Voyager®
Primo® Integration User's Guide

April 2009

ExLibris The bridge to knowledge

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, URM, Voyager, 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.

Document released: April 2009

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

Contents

Contents

Contents

Contents

Figures

2 Data Extraction

3 Patron Services

4 Course Reserves

5 Real Time Availability

6 Universal Borrowing with Primo

Figures

Tables

Tables

About This Document

Purpose

The purpose of the *Voyager® Primo® Integration User's Guide* is to provide you with the information that you need to do the following.

- Set up/modify configuration files.
- Run batch files.
- Set up user authentication with Patron Directory Services.
- Set up the system for processing user requests.

Intended Audience

This document is written for a technical audience responsible for the following:

- Configuration file setup.
- Initial upload of Voyager data into Primo.
- Writing scripts and scheduling batch jobs for running updates.
- User authentication setup.
- Review/analysis of message logs as needed.

Reason for Reissue

This guide incorporates and is being reissued for the following reasons:

- Addition of cross references to the circulation batch jobs 34 and 36 described in the *Voyager Technical User's Guide*. See the note in CourseReserves= on page 4-2.
- Correction to directory path referenced in step 2 of Real Time Availability Processing that starts on page 5-1.
- Addition of Universal Borrowing with Primo on page 6-1.
- Addition of Voyager extract to Primo that includes Rosetta information. Additional parameters are provided to enable this function. See the following:

  - [Enrichment Tags] on page 2-26 for information regarding the `InsertDPSTag`

  - [DPS] on page 2-27 for information regarding the new stanza implemented for this function

  - Figure 2-1 on page 2-6, Figure 2-2 on page 2-12, and Figure 2-3 on page 2-17 for examples of the `.ini` configuration files with the new `[DPS]` stanza and updated `[Enrichment Tags]` stanza.

Document Summary

Chapter 1	"Getting Started" This chapter describes the prerequisites for getting started with the Voyager Primo integration setup.
Chapter 2	"Data Extraction" This chapter provides information needed to process the initial data extraction from Voyager and ongoing updates.
Chapter 3	"Patron Services" This chapter provides information about patron services such as user authentication and processing user requests.
Chapter 4	"Course Reserves" This chapter provides information regarding the extraction of course reserves data from the Voyager database.
Chapter 5	"Real Time Availability" This chapter provides information about real time availability processing with Voyager Primo integration.

Chapter 6	"Universal Borrowing with Primo" This chapter provides information about implementing the Voyager Universal Borrowing (UB) function with Primo as the patron's user interface.
Index	The Index is an alphabetical, detailed cross-reference of topics.

Conventions Used in This Document

The following conventions are used throughout this document:

- Names of commands, variables, stanzas, files, and paths (such as `/dev/tmp`), as well as selectors and typed user input, are displayed in `constant width` type.
- Commands or other keyboard input that must be typed exactly as presented are displayed in **`constant width bold`** type.
- Commands or other keyboard input that must be supplied by the user are displayed in ***`constant width bold italic`*** type.
- System-generated responses such as error messages are displayed in `constant width` type.
- Variable *portions* of system-generated responses are displayed in *`constant width italic`* type.
- Keyboard commands (such as **Ctrl** and **Enter**) are displayed in **bold**.
- Required keyboard input such as "Enter **`vi`**" is displayed in **`constant width bold`** type.
- Place holders for variable portions of user-defined input such as **`ls -l`** ***`filename`*** are displayed in ***`italicized constant width bold`*** type.
- The names of menus or status display pages and required selections from menus or status display pages such as "From the **Applications** drop-down menu, select **System-wide**," are displayed in **bold** type.
- Object names on a window's interface, such as the **Description** field, the **OK** button, and the **Metadata** tab, are displayed in **bold** type.
- The titles of documents such as *Acquisitions User's Guide* are displayed in *italic* type.
- Caution, and important notices are displayed with a distinctive label such as the following:

NOTE:
Extra information pertinent to the topic.

IMPORTANT:
Information you should consider before making a decision or configuration.

CAUTION:
Information you must consider before making a decision, due to potential loss of data or system malfunction involved.

TIP:
Helpful hints you might want to consider before making a decision.

RECOMMENDED:
Preferred course of action.

OPTIONAL:
Indicates course of action which is not required, but may be taken to suit your library's preferences or requirements.

Document Reproduction/Photocopying

Photocopying the documentation is allowed under your contract with Ex Libris (USA) Inc. It is stated below:

> All documentation is subject to U.S. copyright protection. CUSTOMER may copy the printed documentation only in reasonable quantities to aid the employees in their use of the SOFTWARE. Limited portions of documentation, relating only to the public access catalog, may be copied for use in patron instruction.

Comment on This Document

Please contact Customer Support to provide us with your feedback. For Customer Support contact information, see SupportWeb at:

`http://www.exlibrisgroup.com/support_center.htm`.

To Submit Comments by E-mail

To submit comments by e-mail, please send your message to:

`docmanager@exlibrisgroup.com`

Getting Started

1

Getting Started

1

Introduction

The *Voyager*® *Primo*® *Integration User's Guide* is designed to provide you with the information that you need to successfully utilize Voyager data in a Primo user environment. The following components are a part of the Voyager Primo integration.

- Configuration files for data extraction.
- Batch files to run jobs.
- User authentication with Patron Directory Services (PDS).
- User information provided with Library Card.
- User request processing with Primo's GetIt! function.

Purpose of this Chapter

The purpose of this chapter is to identify the skills and preparations necessary for setting up your environment for Voyager Primo integration.

Prerequisite Skills and Knowledge

To use this document effectively and set up your systems for Voyager Primo integration, you need to have a general understanding of the following.

- Microsoft® Windows-based applications.
- XML and XML-format data as well as editors and related software.
- UNIX® operating system commands and file system.
- At least one UNIX-based text editor such as `ed` or `vi`.
- Local procedures.

This user's guide assumes you have a working knowledge of Primo setup and access to Primo documentation.

Before You Begin

Before you can begin your Voyager Primo integration, you need to have the following software installed.

- Voyager (the base product)
- Primo
- Voyager Primo integration feature component

Checklist

In preparation for setting up Voyager Primo integration, gather the following information.

- Identify the elements needed to customize configuration files.
- Identify the schedule/times for processing updates.
- Identify excluded happening locations.
- Identify excluded operator IDs.
- Identify institution ID.
- Identify statuses to be made available to Primo.

Port Settings

Voyager port settings are defined one time at installation.

NOTE:
The Real Time Availability (RTA) port is different from the OPAC port. The RTA port is also defined one time at installation.

Refer to the *Primo Interoperability Guide* for additional information regarding port settings.

Data Extraction

2

Data Extraction

2

Introduction

Primo discovery takes advantage of the rich data stored in the Voyager database. To do this, bibliographic data is extracted from the Voyager database for an initial load and then subsequently kept current via regularly scheduled updates with Primo. This chapter discusses the data extraction process.

Purpose of this Chapter

The purpose of this chapter is to describe the components needed for extracting Voyager data for Primo to harvest.

Data Extraction Process Overview

The Voyager Primo integration data extraction tools are used for the following activities.

- Initial load of Voyager data for a new installation of Primo.
- Ongoing updates to the Primo database for an existing installation of Primo harvesting Voyager data.
- Other special purposes.

Initial Primo Data Load

For the initial Primo database load, you need to do the following.

- Tailor the configuration file to your environment/needs.

  Use the `PrimoExp-Publishing.ini` as a starting point. See Publishing Configuration File on page 2-11.

- Run the `Pprimoexp` batch job.

  Using the parameters set in the configuration file, this step identifies the Voyager records to be uploaded to the Primo database.

  A separate script, `PbundlePrimo`, wraps the Voyager records in a `tar.gz` file for Primo to access. The `tar.gz` files are located in `/m1/voyager/yyydb/primo/Publishing/bundled` where `yyydb` is the database name.

- Permit Primo to securely access the directory in which the generated files from Voyager are placed. This is the output location for the `tar.gz` files.

  NOTE:
  A number of options exist for moving the data to the Primo server for the initial load such as using FTP, a shared NFS mount, manually copying the data, or utilizing a cron/script to copy the data. For additional information, see your implementation support contact to discuss the alternatives for Primo to access Voyager data.

Initial Load Generation - Excluded Records

When identifying bibliographic records for the initial data load, the following criteria is used to identify records to be excluded.

- If a bibliographic record is suppressed from the OPAC, a MARCXML record is not generated.
- If a bibliographic record was most recently modified at one of the excluded happening locations as identified in the configuration file, a MARCXML record is not generated.
- If a bibliographic record was most recently modified by one of the excluded operators as identified in the configuration file, a MARCXML record is not generated.

See Stanza Definitions for Voyager Primo Integration Configuration Files on page 2-23 for more information about customizing the configuration file.

Ongoing Updates to Primo Database

You need to evaluate how frequently to process updates to the Primo database. Consider the following.

- Level of activity within your Voyager database.
- Frequency of bibliographic record changes.
- Frequency of item status changes.
- Time of day when system usage is greatest.
- Time of day when system usage is the least.

Pipes

Primo uses separate pipes for harvesting Voyager data.

One pipe is used to harvest Voyager data with bibliographic record changes. This is the publishing pipe.

The other pipe is used to harvest Voyager data with holdings availability changes. This is the availability pipe.

Given this approach for harvesting data, separate and customized versions of configuration files are provided to you for publishing and availability purposes as follows.

- `PrimoExp-Publishing.ini`. See Publishing Configuration File on page 2-11 for more information.
- `PrimoExp-Availability.ini`. See Availability Configuration File on page 2-6 for more information.

These are located in `/m1/voyager/yyydb/ini` where `yyydb` is the database name.

These configuration files are the same except for the parameter setting differences required for each purpose. See Stanza Definitions for Voyager Primo Integration Configuration Files on page 2-23 for more information about the parameters that affect which records are generated for harvesting.

NOTE:
The type of content in these generated records is the same. However, different records get generated for Primo harvesting as identified by the triggers (parameters) specified in the configuration files.

Corresponding copies of `Pprimoexp` batch files for publishing and availability purposes can be created in order to automate the harvesting process using a cron. See Pprimoexp Batch Job on page 2-33 for more information about `Pprimoexp` batch processing.

Having separate functional configuration files, batch files, and directories for generated files enables your flexibility in scheduling production processing of data changes in the Voyager Primo integrated environment. This allows you to schedule Primo harvesting of bibliographic record changes from Voyager at an appropriate time and separate from scheduling harvesting of Voyager data with holdings availability changes.

Special Purpose/Selective Database Extractions

Aside from the initial Primo data load and ongoing updates, you may want to set up the Voyager Primo integration tools for other special purposes that meet your institution's specific requirements.

For example, you may want to set up the Voyager Primo integration tools to create a testing subset before doing a complete initial load of the database. This can be done using parameters like `BibRangeBegin` and `BibRangeEnd` or `BibsFromFile`.

The `PrimoExp-Selective.ini` configuration file is provided as one method for creating a testing subset. It is located in `/m1/voyager/yyydb/ini` where `yyydb` is the database name.

For Selective record processing and storing `tar.gz` files, the following directory is provided where `yyydb` is the database name.

- `/m1/voyager/yyydb/primo/Selective/bundled`.

  Insure that Primo has access to this directory.

Primo Transaction / Record Normalization

Primo originates an FTP transaction to Voyager to harvest the records selected by the `Pprimoexp` processing for inclusion in the Primo database. Subsequently within Primo, these records are normalized into the Primo Normalized XML (PNX) format.

NOTE:
A number of options exist for moving the data to the Primo server such as using a shared NFS mount, manually copying the data, or utilizing a cron/script to copy the data. See your implementation support contact to discuss alternatives for harvesting data.

Configuration Files

Primo uses data extracted from the Voyager database. Configuration files are used to define what is extracted. The following customized configuration files are provided with your installation.

- `PrimoExp-Availability.ini`. See Availability Configuration File on page 2-6.
- `PrimoExp-Publishing.ini`. See Publishing Configuration File on page 2-11.
- `PrimoExp-Selective.ini`. Selective Configuration File on page 2-17.

They are located in `/m1/voyager/yyydb/ini` where `yyydb` is the database name.

These configuration files are provided as examples for publishing, availability, and selective requirements with parameters set to what many customers use. However, copies of the configuration files may be made and tailored to meet your unique needs.

The configuration files define the following elements.

- Item statuses to check for initial load and ongoing updates to the Primo database.
- Happening locations for exclusion. Records associated with these happening locations are excluded from the initial load and/or updates for Primo harvesting.
- Records to be excluded from the initial load and/or ongoing updates determined by operator ID that processed the record. See [Excluded Operator Ids] on page 2-25.
- Institution ID to be included in the availability tag.
- Location definitions for consistency within the Primo database. See [Location to Primo] on page 2-25.
- Tags to identify unique field usage within your institution's database specific to the database extract. See [Enrichment Tags] on page 2-26.
- Extract processing considerations such as date/time, record ID(s), filters, and so on. See [PrimoExp] on page 2-27.
- Additional processing considerations for uploading the Voyager records identified by the configuration file processing. See [XMLWriterProtocol] on page 2-32.

