



Voyager[®] 10.3

Technical User's Guide

May 2021

Ex Libris Confidential

CONFIDENTIAL INFORMATION

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

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

DISCLAIMER

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

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

TRADEMARKS

"Ex Libris," the Ex Libris Bridge to Knowledge , Primo, Aleph, Voyager, SFX, MetaLib, Verde, DigiTool, Rosetta, bX, URM, Alma , 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, 2019. All rights reserved.

Document released: May 2021

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

Contents

About This Document

• Purpose	xxiii
• Intended Audience	xxiii
• Reason for Reissue	xxiii
• Document Summary	xxiv
• Conventions Used in This Document	xxvi
• Document Reproduction/Photocopying	xxvii
• Comment on This Document	xxvii

1 Getting Started

• Introduction	1-1
• Prerequisite Skills and Knowledge	1-1
• Before You Begin	1-1

2 Overview of the Data Conversion Process

• Introduction	2-1
• Purpose of this Chapter	2-2
• Bibliographic Data	2-2
• Authority Data	2-2
• Patron Data	2-3
• Circulation Transactions	2-3
• Vendor Data	2-4

3 Server Activities in the Voyager System

• Introduction	3-1
• Purpose of This Chapter	3-1

Contents

• Services, Scripts, Service Dependencies and Rationale	3-2
Scripts	3-2
Service Dependencies	3-3
• Solaris, Linux, and AIX Servers	3-4
Starting the Server	3-4
Rebooting or Shutting Down the Server	3-4
Solaris Servers	3-4
Linux or AIX Servers	3-5
Stopping and Starting Oracle Services	3-6
Stopping and Starting Voyager	3-8
Stopping and Starting the Apache Web Server	3-10
Database Backup for Solaris and Linux Servers	3-11
Backup Process	3-11
Determining the Backup Device Driver	3-12
Identifying your Data File System/Volume	3-15
Reviewing Cron Backup Results (Solaris and Linux)	3-18
• Restarting tomcat	3-18
• Stopping/Restarting Voyager PDS	3-19
• Restoring from Backup	3-19
• Crons Set Up at Installation	3-19

4 Voyager Client Installation and the Voyager.ini File

• Introduction	4-1
Prerequisites	4-2
• Client Installation Options	4-2
• Transition Considerations for AutoUpdate	4-2
• Downloading Voyager Client Files	4-3
• Installing Voyager Clients Using the Manual Method	4-4
• Modifying (Adding or Deleting Clients) an Installation	4-9
• Repairing an Installation of Voyager Clients	4-13
• Uninstalling the Voyager Clients	4-15
• The Voyager.ini File on the PC	4-17
Client Module Stanzas	4-20
Circulation Charge Timeout	4-21
[GlobalLog] Stanza	4-22
Single Client Login	4-23
Staff Client Re-Sort	4-24

Contents

[E-mail] Stanza	4-27
[MARC Posting] Stanza	4-27
MARC Record Posting Feature	4-28
[SearchURI] Stanza	4-31
Adding a Button to the Search Dialog Box	4-31
[Upgrade] Stanza - Voyager 7	4-32
New Software Version Detected?	4-34
Traditional Method	4-35
HTTP Method	4-36
FTP Method	4-39
.BAT File Method	4-39
[Upgrade] Stanza - Voyager 8 and Later	4-44
AutoUpdate Characteristics	4-45
AcquisitionsStart.exe and CatalogStart.exe	4-47
[HelpMenuLink] Stanza	4-48
• Additional Files Installed with the Client Installation	4-48
• AutoUpdate on the Server	4-51
Server Files	4-51
Location of the Files to be Downloaded	4-51
AutoUpdate Configuration File	4-52
Apache Configuration	4-52
User Access for AutoUpdate	4-54
Adding a Login/Password to the Existing File	4-54
Modifying a Login/Password for an Existing User	4-55
Deleting a Login/Password for an Existing User	4-55
Test AutoUpdate	4-56
Client Setup	4-58
Voyager.ini Setup	4-58
Customize AutoUpdate Directories	4-59
Scenario 1 - Deploy Custom Bmarc9xx.cfg for Marc21 Tag Table	4-59
Scenario 2 - Deploy Prelinked Institution-Specific reports.mdb4-	60
• File Transfer Protocol (FTP) - Voyager Client Installations	4-61

5

Patron Batch Jobs in XML

• Overview	5-1
• Batch Processing - UNIX Shell Scripts	5-2

Contents

PpatronImpX (Patron Import)	5-2
Processing	5-4
<action> Element	5-5
<overlayAction> Element	5-5
Audit File	5-6
Error File	5-7
Web Service for Patron Import	5-8
PpatronExpX (Patron Export)	5-8
Web Service for Patron Export	5-10
PpatronPrg (Patron Purge)	5-11
Exceptions	5-15
• Web Admin	5-16

6 Patron Extract (Patron SIF Method)

• Introduction	6-1
• Purpose of this Chapter	6-1
• Overview of the Patron Extract Program	6-1
• Parameters	6-2
Running Patron Extract	6-4
• Output File Specification	6-5
• Additional Files	6-5
Audit File	6-5
Error File (Exception File)	6-6

7 Patron Update (Patron SIF Method)

• Introduction	7-1
• Purpose of this Chapter	7-1
• Overview the Patron Update Program	7-1
• Input File Specification	7-3
• Parameters	7-3
Running Patron Update	7-6
• Additional Files	7-7
Audit File	7-7
Error File (Exception File)	7-8

Contents

8 Bursar Transfer System

- Introduction 8-1
- Purpose of this Chapter 8-1
- Overview of the Bursar Transfer System 8-2
- Steps of the Bursar Transfer System 8-3
- Supported One-Way Transfer Types 8-4
 - Transferring Total Patron Balances 8-4
 - Transferring Itemized Patron Fines/Fees 8-4
- Configuration File 8-5
- Parameters 8-7
 - Running Bursar Transfer 8-8
- Output File Specification 8-9
 - Bursar SIF Format 8-9
- Additional Files 8-10
 - Audit File 8-10
 - Error File 8-11
- Bursar Transfers in the Circulation Module 8-11

9 Circulation Batch Jobs

- Introduction 9-1
- Purpose of This Chapter 9-2
- Update Shelving Status (Circjob 1) 9-3
- Archive and Expire Call Slip Requests (Circjob 8) 9-4
- Export OPAC Requests (Circjob 26) 9-4
- Archive Short Loans (Circjob 27) 9-5
- Purge Universal Borrowing (UB) Patron Stub Records (Circjob 29) 9-5
- Accrued Fines and Demerits (Circjob 30) 9-6
- Patron Suspension (Circjob 31) 9-7
- Universal Borrowing (UB) Request Promotion (Circjob 32) 9-8
- Update Remote Circulation Cluster Cache (Circjob 33) 9-9
- Place Items on Active Course Reserve List (Circjob 34) 9-10
- Place Recalls and Holds for Items on Active Course Reserve List (Circjob 35) 9-12
- Take Items on Inactive Course Reserve List Off Reserve (Circjob 36) 9-14
- Forgive Demerits (Circjob 37) 9-16

Contents

• Retain Patron IDs (Circjob 38)	9-17
• Patron Purge (Circjob 39)	9-23
Patron Purge Files	9-23
Patron Purge Command Line (Circjob 39)	9-25
• Forgive Fines by Patron ID (Circjob 40)	9-27
Running circjob 40 from the command line	9-27
Error Logging for Batch Forgive Jobs	9-28
circjob.log	9-28
err.forgive.YYYYMMDD.HHMM	9-29
Audit Report for Batch Forgive Jobs	9-29
• Forgive Fines by Create Date (Circjob 41)	9-30
Running circjob 41 from the command line	9-30
Error Logging for Batch Forgive Jobs	9-32
circjob.log	9-32
err.forgive.YYYYMMDD.HHMM	9-32
Audit Report for Batch Forgive Jobs	9-32
• Forgive Fines by Patron Group and Expire Date (Circjob 42)	9-33
Running circjob 42 from the command line	9-33
Error Logging for Batch Forgive Jobs	9-35
circjob.log	9-35
err.forgive.YYYYMMDD.HHMM	9-35
Audit Report for Batch Forgive Jobs	9-36
• Synchronize Patron Counters for Universal Borrowing (Circjob 43)	9-36
Frequency of Use	9-37
How to Access	9-37
Server Access	9-37
Command Line Access	9-39
WebAdmin Access	9-40
Circjob Log	9-40
Error Conditions	9-40
• Automatically Expire Patron Barcode Status (Circjob 44)	9-41
• Remap Automatically Mapped UB Stub Patrons (Circjob 45)	9-42

10 Media Scheduling Batch Jobs

• Introduction	10-1
• Purpose of this Chapter	10-1
• Retain Patron IDs (Mediajob 5)	10-1

Contents

11	Bulk Export of MARC Records	
	• Introduction	11-1
	• Purpose of this Chapter	11-1
	• Overview of Bulk Export	11-2
	• Parameters	11-3
	Running the Marcexport Program	11-10
	• Output File Specification	11-11
	• Additional Files	11-11
	Audit File	11-11
12	Prebulk Program	
	• Introduction	12-1
	• Purpose of this Chapter	12-2
	• Overview of Prebulk Processing	12-2
	Using Prebulk to Check the Records	12-3
	Using Prebulk to Strip Fields	12-3
	Using Prebulk to Create an Interleaved File	12-4
	Bulk Importing of the Interleaved File Generated by Prebulk	12-4
	Holdings Created from the Interleaved File	12-5
	• Input File	12-5
	• Creating the Configuration File	12-5
	Default Location of the Configuration File	12-6
	Name of the Configuration File	12-6
	Stanza Types	12-7
	Using the Default Call Number	12-7
	Using a Call Number Stored in the Record	12-7
	Overrides Stanza	12-8
	CREATEMFHD	12-8
	DEFAULTCALLNO	12-8
	DEFAULTCALLIND	12-9
	USE001FOR014	12-9
	USE003FORLOC	12-9
	Strip Stanza	12-9
	Non-Standard 035 Field Elimination	12-10
	MFHDTAG Stanza	12-11
	CALLTYPES Stanza	12-12

Contents

LOCATIONS Stanza	12-12
MAPPING Stanza	12-15
008 Stanza	12-16
• Parameters	12-19
Running the Prebulk Program	12-20
• Output File Specification	12-21
• Additional Files	12-21
Log (Audit) File	12-21
Error File	12-22

13 Bulk Import, Replace, and Merge of MARC Records

• Introduction	13-1
• Purpose of this Chapter	13-2
• Overview of the Bulk Import, Replace, Merge MARC Records Program	13-2
Generating Keyword Index Files	13-3
• Parameters	13-4
Running Bulk Import	13-9
• Input File Specification	13-9
• Additional Files	13-9
Log File	13-10
Delete, Discard, Replace, and Reject Files	13-10
Delete	13-10
Discard	13-11
Replace	13-11
Reject	13-11
Error File	13-12
• Messages in Log and Error Files	13-12

14 Global Heading Change Jobs

• Introduction	14-1
• Purpose of This Chapter	14-1
• Overview of GHC	14-1
• Parameters	14-2
Process Global Heading Change Queue Step 1 (Catjob 11)	14-4

Contents

Process Global Heading Change Queue Step 2 (Catjob 12)	14-5
Process Global Heading Change Fields Step 3 (Catjob 13)	14-6

15

Storage Barcode Verify (Pstrgvfy) Program

• Introduction	15-1
• Purpose of This Chapter	15-2
• Changing Location Codes of Item Records or MFHDs	15-2
• Populating the Operator or Location Columns of a MFHD's History Tab	15-3
• Verifying Barcodes of Item Records or MFHDs	15-4
• Pstrgvfy Input File	15-4
• Pstrgvfy Parameters	15-5
• Pstrgvfy Command File	15-8
• Pstrgvfy Log File	15-9
• Sample Pstrgvfy Command Line	15-10

16

Popacjob

• Introduction	16-1
• Purpose of This Chapter	16-1
OPAC_Search_Log Table	16-2
Bib_usage_log Table	16-3
Setting Up OPAC Search Logging	16-4
Setting Up Bibliographic Usage Logging	16-5
Setting up Selective Dissemination of Information (SDI)	16-6
• Accessing Logged Information	16-6
OPAC Search Log Export program	16-6
Interactive Method	16-7
Cron Method	16-8
SDI Searches Program (Tomcat WebVoyage)	16-9
Enabling SDI	16-10
Customizing the Search Results URL Page Sent to Patrons by SDI (sdiemail.ini File)	16-12
Run SDI - Interactive Method	16-13
Run SID - Cron Method	16-14

Contents

SDI Searches Program (Classic WebVoyáage)	16-15
Enabling SDI	16-15
Configuring SDI Options	16-16
Customizing the Search Results URL Page Sent to Patrons by SDI	16-18
Interactive Method	16-20
Cron Method	16-21
OPAC Bib Usage Log Export Program	16-21
Interactive Method	16-21
Cron Method	16-23

17 **Acquisitions Batch Job - Fix Exchange Rates**

• Introduction	17-1
• Fix Exchange Rates	17-1
Purchase Orders Updated with Acqjob 5	17-2
Additional Considerations for Acqjob 5	17-3
• Running Acqjob 5	17-3
Running Acqjob 5 from the Command Line	17-4
Running Acqjob 5 Interactively	17-4
• Log File	17-6

18 **Server-Side Configurations**

• Introduction	18-1
• UB Barcode Lookup	18-1
Configuring the UB Barcode Lookup Feature	18-1
voyager.env	18-2
Creating the UB Barcode Lookup Configuration File	18-2
UB Barcode Lookup Configuration Schema File	18-8
Configuration File Validation	18-11
• Dynamic Noise Word Reduction	18-12
• Configuring the Z39.50 Server for UTF-8 Encoded Records	18-13

Contents

19	Patron Record Standard Interface File	
	• Introduction	19-1
	• Purpose of This Chapter	19-1
	• File Specification	19-2
	• Patron Record SIF Format	19-2
	Base Segment	19-3
	Address Segment	19-10
	Variable Segment	19-12
	End-of-Record Segment	19-13
20	Charge Transaction Record Standard Interface File	
	• Introduction	20-1
	• Purpose of This Chapter	20-1
	• Input File Specification	20-1
	• Charge Transaction Record SIF Format	20-2
21	Item Delete Standard Interface File	
	• Introduction	21-1
	• Purpose of This Chapter	21-1
	• File Format	21-1
22	Vendor Record Standard Interface Format	
	• Introduction	22-1
	• Purpose of This Chapter	22-1
	• Input File Specification	22-1
	• Vendor SIF Format	22-2
	Base Segment	22-3
	Address Segment	22-4
	Variable Segment	22-6
	End-of-Record Segment	22-6

Contents

23	Acquisitions Notices Standard Interface File	
	• Introduction	23-1
	• Purpose of This Chapter	23-1
	• File Specification	23-2
	• Acquisitions Notices SIF Format	23-3
	Base Segment	23-4
	Suffixes	23-5
	Cancellation Notice Suffix (00)	23-6
	Return Notice Suffix (01)	23-6
	Order Claim Notice Suffix (02)	23-7
	Serial Claim Notice Suffix (03)	23-7
	Voucher/Check Request Suffix (04)	23-8
	Cancel Serial Claim Notice Suffix (05)	23-8

24	Acquisitions Reports Standard Interface File	
	• Introduction	24-1
	• Purpose of This Chapter	24-1
	• File Specification	24-1
	• Acquisitions Reports SIF Format	24-3
	Base Segment	24-4
	Suffixes	24-4
	Purchase Orders Report (00)	24-5
	Open Orders Report (01)	24-8
	Global Open Orders Report (02)	24-9
	Fund Snapshot Report Suffix (03)	24-11
	Open Orders Before Rollover Report (04)	24-12
	Open Orders After Rollover Report (05)	24-12
	Fund Snapshot Before Rollover Report (06)	24-12
	Fund Snapshot After Rollover Report (07)	24-13
	Fund Rollover Status Report Suffix (08)	24-13
	Copy Rollover Status Report Suffix (09)	24-13

