



**Voyager[®]
Primo[®] Integration User's Guide**

October 2008

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.

Copyright Ex Libris Limited, 2008. All rights reserved.

Document released: October 2008

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

Contents

About This Document

• Purpose	xi
• Intended Audience	xi
• Reason for Reissue	xii
• Document Summary	xii
• Conventions Used in This Document	xii
• Document Reproduction/Photocopying	xiv
• Comment on This Document	xiv
To Submit Comments by E-mail	xv

1 Getting Started

• Introduction	1-1
• Purpose of this Chapter	1-1
• Prerequisite Skills and Knowledge	1-1
• Before You Begin	1-2
Checklist	1-2
Port Settings	1-2

2 Data Extraction

• Introduction	2-1
• Purpose of this Chapter	2-1
• Data Extraction Process Overview	2-1
Initial Primo Data Load	2-2
Initial Load Generation - Excluded Records	2-2
Ongoing Updates to Primo Database	2-3
Pipes	2-3
Special Purpose/Selective Database Extractions	2-4
Primo Transaction / Record Normalization	2-4
• Configuration Files	2-5

Contents

Availability Configuration File	2-6
Publishing Configuration File	2-12
Selective Configuration File	2-18
• Stanza Definitions for Voyager Primo Integration Configuration Files	2-25
[Item Statuses]	2-25
[Excluded Happening Locations]	2-27
[Excluded Operator Ids]	2-27
[Institution Id]	2-27
[Location to Primo]	2-27
[Enrichment Tags]	2-28
[PrimoExp]	2-29
[XMLWriterProtocol]	2-33
• Pprimoexp Batch Job	2-34
RequiredBib Command Line Option (-R)	2-34
Scheduling	2-35

3 Patron Services

• Introduction	3-1
• Patron Authentication (Patron Directory Services)	3-1
Configuration	3-2
PDS Invoked	3-2
• Patron Information (Primo Library Card)	3-4
• Patron Requests (GetIt!)	3-4

4 Course Reserves

• Introduction	4-1
• Course Reserves Function Overview	4-1
• CourseReserves=	4-2
• InsertCRTag=952 Enrichment Tag	4-3
• Pprimoexp Extract Processing	4-4

Contents

5	Real Time Availability	
	• Introduction	5-1
	• Real Time Availability Processing	5-1
	• Error Conditions	5-3
	• web.xml	5-4
	• PrimoExp.ini	5-5

IN	Index	IN-1
-----------	--------------	-------------

Contents

Figures

2	Data Extraction	
2-1.	PrimoExp-Availability.ini Example	2-6
2-2.	PrimoExp-Publishing.ini Example	2-12
2-3.	PrimoExp-Selective.ini Example	2-19

3	Patron Services	
3-1.	Primo Sign In / PDS Authentication	3-2
3-2.	Get Avail Requests (prompting sign in)	3-3
3-3.	Login dialog box	3-4
3-4.	Select A Request options	3-5

4	Course Reserves	
4-1.	CourseReserves= flag	4-2
4-2.	952 tag	4-3
4-3.	Extract processing without ChangedSince= specified	4-5
4-4.	Extract processing with ChangedSince= specified	4-6

5	Real Time Availability	
5-1.	Example real time availability XML returned to Primo	5-2
5-2.	Example error response for missing bibliographic record ID number	5-3
5-3.	Example error response for incorrect operation parameter	5-3
5-4.	Example error response for record ID that doesn't exist	5-4
5-5.	web.xml example	5-5

Figures

Tables

2	Data Extraction	
	2-1. [Item Statuses] parameters	2-26
	2-2. [PrimoExp] parameter/command descriptions	2-29
	2-3. [XMLWriterProtocol] parameter descriptions	2-33

4	Course Reserves	
	4-1. 952 Extract	4-3

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:

- Changes to the About chapter.
- Addition of port setting notes in [Port Settings](#) on [page 1-2](#).
- Additions to the ChangedSince= description. See [Table 2-2](#) on [page 2-29](#).
- Addition of -R command line option. See [RequiredBib Command Line Option \(-R\)](#) on [page 2-34](#).
- Addition of UsePrettyPrint=. See [Table 2-3](#) on [page 2-33](#).
- Addition of course reserves functionality. See Chapter [4](#).
- Addition of real time availability function. See Chapter [5](#).

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.
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

Introduction	1-1
Purpose of this Chapter	1-1
Prerequisite Skills and Knowledge	1-1
Before You Begin	1-2
• Checklist	1-2
• Port Settings	1-2

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.

Introduction	2-1
Purpose of this Chapter	2-1
Data Extraction Process Overview	2-1
• Initial Primo Data Load	2-2
Initial Load Generation - Excluded Records	2-2
• Ongoing Updates to Primo Database	2-3
Pipes	2-3
• Special Purpose/Selective Database Extractions	2-4
• Primo Transaction / Record Normalization	2-4
Configuration Files	2-5
• Availability Configuration File	2-6
• Publishing Configuration File	2-12
• Selective Configuration File	2-18
Stanza Definitions for Voyager Primo Integration	
Configuration Files	2-25
• [Item Statuses]	2-25
• [Excluded Happening Locations]	2-27
• [Excluded Operator Ids]	2-27
• [Institution Id]	2-27
• [Location to Primo]	2-27
• [Enrichment Tags]	2-28
• [PrimoExp]	2-29

• [XMLWriterProtocol]	2-33
Pprimoexp Batch Job	2-34
• RequiredBib Command Line Option (-R)	2-34
• Scheduling	2-34

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 2-12](#).

- 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 2-25](#) 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 2-12](#) for more information.
- `PrimoExp-Availability.ini`. See [Availability Configuration File](#) on [page 2-6](#) for more information.

These are located in `/ml/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 2-25](#) 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 2-34](#) 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 `/ml/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.

- `/ml/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 2-12](#).
- `PrimoExp-Selective.ini`. [Selective Configuration File](#) on [page 2-18](#).

They are located in `/ml/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 2-27](#).
- Institution ID to be included in the availability tag.
- Location definitions for consistency within the Primo database. See [\[Location to Primo\]](#) on [page 2-27](#).
- Tags to identify unique field usage within your institution's database specific to the database extract. See [\[Enrichment Tags\]](#) on [page 2-28](#).
- Extract processing considerations such as date/time, record ID(s), filters, and so on. See [\[PrimoExp\]](#) on [page 2-29](#).
- Additional processing considerations for uploading the Voyager records identified by the configuration file processing. See [\[XMLWriterProtocol\]](#) on [page 2-33](#).