See Stanza Definitions for Voyager Primo Integration Configuration Files on page 2-23 for a complete description of configuration file options.

Availability Configuration File

The following is the customized availability configuration file provided at installation.

```
# Configuration for Voyager Export for Primo -- Availability

# Voyager status maps to (A)vailable or (U)navailable for Primo
[Item Statuses]
Not Charged=A
Charged=U
Renewed=U
Overdue=U
Recall Request=U
Hold Request=U
On Hold=U
In Transit=U
In Transit Discharged=U
In Transit On Hold=U
Discharged=A
Missing=U
Lost--Library Applied=U
Lost--System Applied=U
Claims Returned=U
Damaged=U
Withdrawn=U
At Bindery=U
Cataloging Review=A
Circulation Review=A
Scheduled=U
In Process=U
Call Slip Request=U
Short Loan Request=U
Remote Storage Request=U

[Excluded Happening Locations]
```

Figure 2-1. PrimoExp-Availability.ini Example

```
# Items whose most recent activity took place in one of these locations will
          not be exported.


[Excluded Operator Ids]
# Items whose most recent activity was recorded by one of these operators will
          not be exported.


[Institution Id]
# This value will be included in the Availability tag
# as subfield a
Id=QAVOY


[Location to Primo]
# This section maps Voyager locations to Primo locations.


[Enrichment Tags]
# tags added to marc record with additional data
# AVA tag contains the availability information
# AUT tag contains name cross-reference information
# SUB tag contains subject cross-reference information
# CR tag contains course reserve information
# DPS tag contains Digital Preservation System (Product Rosetta) information
InsertAVATag=949
InsertAUTTag=950
InsertSUBTag=951
InsertCRTag=952
InsertDPSTag=953


#=========================================================================
# Preservation related object link definition
# This includes the label for the URL link, the URL link, and parameter name.
# The only supported dynamic parameter at this time is id (i.e. bib id).
# The parameter is passed to the third-party application that processes and
          retrieves
# intellectual entities from the Preservation repository.
#=========================================================================
[DPS]
DPSLinkLabel=Rosetta Link:
```

Figure 2-1. PrimoExp-Availability.ini Example (Continued)

```
DPSLinkAddress=http://somecomponent.com/content-aggregator/
          getIEs?system=ilsdb&
DPSLinkParm=id=



[PrimoExp]
# note that stanza.name can be used with -- on the command line
# argv(case insensitive) override
# ini(case insensitive) override
# env (all uppercase)
#
# these parameters provide defaults for this run of the job
# each of them can be overridden using a command-line parameter
#
# BibRangeBegin and BibRangeEnd are the beginning and ending bibliographic ID
          numbers to export.
# -B overrides this parameter on the command line
# BibRangeBegin
#BibRangeBegin=1
# -E overrides this parameter on the command line
#BibRangeEnd=200
#
# BibsFromFile reads the file at the supplied path for a list of bibliographic
          record ids to export.
# The file should contain one bibliographic record ID per line.
# -F overrides this parameter on the command line
# BibsFromFile=/m1/voyager/2007.1.0/qa710db/local/PrimoBibs.txt
#
# ChangedSince exports records changed since the supplied date.
# ChangedSince may be either
#     a timestamp in YYYYMMDD.HHMMSS format
#    or a path to a file containing a timestamp in YYYYMMDD.HHMMSS format on a
          single line
# If a path to a file is supplied, the current run time will be saved in the
          file when
# the job finishes.
# -C overrides this parameter on the command line
# Example:
# ChangedSince=YYYYMMDD.HHMMSS
```

Figure 2-1. PrimoExp-Availability.ini Example (Continued)

```
#  or
# ChangedSince=/m1/voyager/2007.1.0/qa710db/primo/Availability/
          ChangedSince.txt
ChangedSince=/m1/voyager/2007.1.0/qa710db/primo/Availability/ChangedSince.txt
#
#
# HeadingChanges determines whether to export bibliographic records
# whose cross-references have changed
# -H Y|N overrides this parameter on the command line
HeadingChanges=N
#
# AvailFilter detemines whether to export records whose availability
# fields have not changed since the last time the job ran.
# If AvailFilter is Y, only records whose availability has changed are
          exported;
# if AvailFilter is N, all changed records are exported.
# -A Y|N overrides this parameter on the command line
AvailFilter=Y
#
# for filenames
# note that $str$ will be passed to date format so that you can
# specify date in the these files
#
# LogFile is the location of the log file.
# Note that $str$ will be passed to Java's Date.Format()
# function to control the datestamp embedded in the file name.
# -L overrides this parameter on the command line
LogFile=primo.export.$yyMMdd$.log
LogFileDir=/m1/voyager/2007.1.0/qa710db/primo/Availability/logs
:
#
# -v
# the logging level
LogLevel=10
# LogToStdOut determines whether or not to write the log output to standard
          output,
# as well as the log file.
LogToStdOut=N
```

Figure 2-1. PrimoExp-Availability.ini Example (Continued)

```
# Database and the other Database parameters determine which database to
          connect to.
# Ordinarily these are set by the wrapper script based on the environment.
# -d
# DataBase=
#
# DatabaseHost
DatabaseHost=10.100.2.32
#
# DatabasePort
# -e
# UserPass
# The credentials needed to connect to the database.
# Optional parameter, override on command line with -u
# STRONG RECOMMENDATION: do not set this value here, use
# the command line override instead.
# Special note: Oracle TNS Alias (@DB) is _not_ used here
#UserPass=qa710db/qwrite0
#
Database=VGER
Protocol=com.endinfosys.voyager.extract.OAIPMHXmlWriterProtocol

# must do items, then mfhds, then bibs
Task=com.endinfosys.voyager.extract.ExtractItemsPrimo
Task=com.endinfosys.voyager.extract.ExtractMFHDsPrimo
Task=com.endinfosys.voyager.extract.ExtractBibsPrimo
DoItems=Y
DoMfhds=Y

# The Del* Dir and File are used to determine when records are deleted.
# These are standard Voyager locations.
DelBibsDir=/m1/voyager/2007.1.0/qa710db/rpt
DelBibFile=deleted.bib.marc
DelMFHDsDir=/m1/voyager/2007.1.0/qa710db/rpt
DelMFHDsFile=deleted.mfhd.marc
DelItemsDir=/m1/voyager/2007.1.0/qa710db/rpt
DelItemsFile=delete.item
```

Figure 2-1. PrimoExp-Availability.ini Example (Continued)

```
# Set to Y to include course reserve data.
CourseReserves=N


[XMLWriterProtocol]
# OAIPMHXmlWriterProtocol will include the bib id
#  as the last element in the file name before
#  the .xml extension.
# Note the path must be coordinated with the bundlePrimo.ksh
#  script execution.
File=/m1/voyager/2007.1.0/qa710db/primo/Availability/exports/
          primo.export.$yyMMddhhmmss$.xml
XSL=
# max number of records to group in a single tar file
#  all in single group if 0 or undefined
recsPerGroup=1000
# prefix for record identifiers, must be present, may be empty
idPrefix=
#
# Set to Y to enable pretty printing of XML output.
# Caution: may affect data with embedded spaces
UsePrettyPrint=N


#
# EVERYTHING BELOW THIS LINE IS INTERNAL DO NOT MODIFY
#
```

Figure 2-1. PrimoExp-Availability.ini Example (Continued)

Publishing Configuration File

The following is the customized publishing configuration file provided at installation.

```
# Configuration for Voyager Export for Primo -- Publishing


# Voyager status maps to (A)vailable or (U)navailable for Primo
[Item Statuses]
Not Charged=A
Charged=U
Renewed=U
Overdue=U
Recall Request=U
Hold Request=U
On Hold=U
In Transit=U
In Transit Discharged=U
In Transit On Hold=U
Discharged=A
Missing=U
Lost--Library Applied=U
Lost--System Applied=U
Claims Returned=U
Damaged=U
Withdrawn=U
At Bindery=U
Cataloging Review=A
Circulation Review=A
Scheduled=U
In Process=U
Call Slip Request=U
Short Loan Request=U
Remote Storage Request=U


[Excluded Happening Locations]
# Items whose most recent activity took place in one of these locations will
          not be exported.


[Excluded Operator Ids]
# Items whose most recent activity was recorded by one of these operators will
          not be exported.
```

Figure 2-2. PrimoExp-Publishing.ini Example

```
[Institution Id]
# This value will be included in the Availability tag
# as subfield a
Id=QAVOY

[Location to Primo]
# This section maps Voyager locations to Primo locations.

[Enrichment Tags]
# tags added to marc record with additional data
# AVA tag contains the availability information
# AUT tag contains name cross-reference information
# SUB tag contains subject cross-reference information
# CR tag contains course reserve information
# DPS tag contains Digital Preservation System (Product Rosetta) information
InsertAVATag=949
InsertAUTTag=950
InsertSUBTag=951
InsertCRTag=952
InsertDPSTag=953

#=========================================================================
# Preservation related object link definition
# This includes the label for the URL link, the URL link, and parameter name.
# The only supported dynamic parameter at this time is id (i.e. bib id).
# The parameter is passed to the third-party application that processes and
          retrieves
# intellectual entities from the Preservation repository.
#=========================================================================
[DPS]
DPSLinkLabel=Rosetta Link:
DPSLinkAddress=http://somecomponent.com/content-aggregator/
          getIEs?system=ilsdb&
DPSLinkParm=id=



[PrimoExp]
```

Figure 2-2. PrimoExp-Publishing.ini Example (Continued)

```
# note that stanza.name can be used with -- on the command line
# argv(case insensitive) override
# ini(case insensitive) override
# env (all uppercase)
#
# these parameters provide defaults for this run of the job
# each of them can be overridden using a command-line parameter
#
# BibRangeBegin and BibRangeEnd are the beginning and ending bibliographic ID
          numbers to export.
# -B overrides this parameter on the command line
# BibRangeBegin
#BibRangeBegin=1
# -E overrides this parameter on the command line
#BibRangeEnd=200
#
# BibsFromFile reads the file at the supplied path for a list of bibliographic
          record ids to export.
# The file should contain one bibliographic record ID per line.
# -F overrides this parameter on the command line
# BibsFromFile=/m1/voyager/2007.1.0/qa710db/local/PrimoBibs.txt
#
# ChangedSince exports records changed since the supplied date.
# ChangedSince may be either
#     a timestamp in YYYYMMDD.HHMMSS format
#    or a path to a file containing a timestamp in YYYYMMDD.HHMMSS format on a
          single line
# If a path to a file is supplied, the current run time will be saved in the
          file when
# the job finishes.
# -C overrides this parameter on the command line
# Example:
# ChangedSince=YYYYMMDD.HHMMSS
#  or
# ChangedSince=/m1/voyager/2007.1.0/qa710db/primo/Availability/
          ChangedSince.txt
ChangedSince=/m1/voyager/2007.1.0/qa710db/primo/Publishing/ChangedSince.txt
#
#
```

Figure 2-2. PrimoExp-Publishing.ini Example (Continued)

```
# HeadingChanges determines whether to export bibliographic records
# whose cross-references have changed
# -H Y|N overrides this parameter on the command line
HeadingChanges=Y
#
# AvailFilter detemines whether to export records whose availability
# fields have not changed since the last time the job ran.
# If AvailFilter is Y, only records whose availability has changed are
          exported;
# if AvailFilter is N, all changed records are exported.
# -A Y|N overrides this parameter on the command line
AvailFilter=N
#
# for filenames
# note that $str$ will be passed to date format so that you can
# specify date in the these files
#
# LogFile is the location of the log file.
# Note that $str$ will be passed to Java's Date.Format()
# function to control the datestamp embedded in the file name.
# -L overrides this parameter on the command line
LogFile=primo.export.$yyMMdd$.log
LogFileDir=/m1/voyager/2007.1.0/qa710db/primo/Publishing/logs
#
#
# -v
# the logging level
LogLevel=10
# LogToStdOut determines whether or not to write the log output to standard
          output,
# as well as the log file.
LogToStdOut=N


# Database and the other Database parameters determine which database to
          connect to.
# Ordinarily these are set by the wrapper script based on the environment.
# -d
# DataBase=
#
```