Contents

25	Cataloging Reports Standard Interface File	
	• Introduction	25-1
	• Purpose of This Chapter	25-1
	• File Specification	25-1
	• Cataloging Reports SIF Format	25-2
	Base Segment	25-4
	Suffixes	25-4
	Unauthorized Subject Headings (00)	25-5
	Unauthorized Name Headings (01)	25-5
	Unauthorized Title Headings (02)	25-5
	Unauthorized Name/Title Headings (03)	25-6
	Unauthorized Subdivision Headings (04)	25-6
	Duplicate Authority Records (05)	25-6
	'See' References with Linked Bib Records (06)	25-7
	'See' References Authorized in Another Authority Record (07)	25-7
	'See' References without Corresponding Authority Record (08)	25-8
	856 Link Failure Report Suffix (09)	25-8
26	Circulation Notices Standard Interface File	
	• Introduction	26-1
	• Purpose of This Chapter	26-1
	• File Specification	26-2
	• Circulation Notices SIF Format	26-3
	Base Segment	26-4
	Suffixes	26-5
	Cancellation Notice Suffix (00)	26-6
	Item Available Notice Suffix (01)	26-6
	Overdue Notice Suffix (02)	26-6
	Recall Notice Suffix (03)	26-7
	Recall-Overdue Notice Suffix (04)	26-7
	Fine/Fee Notice Suffix (05)	26-7
	Statement of Fines and Fees Suffix (06)	26-8
	Courtesy (Due) Notice Suffix (07)	26-9

Contents

27	Circulation Reports Standard Interface File	
	• Introduction	27-1
	• Purpose of This Chapter	27-1
	• File Specification	27-1
	• Circulation Reports SIF Format	27-2
	Base Segment	27-4
	Suffixes	27-4
	Reserved Items Active Report (00)	27-5
	Reserved Items Expired Report (01)	27-5
	Hold Shelf Expired Report (02)	27-5
	Missing in Transit Report (03)	27-6
	Circulation Statistics Report (04)	27-6
	Circulation Item-Related Exceptions Report (05)	27-7
	Circulation Patron-Related Exceptions Report (06)	27-7
	Circulation Transaction-Related Exceptions Report (07)	27-7
	Global Circulation Statistics Report (08)	27-8
	Distribution Item Order List Report (09)	27-8
28	Media Scheduling Notices Standard Interface File	
	• Introduction	28-1
	• Purpose of This Chapter	28-1
	• File Specification	28-2
	• Media Scheduling Notices SIF Format	28-3
	Base Segment	28-3
	Suffixes	28-4
	Overdue Notice Suffix (00)	28-5
29	Media Scheduling Reports Standard Interface File	
	• Introduction	29-1
	• Purpose of This Chapter	29-1
	• File Specification	29-1
	• Media Scheduling Reports SIF Format	29-2
	Base Segment	29-3

Contents

Suffixes	29-3
Media Equipment Inventory Report (00)	29-4
Booking Statistics Report (01)	29-4
Booking Exceptions Report (02)	29-5
Booking Charge Statistics Report (03)	29-5

30

Database Views

• Introduction	30-1
• Purpose of This Chapter	30-2
• Views	30-2
Authblob_vw	30-2
Authhistory_vw	30-3
Authheading_vw	30-3
Authority1xx4xx_vw	30-4
Authority5xx1xx_vw	30-4
Authoritydupe_vw	30-5
Bib_vw	30-5
Bibblob_vw	30-7
Bibhistory_vw	30-7
Bibloc_vw	30-8
Circcharges_vw	30-8
Circrenewal_vw	30-10
Fundledger_vw	30-12
Heading_vw	30-14
Issues_vw	30-15
Item_vw	30-16
LCclass_vw	30-18
Marccomputer_vw	30-19
Marcbook_vw	30-19
Marcmap_vw	30-20
Marcmusic_vw	30-20
Marcserial_vw	30-21
Marcvisual_vw	30-22
MFHDblob_vw	30-22
MFHDhistory_vw	30-23
NLMclass_vw	30-23
Recordcount_vw	30-24

Contents

Serials_vw	30-24
Sudocclass_vw	30-25
Vendorinvoice_vw	30-26
Vendororder_vw	30-28

31

WebAdmin

• Introduction	31-1
• The Purpose of this Chapter	31-2
• Setting Up WebAdmin	31-2
System Requirements	31-2
Starting the Server Dæmon	31-3
Creating WebAdmin Users and Passwords	31-3
Adding the WebAdmin Authorization Section to the httpd.conf File	31-5
• Starting the Voyager Server Utilities	31-6
• Using WebAdmin	31-7
Voyager Server Utilities Page	31-7
Acquisitions Utilities	31-9
Cataloging Utilities	31-12
Bulk Import MARC Records	31-12
Bulk Export MARC Records	31-17
Cataloging Reports	31-22
Circulation Utilities	31-24
Reports and Notices - Forgiving Fines and Fees	31-24
Reports and Notices - XML Patron Purge	31-29
Bursar Transfer	31-30
Media Scheduling Utilities	31-34
OPAC Reports Utilities	31-36
Popacjob	31-36
System Admin Utilities	31-38
Log Files Utility	31-38
Report Files Utility	31-39
FPC Admin Utilities	31-39
Enabling FPC Admin Utilities	31-42

Contents

32	Oracle Stored Functions	
	• Introduction	32-1
	MARC Functions	32-2
	Advanced MARC Functions	32-7
	Miscellaneous Functions	32-12
33	SMS Messaging	
	• Introduction	33-1
	• Implementation	33-2
	• SMS Gateway/Service Providers	33-3
	• Ex Libris SMS Proxy	33-4
	• Voyager Setup/Configuration	33-4
	Circulation Notices	33-5
	webvoyage.properties	33-5
	voyager.env	33-6
	SmsProxyConf.xml	33-7
	Pcircjob/Pmediajob	33-9
	• Patron SMS Number	33-9
	• SMS Send Status Log	33-11
34	System Monitoring Job	
	• Introduction	34-1
	• Purpose of This Chapter	34-1
	• Overview	34-2
	• Parameters	34-2
	• Examples	34-3
	• Setup	34-3
A	UseMARCON Configuration for Use with Voyager A-1	
	• What is UseMARCON?	A-1
	In This Chapter	A-1

Contents

• Supported MARC Formats	A-2
• UseMARCON File Structure	A-2
UseMARCON Conversion Directories	A-2
UseMARCON Initialization (ini) File	A-3
Required Files	A-4
Sample UseMARCON Initialization File	A-5
• UseMARCON Log File	A-5
Fatal/Non-Fatal UseMARCON Errors	A-7
Fatal Errors	A-7
Non-Fatal Errors	A-8
Sample Fatal Error (END non OK) Message in UseMARCON Log File	A-8
Troubleshooting UseMARCON Error Conditions	A-8
UseMARCON Log File Example	A-9
• Bulk Import and UseMARCON	A-10
Overview	A-11
Bulk Import Example	A-11
Bulk Import Error Reporting	A-11
• MARC Export and UseMARCON	A-12
Overview	A-12
MARC Export Example	A-12
MARC Export Error Reporting	A-13
• Z39.50 Server and UseMARCON Interaction	A-13
Overview	A-13
The z3950svr.ini File	A-13
Sample [usemarcon translations] Stanza in z3950svr.ini File (Lines 1-3)	A-15
UseMARCON/Z39.50 Error Reporting	A-15
• UseMARCON Error Numbers	A-16
General Error Numbering	A-16
Error Message Numbers	A-17

B WebVoyage Patron Authentication Adapter FeatureB-1

• Overview	B-1
• Detailed WebVoyage Patron Adapter Interaction	B-3
• WebVoyage Configuration	B-6
Enabling and Disabling Webvoyage Adapter Behavior	B-7
Setting the URL to the External Authentication System	B-8

Contents

Enabling and Disabling the WebVoyage Patron Login Bypass	B-8
Setting the Text of the External Authentication Link Button	B-8
Format of the Redirect URL from the Authentication Adapter to Web-	
Voyage	B-8
• Example	B-9

Contents

About This Document

Purpose

This document provides information regarding basic server procedures and information on many server batch jobs used in Voyager®. Additionally, search logging in Voyager, WebAdmin, and the various input and output standard interface file (SIF) formats are covered.

Intended Audience

This document is intended for Voyager customers, especially those responsible for working with the server where your Voyager database resides.

Reason for Reissue

This user's guide incorporates and is being reissued for the following reason:

- Updated the section [Installing Voyager Clients Using the Manual Method](#) on [page 4-4](#).

Document Summary

This document consists of the following chapters:

- Chapter 1 [“Getting Started”](#) This chapter discusses some of the data loads in the data conversion process.
- Chapter 2 [“Overview of the Data Conversion Process.”](#) This chapter discusses some of the data loads in the data conversion process.
- Chapter 3 [“Server Activities in the Voyager System.”](#) This chapter provides the correct procedures for some server activities.
- Chapter 4 [“Voyager Client Installation and the Voyager.ini File.”](#) This chapter covers how to download and install the Voyager clients.
- Chapter 5 [“Patron Batch Jobs in XML.”](#) This chapter describes the patron import, export, and purge batch jobs that use XML input and output files.
- Chapter 6 [“Patron Extract \(Patron SIF Method\).”](#) This chapter provides information about the patron extract program.
- Chapter 7 [“Patron Update \(Patron SIF Method\).”](#) This chapter provides information about the patron update program.
- Chapter 8 [“Bursar Transfer System.”](#) This chapter discusses the bursar transfer program.
- Chapter 9 [“Circulation Batch Jobs.”](#) This chapter provides information on Circulation batch jobs.
- Chapter 10 [“Media Scheduling Batch Jobs.”](#) This chapter provides information on Media Scheduling batch jobs.
- Chapter 11 [“Bulk Export of MARC Records.”](#) This chapter provides information about the MARC extract program.
- Chapter 12 [“Prebulk Program.”](#) This chapter discusses the Prebulk program.
- Chapter 13 [“Bulk Import, Replace, and Merge of MARC Records.”](#) This chapter provides information on the bulk import program.
- Chapter 14 [“Global Heading Change Jobs.”](#) This chapter covers Global Headings Change batch jobs.
- Chapter 15 [“Storage Barcode Verify \(Pstrgvfy\) Program.”](#) This chapter provides information about the storage barcode verify program.
- Chapter 16 [“Popacjob.”](#) This chapter provides information on logging in Voyager.
- Chapter 17 [“Acquisitions Batch Job - Fix Exchange Rates.”](#) This discusses the Acquisitions batch job that updates commitments of Purchase Orders that use foreign currency.

- Chapter 18 [“Server-Side Configurations.”](#) This chapter describes the set up information for features that require configuration changes on the server.
- Chapter 19 [“Patron Record Standard Interface File.”](#) This chapter provides information on this SIF.
- Chapter 20 [“Charge Transaction Record Standard Interface File.”](#) This chapter provides information on this SIF.
- Chapter 21 [“Item Delete Standard Interface File.”](#) This chapter provides information on this SIF.
- Chapter 22 [“Vendor Record Standard Interface Format.”](#) This chapter provides information on this SIF.
- Chapter 23 [“Acquisitions Notices Standard Interface File.”](#) This chapter provides information on this SIF.
- Chapter 24 [“Acquisitions Reports Standard Interface File.”](#) This chapter provides information on this SIF.
- Chapter 25 [“Cataloging Reports Standard Interface File.”](#) This chapter provides information on this SIF.
- Chapter 26 [“Circulation Notices Standard Interface File.”](#) This chapter provides information on this SIF.
- Chapter 27 [“Circulation Reports Standard Interface File.”](#) This chapter provides information on this SIF.
- Chapter 28 [“Media Scheduling Notices Standard Interface File.”](#) This chapter provides information on this SIF.
- Chapter 29 [“Media Scheduling Reports Standard Interface File.”](#) This chapter provides information on this SIF.
- Chapter 30 [“Database Views.”](#) This chapter provides information on the database views, simplified ways of retrieving data from the database.
- Chapter 31 [“WebAdmin.”](#) This chapter discusses the WebAdmin utility.
- Chapter 32 [“Oracle Stored Functions.”](#) This chapter describes Oracle stored functions that may be called in an SQL query and used in a Voyager environment.
- Chapter 33 [“SMS Messaging.”](#) This chapter describes the SMS feature and the setup required to implement this feature.
- Chapter 34 [“System Monitoring Job.”](#) This chapter describes the system monitor program that enables a Voyager system administrator to receive alerts when certain system thresholds are met.
- Appendix A [UseMARCON Configuration for Use with Voyager.”](#) This chapter discusses UseMARCON.

Appendix B	WebVoyage Patron Authentication Adapter Feature .” This chapter discusses patron authentication.
Index	The Index is a detailed, alphabetical cross-reference of topics about which this document contains information.

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 *Curator Web Client 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

To provide feedback regarding this document, use the Ex Libris eService or send your comments in an e-mail message to docmanager@exlibrisgroup.com.

Introduction

This user's guide discusses a variety of server, and client related activities. It provides step-by-step instructions for completing some server maintenance tasks, client installation, and running batch jobs on the server.

Prerequisite Skills and Knowledge

To use this document effectively, you need knowledge of the following:

- Basic Microsoft® interface navigation
- Basic UNIX® commands and navigation

Before You Begin

This user's guide is primarily intended for system administrators. To accomplish most of the tasks described in this document you must have access to your Voyager server.

Overview of the Data Conversion Process

2

Introduction

The data conversion process occurs in four main steps.

1. Data extraction. Bibliographic, authority, and holdings information is extracted from the current system. This step is completed by the site.
2. Data processing. The data is processed such that the holdings and item information is made into a loadable format.
3. Test load of your data. Your institution is provided with a test load of your site's data for your review. All of this information is overwritten when the production database is installed.
4. Production load of your data. This step loads the bibliographic, authority, patron, circulation transactions, and vendor data. There is an order in which the data must be loaded. Specifically, all bibliographic data (which includes holdings, item records, and authority records) must be loaded first, before the other data types of patron and vendor data, and circulation transactions.

In addition to the data loads discussed in the following sections, sites may also load fine/fee information, orders, serial publication patterns, as well as hold and recall information.

Purpose of this Chapter

This chapter provides an overview of the following types of data loads.

- Bibliographic data
- Authority data
- Patron data
- Circulation Transactions
- Vendor data

Bibliographic Data

Bibliographic (and authority) data must be loaded before any other data types. If necessary, your data will be converted to the UTF-8 encoding during the load process.

Before the bibliographic data is loaded, do not define or create anything in the System Administration module. Any information present will be completely overwritten. You will not be able to recover it.

After the bibliographic data has been loaded, the **Location Code** field is populated in System Administration. This value was defined by your institution's responses to the Data Migration Questionnaire.

To add the **Location Name**, **Spine Label Name**, and **OPAC Display Name**, go in to the System Administration module and select **System> Locations**.

In addition, the **Item Type Code** is also populated in System Administration. This value was also defined by your institution's responses to the Data Migration Questionnaire.

To add the **Item Type Name**, and the **Display Name**, go in to the System Administration module and select **System> Item Types**.

Authority Data

Authority data should be loaded at the same time as the bibliographic data. If necessary, your data will be converted to the UTF-8 encoding during the load process.

Before the authority data is loaded, do not define or create anything in the System Administration module. Any information present will be completely overwritten. You will not be able to recover it.

There is no information that is automatically created in any module from the loaded authority data.

Patron Data

Patron data must be loaded after all bibliographic data has been loaded.

Before the patron data is loaded, do not define or create any of the following:

- Patron Groups in System Administration
- Statistical Categories in System Administration
- Patron Records in Circulation

After the patron data has been loaded, the **Patron Group Code** field is populated in System Administration.

To add the **Patron Group Name**, and the **Display Name**, go in to the System Administration module and select **Circulation> Patron Groups**.

In addition, the **Patron** and **Item Statistical Category Code** fields are populated in System Administration.

To add the **Patron Category Name**, and the **Item Category Name**, go in to the System Administration module and select **System> Statistical Categories> Patron** or **Item tab**.

Patron Records are added to the system as well. The patron data is provided to Ex Libris in API format or in a delimited data file. For information regarding the specific file format see [Patron Update \(Patron SIF Method\)](#) on [page 7-1](#).

Circulation Transactions

Circulation transaction data must be loaded after all bibliographic and patron data has been loaded.

Before the circulation transaction data is loaded, circulation policy definitions and circulation matrix definitions must be created in the System Administration module.

To create Circulation Policy Definitions for your institution, in System Administration, select **Circulation> Policy Definitions**. Be sure to define at least one circulation happening location during this process. See the *Voyager System Administration User's Guide, Circulation Locations (Circulation Happening Locations)*, for more information. Transactions will not load if there is not a circulation happening location defined.

The transaction loader uses the defined circulation happening locations for every transaction it adds to the database.

There can be multiple circulation locations in any given policy group. Given this, when doing transaction loads, the site should:

1. Define Circulation policy groups so that there is at least one Circulation happening location for each group. This Circulation location should be the one that is correct for the transactions that are being loaded.
2. After the transactions have been loaded, the site can define additional Circulation locations for the Circulation policy groups.

Before the circulation transactions data is loaded, you must also create Circulation Policy Matrix Definitions. To do this in System Administration, select **Circulation> Policy Definitions> Matrix**. You must create these Circulation Definitions in order for Voyager to determine appropriate actions for specific patrons and item types. See the *Voyager System Administration User's Guide, Circulation Policy Matrix*, for more information.