See [Stanza Definitions for Voyager Primo Integration Configuration Files](#) on [page 2-25](#) for a complete description of configuration file options.

Availability Configuration File

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

Line#	
1	# Configuration for Voyager Export for Primo Availability
2	
3	[Item Statuses]
4	# Voyager status maps to (A)available or (U)navailable for Primo
5	Not Charged=A
6	Charged=U
7	Renewed=U
8	Overdue=U
9	Recall Request=U
10	Hold Request=U
11	On Hold=U
12	In Transit=U
13	In Transit Discharged=U
14	In Transit On Hold=U
15	Discharged=A
16	Missing=U
17	Lost--Library Applied=U
18	Lost--System Applied=U
19	Claims Returned=U
20	Damaged=U
21	Withdrawn=U
22	At Bindery=U
23	Cataloging Review=A
24	Circulation Review=A
25	Scheduled=U
26	In Process=U

Figure 2-1. PrimoExp-Availability.ini Example

```
Line#
27 Call Slip Request=U
28 Short Loan Request=U
29 Remote Storage Request=U
30
31 [Excluded Happening Locations]
32 # Items whose most recent activity took place in one of
    these locations will not be exported.
33
34 [Excluded Operator Ids]
35 # Items whose most recent activity was recorded by one of
    these operators will not be exported.
36
37 [Institution Id]
38 # This value will be included in subfield a of each
    Availability field
39 Id=@DBNAME@
40
41 [Location to Primo]
42 # This section maps Voyager locations to Primo locations.
43
44
45 [Enrichment Tags]
46 # tags added to marc record with additional data
47 # AVA tag contains the availability information
48 # AUT tag contains name cross-reference information
49 # SUB tag contains subject cross-reference information
50 InsertAVATag=949
51 InsertAUTTag=950
52 InsertSUBTag=951
53
54
55 [PrimoExp]
```

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

```
Line#
56 # these parameters provide defaults for this run of the
    job
57 # each of them can be overridden using a command-line
    parameter
58
59 # note that stanza.name can be used with -- on the command
    line
60 # argv(case insensitive) override
61 # ini(case insensitive) override
62 # env (all uppercase)
63
64 # BibRangeBegin and BibRangeEnd are the beginning and
    ending bibliographic ID numbers to export.
65 # -B overrides this parameter on the command line
66 # BibRangeBegin
67 #BibRangeBegin=1
68 # -E overrides this parameter on the command line
69 #BibRangeEnd=200
70 #
71
72 # BibsFromFile reads the file at the supplied path for a
    list of bibliographic record ids to export.
73 # The file should contain one bibliographic record ID per
    line.
74 # -F overrides this parameter on the command line
75 #BibsFromFile=@VOYAGER@/@DATABASE@/local/PrimoBibs.txt
76 #
77
78 # ChangedSince exports records changed since the supplied
    date.
79 # ChangedSince may be either
80 #         a timestamp in YYYYMMDD.HHMMSS format
81 #         or a path to a file containing a timestamp in
    YYYYMMDD.HHMMSS format on a single line
```

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


```
Line#
82 # If a path to a file is supplied, the current run time
    # will be saved in the file when
83 # the job finishes.
84
85 # -C overrides this parameter on the command line
86 # changes since a date or date & time
87 #ChangedSince=YYYYMMDD.HHMMSS
88 # or
89 # changes since the last (saved) run
90 # current run time will be saved
91 ChangedSince=@VOYAGER@/@DATABASE@/primo/Availability/
    ChangedSince.txt
92 #
93 # HeadingChanges determines whether to export
    # bibliographic records whose cross-references have
    # changed
94 # -H Y|N overrides this parameter on the command line
95 HeadingChanges=N
96 #
97 # AvailFilter determines whether to export records whose
    # availability fields have not changed since the
    # last time
98 # the job ran. If AvailFilter is Y, only records whose
    # availability
99 # has changed are exported; if AvailFilter is N, all
    # changed records are exported.
100 # Whether to filter records to prevent sending records
    # that have
101 # had no change in their availability tags. Defaults to
    # N.
102 # -A Y|N overrides this parameter on the command line
103 AvailFilter=Y
104
105 # Logfile is the location of the logfile.
```

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

```
Line#
106 # Note that $str$ will be passed to java's Date.Format()
      function to control the datestamps embedded in
      the file name.
107
108 # -L overrides this parameter on the command line
109 LogFile=@VOYAGER@/@DATABASE@/primo/Availability/logs/
      primo.export.$yyMMdd$.log
110 #LogFile=/dev/null
111
112 # -v
113 # the logging level
114 LogLevel=10
115
116 # LogToStdOut determines whether or not to write the log
      output to standard output, as well as the log
      file.
117 # whether to duplicate log output to stdout
118 LogToStdOut=N
119
120 # Database and the other Database parameters determine
      which database to connect to.
121 # Ordinarily these are set by the wrapper script based on
      the environment.
122 # -d
123 # DataBase=
124 #
125 # DatabaseHost
126 DatabaseHost=@DBHOST@
127 #DatabaseHost=208.178.237.40
128 # DatabasePort
129 # -e
130 #UserPass=@DBUSER@/@DBPASS@
131 Database=VGER
```

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

```
Line#
132 Protocol=com.endinfosys.voyager.extract.OAIPMHxmlWriterPr
      otocol
133
134 # must do items, then mfhds, then bibs
135 Task=com.endinfosys.voyager.extract.ExtractItemsPrimo
136 Task=com.endinfosys.voyager.extract.ExtractMFHDSPrimo
137 Task=com.endinfosys.voyager.extract.ExtractBibsPrimo
138 DoItems=Y
139 DoMfhds=Y
140
141 # The Del* Dir and File are used to determine when records
      are deleted.
142 # These are standard Voyager locations.
143 DelBibsDir=@VOYAGER@/@DATABASE@/rpt
144 DelBibFile=deleted.bib.marc
145 DelMFHDSDir=@VOYAGER@/@DATABASE@/rpt
146 DelMFHDSFile=deleted.mfhd.marc
147 DelItemsDir=@VOYAGER@/@DATABASE@/rpt
148 DelItemsFile=delete.item
149
150 [XMLWriterProtocol]
151 # OAIPMHxmlWriterProtocol will include the bib id
152 # as the last element in the file name before
153 # the .xml extension.
154 # Note the path must be coordinated with the
      bundlePrimo.ksh
155 # script execution.
156 # Note that $str$ will be passed to java's Date.Format()
      function to control the timestamps embedded in
      the file name.
157
158 File=@VOYAGER@/@DATABASE@/primo/Availability/exports/
      primo.export.$yyMMddhhmmss$.xml
```