Figure 2-2. PrimoExp-Publishing.ini Example (Continued)

```
# DatabaseHost
DatabaseHost=10.100.2.32
#
# DatabasePort
# -e
# UserPass
# The credentials needed to connect to the database.
# Optional parameter, override on command line with -u
# STRONG RECOMMENDATION: do not set this value here, use
# the command line override instead.
# Special note: Oracle TNS Alias (@DB) is _not_ used here
#UserPass=qa710db/qwrite0
#
Database=VGER
Protocol=com.endinfosys.voyager.extract.OAIPMHXmlWriterProtocol

# must do items, then mfhds, then bibs
Task=com.endinfosys.voyager.extract.ExtractItemsPrimo
Task=com.endinfosys.voyager.extract.ExtractMFHDsPrimo
Task=com.endinfosys.voyager.extract.ExtractBibsPrimo
DoItems=Y
DoMfhds=Y

# The Del* Dir and File are used to determine when records are deleted.
# These are standard Voyager locations.
DelBibsDir=/m1/voyager/2007.1.0/qa710db/rpt
DelBibFile=deleted.bib.marc
DelMFHDsDir=/m1/voyager/2007.1.0/qa710db/rpt
DelMFHDsFile=deleted.mfhd.marc
DelItemsDir=/m1/voyager/2007.1.0/qa710db/rpt
DelItemsFile=delete.item

# Set to Y to include course reserve data.
CourseReserves=N

[XMLWriterProtocol]
# OAIPMHXmlWriterProtocol will include the bib id
```

Figure 2-2. PrimoExp-Publishing.ini Example (Continued)

```
#  as the last element in the file name before
#  the .xml extension.
# Note the path must be coordinated with the bundlePrimo.ksh
#  script execution.
File=/m1/voyager/2007.1.0/qa710db/primo/Publishing/exports/
          primo.export.$yyMMddhhmmss$.xml
XSL=
# max number of records to group in a single tar file
#  all in single group if 0 or undefined
recsPerGroup=1000
# prefix for record identifiers, must be present, may be empty
idPrefix=
#
# Set to Y to enable pretty printing of XML output.
# Caution: may affect data with embedded spaces
UsePrettyPrint=N


#
# EVERYTHING BELOW THIS LINE IS INTERNAL DO NOT MODIFY
#
```

Figure 2-2. PrimoExp-Publishing.ini Example (Continued)

Selective Configuration File

The following is the customized selective configuration file provided at installation.

```
# Configuration for Voyager Export for Primo -- Selective


# Voyager status maps to (A)vailable or (U)navailable for Primo
[Item Statuses]
Not Charged=A
Charged=U
Renewed=U
Overdue=U
```

Figure 2-3. PrimoExp-Selectivew.ini Example

```
Recall Request=U
Hold Request=U
On Hold=U
In Transit=U
In Transit Discharged=U
In Transit On Hold=U
Discharged=A
Missing=U
Lost--Library Applied=U
Lost--System Applied=U
Claims Returned=U
Damaged=U
Withdrawn=U
At Bindery=U
Cataloging Review=A
Circulation Review=A
Scheduled=U
In Process=U
Call Slip Request=U
Short Loan Request=U
Remote Storage Request=U


[Excluded Happening Locations]
# Items whose most recent activity took place in one of these locations will
          not be exported.


[Excluded Operator Ids]
# Items whose most recent activity was recorded by one of these operators will
          not be exported.


[Institution Id]
# This value will be included in the Availability tag
# as subfield a
Id=QAVOY


[Location to Primo]
# This section maps Voyager locations to Primo locations.
```

Figure 2-3. PrimoExp-Selectivew.ini Example (Continued)

```
[Enrichment Tags]
# tags added to marc record with additional data
# AVA tag contains the availability information
# AUT tag contains name cross-reference information
# SUB tag contains subject cross-reference information
# CR tag contains course reserve information
# DPS tag contains Digital Preservation System (Product Rosetta) information
InsertAVATag=949
InsertAUTTag=950
InsertSUBTag=951
InsertCRTag=952
InsertDPSTag=953


#============================================================================
# Preservation related object link definition
# This includes the label for the URL link, the URL link, and parameter name.
# The only supported dynamic parameter at this time is id (i.e. bib id).
# The parameter is passed to the third-party application that processes and
          retrieves
# intellectual entities from the Preservation repository.
#============================================================================
[DPS]
DPSLinkLabel=Rosetta Link:
DPSLinkAddress=http://somecomponent.com/content-aggregator/
          getIEs?system=ilsdb&
DPSLinkParm=id=



[PrimoExp]
# note that stanza.name can be used with -- on the command line
# argv(case insensitive) override
# ini(case insensitive) override
# env (all uppercase)
#
# these parameters provide defaults for this run of the job
# each of them can be overridden using a command-line parameter
#
# BibRangeBegin and BibRangeEnd are the beginning and ending bibliographic ID
          numbers to export.
```

Figure 2-3. PrimoExp-Selectivew.ini Example (Continued)

```
# -B overrides this parameter on the command line
# BibRangeBegin
#BibRangeBegin=1
# -E overrides this parameter on the command line
#BibRangeEnd=200
#
# BibsFromFile reads the file at the supplied path for a list of bibliographic
          record ids to export.
# The file should contain one bibliographic record ID per line.
# -F overrides this parameter on the command line
BibsFromFile=/m1/voyager/2007.1.0/qa710db/local/PrimoBibs.txt
#
# ChangedSince exports records changed since the supplied date.
# ChangedSince may be either
#     a timestamp in YYYYMMDD.HHMMSS format
#    or a path to a file containing a timestamp in YYYYMMDD.HHMMSS format on a
          single line
# If a path to a file is supplied, the current run time will be saved in the
          file when
# the job finishes.
# -C overrides this parameter on the command line
# Example:
# ChangedSince=YYYYMMDD.HHMMSS
#  or
# ChangedSince=/m1/voyager/2007.1.0/qa710db/primo/Availability/
          ChangedSince.txt
#
#
#
# HeadingChanges determines whether to export bibliographic records
# whose cross-references have changed
# -H Y|N overrides this parameter on the command line
HeadingChanges=N
#
# AvailFilter detemines whether to export records whose availability
# fields have not changed since the last time the job ran.
# If AvailFilter is Y, only records whose availability has changed are
          exported;
# if AvailFilter is N, all changed records are exported.
```

Figure 2-3. PrimoExp-Selectivew.ini Example (Continued)

```
# -A Y|N overrides this parameter on the command line
AvailFilter=N
#
# for filenames
# note that $str$ will be passed to date format so that you can
# specify date in the these files
#
# LogFile is the location of the log file.
# Note that $str$ will be passed to Java's Date.Format()
# function to control the datestamp embedded in the file name.
# -L overrides this parameter on the command line
LogFile=primo.export.$yyMMdd$.log
LogFileDir=/m1/voyager/2007.1.0/qa710db/primo/Selective/logs
#
#
# -v
# the logging level
LogLevel=10
# LogToStdOut determines whether or not to write the log output to standard
          output,
# as well as the log file.
LogToStdOut=N


# Database and the other Database parameters determine which database to
          connect to.
# Ordinarily these are set by the wrapper script based on the environment.
# -d
# DataBase=
#
# DatabaseHost
DatabaseHost=10.100.2.32
#
# DatabasePort
# -e
# UserPass
# The credentials needed to connect to the database.
# Optional parameter, override on command line with -u
# STRONG RECOMMENDATION: do not set this value here, use
```

Figure 2-3. PrimoExp-Selectivew.ini Example (Continued)

```
# the command line override instead.
# Special note: Oracle TNS Alias (@DB) is _not_ used here
#UserPass=qa710db/qwrite0
#
Database=VGER
Protocol=com.endinfosys.voyager.extract.OAIPMHXmlWriterProtocol

# must do items, then mfhds, then bibs
Task=com.endinfosys.voyager.extract.ExtractItemsPrimo
Task=com.endinfosys.voyager.extract.ExtractMFHDsPrimo
Task=com.endinfosys.voyager.extract.ExtractBibsPrimo
DoItems=Y
DoMfhds=Y

# The Del* Dir and File are used to determine when records are deleted.
# These are standard Voyager locations.
DelBibsDir=/m1/voyager/2007.1.0/qa710db/rpt
DelBibFile=deleted.bib.marc
DelMFHDsDir=/m1/voyager/2007.1.0/qa710db/rpt
DelMFHDsFile=deleted.mfhd.marc
DelItemsDir=/m1/voyager/2007.1.0/qa710db/rpt
DelItemsFile=delete.item

# Set to Y to include course reserve data.
CourseReserves=N

[XMLWriterProtocol]
# OAIPMHXmlWriterProtocol will include the bib id
#  as the last element in the file name before
#  the .xml extension.
# Note the path must be coordinated with the bundlePrimo.ksh
#  script execution.
File=/m1/voyager/2007.1.0/qa710db/primo/Selective/exports/
          primo.export.$yyMMddhhmmss$.xml
XSL=
# max number of records to group in a single tar file
#  all in single group if 0 or undefined
recsPerGroup=1000
```

Figure 2-3. PrimoExp-Selectivew.ini Example (Continued)

```
# prefix for record identifiers, must be present, may be empty
idPrefix=
#
# Set to Y to enable pretty printing of XML output.
# Caution: may affect data with embedded spaces
UsePrettyPrint=N


#
# EVERYTHING BELOW THIS LINE IS INTERNAL DO NOT MODIFY
#
```

Figure 2-3. PrimoExp-Selectivew.ini Example (Continued)

Stanza Definitions for Voyager Primo Integration Configuration Files

This section defines the configuration file stanzas that need to be customized for Voyager Primo integration.

The following is a list of the stanzas.

- [Item Statuses]
- [Excluded Happening Locations]
- [Excluded Operator Ids]
- [Institution Id]
- [Location to Primo]
- [Enrichment Tags]
- [DPS]
- [PrimoExp]
- [XMLWriterProtocol]

[Item Statuses]

The `[Item Statuses]` stanza identifies which item statuses in Voyager are defined as Available (A) or Unavailable (U) for Primo's purposes. A change in status causes a record to be selected for the availability pipe.

See Table 2-1 for a list of the item statuses and default settings stored in the `[Item Statuses]` stanza of the configuration file.

Table 2-1. [Item Statuses] parameters

Parameter	Default Setting
Not Charged=	A
Charged=	U
Renewed=	U
Overdue=	U
Recall Request=	U
Hold Request=	U
On Hold=	U
In Transit=	U
In Transit Discharged=	U
In Transit On Hold=	U
Discharged=	A
Missing=	U
Lost--Library Applied=	U
Lost--System Applied=	U
Claims Returned=	U
Damaged=	U
Withdrawn=	U
At Bindery=	U
Cataloging Review=	A
Circulation Review=	A
Scheduled=	U
In Process=	U
Call Slip Request=	U
Short Loan Request=	U
Remote Storage Request=	U

CAUTION:
Changes to these parameters after your institution has an established

ongoing update/upload process may cause a significant data upload to the Primo database utilizing system resources as needed to complete the process.

[Excluded Happening Locations]

The `[Excluded Happening Locations]` stanza is used to identify the records associated with these locations to be excluded from consideration for uploading to the Primo database. These are location codes as established in Voyager System Administration.

The locations identified in the default configuration file are simply there as an example. Delete any that do not apply to your institution and enter your own location codes. Enter one location code per line.

[Excluded Operator Ids]

The `[Excluded Operator Ids]` stanza is used to identify the records associated with these operator ID(s) to be excluded from consideration for uploading to the Primo database. Enter one operator ID per line.

[Institution Id]

The `[Institution Id]` stanza is used to specify your institution. The institution ID is included as a subfield in the Availability tag of records prepared for the Primo database and is used by Primo to determine ownership of the records.

The Voyager institution ID is set one time at system installation. Verify this setting to insure its accuracy.

[Location to Primo]

The `[Location to Primo]` stanza is used to maintain consistency within Primo for item location codes. Primo is capable of retrieving records from many sources which may not be consistent in naming item location codes. The `[Location to Primo]` stanza can also be used by libraries to collapse locations or split locations for display in Primo.

Enter the Voyager item location code first and the Primo equivalent using the following format.

[Voyager Location Name]=[Primo Location Name]

See the following for an example.

```
Music Reserve=Reserve

Science Reserve=Reserve
```

Each Voyager/Primo item location code entry should be on a separate line in the configuration file.

NOTE:
Locations that are not listed in the `[Location to Primo]` stanza pass to Primo unchanged.

[Enrichment Tags]

The `[Enrichment Tags]` stanza is used to specify the tags used to include the availability, author enrichment, subject enrichment, course reserves, and Rosetta information in the extract.

By default, the configuration file makes the following assignments:

```
InsertAVATag=949
InsertAUTTag=950
InsertSUBTag=951
InsertCRTag=952
InsertDPSTag=953
```

The 949 code identifies the placement of availability information in the MARCXML output.

The 950 code identifies the placement of name/author cross-reference information.

The 951 code identifies the placement of subject cross-reference information.

The 952 code is provided for course reserves data extraction. See InsertCRTag=952 Enrichment Tag on page 4-3 for more information.

The 953 code identifies when a Voyager bibliographic record is stored in Rosetta as in the following example:

```
953‡a Exists in Rosetta
```

The defaults in the `[Enrichment Tags]` stanza assume that your institution is not already using 949, 950, 951, 952, and 953 in the MARC records of your database.