IMPORTANT:

Before the circulation transaction data is loaded, do not circulate any items using Voyager.

Also note that item and patron barcode numbers in the circulation transaction records must match the barcode numbers in the previously loaded item and patron records.

This load creates transactions in the circulation module.

Vendor Data

Vendor data must be loaded after all bibliographic data has been loaded. Before the vendor data is loaded, do not create any vendor records in Acquisitions.

This vendor file load is an optional load. Doing the load, as opposed to manually entering the information in the System Administration module, depends on the number of vendors. The vendor file contains current address and contact information of vendors. It can include vendor types as well.

NOTE:

At least one vendor type must be set up in the System Administration module, if they were not included in the vendor file load.

After the vendor data has been loaded, add the appropriate vendor type definitions in the System Administration module. To define these values, select **System> Vendor Types**.

Also, you must define the Voyager base currency to match the default currency specified in the vendor data. To define a base currency, select, in the System Administration module, **System> Base Currency**. Only one currency is loaded, if your site uses more than one currency, add the appropriate additional currencies after the load using the Acquisitions module by selecting **Functions> Currency Maintenance**.

This load creates vendor address records in the Acquisitions module.

Introduction

Voyager is client server software where the database resides on a server and the clients that allow access to the database reside on user's computers.

Voyager® runs on Sun Solaris®, Red Hat Linux, or IBM® AIX®.



IMPORTANT:

It is assumed that the operator has basic knowledge of UNIX. If there are terms or procedures that are unfamiliar, call Ex Libris Customer Support.

Purpose of This Chapter

There are some basic server maintenance activities necessary for efficient functioning of Voyager.

This chapter discusses the following server activities:

- Services, scripts, dependencies and rationale
- Server procedures
 - Oracle Database Management Server (DBMS)
 - Voyager
 - Tomcat

- Apache™ web server procedures
- Backups

This information is specific for sites with a single server. Sites with multiple servers should contact Ex Libris Customer Support for the correct server procedures.

Services, Scripts, Service Dependencies and Rationale

The Voyager system uses the following services:

- Oracle - this is the actual database, the “back-end”, the place where the data is stored.
- Voyager - this is the brains, the way to organize, retrieve, and add to the database, the “front-end”.
- Tomcat - this service enables specific activities to occur. Voyager may run multiple instances of Tomcat. For example, one instance of Tomcat runs Patron Self Registration (PSR) if enabled, and another instance runs Link Finder Plus (LFP)
- Apache - this is a web server, its job is to provide, “serve”, web pages. It runs WebVoyage.

Scripts

There are several scripts provided by Ex Libris that automatically run various processes on the Voyager server to maintain efficient functioning.

The scripts are:

- dbora
- voyager
- httpd2

To run a start up script use the start parameter. To run a shut down script use the stop parameter.

For example:

- **dbora start** to start Oracle or **dbora stop** to shut down Oracle
- **voyager start** to start Voyager or **voyager stop** to shut down Voyager

- `httpd2 start` to start the Apache web server or `Apache stop` to shut down the Apache web server

There is also a backup script, see [Database Backup for Solaris and Linux Servers](#) on [page 3-11](#). Additionally multiple Crons are provided to run some batch jobs, see [Crons Set Up at Installation](#) on [page 3-19](#).

Service Dependencies

Dependencies occur when running a particular script on the server effects multiple services.

There are service dependencies when running some of the shut down scripts.



IMPORTANT:

There are no dependencies among the start up scripts. Therefore, if starting the machine manually each script must be run.

[Table 3-1](#) describes the service dependencies between the shut down scripts and the rationale for these dependencies.

Table 3-1. Service Dependencies

When Running Script...	This service shuts down...	and these dependent services shut down...	Rationale
dbora stop	Oracle	Voyager shuts down. Tomcat instances shut down.	If Oracle is shut down, Voyager and Tomcat cannot work, therefore, they automatically shut down. However, Apache is not dependent, and keeps running because it may be a web server for something other than Web-Voyage. This allows running Oracle without running Voyager.

Table 3-1. Service Dependencies

When Running Script...	This service shuts down...	and these dependent services shut down...	Rationale
voyager stop	Voyager	Tomcat instance for Patron Self Registration (PSR) shuts down.	If Voyager is shut down, then the Tomcat instance used for Patron Self Registration is also shut down because there is no access to Voyager. NOTE: Oracle, Apache, and the Tomcat instance used for LFP are not stopped.
httpd2 stop	Apache web server	No dependencies	Operators have the ability to shut down the web server without affecting Oracle, Voyager, and Tomcat.

Solaris, Linux, and AIX Servers

The following sections contain the server procedures for Sun Solaris, Linux, and IBM AIX servers.

Starting the Server

There is no specific procedure for starting the server. By powering on the server, start up scripts (provided by Ex Libris) run, which start Oracle, Voyager, Tomcat, and Apache.

Rebooting or Shutting Down the Server

Ex Libris recommends rebooting your system at least once a week. Other occasions when you might need to shutdown your server are adding new equipment, moving the server hardware, or installing operating system patches.

Solaris Servers

The procedure for rebooting or shutting down a Solaris server is shown in [Procedure 3-1, Rebooting or Shutting Down a Solaris Server](#).

NOTE:

The behavior of `/etc/init.d/voyager` on Solaris is as follows:

When the `start` argument is used, the script permanently enables services in the Service Management Facility (SMF) unless an additional `temporary` argument is included on the command line.



Procedure 3-1. Rebooting or Shutting Down a Solaris Server

Use the following to reboot or shut down your Solaris server safely.

1. Log in as root.

**CAUTION:**

Logging in as root gives the operator full access to the servers operating system and all of the software installed on the server. Use extreme caution when using this login.

2. Two options are available:
 - a. To reboot safely, that is, to shut down and restart the server, enter the following command:

`init 6`
 - b. To remove power from the server safely, turning the server off, enter the following command:

`init 5`

Result: The system reboots or powers off.

NOTE:

Powering off in this manner automatically runs all shutdown scripts. Rebooting automatically runs all shutdown scripts, except power off. It then resets the machine, running all of the start-up scripts.

Linux or AIX Servers

The procedure for rebooting or shutting down an AIX server is shown in [Procedure 3-2, Rebooting or Shutting Down an AIX or Linux Server](#).



Procedure 3-2. Rebooting or Shutting Down an AIX or Linux Server

Use the following to reboot or shut down your AIX or Linux server safely.

1. Log in as root.



CAUTION:

Logging in as root gives the operator full access to the servers operating system and all of the software installed on the server. Use extreme caution when using this login.

2. Two options are available:
 - a. To reboot safely, that is, to shut down and restart the server, enter the following command:

```
shutdown -r now
```
 - b. To remove power from the server safely, turning the server off, enter the following command:

```
shutdown -h now
```

Result: The system reboots or powers off.

NOTE:

Powering off in this manner automatically runs all shutdown scripts. Rebooting automatically runs all shutdown scripts, except power off. It then resets the machine, running all of the start-up scripts.

Stopping and Starting Oracle Services

Normally, you won't need to stop the Oracle services, however there will be some occasions, for example with a manual backup or to patch the operating system, where you may need to stop Oracle.

The procedure for stopping Oracle is shown in [Procedure 3-3, Stopping Oracle Services on a Solaris, Linux, or AIX Server](#).



Procedure 3-3. Stopping Oracle Services on a Solaris, Linux, or AIX Server

Use the following to stop Oracle.

1. Log in as oracle.
2. Enter the following command to switch the user to root:

```
su - root
```

3. Enter the following command:

```
/etc/init.d/dbora stop
```

Result: Oracle and dependent services stop (see [Service Dependencies](#) on [page 3-3](#)).

The procedure for starting Oracle is shown in [Procedure 3-4, Starting Oracle Services on a Solaris, Linux, or AIX Server](#).



Procedure 3-4. Starting Oracle Services on a Solaris, Linux, or AIX Server

Use the following to start Oracle.

1. Log in as oracle.
2. Enter the following command to switch user to root:

```
su - root
```

3. Enter the following command:

```
/etc/init.d/dbora start
```

Result: Oracle services start.

NOTE:

Only Oracle starts, there are no dependencies.

Stopping and Starting Voyager

Ex Libris suggests stopping Voyager once a day to kill any stale Voyager processes. This happens automatically when the nightly backup occurs since the backup stops Voyager (presuming you are running the Ex Libris supplied backup script.)

The procedure for stopping Voyager is shown in [Procedure 3-5, Stopping Voyager on a Solaris Server](#).



Procedure 3-5. Stopping Voyager on a Solaris Server

Use the following to stop Voyager.

1. Log in as root.
2. Enter the following command:

```
/etc/init.d/voyager stop
```

Result: Voyager and its dependent services stop (see [Service Dependencies on page 3-3](#)). For servers running Solaris 10, the system calls `inetadm` to disable Voyager ports.

The procedure for starting Voyager is shown in [Procedure 3-6, Starting Voyager on a Solaris Server](#).



Procedure 3-6. Starting Voyager on a Solaris Server

Use the following to start Voyager.

1. Log in as root.
2. Enter the following command:

```
/etc/init.d/voyager start
```

Result: Voyager services start.

NOTE:

The Voyager service starts. For servers running Solaris 10, the system calls `inetadm` to enable Voyager ports.

The procedure for stopping Voyager is shown in [Procedure 3-7, Stopping Voyager on a Linux or AIX Server](#).



Procedure 3-7. Stopping Voyager on a Linux or AIX Server

Use the following to stop Voyager.

1. Log in as root.
2. Enter the following command:

```
/etc/init.d/voyager stop
```

Result: Voyager stops, and its dependent services stop, see [Service Dependencies](#) on [page 3-3](#).

The procedure for starting Voyager is shown in [Procedure 3-8, Starting Voyager on a Linux or AIX Server](#).



Procedure 3-8. Starting Voyager on a Linux or AIX Server

Use the following to start Voyager.

1. Log in as root.
2. Enter the following command:

```
/etc/init.d/voyager start
```

NOTE:

Only Voyager starts, there are no dependencies.

Stopping and Starting the Apache Web Server

Ex Libris installs the Apache web server for the purpose of running WebVoyage, the web-based OPAC.

The procedure for stopping the Apache web server is shown in [Procedure 3-9, Stopping the Apache Web Server](#).



Procedure 3-9. Stopping the Apache Web Server

Use the following to stop the Apache web server.

1. Log in as oracle.
2. Enter the following command to switch user to root:

```
su - root
```

3. Enter the following command:

```
/etc/init.d/httpd2 stop
```

Result: The Apache web server stops.

The procedure for starting the Apache web server is shown in [Procedure 3-10, Starting the Apache Web Server](#).



Procedure 3-10. Starting the Apache Web Server

Use the following to start the Apache web server.

1. Log in as oracle.
2. Enter the following command to switch user to root:

```
su - root
```

3. Enter the following command:

```
/etc/init.d/httpd2 start
```

Result: The Apache web server starts.

Database Backup for Solaris and Linux Servers

Ex Libris requires sites to complete a backup of their data once each day. The administrator should perform the backup during off hours while the system is not operational (cold backup).

NOTE:

Backups that run while the system is still operational (hot backups) are generally not recommended because they require additional hardware and are more complex than a cold backup. Sites that want to run non-supported backups should contact Ex Libris Customer Support for approval.

Sites can run a backup of their database either manually by running the `/m1/utility/dailybackup` script or automatically by using the cron provided by Ex Libris.

At installation of your Voyager system, a cron is set up to perform the standard daily backup. This cron specifies the time that the backup should run and the appropriate commands to complete the backup. If you need assistance editing (or creating) a backup cron, contact Ex Libris Customer Support.

After the system runs the cron, the results are sent in an email to root. For more information on verifying the results of a cron, see [Reviewing Cron Backup Results \(Solaris and Linux\)](#) on [page 3-18](#).

Before you begin performing backups, verify that the following devices are ready:

- Backup device driver (see [Determining the Backup Device Driver](#) on [page 3-12](#))
- Data volume (see [Identifying your Data File System/Volume](#) on [page 3-15](#))

NOTE:

AIX sites need to contact Ex Libris customer support for assistance with backups. The discussion and procedures here are for the Solaris and Linux operating systems.

Backup Process

The following activities are performed by the system during a backup:

- Shutdown Apache web server, Voyager, and Oracle instances.
- Start snapshots.

- Start Apache web server, Voyager and Oracle instances.
- Dump the data to tape.
- End snapshots.



IMPORTANT:

Your institution is responsible for safeguarding its own data. If database corruption occurs, restoring the data from a backup will insure your data integrity and decrease downtime. If you require Ex Libris to reload your data because of insufficient backups, you will be charged for that service. If you have any problem with backing up your system using ufsdump, we will be able to provide support. We cannot provide assistance for any other means of backup.

Determining the Backup Device Driver

You must determine your backup device driver.

Most tape devices can record at multiple densities, such as low, medium, and high. High density is preferable because it allows for the quick backup of data, as well as more data written to the tape.



Procedure 3-11. Determining the Backup Device Driver for Linux

Use the following to determine the backup device.

1. Insert a tape of the appropriate size into your backup unit.
2. Log in as root.
3. Enter the following command:

```
ls /dev/st*
```

Result: This lists all the tape device drivers for which your server is configured. Your list will look similar to:

```
/dev/st0
```

```
/dev/nst0
```

The digit (0, zero) is the driver number.

The character `n` represents the non-rewinding driver. This is used when backing

up multiple file systems sequentially.

The characters following the digit can be `l`, `m`, `h`, `c`, and `u`. They correspond to the density of the tape drive listed from lowest (`l`) to highest (`u`).

4. Write down the driver number with the highest density mode available.

In the example above, `st0` is the driver number and highest density mode available.

5. You need to determine the density at which your system reads/writes tapes by testing the drivers. Test the high density driver first by entering:

```
mt -f /dev/### status
```

Where the `##` is the driver number/character combination for your highest density driver.

Therefore, following the example above, the command would be:

```
mt -f /dev/st0 status
```

Result: The system should respond by identifying the type of tape backup unit and if the unit is available ([Figure 3-1](#)). If available, then that is the density to which your system can read/write tapes and the driver number/character combination to use in your dump back-up command (manual backups). See [Database Backup for Solaris and Linux Servers](#) on [page 3-11](#).

```
SCSI 2 tape drive:
File number=0, block number=0, partition=0.
Tape block size 512 bytes. Density code 0x30 (AIT-1 or MLR3).
Soft error count since last status=0
General status bits on (41010000):
BOT ONLINE IM_REP_EN
```

Figure 3-1. Example output

If you get the message, `no tape loaded` or `drive offline`, that means you cannot read/write tapes at that density. Try the same test with a medium density driver, and so on until you find a density where you can read/write tapes.

6. Make a note of which device driver worked. You need it if you run the `dump backup` command manually.



Procedure 3-12. Determining the Backup Device Driver for Solaris

Use the following to determine the backup device.

1. Insert a tape of the appropriate size into your backup unit.
2. Log in as root.
3. Enter the following command:

```
ls /dev/rmt/*
```

Result: This lists all the tape device drivers for which your server is configured. Your list will look similar to:

```
/dev/rmt/0  
/dev/rmt/0b  
/dev/rmt/0bn  
/dev/rmt/0u  
/dev/rmt/0un
```

The digit (0, zero) is the driver number.

The character `n` represents the non-rewinding driver. This is used when backing up multiple file systems sequentially.

The characters following the digit can be `l`, `m`, `h`, `c`, and `u`. They correspond to the density of the tape drive listed from lowest (`l`) to highest (`u`).

The character `b` represents Berkeley-style tape positioning and should not be used.

4. Write down the driver number with the highest density mode available. In the example above, `0u` (zero `u`) is the driver number and highest density mode available.
5. You need to determine the density at which your system reads/writes tapes by testing the drivers.

Test the high density driver first by entering:

```
mt -f /dev/rmt/## status
```

Where the ## is the driver number/character combination for your highest density driver.

Therefore, following the example above, the command would be:

```
mt -f /dev/rmt/0u status
```

Result: The system should respond by identifying the type of tape backup unit and if the unit is available ([Figure 3-2](#)). If available, then that is the density to which your system can read/write tapes and the driver number/character combination to use in your `ufsdump` backup command (manual backups). See [Database Backup for Solaris and Linux Servers](#) on [page 3-11](#).

```
#mt -f /dev/rmt/0u status
Quantum DLT8000 tape drive:
  sense key(0x6)=Unit Attention residual=0 retries=0
  file no=0 block no=0
#
```

Figure 3-2. Example output

If you get the message, `no tape loaded` or `drive offline`, that means you cannot read/write tapes at that density. Try the same test with a medium density driver, and so on until you find a density where you can read/write tapes.

