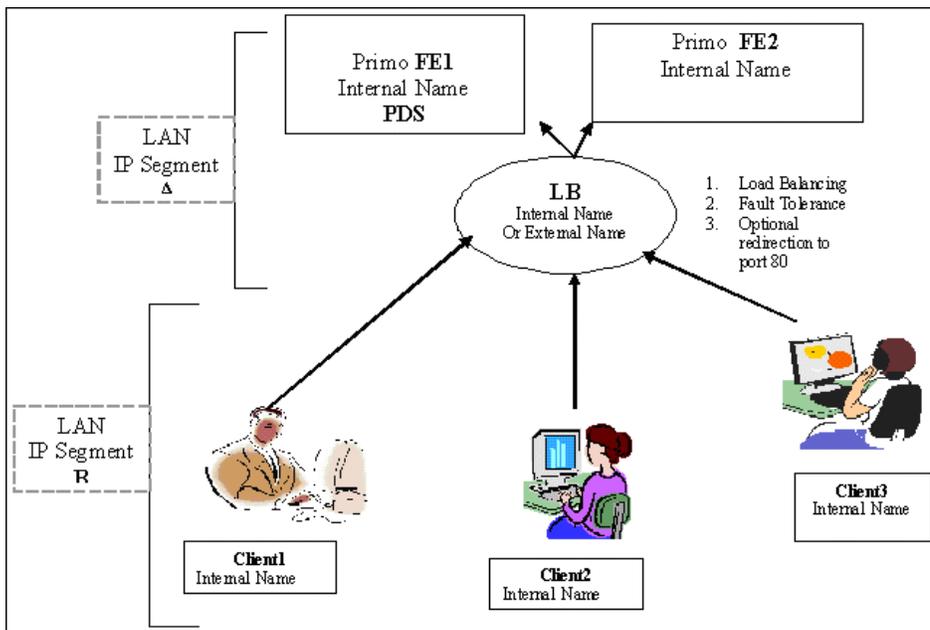
Load Balancer Monitor Guidelines

Ex Libris recommends that your load balancer provide a monitor utility that gathers the following statistics for each server in each group:

- Status
 - Number of requests performed
 - Cumulative size of the data sent
 - Cumulative size of the data received
 - Current connections illustrate the trend of successful connections over time
-

Load Balancer Test Case

The following figure shows an MFE load balancing test case performed by Ex Libris:
Figure 40:



For image detail, press Ctrl++ Load Balancer Test Case

NOTES:

- 1. Load Balancer brand, model, type, configuration, and infrastructure setup is under full customer responsibility.
- 2. The LB must support sticky sessions in order for Primo to work properly.

ⓘ The TimeOut parameter is necessary for remote search.

ⓘ Ex Libris recommends using a hardware load balancer for the Production environment.

Configuring SSL for Primo

This section details the procedure required to implement Secure Socket Layers (SSL) for the following:

- Provide secure authentication via PDS. For more information on PDS, refer to the Patron Directory Services Guide.
- Provide secure communication between Primo and Aleph's DLF API.

Implementing SSL for PDS Authentication

The Ex Libris staff and site staff must confirm the exact path to the SSL certificates and names of the SSLCertificateFile and SSLCertificateKeyFile files. The certificates are located in the following directory:

```
$httpd_root/SSLconf/conf
```

If the local installation is the first installation on the server, the \$httpd_root environment parameter in Primo typically refers to the following path:

```
/exlibris/primo/p1_1/primoe/apache
```

To implement the SSL changes:

1 Edit the following files:

- \$httpd_root/conf/ssl.conf

- \$httpd_root/conf/ssl.conf.tml

2 Make the relevant configuration changes in each of the files:

- Confirm or change the port (the default port within the ssl.conf table is 443). For example, change the following line:

```
Listen @_HTTPS_PORT
```

to:

```
Listen 443
```

- Change the paths and names of the SSLCertificateFile and SSLCertificateKeyFile parameters. For example, change the following lines, where <r> indicates the release in which Primo was initially installed and <c> indicates the Primo copy:

```
SSLCertificateFile /exlibris/primo/p<r><c>/primoe/apache/SSLconf/conf/@_SSL_CERTFILE
```

```
SSLCertificateFile /exlibris/primo/p<r><c>/primoe/apache/SSLconf/conf/@_SSL_CERTKEY
```

to:

```
SSLCertificateFile /exlibris/primo/p<r><c>/primoe/apache/SSLconf/conf/new.cert.cert
```

```
SSLCertificateKeyFile /exlibris/primo/p<r><c>/primoe/apache/SSLconf/conf/new.cert.key
```