If, for example, your institution currently uses MARC 949 for another purpose, you can specify a different, unused 9XX number for the `InsertAVATag` in the `[Enrichment Tags]` stanza. The same logic applies to the other Enrichment Tags. Changing the default requires a matching change in the Primo configuration for harvesting.

[DPS]

The `[DPS]` stanza is used to identify the link to Rosetta information when it exists.

See Table 2-2 for a description of the parameters used to construct the 856 tag with Rosetta link information.

Table 2-2. [DPS] parameter descriptions

Parameter	Description
DPSLinkLabel	Use this parameter to specify the label text. The default text is as follows: `Rosetta Link:`
DPSLinkAddress	Use this parameter to identify the link address for Rosetta information. The following is an example of the link format: `http://somecomponent.com/content-aggregator/getIEs?system=ilsdb&`
DPSLinkParm	Use this to specify the link parameter. The default is as follows: `id=`

[PrimoExp]

The `[PrimoExp]` stanza is used to identify database extract processing considerations such as number of records to process, where to log messages, whether to use filters and so on.

The parameters identified in the `[PrimoExp]` stanza may be entered on the command line or set within the configuration file. Parameters specified on the command line override any settings within the configuration file.

See Table 2-3 for a description of these parameters.

Table 2-3. [PrimoExp] parameter/command descriptions

Parameter/Command	Description
BibRangeBegin= -B	Use this parameter to specify the bibliographic record ID number for the beginning of a range of bibliographic record IDs. Use this with -E. Used in combination with -E, you are able to process a subset of records from the entire database. **IMPORTANT:** *HeadingChanges must be set equal to N when BibRangeBegin is used.*
BibRangeEnd= -E	Use this parameter to specify the bibliographic record ID number for the end of a range of bibliographic record IDs. Use this with -B. Used in combination with -B, you are able to process a subset of records from the entire database. **IMPORTANT:** *HeadingChanges must be set equal to N when BibRangeEnd is used.*
BibsFromFile= -F	Use this parameter to identify the name of a file that contains bibliographic records to be processed/stored in a file that is separate from the database. This assumes that the file is formatted with one bibliographic record ID per line/row.

Table 2-3. [PrimoExp] parameter/command descriptions

Parameter/Command	Description
ChangedSince= -C	Use this parameter to indicate which Voyager records to evaluate for export to Primo based on a date/timestamp format as follows. YYYYMMDDHHMMSS where ... YYYY=year MM=Month DD=Day HH=Hours MM=Minutes SS=Seconds If no time is specified, 12:00AM is the default setting. Alternatively, store a date such as ***20070711*** and time in a file called since.txt with a path of your choosing and reference the since.txt file in [PrimoExp]. For example: ChangedSince=./test/work/since.txt **NOTE:** The system logs the following warnings/errors with the use of the ChangedSince= parameter: • If the ChangedSince= date is more than 7 days in the past, a warning is logged. • If the ChangedSince= date is more than 30 days in the past, an error message is logged; but the extract runs. • If the ChangedSince= date is more than 180 days in the past, the extract logs an error and stops.

Table 2-3. [PrimoExp] parameter/command descriptions

Parameter/Command	Description
HeadingChanges= -H	Use this parameter to indicate the following. • Yes (Y), find changed headings and include all records affected. • No (N), do not look for changed headings. This parameter should be set to Y when the configuration file is used to harvest records for the Primo publishing pipe. This parameter should be set to N when the configuration file is used to harvest records for the Primo availability pipe. **IMPORTANT:** *The HeadingChanges parameter requires that a setting is specified for the ChangedSince parameter. HeadingChanges can only be process if there is a ChangedSince date.*
AvailFilter= -A	Use this parameter to indicate which Voyager records to evaluate for export to Primo based on availability changes. Y=Yes (filter for availability changes) N=No (do not filter for availability changes) This parameter should be set to Y when the configuration file is used to harvest records for the Primo availability pipe. This parameter should be set to N when the configuration file is used to harvest records for the Primo publishing pipe.

LogFile=
-L
	Use this parameter to set the file name for the log file.
	For example:
	`primo.export.$yyMMdd$.log`
	The system automatically updates the `$yyMMdd$` with the date/time component of the file name.
	To specify that no log file is to be created, use the following parameter setting.
	`LogFile=/dev/null`
	TIP:
	Use a naming convention that reflects whether the configuration file is being used for Voyager records to be exported to the Primo publishing pipe or the Primo availability pipe.

Table 2-3. [PrimoExp] parameter/command descriptions

Parameter/Command	Description
LogFileDir=	Use this parameter to set the path for the log file.
LogLevel= -v	Leave the `LogLevel` parameter setting as specified in the configuration file unless directed otherwise by Ex Libris support staff.
LogToStdOut=	Leave the `LogToStdOut` parameter setting as specified in the configuration file unless directed otherwise by Ex Libris support staff.
Database= -d	Use this parameter to set the Oracle database name. The default is **`VGER`**. The `-d` command is used to combine the Database, DatabaseHost, and DatabasePort parameters into a single string. For example: `-d 208.178.237.40:1521@VGER`
DatabaseHost= -d	Use this parameter to set the database host IP address (the same as in `voyager.env`). The `DatabaseHost` parameter is set one time at system installation. Verify this setting to insure its accuracy. The `-d` command is used to combine the Database, DatabaseHost, and DatabasePort parameters into a single string. For example: `-d 208.178.237.40:1521@VGER`
DatabasePort= -d	Use this parameter to set the database port. The `-d` command is used to combine the Database, DatabaseHost, and DatabasePort parameters into a single string. For example: `-d 208.178.237.40:1521@VGER`
UserPass= -e	Use this parameter to specify the Oracle database user name and password. For example: `-e user/pass`
DelBibsDir=	Edit the parameter provided in the configuration file to match your Voyager configuration.
DelBibFile=	Leave this parameter setting as specified in the configuration file.

Table 2-3. [PrimoExp] parameter/command descriptions

Parameter/Command	Description
DelMFHDsDir=	Edit the parameter provided in the configuration file to match your Voyager configuration.
DelMFHDsFile=	Leave this parameter setting as specified in the configuration file.
DelItemsDir=	Edit the parameter provided in the configuration file to match your Voyager configuration.
DelItemsFile=	Leave this parameter setting as specified in the configuration file.

[XMLWriterProtocol]

The `[XMLWriterProtocol]` stanza defines the storage location of the XML records, number of records in the tar.gz file, and so forth for the XML records targeted for export to the Primo database.

See Table 2-4 for a description of the parameters used in the `[XMLWriterProtocol]` stanza.

Table 2-4. [XMLWriterProtocol] parameter descriptions

Parameter	Description

File=	Use this parameter to identify the file path for storing the XML files generated from harvesting Voyager records for export to the Primo database.
	
	  **TIP:**
	Use a naming convention that reflects whether the configuration file is being used for Voyager records to be exported to the Primo publishing pipe or the Primo availability pipe.
	
	For example:
	`./VPrimoExport/Pub/primo.export.$yyMMddhhmmss$.xml`
	`./VPrimoExport/Avail/primo.export.$yyMMddhhmmss$.xml`
	
	NOTE:
	The path you specify needs to be coordinated with the execution of the `bundlePrimo.ksh` script. The `bundlePrimo.ksh` script wraps the individual records into the tar.gz files that are accessed by the Primo publishing and availability pipes for import.

XSL=	Leave this parameter blank. No setting required.

Table 2-4. [XMLWriterProtocol] parameter descriptions

Parameter	Description
recsPerGroup=	Use this parameter to identify the number of records you prefer to be saved in each tar.gz file. Check to verify that this setting is consistent with the Primo limit.
idPrefix=	Leave this parameter blank. No setting required.
UsePrettyPrint=	Use this parameter, set to `Y` (Yes), for more readable XML files. **NOTE:** Using the pretty print `Y` option risks altering data with leading and trailing spaces. Use the `N` (No) option for machine-readable files when human review is not a requirement. The `N` (No) option is the default.

Pprimoexp Batch Job

The `Pprimoexp` batch job creates XML files for export that conform to the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) schema and places them in the location identified by the ***`File=`*** parameter set in the `[XMLWriterProtocol]` stanza. See Table 2-4.

The `Pprimoexp` batch job is located in `/m1/voyager/yyydb/sbin` where `yyydb` is the database name.

`Pprimoexp` runs the `PbundlePrimo` script that handles, for example, bundling individual records into `tar.gz` files and cleans up the exports directory.

RequiredBib Command Line Option (-R)

When running the `Pprimoexp` batch job, you may optionally specify one or more Voyager bibliographic record IDs (similar to a selective extraction).

The format is as follows:

1. For a single ID number where ### is the bibliographic record ID number.

   `-R #####`

2. For multiple ID numbers where ### is the bibliographic record ID number commas need to separate the ID numbers.

   `-R #####,#####,#####,#####`

Scheduling

The Voyager Primo integration setup is designed to give you the flexibility to establish more than one production schedule.

- One production job may be set to run once per day to allow for Primo harvesting of Voyager data with bibliographic record changes through the publishing pipe.

  IMPORTANT:
  This requires an update to the bibliographic index in the Primo environment.

- Another production schedule may be set to run several times per day such as once per hour to allow for exporting Voyager holdings records with availability changes.

  NOTE:
  Voyager availability data harvested through the Primo availability pipe only updates the availability information without changing the bibliographic information in Primo.

The `Pprimoexp` batch job schedule determines when Voyager records are examined for export based on the criteria set in the configuration file.

Separately, the Primo publishing and availability pipes are scheduled to FTP whatever `tar.gz` files are available for export. For more information, refer to Schedule Tasks in the *Primo Administrator Guide*.

```
# Configuration for Voyager Export for Primo -- Selective

# Voyager status maps to (A)vailable or (U)navailable for Primo
[Item Statuses]
Not Charged=A
Charged=U
Renewed=U
Overdue=U
Recall Request=U
Hold Request=U
On Hold=U
In Transit=U
In Transit Discharged=U
In Transit On Hold=U
Discharged=A
Missing=U
Lost--Library Applied=U
Lost--System Applied=U
Claims Returned=U
Damaged=U
Withdrawn=U
At Bindery=U
Cataloging Review=A
Circulation Review=A
Scheduled=U
In Process=U
Call Slip Request=U
Short Loan Request=U
Remote Storage Request=U

[Excluded Happening Locations]
# Items whose most recent activity took place in one of these
        locations will not be exported.

[Excluded Operator Ids]
```

```
# Items whose most recent activity was recorded by one of these
        operators will not be exported.


[Institution Id]
# This value will be included in the Availability tag
# as subfield a
Id=QAVOY


[Location to Primo]
# This section maps Voyager locations to Primo locations.


[Enrichment Tags]
# tags added to marc record with additional data
# AVA tag contains the availability information
# AUT tag contains name cross-reference information
# SUB tag contains subject cross-reference information
# CR tag contains course reserve information
# DPS tag contains Digital Preservation System (Product
        Rosetta) information
InsertAVATag=949
InsertAUTTag=950
InsertSUBTag=951
InsertCRTag=952
InsertDPSTag=953


#============================================================
        ==========
# Preservation related object link definition
# This includes the label for the URL link, the URL link, and
        parameter name.
# The only supported dynamic parameter at this time is id (i.e.
        bib id).
# The parameter is passed to the third-party application that
        processes and retrieves
# intellectual entities from the Preservation repository.
#============================================================
        ==========
```

```
[DPS]
DPSLinkLabel=Rosetta Link:
DPSLinkAddress=http://somecomponent.com/content-aggregator/
        getIEs?system=ilsdb&
DPSLinkParm=id=



[PrimoExp]
# note that stanza.name can be used with -- on the command line
# argv(case insensitive) override
# ini(case insensitive) override
# env (all uppercase)
#
# these parameters provide defaults for this run of the job
# each of them can be overridden using a command-line parameter
#
# BibRangeBegin and BibRangeEnd are the beginning and ending
        bibliographic ID numbers to export.
# -B overrides this parameter on the command line
# BibRangeBegin
#BibRangeBegin=1
# -E overrides this parameter on the command line
#BibRangeEnd=200
#
# BibsFromFile reads the file at the supplied path for a list of
        bibliographic record ids to export.
# The file should contain one bibliographic record ID per line.
# -F overrides this parameter on the command line
BibsFromFile=/m1/voyager/2007.1.0/qa710db/local/PrimoBibs.txt
#
# ChangedSince exports records changed since the supplied date.
# ChangedSince may be either
#     a timestamp in YYYYMMDD.HHMMSS format
#     or a path to a file containing a timestamp in
        YYYYMMDD.HHMMSS format on a single line
# If a path to a file is supplied, the current run time will be
        saved in the file when
```

```
# the job finishes.
# -C overrides this parameter on the command line
# Example:
# ChangedSince=YYYYMMDD.HHMMSS
#  or
# ChangedSince=/m1/voyager/2007.1.0/qa710db/primo/Availability/
        ChangedSince.txt
#
#
#
# HeadingChanges determines whether to export bibliographic
        records
# whose cross-references have changed
# -H Y|N overrides this parameter on the command line
HeadingChanges=N
#
# AvailFilter detemines whether to export records whose
        availability
# fields have not changed since the last time the job ran.
# If AvailFilter is Y, only records whose availability has
        changed are exported;
# if AvailFilter is N, all changed records are exported.
# -A Y|N overrides this parameter on the command line
AvailFilter=N
#
# for filenames
# note that $str$ will be passed to date format so that you can
# specify date in the these files
#
# LogFile is the location of the log file.
# Note that $str$ will be passed to Java's Date.Format()
# function to control the datestamp embedded in the file name.
# -L overrides this parameter on the command line
LogFile=primo.export.$yyMMdd$.log
LogFileDir=/m1/voyager/2007.1.0/qa710db/primo/Selective/logs
#
#
```

```
# -v
# the logging level
LogLevel=10
# LogToStdOut determines whether or not to write the log output
        to standard output,
# as well as the log file.
LogToStdOut=N


# Database and the other Database parameters determine which
        database to connect to.
# Ordinarily these are set by the wrapper script based on the
        environment.
# -d
# DataBase=
#
# DatabaseHost
DatabaseHost=10.100.2.32
#
# DatabasePort
# -e
# UserPass
# The credentials needed to connect to the database.
# Optional parameter, override on command line with -u
# STRONG RECOMMENDATION: do not set this value here, use
# the command line override instead.
# Special note: Oracle TNS Alias (@DB) is _not_ used here
#UserPass=qa710db/qwrite0
#
Database=VGER
Protocol=com.endinfosys.voyager.extract.OAIPMHXmlWriterProtocol


# must do items, then mfhds, then bibs
Task=com.endinfosys.voyager.extract.ExtractItemsPrimo
Task=com.endinfosys.voyager.extract.ExtractMFHDsPrimo
Task=com.endinfosys.voyager.extract.ExtractBibsPrimo
DoItems=Y
```

```
DoMfhds=Y

# The Del* Dir and File are used to determine when records are
        deleted.
# These are standard Voyager locations.
DelBibsDir=/m1/voyager/2007.1.0/qa710db/rpt
DelBibFile=deleted.bib.marc
DelMFHDsDir=/m1/voyager/2007.1.0/qa710db/rpt
DelMFHDsFile=deleted.mfhd.marc
DelItemsDir=/m1/voyager/2007.1.0/qa710db/rpt
DelItemsFile=delete.item

# Set to Y to include course reserve data.
CourseReserves=N

[XMLWriterProtocol]
# OAIPMHXmlWriterProtocol will include the bib id
#  as the last element in the file name before
#  the .xml extension.
# Note the path must be coordinated with the bundlePrimo.ksh
#  script execution.
File=/m1/voyager/2007.1.0/qa710db/primo/Selective/exports/
        primo.export.$yyMMddhhmmss$.xml
XSL=
# max number of records to group in a single tar file
#  all in single group if 0 or undefined
recsPerGroup=1000
# prefix for record identifiers, must be present, may be empty
idPrefix=
#
# Set to Y to enable pretty printing of XML output.
# Caution: may affect data with embedded spaces
```

```
UsePrettyPrint=N


#
# EVERYTHING BELOW THIS LINE IS INTERNAL DO NOT MODIFY
#
```