6. Make a note of which device driver worked. You will need it if you run the `ufsdump` backup command manually.
-

Identifying your Data File System/Volume

You need to identify in which file system(s) your data resides so that you can backup the correct file system(s).

For Voyager 2006.2 and earlier installations, there is an `/m1` file system (which is where Voyager resides) and an `/oracle` file system.

For Voyager 2006.5 and later installations, there is an `/m1` filesystem (where Voyager resides), an `/oracle` file system, an `/oracle/oradata` file system, and possibly more depending on the configuration.

Earlier installations of Voyager may have the `/m1` file system only.

**IMPORTANT:**

If you have more than one file system, such as `/m1` and `/oracle`, backups should be performed on each file system (such as `/m1` and `/oracle`). Otherwise, perform the backup on just the `/m1` file system.



Procedure 3-13. Identifying your Data File System for Linux

Use the following to determine the file system structure and, hence, which backup procedures your site should follow.

1. Type `df -k` and press **enter**.

Result: The output should look similar to the values in [Table 3-2](#). These values illustrate the file systems on your server along with the amount of space available on them.

Table 3-2. File systems on the server for Linux

File System	1K Blocks	Used	Available	Use %	Mounted On
/dev/mapper/rootvg	58405020	2964296	52426068	6%	/
/dev/xvda1	101086	13431	82436	15%	/boot
tmpfs	4599528	0	4599528	0%	/dev/shm
/dev/mapper/elvgg-m1	98051740	26123388	67943764	28%	/m1
/dev/mapper/elvgg-oracle	98051740	52246776	33971882	56%	/oracle

2. From the listing in Table 3-2, one can see that this system has both an `/oracle` and `/m1` file system.
3. If `/m1` is not listed, enter the following command:

```
ls /
```

Result: You should see `/m1` and possibly `/oracle` as well. If `/m1` is not present, call Ex Libris Customer Support for assistance.



Procedure 3-14. Identifying your Data File System for Solaris

Use the following to determine the file system structure and, hence, which backup procedures your site should follow.

1. Type `df -k` and press **enter**.

Result: The output should look similar to the values in [Table 3-3](#). These values illustrate the file systems on your server, along with the amount of space available on them.

Table 3-3. Filesystems on the server for Solaris

File System	Kbytes	Used	Available	Capacity	Mounted On
/dev/dsk/c0t3d0s0	620710	267758	290882	48%	/
/proc	0	0	0	0%	/proc
fd	0	0	0	0%	/dev/fd
/dev/dsk/c0t3d0s7	96455	40190	46625	47%	/export/home
/dev/md/dsk/d0	16684646	9558112	5458074	64%	/m1
/dev/md/dsk/d1	17654241	12912680	5241561	70%	/oracle
swap	304044	8	304036	1%	/tmp

2. From the listing in [Table 3-3](#) one can see that this system has both an `/oracle` and `/m1` file system.
3. If `/m1` is not listed, enter the following command:

```
ls /
```

Result: You should see `/m1` and possibly `/oracle` as well. If `/m1` is not present, call Ex Libris Customer Support for assistance.

Reviewing Cron Backup Results (Solaris and Linux)

After the system performs a scheduled backup, it sends an email containing the results of the backup to root. This email contains basic information and error messages that may occur during the backup.



Procedure 3-15. Reviewing the Email Generated by the Cron Backup

Use the following to access the e-mail message produced when the cronned backup runs.

1. Log in as root.
2. Enter the following command:

`mailx`
3. Select the corresponding e-mail number for the backup output.
4. Review the backup output message.

NOTE:

If you see messages that include the words *abort* or *failure*, this *could* indicate a problem. Call Ex Libris Customer Support during regular business hours to discuss this with a technician.

5. Type `q` and press **enter** to exit the backup output message.
-

Restarting tomcat

There are times when you may need to restart the tomcat service such as when making certain changes in Voyager System Administration. The procedure for restarting tomcat is shown in [Procedure 3-16, Restarting tomcat](#).

NOTE:

The `tsvrcctl` script used for restarting the tomcat service has been modified to retain log files from the past seven days instead of deleting all the log files when the tomcat service starts.



Procedure 3-16. Restarting tomcat

Use the following to restart the tomcat service.

1. Logon to the server as voyager.
2. Enter the following:

```
/m1/voyager/xxxdb/tomcat/tsvrcctl start
```

```
/m1/voyager/xxxdb/tomcat/tsvrcctl stop
```

NOTE:

Initiating a `tsvrcctl stop` and `tsvrcctl start` impacts availability of all standard tomcat services for vpds, PSR, vwebv, vxws and vprimo.

Stopping/Restarting Voyager PDS

If you need to stop or restart the Voyager PDS service, use:

- `/m1/voyager/xxxdb/pds/apache/bin/apachectl stop`
- `/m1/voyager/xxxdb/pds/apache/bin/apachectl start`

Restoring from Backup

If you need to restore your data from a backup contact Ex Libris Customer Support. For contact information, see http://www.exlibrisgroup.com/support_center.htm.

Crons Set Up at Installation

In addition to the backup cron provided at installation of your Voyager system, additional crons are set up to perform the following batch jobs daily:

- Acquisitions batch job 1, Order Claim/Cancel processing
- Circulation batch job 12, All Daily jobs
- Circulation batch job 24, Transaction Exceptions

See the *Voyager Reporter User's Guide for Order Claim/Cancel Processing (Acqjob 1)*, *All Daily Jobs (Circjob 12)*, and *Transaction Exceptions (Circjob 24)* that contains additional information about these jobs.

NOTE:

For any cron to run, both Oracle and Voyager must be up and running.

Voyager Client Installation and the Voyager.ini File

4

Introduction

Voyager software is modular and comprised of components that reside on servers as well as individual PCs. This chapter focuses on the download and installation of the Voyager software components that reside on a user's PC.

The following core Voyager clients can be installed on the user's PC:

- Acquisitions
- Call Slip
- Cataloging
- Circulation
- Global Data Change
- Reporter
- System Administration

Other files related to the Voyager client software are also installed on the user's PC. See [Additional Files Installed with the Client Installation](#) on [page 4-48](#).

Voyager extension modules may also be installed on the user's PC. Refer to the individual module guides for more information.

Prerequisites

For Voyager client prerequisites, see the Ex Libris Documentation Center for the most up-to-date list of supported PC operating system and hardware information.

Client Installation Options

With Voyager 8.0 and later, you have the following options for installing core Voyager client software on users' PCs:

- Traditional (manual method)

See [Downloading Voyager Client Files](#) on [page 4-3](#) and [Installing Voyager Clients Using the Manual Method](#) on [page 4-4](#).

- Automated

See [\[Upgrade\] Stanza - Voyager 7](#) on [page 4-32](#).

Once you have downloaded the core Voyager 8.0 and later client software to users' PCs, you have the option to configure AutoUpdate, a new client installation option, for subsequent releases of core Voyager client software. See [\[Upgrade\] Stanza - Voyager 8 and Later](#) on [page 4-44](#).



IMPORTANT:

With the Voyager 8.0 and later release of client software, the `[Upgrade]` stanza (in the `Voyager.ini` file) is being repurposed. If you use the Voyager 7 automated method of installing clients and want to use the Voyager 8 and later AutoUpdate method, you must configure the Voyager 8 and later `[Upgrade]` stanza using the instructions provided in [\[Upgrade\] Stanza - Voyager 8 and Later](#) on [page 4-44](#). For additional information regarding this change, see [Transition Considerations for AutoUpdate](#) on [page 4-2](#).

Transition Considerations for AutoUpdate

With Voyager 8.0 and later, the transition to the AutoUpdate method for the automatic downloading of core Voyager 8 and later client files to the users' PCs begins. The workflow for this transition is as follows:

1. Download the Voyager 8.0 and later clients using the same methods available with Voyager 7.
2. Modify the [Upgrade] stanza in the `Voyager.ini` file as defined in [\[Upgrade\] Stanza - Voyager 8 and Later](#) on [page 4-44](#).

This enables the PC to AutoUpdate the next version of Voyager 8.x and later software.

3. Point the Acquisitions and Cataloging client startup icons to the following .exe files:
 - `AcquisitionsStart.exe`
 - `CatalogStart.exe`

This is done by default when you install Voyager 8.0 and later for the first time.

See [AcquisitionsStart.exe and CatalogStart.exe](#) on [page 4-47](#) for additional information.

4. Download the new clients available after Voyager 8.0 and later using the AutoUpdate capability.

See [AutoUpdate on the Server](#) on [page 4-51](#) for information specific to the server environment.

NOTE:

In the case of a failed AutoUpdate, uninstall the clients (see [Uninstalling the Voyager Clients](#) on [page 4-15](#)) and use the manual installation method (see [Installing Voyager Clients Using the Manual Method](#) on [page 4-4](#)) to re-install the clients.

Downloading Voyager Client Files

The core Voyager files are packaged in the following .exe and .msi executable files:

- `VoyagerInstall.exe`
This installation method requires administrator privileges.
- `VoyagerInstall.msi` (Windows installer package version)

In a Windows XP environment, the .msi method of installation does not require administrator privileges to complete the installation process.

NOTE:

Administrator privileges are required for both methods of installation (.exe or .msi).

The executable files may be downloaded to your Voyager server from the Ex Libris FTP server. See [File Transfer Protocol \(FTP\) - Voyager Client Installations](#) on [page 4-61](#) for instructions regarding the FTP process. Once you have downloaded one of the Voyager client installation executable files, begin the installation process using the instructions in [Installing Voyager Clients Using the Manual Method](#) on [page 4-4](#).

For extension module clients such as Media Scheduling and ILL, refer to the client download and installation instructions provided in the user's guides available for those clients.

Installing Voyager Clients Using the Manual Method

The procedure for the manual method of installing the Voyager clients using the VoyagerInstall.exe installation file, for example, is shown in [Procedure 4-1, Installing Voyager Clients](#).



Procedure 4-1. Installing Voyager Clients

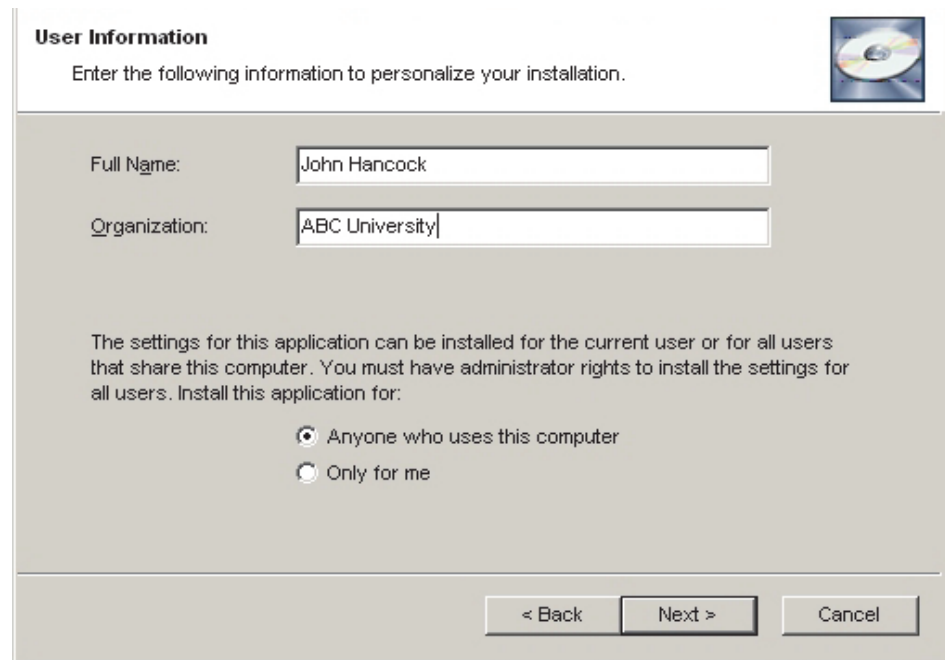
Use the following steps to install the Voyager clients on your computer using the VoyagerInstall.exe installation file. If you have a previous version of Voyager installed, please see Step 7 before proceeding with Step 1.

1. Right-click the VoyagerInstall.exe installation file that you saved and select **Run as Administrator**.

Result: The Windows Installer extracts the necessary files and displays the setup **Welcome** dialog box.

2. Read the setup Welcome information, and click the **Next** button when you are ready.

Result: The **User Information** dialog box opens. See [Figure 4-1](#).

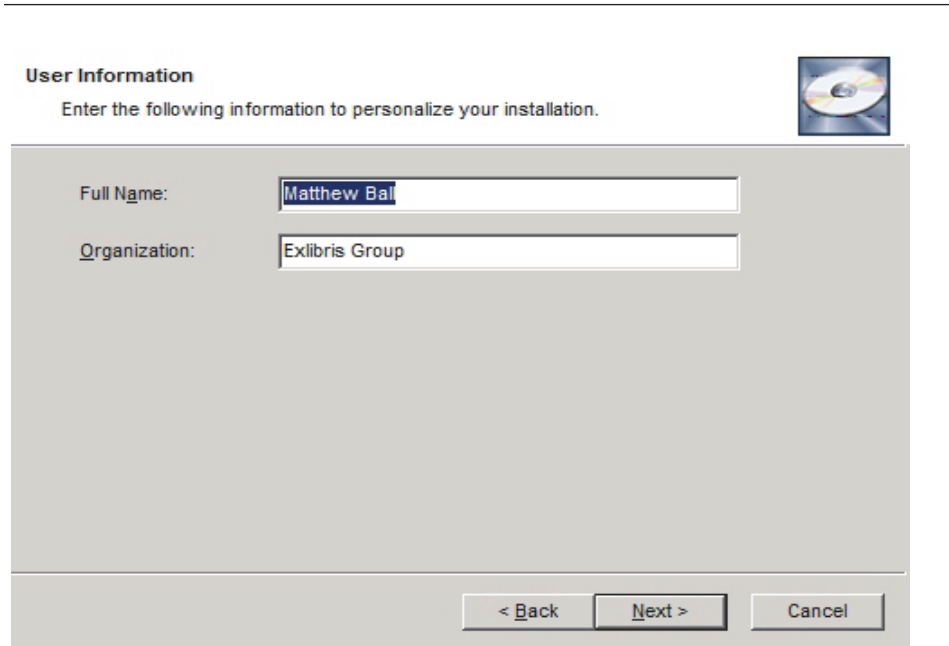


The dialog box is titled "User Information" and contains the instruction "Enter the following information to personalize your installation." It features two text input fields: "Full Name:" with the value "John Hancock" and "Organization:" with the value "ABC University". Below these fields, a message states: "The settings for this application can be installed for the current user or for all users that share this computer. You must have administrator rights to install the settings for all users. Install this application for:". There are two radio button options: "Anyone who uses this computer" (which is selected) and "Only for me". At the bottom right, there are three buttons: "< Back", "Next >", and "Cancel". A CD-ROM icon is located in the top right corner of the dialog box.

Figure 4-1. Voyager installation User Information dialog box

NOTE:

[Figure 4-1](#) illustrates the **User Information** dialog box presented to a user with administrator privileges. See [Figure 4-2](#) for an example of the **User Information** dialog box displayed for users without administrator privileges (using the `VoyagerInstall.msi` installation method).



The image shows a 'User Information' dialog box. At the top, it says 'User Information' and 'Enter the following information to personalize your installation.' There is a CD-ROM icon in the top right corner. Below the text, there are two text input fields. The first is labeled 'Full Name:' and contains the text 'Matthew Ball'. The second is labeled 'Organization:' and contains the text 'Exlibris Group'. At the bottom of the dialog box, there are three buttons: '< Back', 'Next >', and 'Cancel'.

Figure 4-2. User Information dialog box without administrator privileges

3. Enter Full Name, Organization, and select one of the following:
 - Anyone who uses this computer
 - Only for me

4. Click **Next**.

Result: The **Destination Folder** dialog box opens.

5. Identify the Destination Folder. The default folder for installation is C:\Voyager.
 - a. If this is the directory you want, click the **Next>** button.
 - b. If this is not the directory you want, click the **Browse** button and navigate to the correct directory and then click the **Next>** button. Do not choose a folder on a shared or network drive. The folder needs to be on a local drive.

Result: The directory into which the clients are installed is defined and the **Select Installation Type** dialog box opens. See [Figure 4-3](#).