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

```
Line#
159 XSL=
160 # max number of records to group in a single tar file
161 # all in single group if 0 or undefined
162 recsPerGroup=1000
163 # prefix for record identifiers, must be present, may be
    empty
164 idPrefix=
165
166 #
167 # EVERYTHING BELOW THIS LINE IS INTERNAL DO NOT MODIFY
168 #
```

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

Publishing Configuration File

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

```
Line#
1 # Configuration for Voyager Export for Primo Publishing
2
3 [Item Statuses]
4 # Voyager status maps to (A)vailable or (U)navailable for
    Primo
5 Not Charged=A
6 Charged=U
7 Renewed=U
8 Overdue=U
9 Recall Request=U
10 Hold Request=U
11 On Hold=U
```

Figure 2-2. PrimoExp-Publishing.ini Example

```
Line#
12   In Transit=U
13   In Transit Discharged=U
14   In Transit On Hold=U
15   Discharged=A
16   Missing=U
17   Lost--Library Applied=U
18   Lost--System Applied=U
19   Claims Returned=U
20   Damaged=U
21   Withdrawn=U
22   At Bindery=U
23   Cataloging Review=A
24   Circulation Review=A
25   Scheduled=U
26   In Process=U
27   Call Slip Request=U
28   Short Loan Request=U
29   Remote Storage Request=U
30
31   [Excluded Happening Locations]
32   # Items whose most recent activity took place in one of
      these locations will not be exported.
33
34   [Excluded Operator Ids]
35   # Items whose most recent activity was recorded by one of
      these operators will not be exported.
36
37   [Institution Id]
38   # This value will be included in subfield a of each
      Availability field
39   Id=@DBNAME@
40
```

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

```
Line#
41 [Location to Primo]
42 # This section maps Voyager locations to Primo locations.
43
44
45 [Enrichment Tags]
46 # tags added to marc record with additional data
47 # AVA tag contains the availability information
48 # AUT tag contains name cross-reference information
49 # SUB tag contains subject cross-reference information
50 InsertAVATag=949
51 InsertAUTTag=950
52 InsertSUBTag=951
53
54
55 [PrimoExp]
56 # these parameters provide defaults for this run of the
57 #   job
58 # each of them can be overridden using a command-line
59 #   parameter
60
61 # note that stanza.name can be used with -- on the command
62 #   line
63 # argv(case insensitive) override
64 # ini(case insensitive) override
65 # env (all uppercase)
66
67 # BibRangeBegin and BibRangeEnd are the beginning and
68 #   ending bibliographic ID numbers to export.
69 # -B overrides this parameter on the command line
70 #BibRangeBegin=1
71 # -E overrides this parameter on the command line
72 #BibRangeEnd=200
73 #
```

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

```
Line#
70
71 # BibsFromFile reads the file at the supplied path for a
    # list of bibliographic record ids to export.
72 # The file should contain one bibliographic record ID per
    # line.
73 # -F overrides this parameter on the command line
74 #BibsFromFile=@VOYAGER@/@DATABASE@/local/PrimoBibs.txt
75 #
76
77 # ChangedSince exports records changed since the supplied
    # date.
78 # ChangedSince may be either
79 #     a timestamp in YYYYMMDD.HHMMSS format
80 #     or a path to a file containing a timestamp in
    #     YYYYMMDD.HHMMSS format on a single line
81 # If a path to a file is supplied, the current run time
    # will be saved in the file when
82 # the job finishes.
83
84 # -C overrides this parameter on the command line
85 # changes since a date or date & time
86 #ChangedSince=YYYYMMDD.HHMMSS
87 # or
88 # changes since the last (saved) run
89 # current run time will be saved
90 ChangedSince=@VOYAGER@/@DATABASE@/primo/Publishing/
    ChangedSince.txt
91 #
92 # HeadingChanges determines whether to export
    # bibliographic records whose cross-references have
    # changed
93 # -H Y|N overrides this parameter on the command line
94 HeadingChanges=Y
95 #
```

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

```
Line#
 96 # AvailFilter detemines whether to export records whose
    # availability fields have not changed since the
    # last time
 97 # the job ran. If AvailFilter is Y, only records whose
    # availability
 98 # has changed are exported; if AvailFilter is N, all
    # changed records are exported.
 99 # Whether to filter records to prevent sending records
    # that have
100 # had no change in their availability tags. Defaults to N.
101 # -A Y|N overrides this parameter on the command line
102 AvailFilter=N
103
104 # Logfile is the location of the logfile.
105 # Note that $str$ will be passed to java's Date.Format()
    # function to control the datestamps embedded in the
    # file name.
106
107 # -L overrides this parameter on the command line
108 LogFile=@VOYAGER@/@DATABASE@/primo/Publishing/logs/
    primo.export.$yyMMdd$.log
109 #LogFile=/dev/null
110
111 # -v
112 # the logging level
113 LogLevel=10
114
115 # LogToStdOut determines whether or not to write the log
    # output to standard output, as well as the log
    # file.
116 # whether to duplicate log output to stdout
117 LogToStdOut=N
118
119 # Database and the other Database parameters determine
    # which database to connect to.
```

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


```
Line#
120 # Ordinarily these are set by the wrapper script based on
    # the environment.
121 # -d
122 # DataBase=
123 #
124 # DatabaseHost
125 DatabaseHost=@DBHOST@
126 #DatabaseHost=208.178.237.40
127 # DatabasePort
128 # -e
129 #UserPass=@DBUSER@/@DBPASS@
130 Database=VGER
131 Protocol=com.endinfosys.voyager.extract.OAIPMHxmlWriterPr
    otocol
132
133 # must do items, then mfhds, then bibs
134 Task=com.endinfosys.voyager.extract.ExtractItemsPrimo
135 Task=com.endinfosys.voyager.extract.ExtractMFHDsPrimo
136 Task=com.endinfosys.voyager.extract.ExtractBibsPrimo
137 DoItems=Y
138 DoMfhds=Y
139
140 # The Del* Dir and File are used to determine when records
    are deleted.
141 # These are standard Voyager locations.
142 DelBibsDir=@VOYAGER@/@DATABASE@/rpt
143 DelBibFile=deleted.bib.marc
144 DelMFHDsDir=@VOYAGER@/@DATABASE@/rpt
145 DelMFHDsFile=deleted.mfhd.marc
146 DelItemsDir=@VOYAGER@/@DATABASE@/rpt
147 DelItemsFile=delete.item
148
```