Patron Services

3

Patron Services

3

Introduction

The following patron services are provided by Voyager Primo integration.

- Patron authentication with Patron Directory Services.
- Patron information with Primo Library Card.
- Patron requests with Primo GetIt!.

Patron Authentication (Patron Directory Services)

Consistent with other Ex Libris products, Primo utilizes Patron Directory Services (PDS) to process patron authentication.

PDS does not have a patron database of its own. Instead, it provides the flexibility to pass authentication credentials to a designated target or redirect the user to an external authentication page.

The Voyager Primo environment provides the following PDS authentication functions against the Voyager patron database.

- Patron authentication using credentials sent from PDS.
- Borrower information sent to PDS.
- PDS Single Sign On (SSO) support for Primo.

Configuration

There are two server configuration components needed to enable user authentication with PDS.

1. The main PDS configuration table called tab_service.<institute> needs to be configured for communicating with the Voyager web service application.

   More information about PDS configuration is available in the *Patron Directory Services* manual.

2. The Voyager server XML needs to be configured with database connection information such as username, password, host, SID (system ID), and so forth. This can be set up once and is handled by the Ex Libris Voyager installation team.

PDS Invoked

Primo invokes PDS when users attempt to sign in. See Figure 3-1.

Figure 3-1. Primo Sign In / PDS Authentication

If a patron has not signed in prior to selecting **Get Avail Requests** (see Figure 3-2), clicking this option displays the Login dialog box. See Figure 3-3.

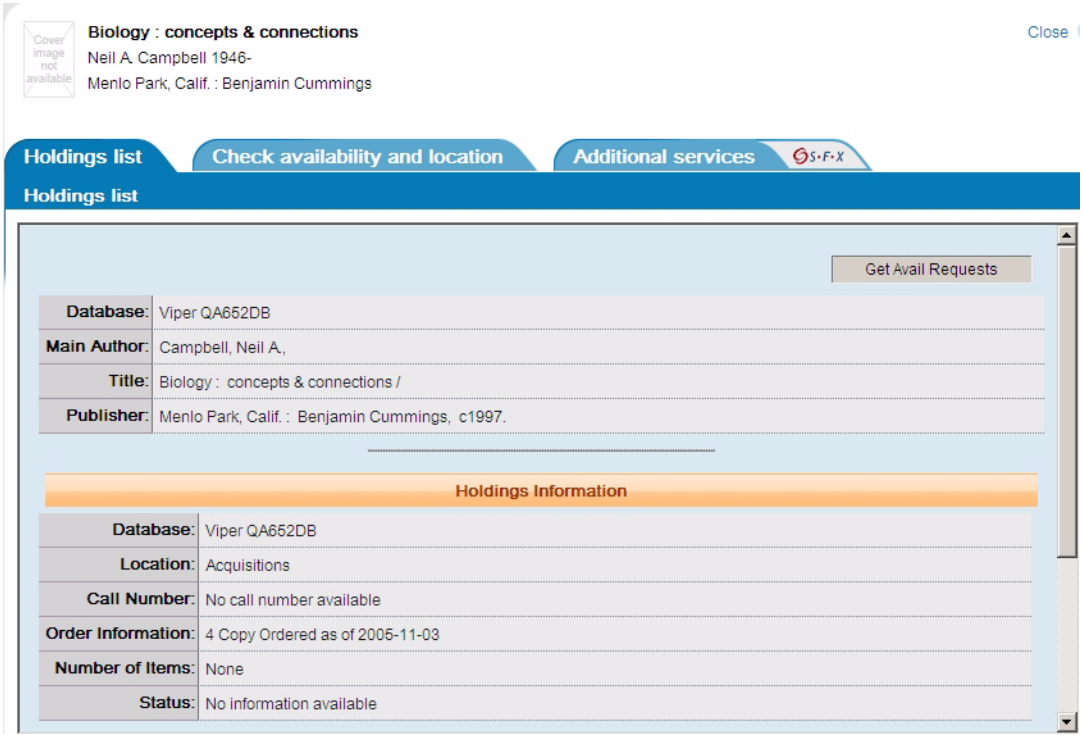

Figure 3-2. Get Avail Requests (prompting sign in)

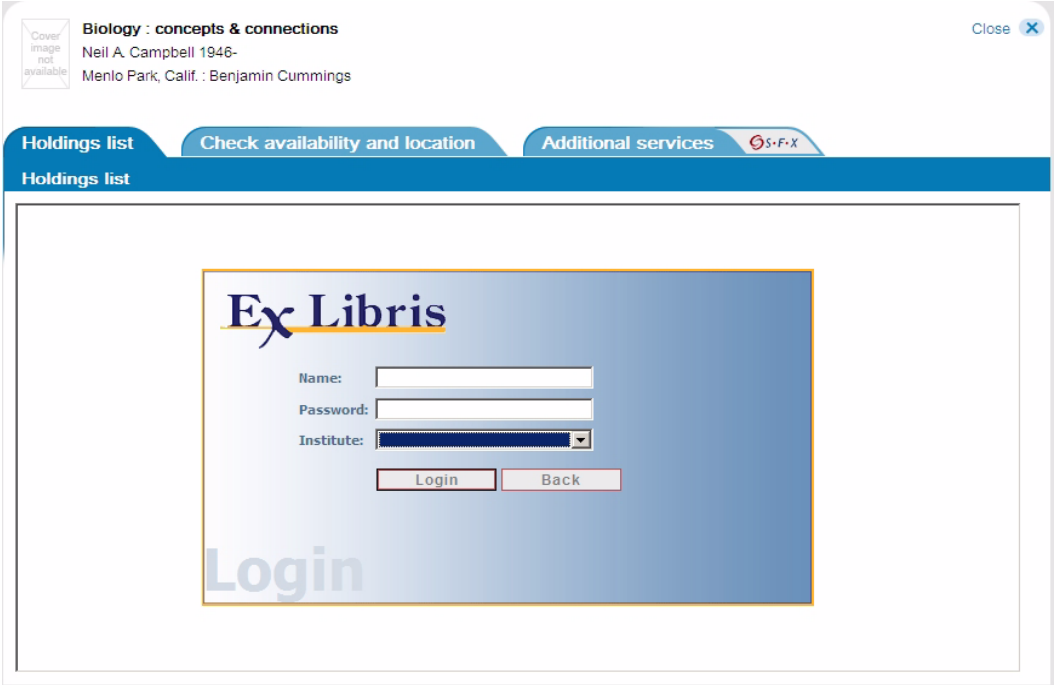

Figure 3-3. Login dialog box

Patron Information (Primo Library Card)

In the Voyager Primo environment, the Primo Library Card utilizes a deep link to Voyager that provides patrons with the following.

- Display of patron information.
- Option to renew charged items.
- Option to cancel requests.

This functionality is accessed through the My Library Card link in Primo.

Patron Requests (GetIt!)

When the patron identifies a record of interest in Primo (from the Voyager database), clicking the GetIt! link displays the **Holdings List** tab in Primo.

The **Holdings List** tab display contains the Voyager bibliographic, holdings, and item information. There is also the **Select a Request** drop-down list (see Figure 3-4) providing the patron with a number of request options if the patron has signed in.

If the patron has not signed in, **Get Avail Requests** displays. See PDS Invoked on page 3-2 for the sign in and PDS authentication steps.

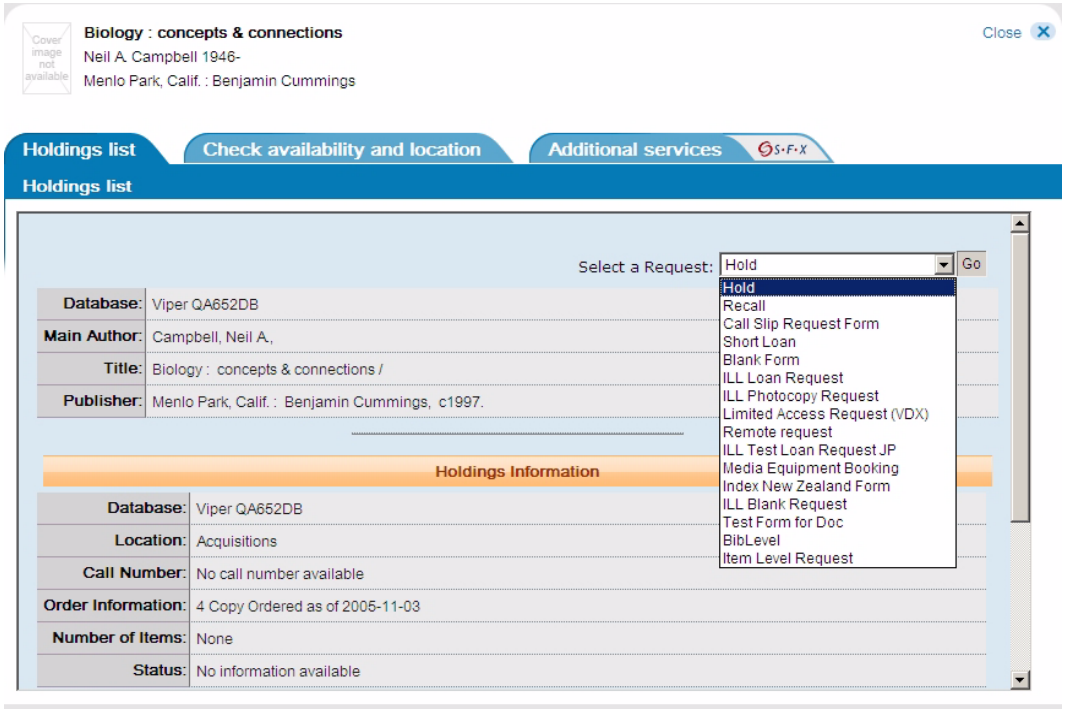

Figure 3-4. Select A Request options

The GetIt! link is configured by the Primo system administrator with the necessary linking information to the Voyager system. See the *Primo Administrator Guide* for details regarding how to configure GetIt!. This provides a deep link that enables the patron to confirm availability and process requests.