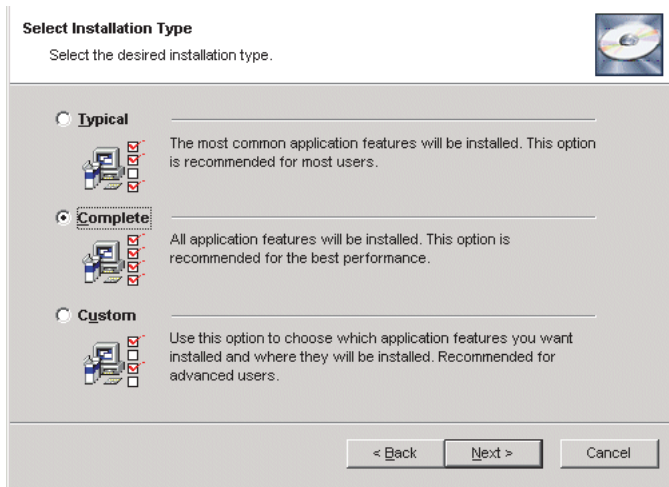


Figure 4-3. Voyager installation Select an Installation Type dialog box

6. Select the radio button corresponding to the type of installation you want to process, and click the **Next>** button.
 - **Typical** installs the Acquisitions, Circulation, and Cataloging clients.
 - **Complete** installs all the Voyager clients.
 - **Custom** allows the user to select specific clients. The Acquisitions, Circulation, and Cataloging clients are selected by default.

Result: The **Ready to install the Application** dialog box opens warning the user that files may be overwritten. See [Figure 4-4](#).

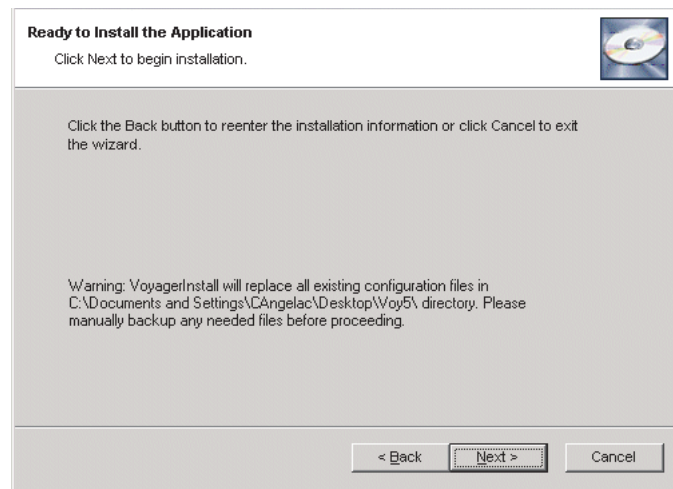


Figure 4-4. Warning Message

OPTIONAL:

7. *Back up any files that you may want to keep. For example, you may want to save any pre-existing configuration files like the Voyager.ini file to a different directory for later reference.*



IMPORTANT:

VoyagerInstall replaces all existing configuration files in the [C:\Voyager\] directory.

8. Click the **Next** button.

Result: The installation begins. See [Figure 4-5](#).

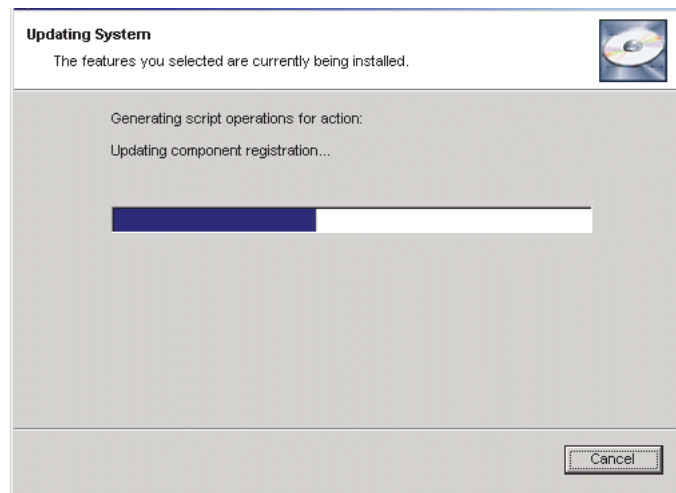


Figure 4-5. Installation in process

9. When the installation is complete, a dialog box displays indicating the build has been successfully installed.
10. Click the **Finish** button to exit the installation program.

Result: Voyager is installed on the user's PC.

11. Edit the `Voyager.ini` file to include the appropriate server and port information. See [The Voyager.ini File on the PC](#) on [page 4-17](#) for more information.

Result: The Voyager clients are installed and customized for your institution's use.

Modifying (Adding or Deleting Clients) an Installation

After installing the Voyager clients, users can modify their installation. They can add or delete Voyager modules.

For example, if you ran the typical installation of the Voyager clients that installs the Acquisitions, Cataloging, and Circulation clients, and you want to include the Callslip client, you can add it. Or, if you ran the complete installation, but do not need the System Administration module, you can delete it.

The procedure for modifying the Voyager installation is shown in [Procedure 4-2, Adding or Deleting Voyager Clients](#).



Procedure 4-2. Adding or Deleting Voyager Clients

Use the following to add or delete Voyager clients on a computer that has Voyager clients currently installed.

1. Click the **Start** button.
2. Select **Run** from the Window's **Start** menu.

Result: The **Run** dialog box displays and the cursor defaults to the **Open** field.

3. Enter *VoyagerInstall.exe*, and click **OK**.

Result: Files are extracted to the local computer and the **Select Installation Type** dialog box displays. See [Figure 4-6](#).

NOTE:

The **Modify** radio button is selected by default.

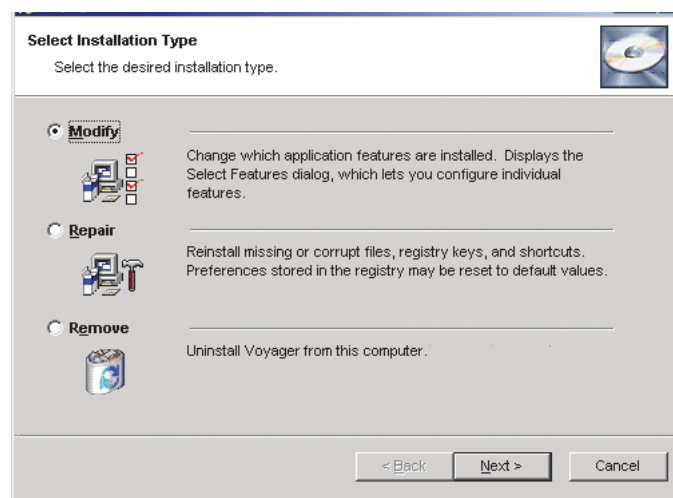


Figure 4-6. Select Installation Type dialog box

4. Click the **Next** button.

Result: The **Select Features** dialog box displays the features that are currently installed on the user's computer. See [Figure 4-7](#).

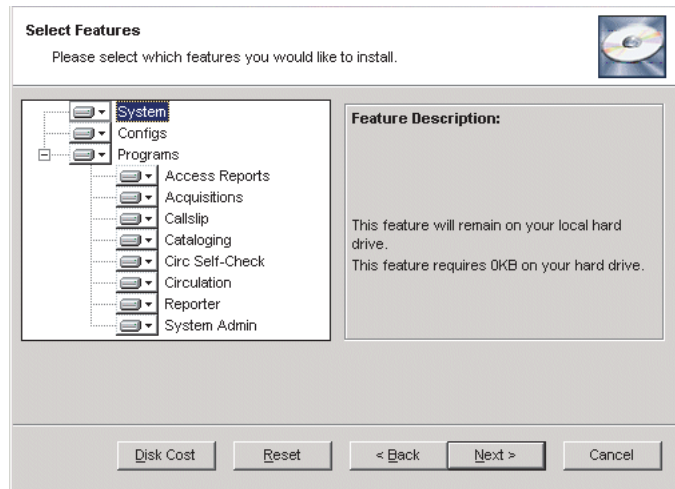


Figure 4-7. Voyager Setup with Selected Features

5. Select the clients you want to add or delete.

For this procedure, the example used is to delete the Voyager System Administration client. See [Figure 4-8](#).

Result: The System Administration client is marked for deletion. See [Figure 4-9](#).

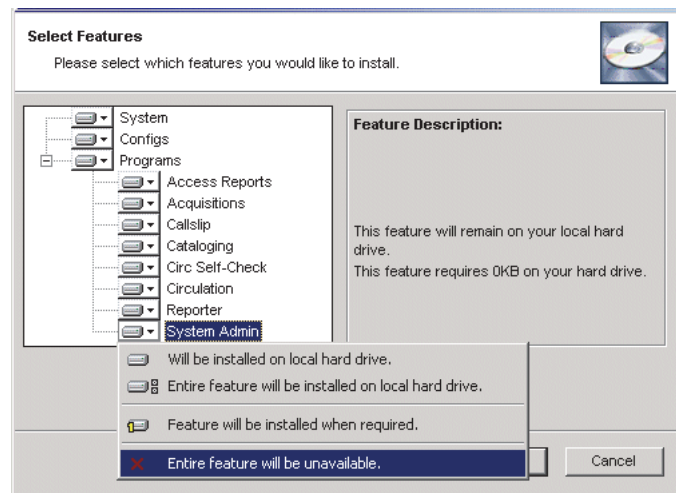


Figure 4-8. Selecting System Administration client for deletion

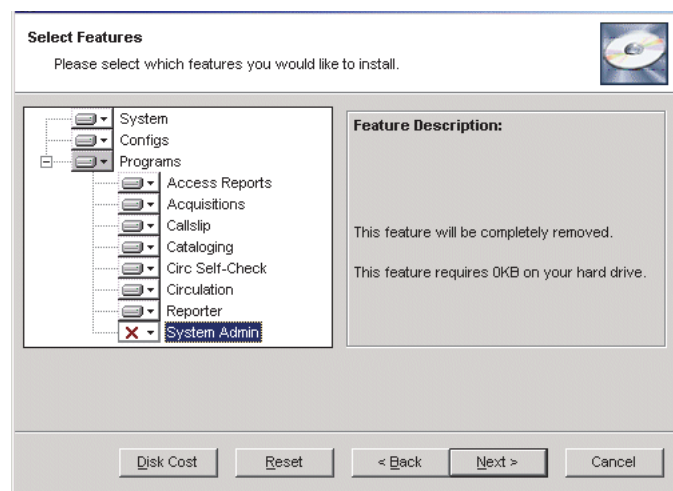


Figure 4-9. System Administration client identified for deletion

6. Click the **Next** button.

Result: The system is ready to modify the application. See [Figure 4-10](#).

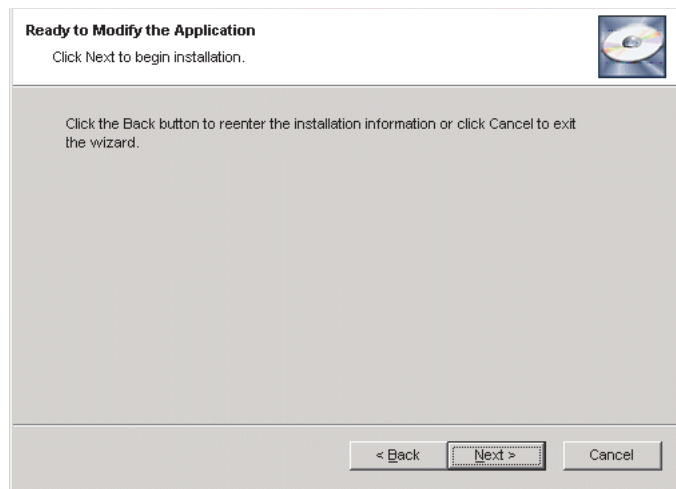


Figure 4-10. Voyager ready to modify

7. Click the **Next** button.

Result: The modification of the installation occurs and displays a confirmation message that it was successful.

8. Click **Finish**.

Result: The installation is modified.

Repairing an Installation of Voyager Clients

If instructed by Ex Libris's Customer Support department or you suspect that your installation was not complete, you can repair the installation.

The procedure for repairing a Voyager installation is shown in [Procedure 4-3, Repairing an Installation](#).



Procedure 4-3. Repairing an Installation

Use the following to repair a Voyager client installation.

1. Click the **Start** button.
2. Select **Run** from the Window's **Start** menu.

Result: The **Run** dialog box displays and the cursor defaults to the **Open** field.

3. Enter *VoyagerInstall.exe*, and click **OK**.

Result: Files are extracted to the local computer and the **Select Installation Type** dialog box displays.

4. Click the **Repair** radio button and subsequent the **Next** button.

Result: The **Ready to Repair the Application** dialog box displays. See [Figure 4-11](#).

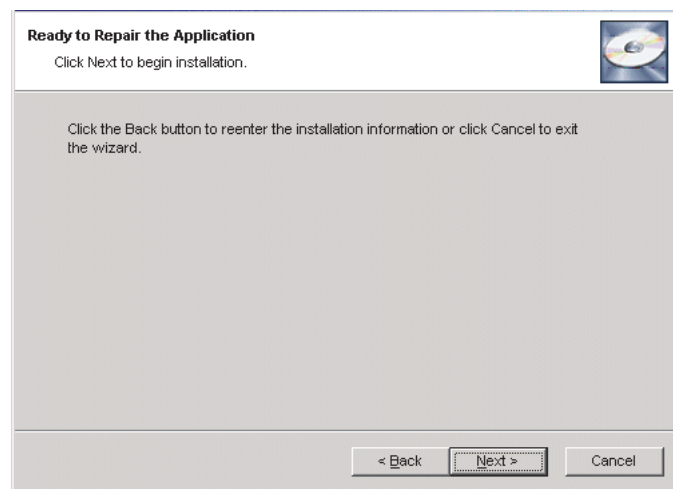


Figure 4-11. Voyager Ready to Repair the Installation

5. Click the **Next** button.

Result: The repair occurs and displays a confirmation message that it was successful.

6. Click **Finish**.

Result: The installation is repaired.

Uninstalling the Voyager Clients

If needed, users should uninstall Voyager clients using Windows, not the installation program.

Before uninstalling the clients, be sure to save any customized files you may have such as tag table files, archived files, and access reports or queries.

The procedure for uninstalling the Voyager clients is shown in [Procedure 4-4](#), [Uninstalling Voyager Clients](#).



Procedure 4-4. Uninstalling Voyager Clients

Use the following to uninstall the Voyager clients.

1. Access the Windows control panel and select **Add/Remove** programs. See [Figure 4-12](#).

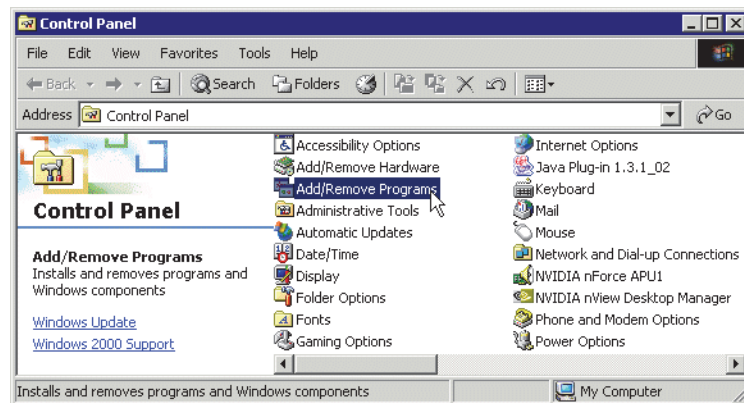


Figure 4-12. Control Panel with Add/Remove Programs Selected

Result: The **Add/Remove** dialog box displays.

2. Select the Voyager program and click the **Remove** button. See [Figure 4-13](#).

NOTE:

The version of the Voyager program that displays is the version that you installed prior to beginning to use AutoUpdate.



Figure 4-13. Add/Remove Dialog Box with Voyager Selected

Result: The system asks for confirmation before removing the program. See [Figure 4-14](#).

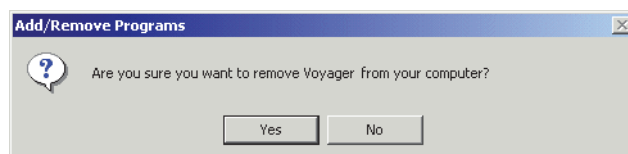


Figure 4-14. Confirmation of Removing Voyager

3. Click **Yes** to remove Voyager.

Result: Voyager is removed and the **Add/Programs** dialog box updates.

4. Close the **Add/Remove** dialog box.

Result: Voyager is uninstalled.

The Voyager.ini File on the PC

The `Voyager.ini` file is an initialization file that contains important connection information that enables your Voyager clients to access the server.

During the download process, the `Voyager.ini` file is placed in the directory with the client software. This is typically the `c:\Voyager` directory.

NOTE:

The `Voyager.ini` file is overwritten each time the `VoyagerInstall.exe` is executed.