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

```
Line#
149 [XMLWriterProtocol]
150 # OAIPMXmlWriterProtocol will include the bib id
151 # as the last element in the file name before
152 # the .xml extension.
153 # Note the path must be coordinated with the
      bundlePrimo.ksh
154 # script execution.
155 # Note that $str$ will be passed to java's Date.Format()
      function to control the datestamps embedded in the
      file name.
156
157 File=@VOYAGER@/@DATABASE@/primo/Publishing/exports/
      primo.export.$yyMMddhhmmss$.xml
158 XSL=
159 # max number of records to group in a single tar file
160 # all in single group if 0 or undefined
161 recsPerGroup=1000
162 # prefix for record identifiers, must be present, may be
      empty
163 idPrefix=
164
165 #
166 # EVERYTHING BELOW THIS LINE IS INTERNAL DO NOT MODIFY
167 #
```

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

Selective Configuration File

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

```
Line#
1      # Configuration for Voyager Export for Primo--Selective,
      with a bibliographic record file
2
3      [Item Statuses]
4      # Voyager status maps to (A)available or (U)navailable
      for Primo
5      Not Charged=A
6      Charged=U
7      Renewed=U
8      Overdue=U
9      Recall Request=U
10     Hold Request=U
11     On Hold=U
12     In Transit=U
13     In Transit Discharged=U
14     In Transit On Hold=U
15     Discharged=A
16     Missing=U
17     Lost--Library Applied=U
18     Lost--System Applied=U
19     Claims Returned=U
20     Damaged=U
21     Withdrawn=U
22     At Bindery=U
23     Cataloging Review=A
24     Circulation Review=A
25     Scheduled=U
26     In Process=U
27     Call Slip Request=U
28     Short Loan Request=U
29     Remote Storage Request=U
```

Figure 2-3. PrimoExp-Selective.ini Example

```
Line#
30
31 [Excluded Happening Locations]
32 # Items whose most recent activity took place in one of
   these locations will not be exported.
33
34 [Excluded Operator Ids]
35 # Items whose most recent activity was recorded by one
   of these operators will not be exported.
36
37 [Institution Id]
38 # This value will be included in subfield a of each
   Availability field
39 Id=@DBNAME@
40
41 [Location to Primo]
42 # This section maps Voyager locations to Primo
   locations.
43
44
45 [Enrichment Tags]
46 # tags added to marc record with additional data
47 # AVA tag contains the availability information
48 # AUT tag contains name cross-reference information
49 # SUB tag contains subject cross-reference information
50 InsertAVATag=949
51 InsertAUTTag=950
52 InsertSUBTag=951
53
54
55 [PrimoExp]
56 # these parameters provide defaults for this run of the
   job
```

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

```
Line#
57 # each of them can be overridden using a command-line
    parameter
58
59 # note that stanza.name can be used with -- on the
    command line
60 # argv(case insensitive) override
61 # ini(case insensitive) override
62 # env (all uppercase)
63
64 # BibRangeBegin and BibRangeEnd are the beginning and
    ending bibliographic ID numbers to export.
65 # -B overrides this parameter on the command line
66 #BibRangeBegin=1
67 # -E overrides this parameter on the command line
68 #BibRangeEnd=200
69 #
70
71 # BibsFromFile reads the file at the supplied path for a
    list of bibliographic record ids to export.
72 # The file should contain one bibliographic record ID
    per line.
73 # -F overrides this parameter on the command line
74 BibsFromFile=@VOYAGER@/@DATABASE@/local/PrimoBibs.txt
75 #
76
77 # ChangedSince exports records changed since the
    supplied date.
78 # ChangedSince may be either
79 #     a timestamp in YYYYMMDD.HHMMSS format
80 #     or a path to a file containing a timestamp in
    YYYYMMDD.HHMMSS format on a single line
81 # If a path to a file is supplied, the current run time
    will be saved in the file when
82 # the job finishes.
```

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

```
Line#
83
84 # -C overrides this parameter on the command line
85 # changes since a date or date & time
86 #ChangedSince=YYYYMMDD.HHMMSS
87 # or
88 # changes since the last (saved) run
89 # current run time will be saved
90 #ChangedSince=@VOYAGER@/@DATABASE@/primo/Selective/
    ChangedSince.txt
91 #
92 # HeadingChanges determines whether to export
    bibliographic records whose cross-references
    have changed
93 # -H Y|N overrides this parameter on the command line
94 HeadingChanges=N
95 #
96 # AvailFilter determines whether to export records whose
    availability fields have not changed since the
    last time
97 # the job ran. If AvailFilter is Y, only records whose
    availability
98 # has changed are exported; if AvailFilter is N, all
    changed records are exported.
99 # Whether to filter records to prevent sending records
    that have
100 # had no change in their availability tags. Defaults to
    N.
101 # -A Y|N overrides this parameter on the command line
102 AvailFilter=N
103 #
104
105 # Logfile is the location of the logfile.
106 # Note that $str$ will be passed to java's Date.Format()
    function to control the datestamps embedded in
    the file name.
```

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

```
Line#
107
108 # -L overrides this parameter on the command line
109 LogFile=@VOYAGER@/@DATABASE@/primo/Selective/logs/
    primo.export.$yyMMdd$.log
110 #LogFile=/dev/null
111
112 # -v
113 # the logging level
114 LogLevel=10
115
116 # LogToStdOut determines whether or not to write the log
    output to standard output, as well as the log
    file.
117 # whether to duplicate log output to stdout
118 LogToStdOut=N
119
120 # Database and the other Database parameters determine
    which database to connect to.
121 # Ordinarily these are set by the wrapper script based
    on the environment.
122 # -d
123 # DataBase=
124 #
125 # DatabaseHost
126 DatabaseHost=@DBHOST@
127 #DatabaseHost=208.178.237.40
128 # DatabasePort
129 # -e
130 #UserPass=@DBUSER@/@DBPASS@
131 Database=VGER
132 Protocol=com.endinfosys.voyager.extract.OAIPMXmlWriter
    Protocol
133
```