NOTE:
Requests in Voyager are only processed after PDS has successfully processed patron authentication.

Course Reserves

4

Course Reserves

4

Introduction

The purpose of this chapter is to describe the characteristics of the course reserves data extraction capability and any Voyager configuration changes required to enable this capability.

Course Reserves Function Overview

With the addition of course reserves data to Voyager bibliographic record extraction, the user discovery experience within Primo is extended. In Primo, facets and search scopes can be created to allow course reserve searching for Voyager customers.

With the course reserves option enabled, extracted Voyager bibliographic records are enriched with course reserve information such as course number, course name, instructor name/title, effective date, expiration date, department name/code, and so on. See Table 4-1 on page 4-3 for the extraction format and description.

NOTE:
Voyager course reserves e-items are not available for extract.

To enable course reserves data extraction, the following is provided in the `PrimoExp<xxx>.ini` configuration file where `xxx` may be `Selective`, `Availability`, or `Publishing`:

- `CourseReserves=`.
- 952 tag in the `[Enrichment Tags]` stanza.

CourseReserves=

In order to specify that Voyager course reserves data should be included in the data extraction process, the `CourseReserves=` flag is provided in the `.ini` configuration file. See Figure 4-1.

```
# Set to Y to include course reserve data.
CourseReserves=N
```

Figure 4-1. CourseReserves= flag

Specify `Y` (Yes) for course reserves data to be evaluated for data extraction. The default specified in the configuration file is `N` (No).

When `Y` is specified for `CourseReserves=`, the value in the Voyager database `ITEM.on_reserve` flag is examined to determine if course reserves data is to be included in the bibliographic record extraction or excluded.

NOTE:
The standard Voyager procedures for keeping the course reserve flag current in the Voyager database must be followed for Primo results to be current, too. See Chapter 8 in the *Voyager Technical User's Guide* for more information regarding circulation batch job 34 (Place Items on Active Course Reserve List) and 36 (Take Items on Inactive Course Reserve List Off Reserve) that are the batch jobs used to maintain currency for course reserves.

If the `ITEM.on_reserve` flag is `Y` (Yes) for one of the bibliographic record's item records, the bibliographic record being generated for extraction includes course reserve data for every list to which the item is linked (when no `ChangeSince=` value has been specified). When there is a `ChangeSince=` value specified, an additional step is taken to evaluate the `RESERVE_ITEM_HISTORY` table to determine if course reserve data is to be extracted. See Pprimoexp Extract Processing on page 4-4 and Figures 4-3 and 4-4 for additional information regarding extract processing.

NOTE:
Item records can be linked to course reserves lists without being on reserve. This allows you to reuse lists without needing to recreate them every semester or term.

Thus, the `ITEM.on_reserve` flag is the authoritative source for determining that an item is on reserve.

InsertCRTag=952 Enrichment Tag

For course reserves data extraction, the `InsertCRTag=952` is provided in the `[Enrichment Tags]` stanza of the `.ini` configuration file. See Figure 4-2.

```
[Enrichment Tags]
# tags added to marc record with additional data
InsertAVATag=949
InsertAUTTag=950
InsertSUBTag=951
InsertCRTag=952
InsertDPSTag=953
```

Figure 4-2. 952 tag

The `[Enrichment Tags]` stanza assumes that your institution is not already using the 952 in the MARC records of your database. If, however, your institution currently uses MARC 952 for another purpose, you can specify a different, unused MARC field number for the `InsertCRTag`. See [Enrichment Tags] on page 2-26 for additional information.

When the course reserves extraction function is enabled, the `InsertCRTag=` option specifies the data extract format. See Table 4-1 for a description of the extracted course reserves subfield data format.

Table 4-1. 952 Extract

952 Subfields	Extract Description
952a	Reserve list name
952b	List effective date
952c	List expiration date
952d	List location display name
952e	Department name
952f	Department code

Table 4-1. 952 Extract

952 Subfields	Extract Description
952g	Instructor last name
952h	Instructor first name
952i	Instructor title
952j	Course name
952k	Course number
952l	Section

IMPORTANT:
Even though you may substitute a different unused MARC field number for 952, the subfield designations in Table 4-1 cannot be changed.

Some subfields may not be included in the extracted data. This is dependent on the course reserves data available in Voyager.

An item record may belong to multiple reserve lists. A 952 field is constructed for each reserve list to which the item belongs.

Pprimoexp Extract Processing

When the `Pprimoexp` batch job is run for data extraction, it checks for the `CourseReserves=` flag to determine if course reserves data extraction should be part of the extraction results. Other parameters set in the batch job may affect the extraction processing end results.

In general, specifying parameters such as `ChangedSince=` or `BibRangeBegin=` and `BibRangeEnd=` indicates that a subset of the entire Voyager database is being processed for extraction.

Specific to course reserves when `ChangedSince=` is specified in the extraction script, the system does a check of the `RESERVE_ITEM_HISTORY` table to determine if one of the bibliographic record's item records has an entry in the table with a date that falls after the `ChangeSince=` value specified before going on to check the `ITEM.on_reserve` value.

See Figure 4-3 on page 4-5 and Figure 4-4 on page 4-6 for illustrations of the extract process workflow.

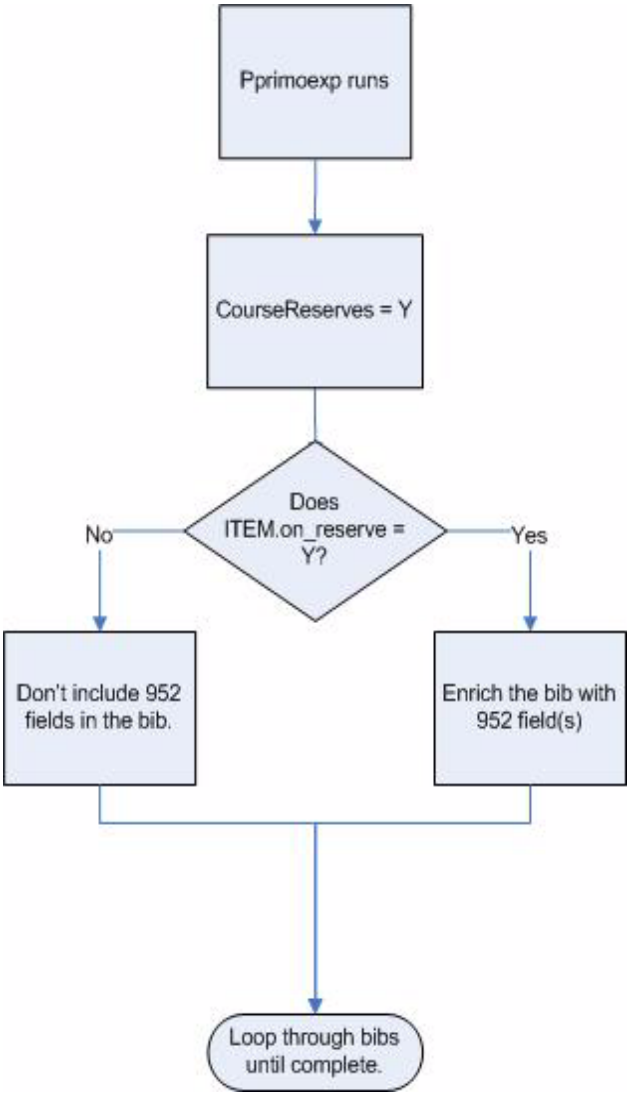

Figure 4-3. Extract processing without ChangedSince= specified

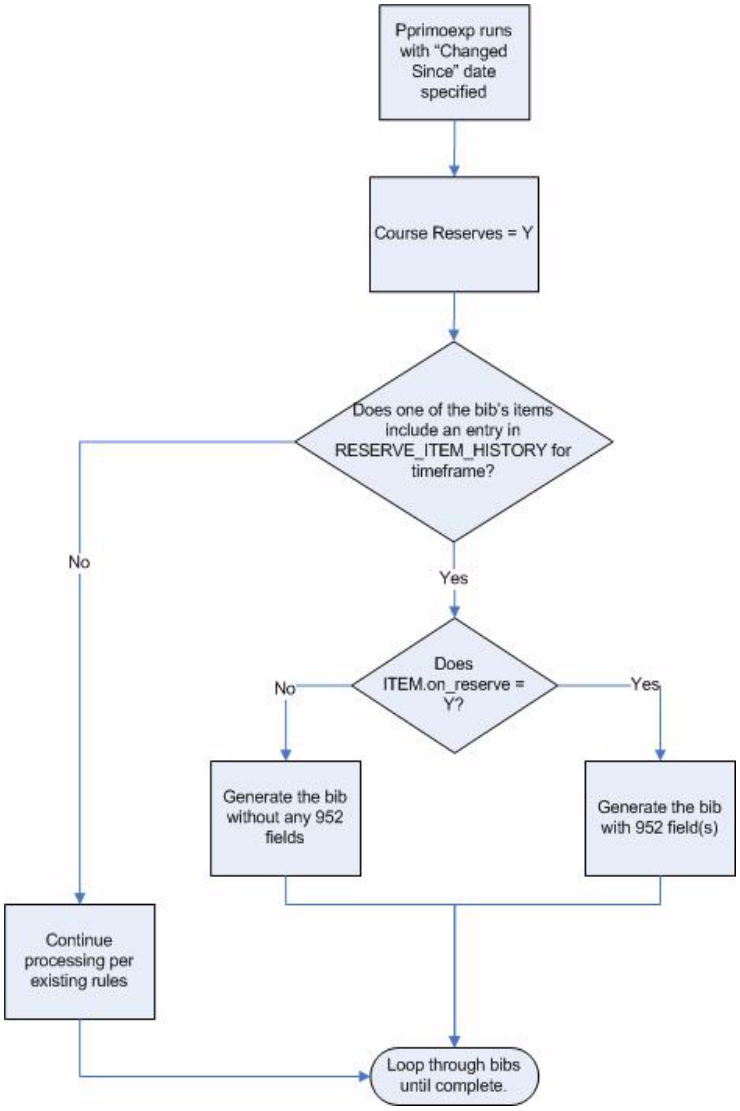

Figure 4-4. Extract processing with ChangedSince= specified

Real Time Availability

5

Real Time Availability

5

Introduction

The Real Time Availability function in Primo provides dynamically-generated Voyager database availability information to the Primo user. Specifically, Primo communicates with Voyager to retrieve availability information for one or more bibliographic records that can be viewed on an individual record display or on a search results page with multiple records.

This chapter provides information regarding real time availability processing with Voyager as a defined ILS in Primo and highlights other considerations like possible error conditions that you may encounter.

For information regarding Primo setup to trigger real time availability communications with Voyager, refer to the *Primo Interoperability Guide*.

Real Time Availability Processing

In general, the Primo/Voyager real time availability processing occurs as follows:

1. Primo sends a request (URL format) to the Voyager ILS. This request includes the following:
    a. An `op` (operation) parameter that must be `publish_avail`.
    b. A `doc_num` parameter with a value of one or more bibliographic record ID numbers in a comma-separated list. The number of IDs is only limited by the practical limit of an http URI.

2. The Voyager availability service (VXWS) processes the request.

   Real time availability processing utilizes the `PrimoExp.ini` configuration file when responding to real time availability queries from Primo. This file is located in `/m1/voyager/yyydb/tomcat/vxws/ini/` where `yyydb` is the database name.

   See PrimoExp.ini on page 5-5 for additional information.

3. Successful processing returns bibliographic record information from the Voyager database in XML format using the Enrichment Tags (see [Enrichment Tags] on page 2-26) to determine the fields/data organization.

See Figure 5-1 for an example of the XML returned from a successfully processed request.

```
<?xml version="1.0" encoding="UTF-8" ?>
 <publish-avail>
 <OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.openarchives.org/OAI/
        2.0/ http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
 <ListRecords>
 <record>
 <header>
 <identifier> 000010000</identifier>
 </header>
 <metadata>
 <record xmlns="http://www.loc.gov/MARC21/slim"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.loc.gov/MARC21/slim
        http://www.loc.gov/standards/marcxml/schema/
        MARC21slim.xsd">
 </metadata>
 </record>
 </ListRecords>
 </publish-avail>
```

Figure 5-1. Example real time availability XML returned to Primo

The value in the `<identifier>` key is the Voyager bibliographic record ID number.

Error Conditions

Error conditions have been identified for the following:

- Missing bibliographic record ID number (`doc_num`) in the URL request.

  See Figure 5-2 for an example of the error response provided.

- URL request that contains an operation parameter other than `publish_avail`.