The `Voyager.ini` file contains the following stanzas:

- [Cataloging]
- [Acquisitions]
- [Circulation]
- [Reports]
- [SysAdmin]
- [CallSlip]
- [Media Scheduling]
- [GlobalDataChange]
- [GlobalLog]
- [SearchURI]
- [MARC POSTing]
- [E-mail]
- [Upgrade]
- [HelpMenuLink]

[Figure 4-15](#) shows an example of a `Voyager.ini` file.

```
[Cataloging]
Server=xxx.xxx.xxx.xxx
Port=7010
Timeout=60

[Acquisitions]
Server=xxx.xxx.xxx.xxx
Port=7020
Timeout=60

[Circulation]
Server=xxx.xxx.xxx.xxx
Port=7030
Timeout=60
ChargeTimeout=60

[Reports]
Server=xxx.xxx.xxx.xxx
Port=7040
Timeout=60

[SysAdmin]
Server=xxx.xxx.xxx.xxx
Port=7050
Timeout=60

[CallSlip]
Server=xxx.xxx.xxx.xxx
Port=7080
Timeout=60

[MediaScheduling]
Server=xxx.xxx.xxx.xxx
Port=7085
Timeout=60
```

Figure 4-15. Sample Voyager.ini file

```
[GlobalDataChange]
Server=xxx.xxx.xxx.xxx
Port=7015
Timeout=60000

[GlobalLog]
SingleLogin=Y
Encrypt=N
ServerSortList=Y
ASCIISortList=Y
ASCIISortColumn=Y

[SearchURI]
#Name=Barn&esNoble
#URI=http://search.barnesandnoble.com
#Copy=Y
#SearchSyntax=/booksearch/results.asp?WRD=<searchtext>
#Name=WebVoyage
#URI=http://xxx.xxx.xxx.xxx/
#Copy=Y
#SearchSyntax=/vwebv/
        search?searchArg=<searchtext>&searchCode=TALL&limitTo=none&recCount=
        10&searchType=0
#Name=Amazon
#URI=http://www.amazon.com
#ECopy=Y
#SearchSyntax=/exec/obidos/external-search?keyword=<searchtext>&mode=blended
#Name=Worldcat
#URI=http://www.worldcat.org/
#Copy=Y
#SearchSyntax=search?&q=
#Name=Google
#URI=http://www.google.com/
#Copy=Y
#SearchSyntax=search?&q=

[MARC POSTing]
#Classic WebVoyage
```

Figure 4-15. Sample Voyager.ini file (Continued)

```
#WebVoyage="http://xxx.xxx.xxx.xxx/cgi-bin/Pbibredirect.cgi"
#Tomcat WebVoyage
WebVoyage="http://xxx.xxx.xxx.xxx/vwebv/holdingsInfo"
#Classic WebVoyage
#LinkResolver="http://xxx.xxx.xxx.xxx/cgi-bin/Phttplinkresolver.cgi"
#Tomcat WebVoyage
LinkResolver="http://xxx.xxx.xxx.xxx/vwebv/linkResolver"

[E-mail]
Server=xxx.xxx.xxx.xxx
Port=25

# This stanza replaces the per-stanza NewVersion= in Voyager 7.0
[Upgrade]
Voyager=
Media=

[HelpMenuLink]
Ex Libris Doc Portal=http://www.customercenter.exlibrisgroup.com/
```

Figure 4-15. Sample Voyager.ini file (Continued)

Client Module Stanzas

The client module stanzas set up the client/server connections for each module.

[Table 4-1](#) lists the parameters that can be specified for the following client module stanzas:

- [Acquisitions]
- [Call Slip]
- [Cataloging]
- [Circulation]
- [GlobalDataChange]
- [MediaScheduling]

- [Reports]
- [Sysadmin]

Table 4-1. Module Stanza Parameters

Parameter	Description
Server= <i>value</i>	This parameter contains the IP address of the Voyager server.
Port= <i>value</i>	This parameter corresponds to each module's designated port, as defined by the <code>/etc/services</code> file on the Voyager server.
Timeout= <i>value</i>	This parameter specifies the number of seconds that are allowed to elapse while attempting to connect the client to the Voyager server.
ChargeTimeout= <i>value</i>	This parameter defines the charge timeout value for the [Circulation] stanza only. For more information, see Circulation Charge Timeout on page 4-21 .

Circulation Charge Timeout

When charging items in Voyager's Circulation module, the system is configured such that the **Charge** workspace closes automatically. This occurs if, after entering the charging patron's information, a set amount of time passes with no activity occurring in the **Charge** workspace. This set amount of time is the charge time-out value.

The `ChargeTimeout=` key is in the [Circulation] stanza of the `Voyager.ini` file. The default charge time-out value is 60 seconds (see [Figure 4-15](#) on [page 4-18](#)).

[Table 4-2](#) shows the possible time-out values for the `ChargeTimeout=` key and Voyager's response.

Table 4-2. Circulation ChargeTimeout Key Values

ChargeTimeout Value	System Response
Blank	No charge timeout, the timer will not count.
0	No charge timeout, the timer will not count.

Table 4-2. Circulation ChargeTimeout Key Values

ChargeTimeout Value	System Response
1 - 4	The system will count 5 seconds.
> 4	The timer will count the number of seconds specified. There is no maximum value. NOTE: The default charge timeout value is 60 seconds.

See *Configuring the Voyager.ini file for Charge time-out* in the *Voyager Circulation User's Guide* for additional information on Circulation Charge Time-out.

[GlobalLog] Stanza

The [GlobalLog] stanza allows you to configure other options (such as single login, encryption, and staff client re-sort).

[Table 4-3](#) describes the parameters in the [GlobalLog] stanza.

Table 4-3. [GlobalLog] Stanza Parameters

Parameter	Description
SingleLogin= <i>value</i>	This parameter enables a single login screen for all modules so that a user does not have to login for each module. Set the <i>value</i> to Y to enable or N to disable. For more information, see Single Client Login on page 4-23 .
Encrypt= <i>value</i>	The default setting for the Encrypt= parameter is N. NOTE: An Encrypt= value of Y is ignored.

Table 4-3. [GlobalLog] Stanza Parameters

Parameter	Description
ServerSortList= <i>value</i>	This parameter enables the Sort By drop-down menu on search results screens from the Cataloging, Circulation, and Acquisitions modules. From this menu, the following items are made available: • Title • Author • Publish Date • Publish Date Descending • Relevance (relevance searches only) Default <i>value</i> is Y. For more information on Voyager-aware re-sorts, see Additional Files Installed with the Client Installation on page 4-48.
ASCIISortList= <i>value</i>	This parameter enables the Sort By drop-down menu on search results screens from the Cataloging, Circulation, and Acquisitions modules. From this menu, Quicksort items, which are user-defined in the System Administration module, are made available. Default <i>value</i> is N. For more information on ASCII re-sorts, see Additional Files Installed with the Client Installation on page 4-48.
ASCIISortColumn= <i>value</i>	This parameter enables column headers on search results screens to be clicked to re-sort search results from the Cataloging, Circulation, and Acquisitions modules. Default <i>value</i> is N. For more information on ASCII re-sorts, see Additional Files Installed with the Client Installation on page 4-48.

Single Client Login

The Single Client Login feature allows you, after logging in to one of the Voyager modules, to open subsequent staff applications (modules) without having to re-enter your user name or password. Single Client Login is configured on each personal computer, so you can enable it on some computers and not on others. To enable the Single Client Login feature, set the `SingleLogin` parameter in the `[GlobalLog]` stanza of the `Voyager.ini` file to Y. See [\[GlobalLog\] Stanza](#) on [page 4-22](#) for additional configuration information.

Since Single Client Login sends the username and password from module to module, the only visible change after enabling it is that there is no login screen.

The following are issues pertaining to Single Client Login.

- If multiple users are using multiple instances of the same module on the same computer, or if multiple users have access to a single computer, you may not want to enable Single Client Login.
- Single Client Login uses the most recent password entered.
- If a user logs in to one module and later attempts to log in to another module for which that user does not have privileges, the login dialog box is invoked for the second module. After logging in to that module, the second login is used for all subsequent modules opened.

This second login remains in effect until or unless all modules are closed, or another login failure occurs and a new user logs in.

- All normal security checks are performed when passwords and usernames are sent to modules using Single Client Login. Refer to the *Voyager System Administration User's Guide* for more information regarding security.

Staff Client Re-Sort

This feature allows users to re-sort their search results from the Cataloging, Circulation, and Acquisitions modules. The search results may be re-sorted in the following methods:

- Voyager-aware re-sort via drop-down menu.
- Quick Sort via drop-down menu.
- Quick Sort via clickable column header.

The Staff Client Re-sort feature is enabled when the corresponding parameters are set in the `[GlobalLog]` stanza of the `Voyager.ini` file. For more information, see [\[GlobalLog\] Stanza](#) on [page 4-22](#).

Both Voyager-aware and Quick Sorts become available when a bibliographic search (such as a left-anchored title search) returns a list of titles on the results screen (see [Figure 4-16](#)).

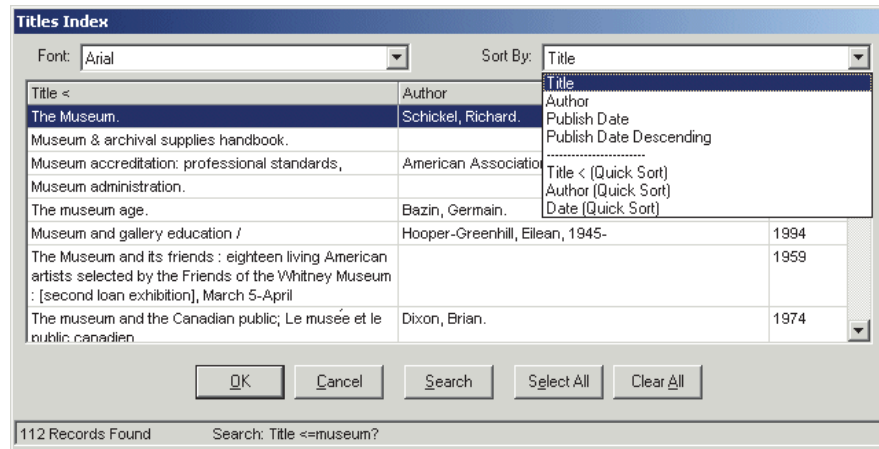


Figure 4-16. Search Results Page with Voyager-Aware and Quick Sort Drop-Down

NOTE:

The Re-sort capability is not available on search results screens for the following: searches that return a list of headings, browse searches, remote Z39.50 connections, remote connections to non-Voyager catalogs, and remote connections to multiple Voyager catalogs.

Voyager-Aware Re-Sorts

After a search is performed, the following menu items display in the **Sort By** drop-down menu (see [Figure 4-17](#)) if the Voyager-Aware feature is enabled:

- Title
- Author
- Publish Date
- Publish Date Descending
- Relevance (appears after relevance searches only)

If the user selects any of these items, the search is performed again, sorting the results by the criteria indicated by the selected item. Unless indicated by the menu item, the results are sorted in ascending order, recognizing non-filing characters.

NOTE:

Since the system queries the database again, search results may be different if more than 10,000 titles were found in the initial search.

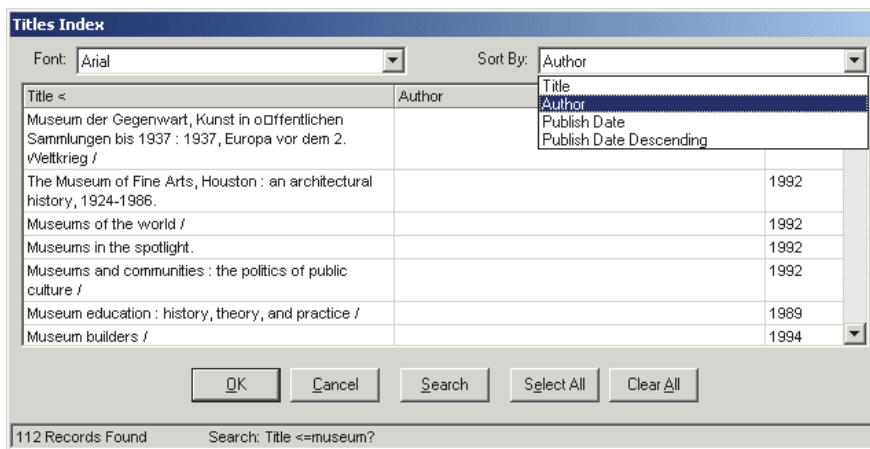


Figure 4-17. Search Results with Voyager-Aware Drop-Down Menu

To enable the Voyager-Aware drop-down menu, set the `ServerSortList` parameter in the `[GlobalLog]` stanza. For more information, see [\[GlobalLog\] Stanza](#) on [page 4-22](#).

Quick Sorts

Quick Sorts allow users to re-sort their search results using a sort criteria and to display the results alphanumerically ignoring non-filing characters.

To perform a quick sort, the user may select a sort criteria from the **Sort By** drop down menu (see [Figure 4-18](#)) or click a column header. The results can be viewed in descending order by selecting the same criteria a second time.

NOTE:

The **Sort By** drop-down menu items are defined in the System Administration module by assigning bib text fields to the search results of a search configuration under the **Search** menu. For more information, see the System Administration User's Guide.

The following parameters in the `[GlobalLog]` stanza are used to enable the Quick Sort features:

- The `ASCIISortList` parameter enables the ASCII drop-down menu.
- The `ASCIISortColumn` parameter enables clickable column headers.

For more information, see [\[GlobalLog\] Stanza](#) on [page 4-22](#).

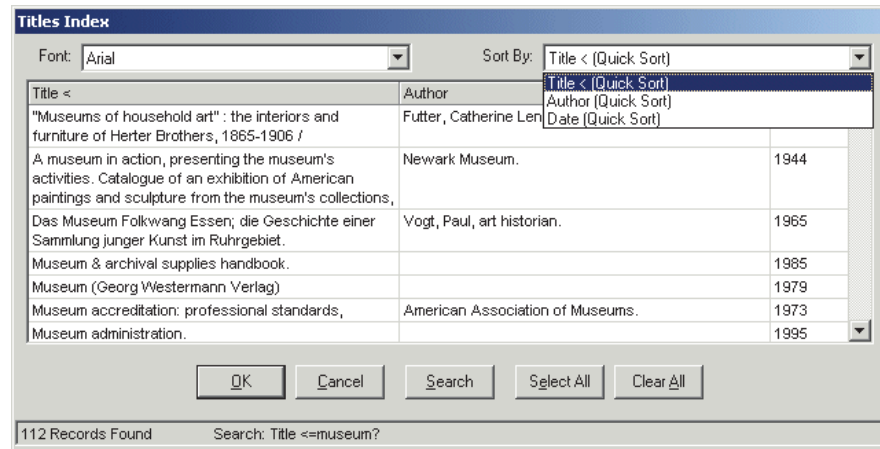


Figure 4-18. Search Results with Quick Sort Drop-Down Menu

[E-mail] Stanza

The [E-mail] stanza specifies the email server and port connections. [Table 4-4](#) describes the parameters in the [E-mail] stanza.

Table 4-4. [E-mail] Stanza Parameters

Parameter	Description
<code>Server=value</code>	Specifies the IP address of the e-mail server.
<code>Port=value</code>	Specifies the port connection on the e-mail server.

This stanza must be configured to allow e-mailing notices produced in the Reporter module. See *Configuring the Voyager.ini file for E-mail* in the *Voyager Reporter User's Guide* for information on this stanza and its configuration.

[MARC Posting] Stanza

The [MARC POSTing] stanza in the `Voyager.ini` file contains a user-defined key-value pair. The key is the text that displays in the module on the drop-down menu listing where the user wants to send and display the MARC record. The value is the URI text to generate the record and an XML-formatted version of the record.

MARC Record Posting Feature

The MARC record posting feature allows the operator to use an http POST request to send a MARC record to a web server for display. The POST request contains:

- the MARC record
- the record type: Bibliographic, Authority, or Holdings
(If you are posting to WebVoyage, only bibliographic records can be posted.)
- the record ID: BibID, AuthID, or MFHDID depending on the record type

For MARCPosting to work, the [MARC POSTing] stanza in the `Voyager.ini` file must be configured; and in order to display the POST, the `Pbibredirect.cgi` (for Classic WebVoyage), `holdingsInfo` (for Tomcat WebVoyage), and `Phttplinkresolver.cgi` scripts are on the Voyager server to receive the records.

Sending a POST request can be done from any of the Voyager modules where a MARC record is visible.

Example [MARC POSTing] Stanza

For example, you can add a key for WebVoyage, and the value would be the IP address to your WebVoyage. A sample [MARC POSTing] stanza is shown in [Figure 4-19](#) (Classic WebVoyage) and [Figure 4-20](#) (Tomcat WebVoyage).

```
[MARC POSTing]
WebVoyage="http://<host>:<port>/cgi-bin/Pbibredirect.cgi"
```

Figure 4-19. Sample [MARC POSTing] stanza (Classic WebVoyage)

```
[MARC POSTing]
WebVoyage="http://<host>:<port>/vwebv/holdingsInfo"
```

Figure 4-20. Sample [MARC POSTing] stanza (Tomcat WebVoyage)

If the `Voyager.ini` contains a `[MARC POSTing]` stanza, from an open record:

- In the Cataloging module, the **Record** menu contains a **Send Record To** option that allows you to select from the key values in the `[MARC POSTing]` stanza ([Figure 4-21](#)).