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

```
Line#
134 # must do items, then mfhds, then bibs
135 Task=com.endinfosys.voyager.extract.ExtractItemsPrimo
136 Task=com.endinfosys.voyager.extract.ExtractMFHDSPrimo
137 Task=com.endinfosys.voyager.extract.ExtractBibsPrimo
138 DoItems=Y
139 DoMfhds=Y
140
141 # The Del* Dir and File are used to determine when
      records are deleted.
142 # These are standard Voyager locations.
143 DelBibsDir=@VOYAGER@/@DATABASE@/rpt
144 DelBibFile=deleted.bib.marc
145 DelMFHDSDir=@VOYAGER@/@DATABASE@/rpt
146 DelMFHDSFile=deleted.mfhd.marc
147 DelItemsDir=@VOYAGER@/@DATABASE@/rpt
148 DelItemsFile=delete.item
149
150 [XMLWriterProtocol]
151 # OAIPMHXmlWriterProtocol will include the bib id
152 # as the last element in the file name before
153 # the .xml extension.
154 # Note the path must be coordinated with the
      bundlePrimo.ksh
155 # script execution.
156 # Note that $str$ will be passed to java's Date.Format()
      function to control the timestamps embedded in
      the file name.
157
158 File=@VOYAGER@/@DATABASE@/primo/Selective/exports/
      primo.export.$yyMMddhhmmss$.xml
159 XSL=
160 # max number of records to group in a single tar file
161 # all in single group if 0 or undefined
```

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


```
Line#
162  recsPerGroup=1000
163  # prefix for record identifiers, must be present, may be
      empty
164  idPrefix=
165
166  #
167  # EVERYTHING BELOW THIS LINE IS INTERNAL DO NOT MODIFY
168  #
```

Figure 2-3. PrimoExp-Selective.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]
- [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, and subject enrichment information in the extract.

By default, the configuration file makes the following assignments.

```
InsertAVATag=949
```

```
InsertAUTTag=950
```

```
InsertSUBTag=951
```

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 defaults in the [Enrichment Tags] stanza assume that your institution is not already using 949, 950, and 951 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 `InsertAUTTag` and the `InsertSUBTag` parameters.

Changing the default requires a matching change in the Primo configuration for harvesting.

[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 2-2](#) for a description of these parameters.

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

Parameter/Command	Description
BibRangeBegin= -B	<p>Use this parameter to specify the bibliographic record ID number for the beginning of a range of bibliographic record IDs.</p> <p>Use this with -E.</p> <p>Used in combination with -E, you are able to process a subset of records from the entire database.</p> <p>▲ IMPORTANT: <i>HeadingChanges must be set equal to N when BibRangeBegin is used.</i></p>
BibRangeEnd= -E	<p>Use this parameter to specify the bibliographic record ID number for the end of a range of bibliographic record IDs.</p> <p>Use this with -B.</p> <p>Used in combination with -B, you are able to process a subset of records from the entire database.</p> <p>▲ IMPORTANT: <i>HeadingChanges must be set equal to N when BibRangeEnd is used.</i></p>
BibsFromFile= -F	<p>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.</p>

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

Parameter/Command	Description
<p>ChangedSince= -C</p>	<p>Use this parameter to indicate which Voyager records to evaluate for export to Primo based on a date/timestamp format as follows.</p> <p>YYYYMMDDHHMMSS where ...</p> <p>YYYY=year</p> <p>MM=Month</p> <p>DD=Day</p> <p>HH=Hours</p> <p>MM=Minutes</p> <p>SS=Seconds</p> <p>If no time is specified, 12:00AM is the default setting.</p> <p>Alternatively, store a date such as <i>20070711</i> and time in a file called <i>since.txt</i> with a path of your choosing and reference the <i>since.txt</i> file in [PrimoExp].</p> <p>For example:</p> <p>ChangedSince=./test/work/since.txt</p> <p>NOTE: The system logs the following warnings/errors with the use of the ChangedSince= parameter:</p> <ul style="list-style-type: none"> • 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 2-2. [PrimoExp] parameter/command descriptions



Parameter/Command	Description
HeadingChanges= -H	<p>Use this parameter to indicate the following.</p> <ul style="list-style-type: none"> • Yes (Y), find changed headings and include all records affected. • No (N), do not look for changed headings. <p>This parameter should be set to Y when the configuration file is used to harvest records for the Primo publishing pipe.</p> <p>This parameter should be set to N when the configuration file is used to harvest records for the Primo availability pipe.</p> <p> IMPORTANT: <i>The HeadingChanges parameter requires that a setting is specified for the ChangedSince parameter. HeadingChanges can only be process if there is a ChangedSince date.</i></p>
AvailFilter= -A	<p>Use this parameter to indicate which Voyager records to evaluate for export to Primo based on availability changes.</p> <p>Y=Yes (filter for availability changes) N=No (do not filter for availability changes)</p> <p>This parameter should be set to Y when the configuration file is used to harvest records for the Primo availability pipe.</p> <p>This parameter should be set to N when the configuration file is used to harvest records for the Primo publishing pipe.</p>
LogFile= -L	<p>Use this parameter to set the file name for the log file.</p> <p>For example:</p> <pre>primo.export.\$yyMMdd\$.log</pre> <p>The system automatically updates the \$yyMMdd\$ with the date/time component of the file name.</p> <p>To specify that no log file is to be created, use the following parameter setting.</p> <pre>LogFile=/dev/null</pre> <p> TIP: <i>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.</i></p>

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

Parameter/Command	Description
LogFileDir=	Use this parameter to set the path for the log file.
LogLevel= -v	Leave the <code>LogLevel</code> parameter setting as specified in the configuration file unless directed otherwise by Ex Libris support staff.
LogToStdOut=	Leave the <code>LogToStdOut</code> 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 <code>-d</code> command is used to combine the <code>Database</code> , <code>DatabaseHost</code> , and <code>DatabasePort</code> parameters into a single string. For example: <code>-d 208.178.237.40:1521@VGER</code>
DatabaseHost= -d	Use this parameter to set the database host IP address (the same as in <code>voyager.env</code>). The <code>DatabaseHost</code> parameter is set one time at system installation. Verify this setting to insure its accuracy. The <code>-d</code> command is used to combine the <code>Database</code> , <code>DatabaseHost</code> , and <code>DatabasePort</code> parameters into a single string. For example: <code>-d 208.178.237.40:1521@VGER</code>
DatabasePort= -d	Use this parameter to set the database port. The <code>-d</code> command is used to combine the <code>Database</code> , <code>DatabaseHost</code> , and <code>DatabasePort</code> parameters into a single string. For example: <code>-d 208.178.237.40:1521@VGER</code>
UserPass= -e	Use this parameter to specify the Oracle database user name and password. For example: <code>-e user/pass</code>
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 2-2. [PrimoExp] parameter/command descriptions

Parameter/Command	Description
DeIMFHDsDir=	Edit the parameter provided in the configuration file to match your Voyager configuration.
DeIMFHDsFile=	Leave this parameter setting as specified in the configuration file.
DellItemsDir=	Edit the parameter provided in the configuration file to match your Voyager configuration.
DellItemsFile=	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 2-3](#) for a description of the parameters used in the [XMLWriterProtocol] stanza.