  See Figure 5-3 for an example of the error response provided.

- Requested bibliographic record ID number does not exist in the Voyager database.

  See Figure 5-4 for an example of the error response provided.

```
<?xml version = "1.0" encoding = "UTF-8"?>
<publish-avail>
<error>doc_num must be included in parameters</error>
</publish-avail>
```

Figure 5-2. Example error response for missing bibliographic record ID number

```
<?xml version = "1.0" encoding = "UTF-8"?>
<login>
<error>Unrecognized op</error>
</login>
```

Figure 5-3. Example error response for incorrect operation parameter

```
<OAI-PMH xmlns=http://www.openarchives.org/OAI/2.0/
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.openarchives.org/OAI/
        2.0/ http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
<ListRecords>
<record>
<header>
<identifier> 999910001</identifier>
</header>
<error>
<error_code>21</error_code>
<error_message>Error reading document</error_message>
</error>
</record> </ListRecords> </OAI-PMH>
```

Figure 5-4. Example error response for record ID that doesn't exist

web.xml

Values in `web.xml` are typically set one time during installation setup. Specfic to real time availability, this file identifies the path for the the `PrimoExp.ini` file that is used in the process. You may want to check to confirm that it is currently accurate.

See Figure 5-5 for an example of `web.xml` content. Any reference to `yyydb` in the example indicates where you would specify your database name.

```
<servlet>
    <servlet-name>VoyagerAvailabilityService</servlet-name>
    <display-name>Voyager Availability Web Service</
     display-name>
    <servlet-
     class>com.endinfosys.voyager.websvc.VoyagerAvailabilit
     yService</servlet-class>
    <init-param>
        <param-name>PrimoExpIniFile</param-name>
        <param-value>
            /m1/voyager/yyydb/ini/PrimoExp.ini
        </param-value>
    </init-param>
</servlet>
```

Figure 5-5. web.xml example

PrimoExp.ini

The real time availability process uses settings below the line ("EVERYTHING BELOW THIS LINE IS INTERNAL DO NOT MODIFY") in the `PrimoExp.ini` file to generate a successful response to Primo.

IMPORTANT:
Do not make changes to this section of the PrimoExp.ini.

The `UserPass=` option in the `[PrimoExp]` stanza must be specified for real time availability to work. This cannot be commented out. See Table 2-3 on page 2-28 for a description of `UserPass=`.

Universal Borrowing with Primo

6

Universal Borrowing with Primo

6

Feature Description

With Primo versions 2.1.3 and higher and Voyager versions 7.1 and higher, support is provided for Universal Borrowing (UB) capability to patrons using Primo as their user interface for discovery. The Universal Borrowing function enables patrons from a library within a consortium to place direct requests on materials from other libraries within the consortium.

Primo provides patrons the following access points for placing UB requests from the discovery display:

- GetIt! link
- Item in the catalog link
- Available location link

Regardless of the access method used, Universal Borrowing policies, patron blocks, and delivery rules are enforced per the policies of the Voyager institution. Also options like Supply Default Pickup Location are also supported.

Process Overview

With the Universal Borrowing with Primo feature, Primo and Voyager work together to provide the patron access to Universal Borrowing resources when the home Voyager database does not have the requested holdings. See Figure 6-1 for a high-level illustration of the process.

Refer to the other sections in this chapter for a more-detail description of the setup required to implement this process.

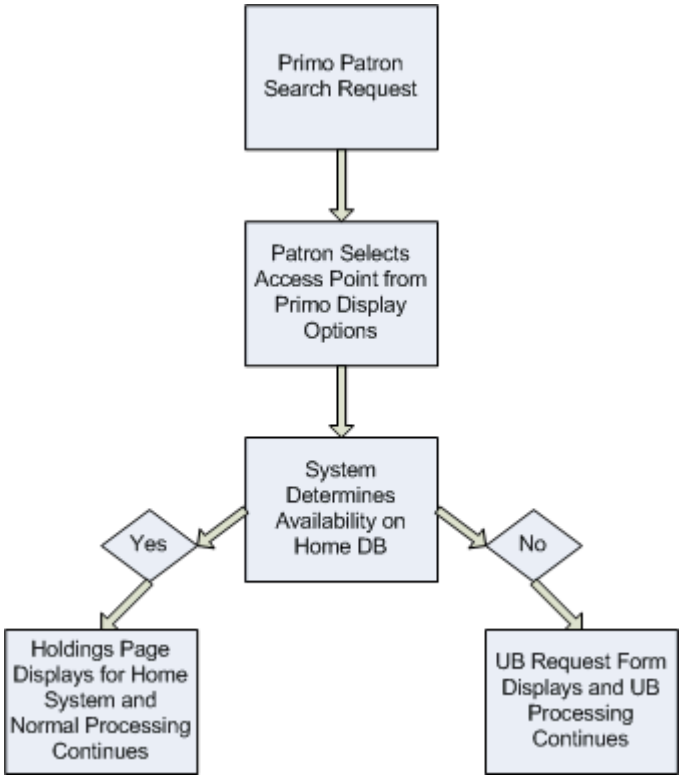

Figure 6-1. Voyager Primo UB process overview

Enabling Universal Borrowing with Primo

The primary setup components for enabling Universal Borrowing with Primo are as follows:

- Configuration files on the Voyager server. Refer to the following sections for setup details:
  - vprimo.properties on page 6-3.
  - requests.properties on page 6-8.
  - messages.properties on page 6-9.

- Additional Voyager database for mapping UB requests from Primo. See Primo UB Connector (PUC) Database on page 6-10 for details regarding setup for this database.
- VPDS setup in Voyager System Administration (Search > Database Definitions). See the *Voyager System Administration User's Guide* for more information regarding database definitions setup.

For information regarding setup on the Primo server for configuration files and so on, refer to the *Primo Interoperability Guide*.

vprimo.properties

The `vprimo.properties` file located in `/m1/voyager/xxxdb/tomcat/vprimo/context/vprimo/ini` (where `xxxdb` is your database name) provides the following UB settings:

- `UBProcessing`
- `TurnOffLibraryDropDown`
- User-defined database key

See an example of this file in Figure 6-2.

The `vprimo.properties` file also contains the holdings status codes for the VPRIMO display status string (not to be modified).

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties version="1.0">
    <comment>definitions for VPRIMO configuration</comment>
    <!-- set this option to Y to enable UB processing, if Voyager has UB      -->
    <entry key="UBProcessing">N</entry>
    <!-- Option to show only one randomly selected library for the item.      -->
    <!-- Default is N to show all libraries that would have the item.         -->
    <entry key="TurnOffLibraryDropDown">N</entry>
    <!--                                                                      -->
    <!-- User defined DB key to the DB code used to retrieve holdings records -->
    <!--                                                                      -->
    <entry key="connectDb.YYYDB">LOCAL</entry>
```

Figure 6-2. Example of vprimo.properties configuration file

```
    <!-- additional example:
    <entry key="connectDb.SomeOtherLibrary">otherdb</entry>
    -->
    <!--                                                                    -->
    <!-- Map "institute" values as returned by PDS to DB_CODE as found in the -->
    <!-- VOYAGER_DATABASES table.                                           -->
    <!--                                                                    -->
    <entry key="institute.YYYDB">LOCAL</entry>
    <!-- additional example:
    <entry key="institute.SomeOtherLibrary">otherdb</entry>
    -->
    <!--                                                                    -->
    <!-- Holdings page message                                              -->
   <entry key="holdings.message.noAvailableItems">There are no circulating items available
          to satisfy your Borrowing Request</entry>
    <!--                                                                    -->
    <!-- Holdings status code to VPRIMO display status string. Please do not  -->
    <!-- modify because the code is defined in the database.                -->
    <!--                                                                    -->
    <entry key="holdings.item.status.1">Not Charged</entry>
    <entry key="holdings.item.status.2">Charged</entry>
    <entry key="holdings.item.status.3">Renewed</entry>
    <entry key="holdings.item.status.4">Overdue</entry>
    <entry key="holdings.item.status.5">Recall Request</entry>
    <entry key="holdings.item.status.6">Hold Request</entry>
    <entry key="holdings.item.status.7">On Hold</entry>
    <entry key="holdings.item.status.8">In Transit</entry>
    <entry key="holdings.item.status.9">In Transit Discharged</entry>
    <entry key="holdings.item.status.10">In Transit On Hold</entry>
    <entry key="holdings.item.status.11">Discharged</entry>
    <entry key="holdings.item.status.12">Missing</entry>
    <entry key="holdings.item.status.13">Lost--Library Applied</entry>
    <entry key="holdings.item.status.14">Lost--System Applied</entry>
    <entry key="holdings.item.status.15">Claims Returned</entry>
    <entry key="holdings.item.status.16">Damaged</entry>
    <entry key="holdings.item.status.17">Withdrawn</entry>
    <entry key="holdings.item.status.18">At Bindery</entry>
    <entry key="holdings.item.status.19">Cataloging Review</entry>
```

Figure 6-2. Example of vprimo.properties configuration file (Continued)

```
    <entry key="holdings.item.status.20">Circulation Review</entry>
    <entry key="holdings.item.status.21">Scheduled</entry>
    <entry key="holdings.item.status.22">In Process</entry>
    <entry key="holdings.item.status.23">Call Slip Request</entry>
    <entry key="holdings.item.status.24">Short Loan Request</entry>
    <entry key="holdings.item.status.25">Remote Storage Request</entry>
</properties>
```

Figure 6-2. Example of vprimo.properties configuration file (Continued)

UBProcessing

Use the `UBProcessing` parameter to turn on the Universal Borrowing with Primo feature on the Voyager server. Set this parameter to `Y` (Yes) to turn it on. The default setting is `N` (No).

TurnOffLibraryDropDown

By default, the system displays all libraries in the consortium that have the item. This is a function of the `TurnOffLibraryDropDown` parameter that has a default setting of `N` (No).

To have the system show only one randomly-selected library for the item, set the `TurnOffLibraryDropDown` parameter to `Y` (Yes).

User-Defined Database Key

The databases (UB libraries) participating in Universal Borrowing with Primo need to be identified in the `vprimo.properties` configuration file. See Figure 6-3 for an example of the format used to define these databases.

```
    <!-- User defined DB key to the DB code used to retrieve holdings records -->
    <!--                                                                      -->
    <entry key="connectDb.YYYdb">LOCAL</entry>
    <entry key="connectDb.qadma710db">DMA71DB</entry>
    <entry key="connectDb.qagwcc710db">GWCC71DB</entry>
    <entry key="connectDb.qarscc710db">RSCC71DB</entry>
```

Figure 6-3. Example connectDb database definitions

Using `<entry key=”connectDb.gadma710db”>DMA71DB</entry>` from the Figure 6-3 example:

`qadma710db` represents the database name.

`DMA71DB` represents the database code as defined in Voyager System Administration. See Figure 6-4 and Figure 6-5.

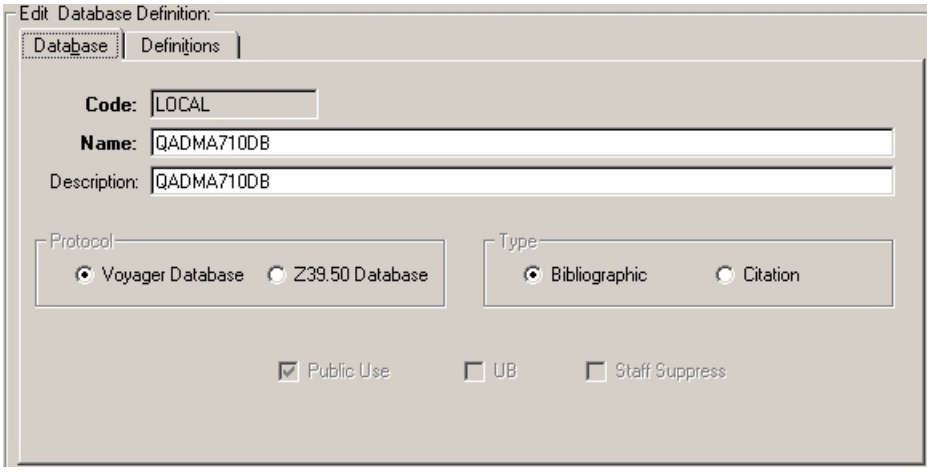

Figure 6-4. Voyager System Administration DB definition - Database tab

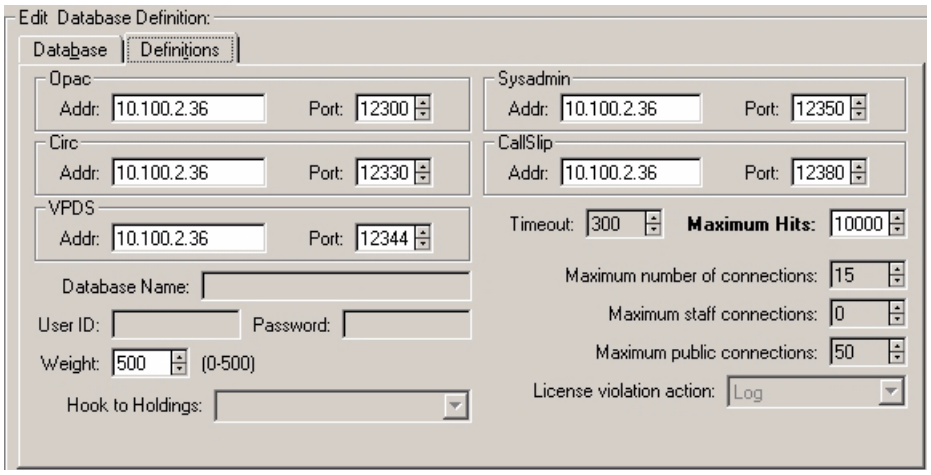

Figure 6-5. Voyager System Administration DB definition - Definitions tab

There needs to be one entry for each UB database participating in Universal Borrowing with Primo.

If `qadma710db` is defined with the LOCAL database code in Voyager System Administration, the entry would be as follows:

```
<entry key="connectDb.gadma710db">LOCAL</entry>
```

NOTE:
There is no LOCAL entry in the PUC `vprimo.properties` configuration file. See Primo UB Connector (PUC) Database on page 6-10 for more information.

Institute Mapping

The institute mapping in the `vprimo.properties` configuration file is used for patron authentication.

Each UB database participating in Universal Borrowing with Primo needs to have an entry in the `vprimo.properties` configuration file where the Primo PDS (Patron Directory Services) code is mapped to the Voyager System Administration's database code. See Figure 6-6 for an example.

NOTE:
The institute definitions are case sensitive and need to be all uppercase.

The institute referenced in this section is the same as the institute identified in the [Institute Id] stanza discussed in [Institution Id] on page 2-25. See Figure 2-1 on page 2-6 for an example of this stanza. The institution ID is stored in Primo in the 949‡a of the Primo PNX.