3 In the /exlibris/primo/p<r><c>/primoe/apache/bin/apachectl_auto file, set the SSL_FLAG parameter to Y.

4 Open the following files for editing:

- /exlibris/primo/p<r><c>/primoe/apache/bin/apachectl

- /exlibris/primo/p<r><c>/primoe/apache/bin/apachectl.tml

5 Add the -DSSL string to each of the files. For example, change the following line:

```
HTTPD="/exlibris/primo/p<r><c>/product/bin/httpd -d
```

```
/exlibris/primo/p<r><c>/primoe/apache"
```

to:

```
HTTPD="/exlibris/primo/p<r><c>/product/bin/httpd -d
```

```
/exlibris/primo/p<r><c>/primoe/apache -DSSL"
```

6 If mod_ssl is shared, it must be loaded explicitly via the the httpd.conf file. To load it explicitly, perform the following steps:

a Enter the following command to see that the mod_ssl.so file exists:

```
ls $primo_product/local/apache/modules
```

b Enter the following commands to edit the httpd.conf file:

```
apcc
```

```
vi httpd.conf
```

cAdd the following line to the httpd.conf file:

```
LoadModule ssl_module $primo_product/local/apache/modules/mod_ssl.so
```

7Enter the following commands to restart the Apache server:

```
apcb
```

```
apachectl stop
```

```
apachectl auto
```

NOTE: If port 443 is used, you will need to restart the Apache server as the root user.

8Enter the following commands to edit the PDSDefinitions file:

```
pdsroot
```

```
cd program
```

```
vi PDSDefinitions
```

9Change the following lines:

```
> our ($server_httpd) = "http://servername:8991";
```

```
> our ($server_httpsd) = "http://servername:443";
```

```
> our ($server_pds) = "http://servername:8991/pds";
```

```
to:
```

```
> our ($server_httpd) = "https://servername:443";
```

```
> our ($server_httpsd) = "https://servername:443";
```

```
> our ($server_pds) = "https://servername:443/pds";
```

10Log on to the Primo Back Office.

11From the Primo Home -> Advanced Configuration -> General Configuration page, select Installation from the subsystem drop-down list.

12Set the PDS_URL field to `https://<server-name>:443/pds`.

Implementing Primo-Aleph Secure Connection

Primo contacts Aleph's JBoss server over HTTP in a method called REST (Representational State Transfer) according to the DLF ILS Discovery Interface API. Because the REST standards do not include session handling or authentication, the data that the API transmits is not protected by any identification or encryption methods. Therefore, customers may configure Aleph and Primo to communicate over SSL (an encrypted and authenticated method).

To configure SSL communication between Primo and Aleph:

1 Perform the following steps on the Primo server:

a Enter the following commands, making sure that the keypass and storepass are the same:

```
keytool -genkey -alias primo -keypass <passwd> -keystore  
/tmp/primo.keystore -storepass <passwd> -dname "cn=<Primo_hostname>, ou=primo, o=<company_name>,  
c=<country_prefix>" -validity 3650  
keytool -selfcert -alias primo -keypass <passwd> -keystore  
/tmp/primo.keystore -storepass <passwd> -dname "cn=<Primo_hostname>, ou=primo, o=<company_name>,  
c=<country_prefix>" -validity 3650  
keytool -list -keypass <passwd> -storepass <passwd>  
-keystore  
/tmp/primo.keystore -rfc > /tmp/primo.crt
```

b Edit the file so that it contains only the certificate information, which includes the following beginning and ending lines:

```
-----BEGIN CERTIFICATE-----  
.  
.  
.  
-----END CERTIFICATE-----
```