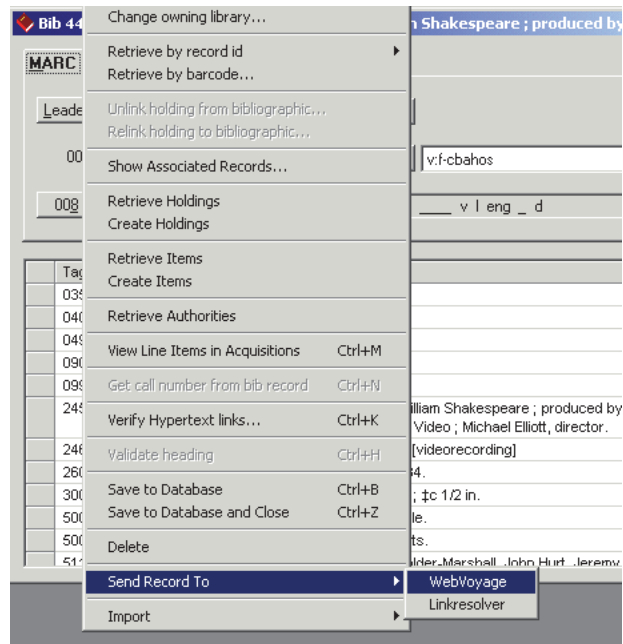


Figure 4-21. Send Record To option

- In all other modules, a **Send Record To** button displays in the MARC record that allows you to select from the key values in the `[MARC POSTing]` stanza ([Figure 4-22](#)).

Tag	Ind 1	Ind 2	Bibliographic Data
010			\$a 70187121
020			\$a 0151478104 \$a 0155492578 (pbk)
035			\$a 70187121
049			\$a SHCM
050	0		\$a P106 \$b .C52 1972
082			\$a 401/.9
092			\$a 410 C54l
100	1		\$a Chomsky, Noam.
245	1	0	\$a Language and mind.
250			\$a Enl. ed.
260			\$a New York, \$b Harcourt Brace Jovanovich \$c [1972]
300			\$a xii, 194 p. \$c 24 cm.

Figure 4-22. Send Record To button and drop-down as seen in the Acquisitions and Circulation modules

NOTE:

If more than one key is provided, they are listed alphabetically in the drop-down menu.

Selecting Send Record To

When the **Send Record To** option (or button) is selected, the system executes the following steps:

1. Generates a temporary HTML file with the record, record type, and record ID.
2. Passes the post the browser, opens the form, and displays the record.

NOTE:

- The MARC data is encoded in UTF-8.
- Characters are HTML escaped as part of the request.

[SearchURI] Stanza

The [SearchURI] stanza of the `Voyager.ini` file configures and enables an additional button on the bibliographic **Search** dialog box to open a Uniform Resource Identifier (URI).

[Table 4-5](#) describes the parameters used in the [SearchURI] stanza.

Table 4-5. [SearchUri] Stanza Parameters

Parameter	Description
Name= <i>value</i>	This parameter provides a label for the additional button on the Search dialog box.
URI= <i>value</i>	This parameter contains the base of the URI. NOTE: The base is the leftmost section of the address, for example, the web address portion.
Copy= <i>value</i>	This parameter is either ' <i>Y</i> ' for yes, or ' <i>N</i> ' for no. If yes, the text in the Search for field is used as the search term(s).
SearchSyntax= <i>value</i>	This parameter provides the syntax (such as <code>/search?&q=<search text></code>) for the search. The system takes the base (URI) and combines it with the search syntax (SearchSyntax) to create the search. NOTE: Users are responsible for the correct search syntax.

Adding a Button to the Search Dialog Box

If you would like to add an additional button to the Search dialog box, configure the [SearchURI] stanza. A sample configuration is shown in [Figure 4-23](#).

```
[SearchURI]
Name=Go to Google
URI=http://www.google.com/
Copy=Y
SearchSyntax=/search?&q=<search text>
```

Figure 4-23. Sample [SearchURI] Stanza

Using the sample configuration, [Figure 4-24](#) shows a “Go to Google” button on the **Search** dialog box. When this button is selected the Google™ website is accessed. The search terms from the **Search** dialog box are inserted into the Google search box and a Google search is executed using those terms.

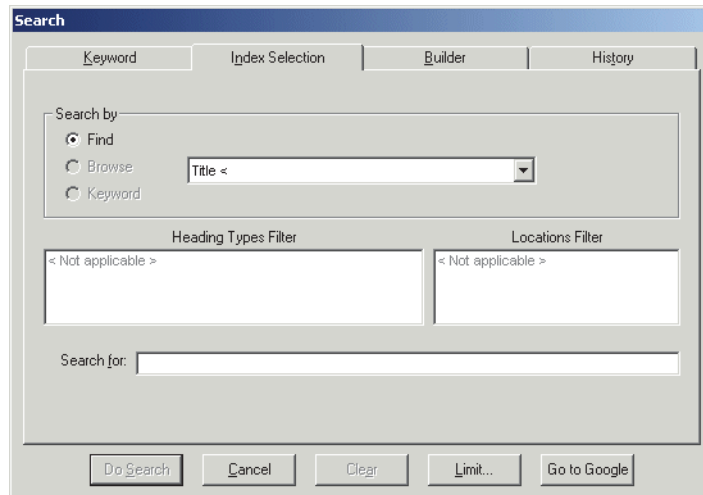


Figure 4-24. “Go to Google” Button Added to Search Dialog Box

[Upgrade] Stanza - Voyager 7

The **Voyager 7 [Upgrade]** stanza provides the facility for automating the download of client software to a user’s PC when a newer version is detected by the user’s PC.

The **[Upgrade]** stanza contains the following parameters:

- **Voyager=**
- **Media=**

See [Figure 4-15](#) on [page 4-18](#) for an illustration of the complete **Voyager.ini** file that includes the **[Upgrade]** stanza.

The **Voyager=** parameter is used to initiate the download and installation of new Voyager client software, and the **Media=** parameter is used to initiate the download and installation of new Media Scheduling software (if this applies to your site).

Your system administrator needs to provide you with the content/format of the parameter option that is specific to your site for accessing the client installation from a server. This points to the location of the installation software. See the examples in [Figure 4-25](#), [Figure 4-26](#), [Figure 4-27](#), and [Figure 4-28](#).

```
[Upgrade]
Voyager=\\Voyager\Installs\VoyagerInstall.exe
Media=\\Voyager\Installs\VoyagerMedia.exe
```

Figure 4-25. [Upgrade] stanza example using a valid UNC resource path

```
[Upgrade]
Voyager=http://10.102.100.91:37908/VoyagerInstall.exe
Media=http://10.102.100.91:37908/MediaInstall.exe
```

Figure 4-26. [Upgrade] stanza example using an http resource

```
[Upgrade]
Voyager=ftp://username:password@10.102.100.91/m1/voyager/clients/VoyagerIn-
stall.exe
Media=ftp://username:password@10.102.100.91/m1/voyager/clients/MediaIn-
stall.exe
```

Figure 4-27. [Upgrade] stanza example using an ftp resource

```
[Upgrade]
Voyager=\\Voyager\Installs\ABCUniversityVoyagerInstall.bat
Media=\\Voyager\Installs\ABCUniversityMediaInstall.bat
```

Figure 4-28. [Upgrade] stanza example using a .bat file

The `Voyager=` and `Media=` parameters may point to either the `.exe` or `.msi` type of installation.

NOTE:

If the PC operating system is Windows XP, using the `.msi` method does not require administrator privileges to complete the installation process. However, for Vista and Windows 7, administrator privileges are required for either method of installation, `.exe` or `.msi`.

The `Voyager=` and `Media=` parameters are considered inactive in the following situations:

- When not included in the stanza.
- When commented out.
- When the value is blank.

New Software Version Detected?

When a Voyager client is started and the system determines that there is a newer version of the client available, the user receives a version available message. See [Figure 4-29](#).

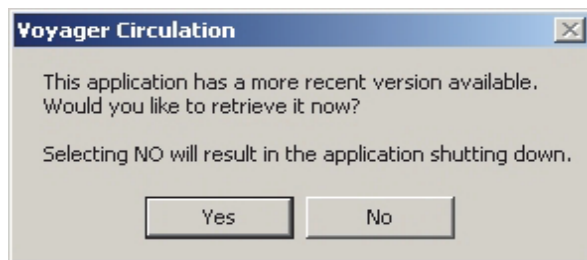


Figure 4-29. Version available message

Selecting Yes

When the user clicks **Yes**, the next step varies depending on the values stored in `Voyager=` and `Media=`.

- When the `Voyager=` value for the traditional method is used as in [Figure 4-25](#), see [Traditional Method](#) on [page 4-35](#).
- When the `Voyager=` value for the HTTP method is used as in [Figure 4-26](#), see [HTTP Method](#) on [page 4-36](#).

- When the `Voyager=` value for the FTP method is used as in [Figure 4-27](#), see [FTP Method](#) on [page 4-39](#).
- When the `Voyager=` value for the .BAT method is used as in [Figure 4-28](#), see [.BAT File Method](#) on [page 4-39](#).

Selecting No - Circulation Offline Option

In general, the application shuts down when the **No** button is selected (see [Figure 4-29](#)). However, when the Circulation client determines there is a new version and the user clicks the **No** button when prompted to install the new software, the system treats this as an unsuccessful attempt to connect to the Circulation module. When this happens, the Offline circulation message displays. See [Figure 4-30](#).

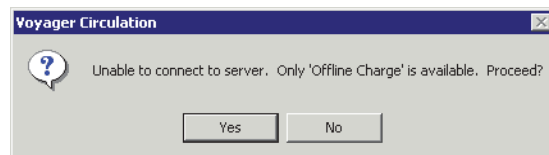


Figure 4-30. Offline Circulation Message

For more information regarding offline circulation, see the *Voyager Circulation User's Guide*.

Traditional Method

The system begins to process the installation executable from the server and displays the installation **Welcome** dialog box. See [Figure 4-31](#).



Figure 4-31. Installation Welcome dialog box

The remainder of the installation follows the steps identified in [Procedure 4-1, Installing Voyager Clients](#), on page [4-4](#). The Media Scheduling installation follows a similar procedure.

HTTP Method

The system needs to validate your security. See [Figure 4-32](#).

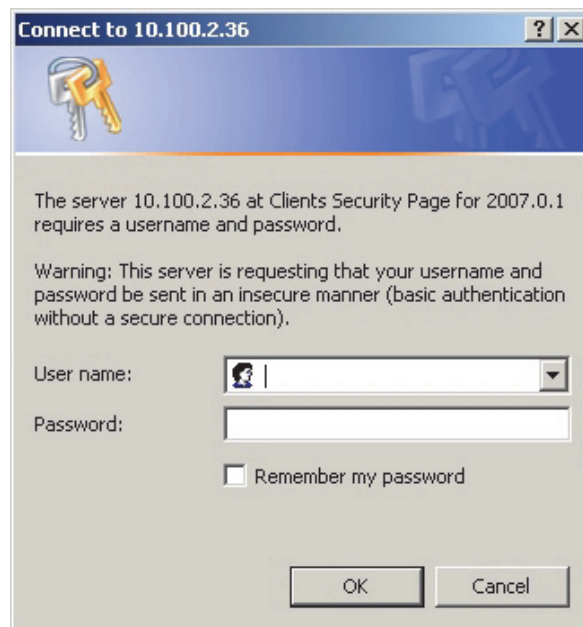


Figure 4-32. Security page (authentication)

Enter the User ID, password, and click OK. The file download dialog box displays. See [Figure 4-33](#).

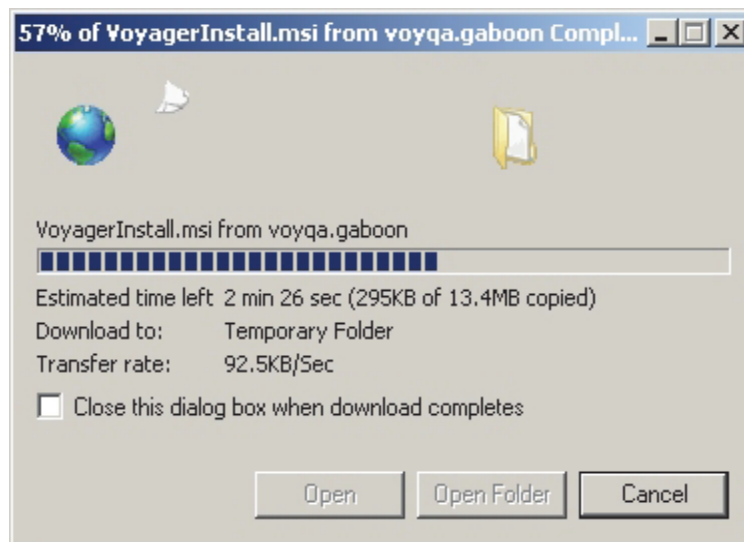


Figure 4-33. File download dialog box

Subsequently, the file download security warning dialog box displays. See [Figure 4-34](#).

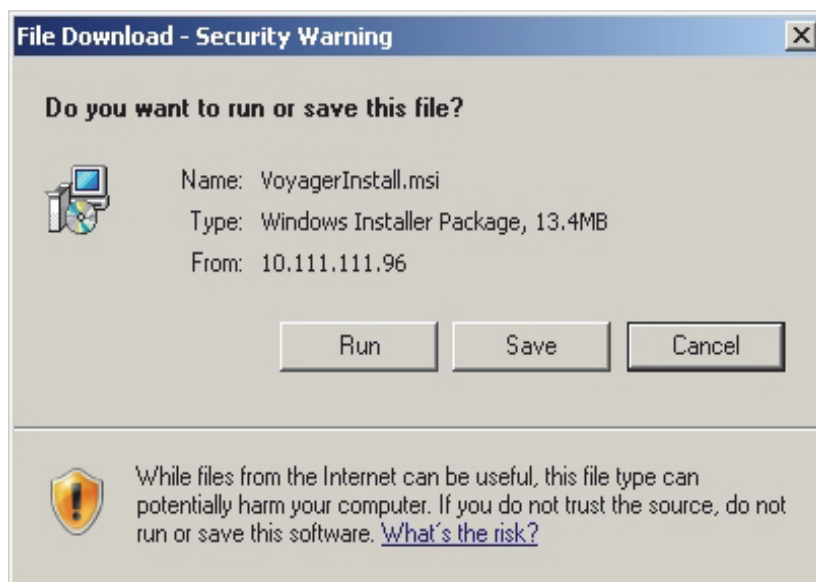


Figure 4-34. File download security warning dialog box

When you click Run, the client software is downloaded for installation and the **Welcome** dialog box is displayed. See [Figure 4-31](#).

The remainder of the installation follows the steps identified in [Procedure 4-1, Installing Voyager Clients](#), on page 4-4. The Media Scheduling installation follows a similar procedure.

FTP Method

The system displays the file download security warning dialog box. See [Figure 4-35](#).

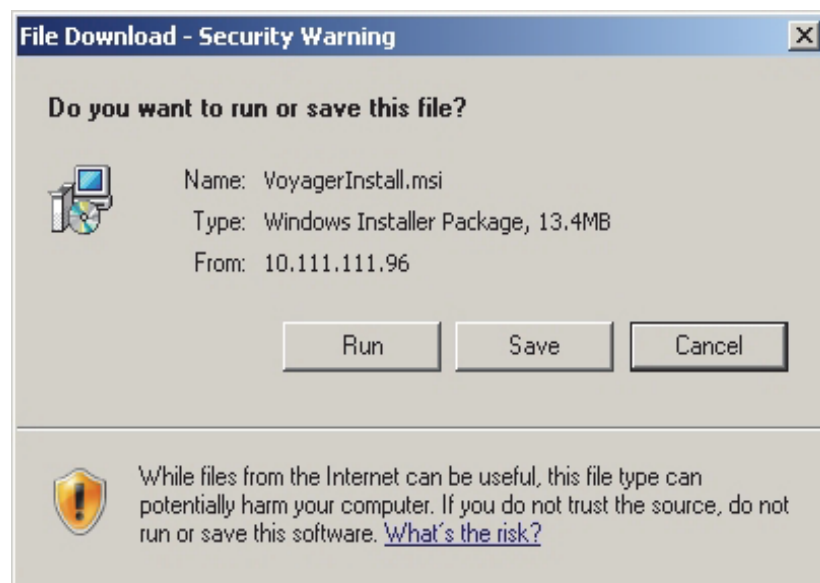


Figure 4-35. File download security warning dialog box

When you click Run, the client software is downloaded for installation and the **Welcome** dialog box is displayed. See [Figure 4-31](#).