```
<!-- Map "institute" values as returned by PDS to DB_CODE as found in the -->
<!-- VOYAGER_DATABASES table.                                              -->
<!--                                                                       -->
<entry key="institute.YYYDB">LOCAL</entry>
<entry key="institute.QADMA710DB">DMA71DB</entry>
<entry key="institute.QAGWCC710DB">GWCC71DB</entry>
<entry key="institute.QARSCC710DB">RSCC71DB</entry>
```

Figure 6-6. Example institute definitions

NOTE:
There is no LOCAL entry in the PUC `vprimo.properties` configuration file for institute definitions. See Primo UB Connector (PUC) Database on page 6-10 for more information.

requests.properties

The `requests.properties` file located in `/m1/voyager/xxxdb/tomcat/vxws/ini` (where `xxxdb` is your database name) provides the following UB settings:

- `SupplyDefaultPickupLocs`
- `SelectPickupLocsMsg`

See an example of this file in Figure 6-7.

```
<!-- the next 2 properties are for universal borrowing requests -->
<entry key="SupplyDefaultPickupLocs">Y</entry>
<entry key="SelectPickupLocsMsg">You must provide a Pick Up Library and Desk before your
          request is complete!</entry>
```

Figure 6-7. Example of UB parameters in request.properties

SupplyDefaultPickupLocs

Use the `SupplyDefaultPickupLocs` to indicate that you want a default pickup location to be displayed for the patron. The default is `Y` (Yes) to turn on this option. Specify `N` (No) to turn it off.

When this key is set to `Y`, the default pickup location is set to the first pickup location in the drop-down list.

When this key is set to `N`, the pickup locations are listed alphabetically in the drop-down list and no default is selected.

SelectPickupLocsMsg

Use the `SelectPickupLocsMsg` key to define the message that displays when no pickup location has been specified by the patron. The message shown in Figure 6-7 is the default that is provided.

messages.properties

The `messages.properties` file located in `/m1/voyager/xxxdb/tomcat/vprimo/context/vprimo/ini` (where `xxxdb` is your database name) provides the various block or error messages that may be returned by the Voyager Primo web services. See Figure 6-8 for an example of stored messages.

These messages are preset and do not require any additional customization. Use the `messages.properties` file as your reference for patron blocked messages.

```
<properties version="1.0">

            <comment>This properties file defines messages to be displayed for various
          block or error messages that might be returned by the Voyager-Primo web
          services.</comment>

            <!-- patron blocked message -->

            <entry key="Requests.patronblocked">You have no requests available in this
          database.</entry>

            <!-- generic request blocked messages -->

            <entry key="Requests.requestblocked-4">There are no circulating items
          available to satisfy your UB Borrowing Request.</entry>

            <entry key="Requests.requestblocked-3">No items are available for Remote
          Storage requests.</entry>

            <entry key="Requests.requestblocked-2">No items are available for CallSlip
          requests.</entry>
```

Figure 6-8. Example from message.properties file

```
    <entry key="Requests.requestblocked-1">No items are available for Hold
requests.</entry>
    <entry key="Requests.requestblocked1">No holdings are available.</entry>
    <entry key="Requests.requestblocked2">The item is on order.</entry>
    <entry key="Requests.requestblocked3">The item is not charged.</entry>
    <entry key="Requests.requestblocked4">The item is missing.</entry>
    <entry key="Requests.requestblocked5">The item is lost.</entry>
    <entry key="Requests.requestblocked6">The item is at the bindery.</entry>
    <entry key="Requests.requestblocked7">No items are available for recall.</
entry>
    <entry key="Requests.requestblocked8">You have already placed a request for
this item.</entry>
    <entry key="Requests.requestblocked9">No recall policy is defined for this
item.</entry>
    <entry key="Requests.requestblocked10">No hold policy is defined for this
item.</entry>
    <entry key="Requests.requestblocked11">This item has been scheduled through
media booking.</entry>
    <entry key="Requests.requestblocked12">This item is not available for
hold.</entry>
```

Figure 6-8. Example from message.properties file (Continued)

Primo UB Connector (PUC) Database

Specific to Universal Borrowing with Primo is the addition of the PUC (Primo UB Connector) database. For each Universal Borrowing consortium, an additional Voyager database (PUC) is created for the purpose of routing Universal Borrowing requests. Key to this capability are the `connectDb` and `institiute` definitions in the `vprimo.properties` file for the PUC system.

Specifically, the PUC `vprimo.properties` file needs to define all the UB databases in the consortium and omit any reference to itself as the LOCAL database. See Figure 6-9 for an example.

```
<!-- User defined DB key to the DB code used to retrieve holdings records -->
    <!--                                                                    -->
    <entry key="connectDb.qadma710db">DMA71DB</entry>
    <entry key="connectDb.qagwcc710db">GWCC71DB</entry>
    <entry key="connectDb.qarscc710db">RSCC71DB</entry>
```

Figure 6-9. Example PUC vprimo.properties connectDB and institute definitions

```
<!--                                                                    -->
<!-- Map "institute" values as returned by PDS to DB_CODE as found in the -->
<!-- VOYAGER_DATABASES table.                                           -->
    <!--                                                                    -->
    <entry key="institute.QADMA710DB">DMA71DB</entry>
    <entry key="institute.QAGWCC710DB">GWCC71DB</entry>
    <entry key="institute.QARSCC710DB">RSCC71DB</entry>
```

Figure 6-9. Example PUC vprimo.properties connectDB and institute definitions (Continued)

Display Voyager Holdings

7

Display Voyager Holdings

7

Files

There are several files that may be tailored to customize the display of Voyager holdings in Primo. See Table 7-1 and Table 7-2 for a list, location, and description of these files.

Table 7-1. Files in /m1/voyager/yyydb/tomcat/vprimo/context/vprimo/xsl/html/default/

File Name	Description
`configs/104X_chronValues.xml`	This file is used to convert numeric values defined in MARC for month and season into chronological names. The element/attribute names in this file are arbitrary, but must match the XPath expression used to reference them from the translateChron template in holdingsDisplay.xsl. Additional languages may be added by creating new language elements and populating them with the appropriate chron elements: • lang id attribute = {language code from 008 field} • chron id attribute = {01-12 for months, 21-24 for seasons} • chron value = {chronological name to be displayed}
`configs/104X_config.xml`	This file is used to define serials format (NISO/Standard) punctuation, pairs of fields to display/not display, and captions.
`configs/104X_display.xsl`	This is the style sheet that pulls together compressed serials information from the 85X/86X fields.
`configs/displaycfg.xml`	This file is used to identify the display fields on a page.
`configs/displaycfg1.xml`	This file is used to identify the display fields on a page.
`configs/displayHoldings.xml`	This file is used to identify display fields such as database, location, call number, and order information for the holdings display record view.
`configs/requestBlockCodes.xsl`	This style sheet controls the display of a patron block message.
`configs/requestBlockMsgs.xsl`	This is the style sheet for request block code variables.

Table 7-1. Files in /m1/voyager/yyydb/tomcat/vprimo/context/vprimo/xsl/html/default/

File Name	Description
configs/shortLoanBlockCodes.xsl	This is the style sheet for request block code variables specific to short loans.
blocks.xsl	This style sheet controls building the block messages that display in MyAccount.
buildHtml.xsl	This style sheet is used to build the html page.
buildMarcDisplay.xsl	This style sheet controls the assembly of MARC records from Voyager (not Primo).
constants.xsl	This style sheet defines the location of style sheets, help, images, and Java script.
constantStrings.xsl	This style sheet manages constant text/labels for requests, copies, forms, UB requests, and so on.
currency.xsl	This style sheet handles the formatting of currency.
display.xsl	This is the style sheet that pulls everything together for display.
getAvailRequests.xsl	This style sheet controls getting available requests from the Voyager database.
marc21slim.xsl	This style sheet is used to build the MARC21 record.
mediaBooking.xsl	This style sheet controls the processing of the media booking request page.
messages.xsl	This style sheet defines a template called by multiple other files like sendRequest.
myAccount.xsl	This style sheet controls the top-level display of My Account page. This is linked to other my*.
myAccountCancel.xsl	This style sheet controls the My Account page when there is a cancel request.
myAccountChargedItems.xsl	This style sheet controls the display of charged items.
myAccountConstants.xsl	This style sheet manages the constants used for My Account page.

Table 7-1. Files in /m1/voyager/yyydb/tomcat/vprimo/context/vprimo/xsl/html/default/

File Name	Description
myAccountLinks.xsl	This style sheet controls the links for My Account, search preferences, personal information, and so on.
myAccountRenew.xsl	This style sheet controls the My Account display for charged items and so on.
myAccountRequests.xsl	This style sheet controls the display of requests.
personalInfo.xsl	This style sheet controls the display of the personal information page.
requestForm.xsl	This style sheet controls the processing of the request form.
sendRequest.xsl	This style sheet controls request processing.
shortLoanReqForm.xsl	This style sheet is used to build the short loan request form.
show-detail.xsl	This is a JavaScript debugging tool. **IMPORTANT:** *Do not edit.*
statusStrings.xsl	This style sheet is used to build the item status string.
UBReqForm.xsl	This style sheet controls building the UB (Universal Borrowing) request page.

Table 7-2. Files in /m1/voyager/yyydb/tomcat/vxws/context/vxws/xsl/

File Name	Description
configs/104X_chronValues.xml	This file is used to convert numeric values defined in MARC for month and season into chronological names. The element/attribute names in this file are arbitrary, but must match the XPath expression used to reference them from the translateChron template in holdingsDisplay.xsl. Additional languages may be added by creating new language elements and populating them with the appropriate chron elements: • lang id attribute = {language code from 008 field} • chron id attribute = {01-12 for months, 21-24 for seasons} • chron value = {chronological name to be displayed}
configs/104X_config.xml	This file is used to define format (NISO/ Standard) punctuation, pairs of fields to display/not display, and captions.
configs/104X_display.xsl	This is the style sheet that pulls together punctuation, caption prefixes, indicator filters, and so on.
configs/emailcfg.xml	This file is used to identify the display fields for e-mail.
configs/emailholdingscfg.xml	This file is used to identify the holdings display fields for e-mail.
configs/requestcfg.xml	This provides the XML for all the display tags such as author, title, subjects, description, publisher and so on.
configs/requestholdingscfg.xml	This provides the XML for the holdings details such as call number and shelving title.
biblevelrequest.xsl	This style sheet controls bibliographic-level requests.
constants.xsl	This style sheet controls cascading style sheets, help files, images, and javascripts.

Table 7-2. Files in /m1/voyager/yyydb/tomcat/vxws/context/vxws/xsl/

File Name	Description
`constantstrings.xsl`	This style sheet manages constant text/labels for requests, copies, forms, UB requests, and so on.
`emailbibrecord.xsl`	This style sheet handles e-mailing bibliographic record requests.
`marc21slim.xsl`	This style sheet is used to build the MARC21 record.

Index

Numerics

A

B

C

D

E

F

G

H

I

L

M

N

O

P

R

S

T

U

V

W

X