2 Perform the following steps on the Aleph server:

a Enter the following commands, making sure that the keypass and storepass are the same:

```
keytool -genkey -alias aleph -keypass <passwd> -keystore  
/exlibris/aleph/a21_8/ng/aleph/home/system/thirdparty/openserver/server/default/conf/aleph.keystore -storepass  
<passwd> -dname "cn=<aleph_hostname>, ou=aleph, o=<company_name>, c=<country_prefix>" -validity 3650  
keytool -selfcert -alias aleph -keypass <passwd> -keystore  
/exlibris/aleph/a21_8/ng/aleph/home/system/thirdparty/openserver/server/default/conf/aleph.keystore -storepass  
<passwd> -dname "cn=<aleph_hostname>, ou=aleph, o=<company_name>, c=<country_prefix>" -validity 3650  
keytool -list -keypass <passwd> -storepass <passwd>  
-keystore  
/exlibris/aleph/a21_8/ng/aleph/home/system/thirdparty/openserver/server/default/conf/aleph.keystore -rfc >  
/tmp/aleph.crt
```

b Edit the file so that it contains only the certificate information, which includes the following beginning and ending lines:

```
-----BEGIN CERTIFICATE-----  
.  
.
```

```
-----END CERTIFICATE-----
```

cEnter the following commands to copy the certificates to and from the Primo server:

```
scp aleph.crt primo@<primo_server>:/tmp  
scp primo@<primo_server>:/tmp/primo.crt /tmp
```

3Enter the following command on the Primo server to import aleph.crt into Primo's keystore:

```
keytool -import -alias aleph -file /tmp/aleph.crt -storepass <passwd> -keystore /tmp/primo.keystore -noprompt
```

4Perform the following steps on the Aleph server:

aEnter the following command to import primo.crt into Aleph's keystore:

```
keytool -import -alias primo -file /tmp/primo.crt  
-storepass <passwd> -keystore /exlibris/aleph/a21_8/ng/aleph/home/system/thirdparty/openserver/server/default/conf/  
aleph.keystore -noprompt
```

bEnter the following commands to access the server.xml file:

```
cd $aleph_dev/ng/aleph/home/system/thirdparty/openserver/server/default/deploy/jbossweb-tomcat.sar
```

If the above directory does not exist, enter the following command:

```
cd $aleph_dev/ng/aleph/home/system/thirdparty/openserver/server/default/deploy/jbossweb.sar
```

cEnter the following command to edit the server.xml file:

```
vi server.xml
```

dIn the following section in the server.xml file, modify the path of the keystore file and its password if needed.

```
<Connector port="8443" address="{jboss.bind.address}"  
maxThreads="100" minSpareThreads="5" maxSpareThreads="15"  
scheme="https" secure="true" clientAuth="true"  
disableUploadTimeout="true"  
keystoreFile="{jboss.server.home.dir}/conf/aleph.keystore"  
keystorePass="{passwd}"  
truststoreFile="{jboss.server.home.dir}/conf/aleph.keystore"  
truststorePass="{passwd}"  
connectionTimeout="20000" sslProtocol="TLS" />
```

NOTE: Make sure that the above lines are not commented out before saving the server.xml file.

When selecting a port for SSL communication, enter the following command to make sure that port is open and free:

```
telnet <server_name_or_ip><ssl_port>
```

In the following example, port 8888 is open for communication to the il-aleph07 server:

```
telnet il-aleph07 8888  
  
Trying 10.1.235.93...  
Connected to il-aleph07.  
Escape character is '^['.
```

For image detail, press Ctrl++ Telnet Example

eEnter the following command to edit the main.properties file:

```
vi $aleph_root/jboss_conf/main.properties
```

fAdd the following line to the main.properties file to make sure that access to the REST API only is available in HTTPS:

```
api.rest.https=Y
```

gEnter the following commands to restart Aleph's JBoss server:

Jbin

jboss_shutdown.sh

jboss_startup.sh

5Perform the following steps in the Primo Back Office:

aOn the Primo Home > Advanced Configuration > General Configuration Wizard page, select the Installation subsystem from the drop-down list.

bOn the General Configuration Wizard page, specify the following parameters:

- `javax.net.ssl.trustStore` – Enter the full path to the `primo.keystore` file.
- `javax.net.ssl.keyStore` – Enter the full path to the `primo.keystore` file.
- `javax.net.ssl.trustStorePassword` – Enter the password used for the truststore.
- `javax.net.ssl.keyStorePassword` – Enter the password used for the keystore.

cClick Save & Continue.

dOn the Primo Home > Ongoing Configuration Wizards > Institution Wizard page, edit your institution.

eOn the Edit Institution page, specify the HTTPS URL in the API field under the Delivery Base URLs section.

fClick Save & Continue.

gDeploy the changes to the institution.

6On the Primo server, restart Primo's Front End.