Table 2-3. [XMLWriterProtocol] parameter descriptions


Parameter	Description
File=	<p>Use this parameter to identify the file path for storing the XML files generated from harvesting Voyager records for export to the Primo database.</p> <p> 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.</p> <p>For example:</p> <pre>./VPrimoExport/Pub/primo.export.\$yyMMddhhmmss\$.xml ./VPrimoExport/Avail/primo.export.\$yyMMddhhmmss\$.xml</pre> <p>NOTE: The path you specify needs to be coordinated with the execution of the <code>bundlePrimo.ksh</code> script. The <code>bundlePrimo.ksh</code> script wraps the individual records into the tar.gz files that are accessed by the Primo publishing and availability pipes for import.</p>
XSL=	Leave this parameter blank. No setting required.

Table 2-3. [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 2-3](#).

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*.

Patron Services

3

Introduction	3-1
Patron Authentication (Patron Directory Services)	3-1
• Configuration	3-2
• PDS Invoked	3-2
Patron Information (Primo Library Card)	3-4
Patron Requests (GetIt!)	3-4

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](#).

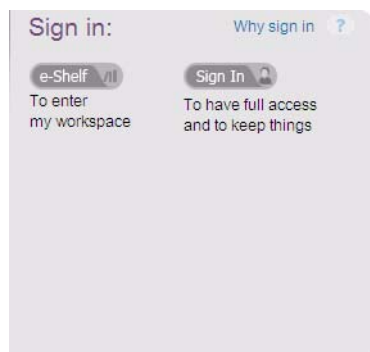


Figure 3-1. Primo Sign In / PDS Authentication

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](#).

Cover image not available

Biology : concepts & connections
Neil A. Campbell 1946-
Menlo Park, Calif. : Benjamin Cummings

Close

Holdings list Check availability and location Additional services SFX

Holdings list

Get Avail Requests

Database:	Viper QA652DB
Main Author:	Campbell, Neil A.,
Title:	Biology : concepts & connections /
Publisher:	Menlo Park, Calif. : Benjamin Cummings, c1997.

Holdings Information

Database:	Viper QA652DB
Location:	Acquisitions
Call Number:	No call number available
Order Information:	4 Copy Ordered as of 2006-11-03
Number of Items:	None
Status:	No information available

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

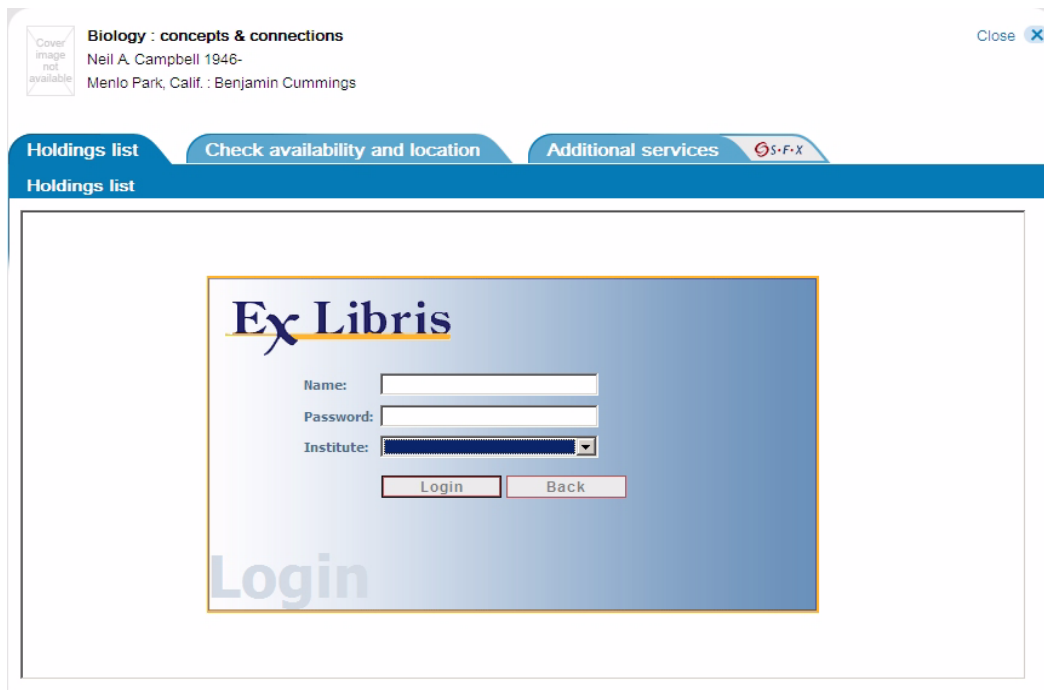


Figure 3-3. Login dialog box

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.

The screenshot shows the Voyager Holdings List interface. At the top, there is a header with a 'Close' button. Below the header, there are three tabs: 'Holdings list', 'Check availability and location', and 'Additional services'. The 'Holdings list' tab is active. The main content area is divided into two sections: 'Bibliographic Information' and 'Holdings Information'. The 'Bibliographic Information' section includes fields for Database (Viper QA652DB), Main Author (Campbell, Neil A.), Title (Biology : concepts & connections /), and Publisher (Menlo Park, Calif. : Benjamin Cummings, c1997). The 'Holdings Information' section includes fields for Database (Viper QA652DB), Location (Acquisitions), Call Number (No call number available), Order Information (4 Copy Ordered as of 2006-11-03), Number of Items (None), and Status (No information available). A 'Select a Request' dropdown menu is open, showing a list of request options: Hold, Recall, Call Slip Request Form, Short Loan, Blank Form, ILL Loan Request, ILL Photocopy Request, Limited Access Request (VDX), Remote request, ILL Test Loan Request JP, Media Equipment Booking, Index New Zealand Form, ILL Blank Request, Test Form for Doc, BibLevel, and Item Level Request. A 'Go' button is next to the dropdown menu.

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

Introduction	4-1
Course Reserves Function Overview	4-1
CourseReserves=	4-2
InsertCRTag=952 Enrichment Tag	4-3
Pprimoexp Extract Processing	4-4

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.

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
```

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 2-28](#) 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
952g	Instructor last name
952h	Instructor first name
952i	Instructor title
952j	Course name

Table 4-1. 952 Extract

952 Subfields	Extract Description
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.

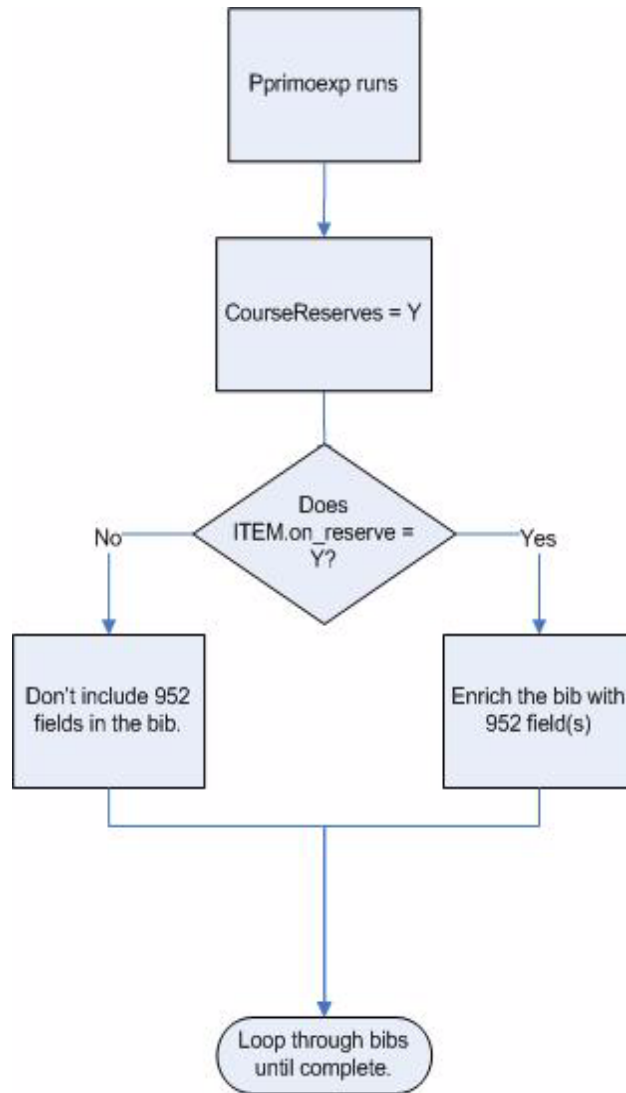


Figure 4-3. Extract processing without ChangedSince= specified

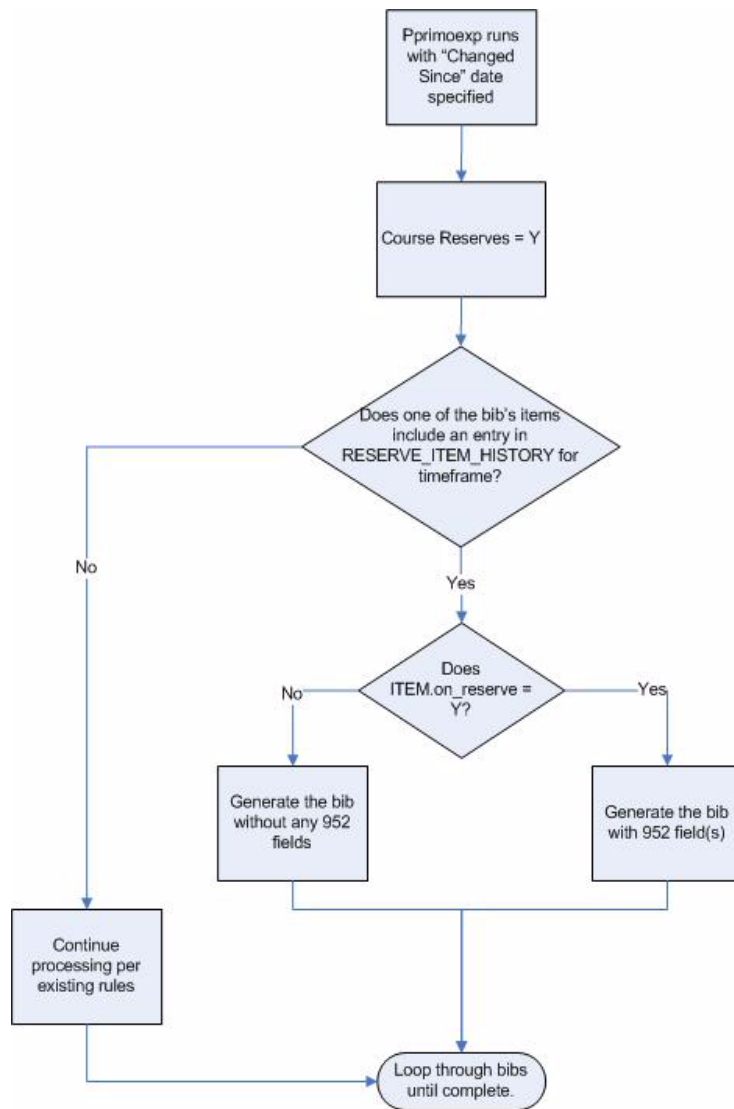


Figure 4-4. Extract processing with ChangedSince= specified

Real Time Availability

5

Introduction	5-1
Real Time Availability Processing	5-1
Error Conditions	5-3
web.xml	5-4
PrimoExp.ini	5-5

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 `/ml/voyager/yyydb/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 2-28](#)) 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. Specific 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 2-2](#) on [page 2-29](#) for a description of `UserPass=`.

Index

Numerics

949, [2-28](#)
 950, [2-28](#)
 951, [2-28](#)

A

about this document, [xi](#)
 At Bindery=, [2-26](#)
 audience
 of this document, [xi](#)
 availability pipe, [2-3](#), [2-25](#), [2-35](#)
 AvailFilter=, [2-31](#)

B

batch job
 Pprimoexp, [2-4](#), [2-34](#), [4-4](#)
 BibRangeBegin, [2-4](#)
 BibRangeBegin=, [2-29](#)
 BibRangeEnd, [2-4](#)
 BibRangeEnd=, [2-29](#)
 BibsFromFile, [2-4](#)
 BibsFromFile=, [2-29](#)

C

Call Slip Request=, [2-26](#)
 Cataloging Review=, [2-26](#)
 ChangedSince=, [2-30](#), [4-4](#)
 Charged=, [2-26](#)
 checklist, [1-2](#)
 Circulation Review=, [2-26](#)

Claims Returned=, [2-26](#)
 comments
 about this document, [xiv](#)
 configuration file
 PrimoExp.ini, [5-2](#), [5-4](#), [5-5](#)
 PrimoExp-Availability.ini, [2-3](#), [2-5](#)
 PrimoExp-Publishing.ini, [2-3](#), [2-5](#)
 PrimoExp-Selective.ini, [2-4](#), [2-5](#)
 conventions used
 in this document, [xii](#)
 course reserves, [4-1](#)
 CourseReserves=, [4-2](#), [4-4](#)

D

Damaged=, [2-26](#)
 data extraction, [2-1](#)
 Database=, [2-32](#)
 DatabaseHost=, [2-32](#)
 DatabasePort=, [2-32](#)
 DelBibFile=, [2-32](#)
 DelBibsDir=, [2-32](#)
 DelltemsDir=, [2-33](#)
 DelltemsFile=, [2-33](#)
 DelMFHDsDir=, [2-33](#)
 DelMFHDsFile=, [2-33](#)
 Discharged=, [2-26](#)
 document summary, [xii](#)

E

[Enrichment Tags], [2-28](#), [4-2](#), [4-3](#)
 [Excluded Happening Locations], [2-27](#)
 [Excluded Operator Ids], [2-27](#)

F

feedback, customer, [xiv](#)
 File=, [2-33](#)

G

GetIt!, [3-4](#)
Getting Started, [1-1](#)
 prerequisite skills and knowledge, [1-1](#)
getting started
 checklist, [1-2](#)

H

harvesting, [2-3](#)
Heading Changes=, [2-31](#)
Hold Request=, [2-26](#)

I

idPrefix=, [2-34](#)
In Process=, [2-26](#)
In Transit Discharged=, [2-26](#)
In Transit On Hold=, [2-26](#)
In Transit=, [2-26](#)
initial data load, [2-2](#)
InsertAUTTag, [2-28](#)
InsertAVATag, [2-28](#)
InsertSUBTag, [2-28](#)
[Institution Id], [2-27](#)
intended audience
 of this document, [xi](#)
[Item Statuses], [2-25](#)
[Item Statuses] parameter
 At Bindery=, [2-26](#)
 Call Slip Request=, [2-26](#)
 Cataloging Review=, [2-26](#)
 Charged=, [2-26](#)
 Circulation Review=, [2-26](#)
 Claims Returned=, [2-26](#)
 Damaged=, [2-26](#)
 Discharged=, [2-26](#)
 Hold Request=, [2-26](#)
 In Process=, [2-26](#)
 In Transit Discharged=, [2-26](#)
 In Transit On Hold=, [2-26](#)
 In Transit=, [2-26](#)

Lost--Library Applied=, [2-26](#)
Lost--System Applied=, [2-26](#)
Missing=, [2-26](#)
Not Charged=, [2-26](#)
On Hold=, [2-26](#)
Overdue=, [2-26](#)
Recall Request=, [2-26](#)
Remote Storage Request=, [2-26](#)
Renewed=, [2-26](#)
Scheduled=, [2-26](#)
Short Loan Request=, [2-26](#)
Withdrawn=, [2-26](#)
ITEM.on_reserve, [4-4](#)

L

Library Card, [3-4](#)
[Location to Primo], [2-27](#)
LogFile=, [2-31](#)
LogFileDir=, [2-32](#)
LogLevel=, [2-32](#)
LogToStdOut=, [2-32](#)
Lost--Library Applied=, [2-26](#)
Lost--System Applied=, [2-26](#)

M

Missing=, [2-26](#)

N

Not Charged=, [2-26](#)

O

On Hold=, [2-26](#)
OPAC port, [1-2](#)
Overdue=, [2-26](#)

P

patron authentication, 3-1
 Patron Directory Services (PDS), 3-1
 patron information, 3-4
 patron requests, 3-4
 PbundlePrimo, 2-2, 2-34
 PDS, 3-1
 photocopying
 documentation, xiv
 pipes, 2-3
 PNX, 2-4
 port setting, 1-2
 Pprimexp, 2-4, 2-34, 4-4
 Primo database
 initial data load, 2-2
 ongoing updates, 2-3
 Primo Library Card, 3-4
 Primo Normalized XML (PNX), 2-4
 [PrimoExp], 2-29, 5-5
 [PrimoExp] parameter
 AvailFilter=, 2-31
 BibRangeBegin=, 2-29
 BibRangeEnd=, 2-29
 BibsFromFile=, 2-29
 ChangedSince=, 2-30, 4-4
 CourseReserves=, 4-2, 4-4
 Database=, 2-32
 DatabaseHost=, 2-32
 DatabasePort=, 2-32
 DelBibFile=, 2-32
 DelBibsDir=, 2-32
 DelItemsDir=, 2-33
 DelItemsFile=, 2-33
 DelMFHDsDir=, 2-33
 DelMFHDsFile=, 2-33
 HeadingChanges=, 2-31
 LogFile=, 2-31
 LogFileDir=, 2-32
 LogLevel=, 2-32
 LogToStdOut=, 2-32
 UserPass=, 2-32
 PrimoExp.ini, 5-2, 5-4, 5-5
 PrimoExp-Availability.ini, 2-3, 2-5
 PrimoExp-Publishing.ini, 2-3, 2-5
 PrimoExp-Selective.ini, 2-4, 2-5
 publishing pipe, 2-3, 2-35
 purpose
 of this document, xi

R

real time availability, 5-1
 Real Time Availability (RTA) port, 1-2
 Recall Request=, 2-26
 recsPerGroup=, 2-34
 reissue
 reason for, xii
 Remote Storage Request=, 2-26
 Renewed=, 2-26
 reproduction, of documentation, xiv
 RESERVE_ITEM_HISTORY, 4-4
 RTA port, 1-2

S

Scheduled=, 2-26
 scheduling, 2-35
 script
 PbundlePrimo, 2-2, 2-34
 Short Loan Request=, 2-26
 stanza
 [Enrichment Tags], 2-28, 4-2, 4-3
 [Excluded Happening Locations], 2-27
 [Excluded Operator Ids], 2-27
 [Institution Id], 2-27
 [Item Statuses], 2-25
 [Location to Primo], 2-27
 [PrimoExp], 2-29, 5-5
 [XMLWriterProtocol], 2-33

T

tab_service.<institute>, 3-2
 table
 ITEM.on_reserve, 4-4
 RESERVE_ITEM_HISTORY, 4-4
 tar.gz files, 2-4

U

UsePrettyPrint=, [2-34](#)

UserPass=, [2-32](#)

V

Voyager institution ID, [2-27](#)

W

web.xml, [5-4](#)

Withdrawn=, [2-26](#)

X

[XMLWriterProtocol], [2-33](#)

[XMLWriterProtocol] parameter

File=, [2-33](#)

idPrefix=, [2-34](#)

recsPerGroup=, [2-34](#)

UsePrettyPring=, [2-34](#)

XSL=, [2-33](#)

XSL=, [2-33](#)