The remainder of the installation follows the steps identified in [Procedure 4-1, Installing Voyager Clients](#), on page 4-4. The Media Scheduling installation follows a similar procedure.

.BAT File Method

The steps relative to the .bat method are site specific. The .bat file provides for any number of installation options to be coded.

The following example illustrates the use of the `.bat` method:

```
VoyagerInstall.msi /qb+! /L*v "c:\testvoy\voy.log"
INSTALLDIR="c:\testvoy" INSTALLLEVEL="4"
```

Figure 4-36. Example code for .bat method

The example in [Figure 4-36](#) does the following:

- This causes a progress dialog box (with no cancel button) and a completed dialog box to display.
- All messages are written to `C:\testvoy\voy.log`
- The software is installed in `C:\testvoy`
- A Complete (versus Custom or Typical) installation is performed.

See [Table 4-6](#) for a description of possible options using this approach.

Table 4-6. Options for .bat file method

Option	Parameters	Description
/x		Uninstalls a product. Example: <code>msiexec /x VoyagerInstall.msi</code>

Table 4-6. Options for .bat file method

Option	Parameters	Description
/L	[i w e a r u c m o p v + !] Logfile	Specifies path to the log file, and the flags indicate which information to log. - i - Status messages. - w - Non-fatal warnings. - e - All error messages. - a - Start up of actions. - r - Action-specific records. - u - User requests. - c - Initial UI parameters. - m - Out-of-memory or fatal exit information. - o - Out-of-disk-space messages. - p - Terminal properties. - v - Verbose output. + - Append to existing file. ! - Flush each line to the log. * - Wildcard, log all information except for the v option. To include the v option, specify /L*v. The following example line of code logs all information except for the v option into c:\VoyagerLog.txt. Example: <pre>VoyagerInstall.msi /L* c:\VoyagerLog.txt</pre>

Table 4-6. Options for .bat file method

Option	Parameters	Description
/q	n b r f	Sets user interface level. q , qn - No UI (User Interface). qb - Basic UI . Use qb! to hide the Cancel button. qr - Reduced UI with no modal dialog box displayed at the end of the installation. qf - Full UI and any authored Fatal Error, User Exit, or Exit modal dialog boxes at the end. qn+ - No UI except for a modal dialog box displayed at the end. qb+ - Basic UI with a modal dialog box displayed at the end. The modal box is not displayed if the user cancels the installation. Use qb+! or qb!+ to hide the Cancel button. qb- - Basic UI with no modal dialog boxes. Please note that /qb+- is not a supported UI level. Use qb-! or qb!- to hide the Cancel button. The following example line of code installs the product with no user interface. Example: <code>VoyagerInstall.msi /qn</code> NOTE: The ! option is available with Windows Installer version 2.0 and works only with basic UI. It is not valid with full UI.

Table 4-6. Options for .bat file method

Option	Parameters	Description
INSTALLDIR=	[directory]	Set installation directory. The default installation directory is C:\Voyager. The following example line of code only installs the programs into the D:\Voyager\ directory. Example: <pre>VoyagerInstall.msi INSTALL- DIR="D:\Voyager"</pre> NOTE: This option only works with VoyagerInstall and VoyagerMedia.
EIS_APP=	[A B C D E F G H I J K]	Specifies applications and/or files to be installed from VoyagerInstall.msi. The default applications to be installed for a typical installation are ACDIJ. A - Acquisitions B - Callslip C - Cataloging D - Circulation E - Circ Self Check F - Reporter G - System Admin H - Access Reports I - System Files J - Config Files K - Global Data Change The following example line of code only installs Cataloging, Reporter, and System Administration. Example: <pre>VoyagerInstall.msi EIS_APP="CFG"</pre> NOTE: This option only works with VoyagerInstall.

Table 4-6. Options for .bat file method

Option	Parameters	Description
INSTALLLEVEL=	[#]	Set installation priority level. The default installation priority level is 3. 3 - Typical Install 4 - Complete Install The following example line of code installs all applications. Example: <pre>VoyagerInstall.msi INSTALLLEVEL="4"</pre>

[Upgrade] Stanza - Voyager 8 and Later

The [Upgrade] stanza (see [Figure 4-15](#) on [page 4-18](#) for an illustration of the complete `Voyager.ini` file) provides the facility for AutoUpdate of the core Voyager client software to a user's PC when a new version is detected on the server.

To enable this capability, specify the following for the `Voyager=` and/or `Media=` parameters in the [Upgrade] stanza using the format identified in [Figure 4-37](#).

- Server user name and password
The default user name/password is `clients/Voyag3R`. See [Apache Configuration](#) on [page 4-52](#) for more information regarding access and the default user name and password.
Contact your system administrator for the current user name and password.
- IP and port
- Automatic update path for Voyager core clients and the Media Scheduling client (if applicable).

**IMPORTANT:**

The path specified must end with a slash (see [Table 4-37](#)).

```
[Upgrade]
Voyager=http://<user name>:<password>@<IP>:<port>/autoupdate/voyager/
Media=http://<user name>:<password><IP>:<port>/autoupdate/media/
```

Figure 4-37. [Upgrade] Stanza - Enable Client AutoUpdate Settings

The default values specified for `Voyager=` and `Media=` are:

```
Voyager=http://clients:Voyag3R@<IP|DNS>:7099/autoupdate/
voyager/
Media=http://clients:Voyag3R@<IP|DNS>:7099/autoupdate/media/
```

The `Voyager=` parameter is used to initiate the download of new Voyager client software, and the `Media=` parameter is used to initiate the download of new Media Scheduling software (if this applies to your site).

NOTE:

If you leave the `Voyager=` and the `Media=` parameters blank, you need to manually download/install new versions of the client software. See [Downloading Voyager Client Files](#) on [page 4-3](#) and [Installing Voyager Clients Using the Manual Method](#) on [page 4-4](#).

AutoUpdate Characteristics

When new software files are detected, the user is prompted to download the new software (see [Figure 4-38](#)).

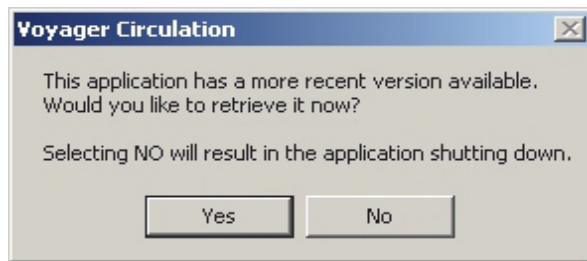


Figure 4-38. Version available message

The new version of the Voyager client files that are available for download to users' PCs are placed on your Voyager server at upgrade in the autoupdate folder (see [Figure 4-39](#)).

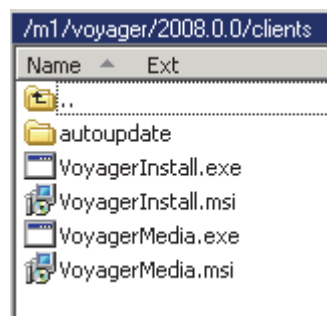


Figure 4-39. autoupdate Folder

If a user declines to download the new software when prompted, the client shuts down. The new files need to be downloaded in order to proceed.

When the user selects **Yes**, all the client files are automatically downloaded when any client is started and determines that there is a new version available.

NOTE:

With the AutoUpdate method, no system administrator privileges are required.

As part of the AutoUpdate process, the software determines which clients are currently available on the PC and replaces those. If a PC for the Circulation Desk only has the Circulation client installed, the software only downloads the Circulation client and no other clients. When the AutoUpdate is complete, it displays a confirmation message (see [Figure 4-40](#)).



Figure 4-40. Client Upgrade Complete

With the AutoUpdate method, there are no options presented during the download such as an option to select a different folder for storing the client files. AutoUpdate assumes that the new version of the clients should be placed in the existing Voyager client folder on the user's PC.

NOTE:

In the case of a failed AutoUpdate, uninstall the clients (see [Uninstalling the Voyager Clients](#) on [page 4-15](#)) and use the manual installation method (see [Installing Voyager Clients Using the Manual Method](#) on [page 4-4](#)) to re-install the clients.

AcquisitionsStart.exe and CatalogStart.exe

For the Acquisitions and Cataloging clients, the startup icons need to point to the following .exe files:

- AcquisitionsStart.exe
- CatalogStart.exe

The first time that new Acquisitions and Cataloging clients are run, system administrator privileges are required. The `AcquisitionsStart.exe` and the `CatalogStart.exe` files are programmed to address this requirement by allowing users with lesser privileges than the Administrator to initially launch the applications. The `*Start.exe` files stand in for a user with Administrator privileges and, subsequently, call the `Acquisitions.exe` or `Catalog.exe`

executables. As a result, for consistently successful execution of these clients, it is recommend that you use the `AcquisitionsStart.exe` and the `CatalogStart.exe` files.

[HelpMenuLink] Stanza

Links to Ex Libris Web content is available from the Help menu with the `[HelpMenuLink]` stanza provided in the `Voyager.ini` file. See [Figure 4-41](#).



Figure 4-41. Help menu example

The Doc Portal option may be edited to change the:

- Menu label
- URL link

Example:

```
[HelpMenuLink]
Ex Libris Doc Portal=http://www.customercenter.exlibrisgroup.com/
```

The menu label is defined to the left of the equal sign, and the URL is defined to the right of the equal sign.

Additional Files Installed with the Client Installation

In addition to the client executable and help files and the `Voyager.ini` file, there are additional folders and files that are installed on the user's computer when installing Voyager. Typically, these files are installed in the `c:\Voyager` directory.

The following folders and files are installed in a complete installation:

- Access Reports folder
 - `Reports.mdb` file

This is a Microsoft Access database containing prepackaged reports and queries. For more information about this file, see *Prepackaged Access Reports Overview* in the *Voyager Reporter User's Guide*.

- Catalog folder
 - MARC Template folder
 - * EditBook.cfg
 - * EditSerial.cfg
 - * NewBook.cfg
 - * NewSerial.cfg
 - * Template.ini

These files control templated input using Citation Server[®]. See the *Voyager Citation Server User's Guide* for more information.

- Tag Table folder
 - * MARC21 folder
 - * OCLC folder
 - * Rlin folder
 - * Bmarcfix.cfg
 - * Lang.cfg
 - * Country.cfg

The tag tables define various types of records, fields, and subfields for bibliographic and authority records. See Appendix A in the *Voyager Cataloging User's Guide* for more information.

- Template folder

The template folder contains default templates for bibliographic, holdings, and authority records. For more information about these templates see the *Voyager Cataloging User's Guide*.

- * MARC21 Repertoire.cfg
- * Special Characters.cfg

For information about these files see the *Voyager Cataloging User's Guide*.

- Circulation folder
 - Offline folder
 - Circ.ini
 - Circbib.cfg
 - CircSC.ini

The Offline folder contains any charge files created when using offline charge. See *Offline Charge* in the *Voyager Circulation User's Guide*.

The `Circ.ini` file contains the templates for all the slips and statements that can be printed in the Circulation module. See *CIRC.INI* in the *Voyager Circulation User's Guide*.

The `Circbib.cfg` is used to configure the **Bib** tab of the **Add Bib/Item** dialog box when creating a bibliographic record in the Circulation module. See *Circbib.cfg file - Customizing the Bib Tab Fields* in the *Voyager Circulation User's Guide*.

The `CircSC.ini` file contains information to configure the Voyager Self-check module. See *CIRCSC.INI* in the *Voyager Circulation User's Guide*.

- Misc folder

- `Acqbib.cfg`

- This file provides the template for brief bibliographic records created in the Acquisitions module. See the *Voyager Acquisitions User's Guide* for more information.

- `Callslip.ini`

- For information about this file, see the *Voyager Call Slip Dæmon User's Guide*.

- `EDI_msg.ini`

- `Limits.ini`

- This configures staff search limits. For more information, see the `limits.ini` section in any of the following user's guides: Acquisitions, Cataloging, or Circulation.

- `Mediahelp.ini`

- `Spinelabel.cfg`

- This file enables the user to configure separate spine and piece label templates for bibliographic, holding, item and serial records. See *Appendix B. Printing Labels* of the *Voyager Circulation User's Guide* for more information.

- Reporter

- Notices

- Reports

- This folder contains Microsoft Access files (`.mdb`).

- System

- This folder contains `.dll` files, fonts, and so on.

AutoUpdate on the Server

AutoUpdate is a feature that allows the clients to be updated on the client workstations without requiring a reinstallation of the clients. Once the initial installation of the clients has been performed as an administrative user, this feature does not require administrative privileges.

When the version of the server software is newer than the version of your clients, you receive a message about updating your clients (see [AutoUpdate Characteristics](#) on [page 4-45](#)). If you choose to upgrade your clients to the latest version, the AutoUpdate process begins and informs you when the process has been completed. This feature can only be executed one time per version. Once your clients are updated to the current version, you are not be able to run the update feature again until the next time your Voyager server software is updated to a newer release.

In order to ensure that AutoUpdate is configured successfully, there is setup required both on the server and in the client software. The server configuration is done at the time of the upgrade and is installed and configured by the Voyager Installation Kit (VIK). The client setup is done after the initial installation of the Voyager 8.0.0 and later clients (see [Transition Considerations for AutoUpdate](#) on [page 4-2](#) and [\[Upgrade\] Stanza - Voyager 8 and Later](#) on [page 4-44](#)). You are then ready for the AutoUpdate feature for the next release or Voyager service pack.

The client feature communicates with an HTTP server on the Voyager application server. You need to ensure that the port on the Voyager application server is available to all staff members that plan to use this feature. The default port is 7099.

Server Files

The server setup consists of a directory that contains the client files that are downloaded, a configuration file that interfaces to the client AutoUpdate feature and identifies what files to download, and the apache configuration files.

The Voyager Installation Kit handles setting up the server configuration.

Location of the Files to be Downloaded

All the files for the Voyager core client AutoUpdate feature are located in the `/m1/voyager/clients/autodupate/voyager` directory.

If you have the Media Scheduling module, those files are located in the `/m1/voyager/clients/autoupdate/media` directory.

The latest version of Voyager setup is in the `/m1/voyager/clients/autoupdate` directory. Previous versions are archived for reference purposes.

AutoUpdate Configuration File

In the `/m1/voyager/clients/autoupdate/voyager` directory, there is a file called `update.txt`. The `update.txt` file contains a list of all of the files that are downloaded when the AutoUpdate process takes place. If a file resides in the Voyager directory but is not in the `update.txt` file, the file is not downloaded to each PC that uses the update feature. If you add a file to the `update.txt` file, you need to make confirm that the file resides in the Voyager directory and in the correct path as you expect to find it on the PC.

The format of the `update.txt` file is:

```
files=\Access Reports\Reports.mdb,  
\Acq.chm,  
...  
\System\vsStr7.ocx,  
\System\vssver.scc,  
\System\VsVIEW6.ocx,  
\vssver.scc
```

Figure 4-42. Example update.txt File

The path in the file refers to the root of the Voyager client installation. If `C:\Voyager` is the directory in which the clients are installed, the file `\Access Reports\Reports.mdb` found in the `update.txt` file is downloaded and installed to the `C:\Voyager\Access Reports\Reports.mdb`.

Apache Configuration

The Voyager Installation Kit installs and configures an HTTP server virtual host on your server. This HTTP server virtual host allows the clients to connect and complete the AutoUpdate process.

The apache configuration can be found in the `/m1/shared/apache2/conf/ConfiguredVirtualHost/clients.conf` file.

```
vi /m1/shared/apache2/conf/ConfiguredVirtualHost/clients.conf

Listen 7099
<VirtualHost *:7099>
    ServerAdmin voyager
    #ServerName 127.0.0.1
    DocumentRoot "/m1/voyager/clients"
    ErrorLog logs/clients_error_log
    CustomLog logs/clients_access_log common

    <Directory "/m1/voyager/clients">
        Options All Indexes FollowSymLinks MultiViews
        # Comment out this
        #Options Indexes FollowSymLinks
        AllowOverride None
        Order allow,deny
        Allow from all
    </Directory>

    <Directory /m1/voyager/clients>
        AuthName "Client Access"
        AuthType Basic
        AuthUserFile /m1/shared/apache2/conf/AuthorizedUsers/
        clients_access
        require valid-user
    </Directory>
</VirtualHost>
```

Figure 4-43. Example clients.conf File

When the virtual host is ready for connections, there is a symbolic link in the `/m1/shared/apache2/conf/ActivatedVirtualHosts` directory.

```
cd /m1/shared/apache2/conf/ActivatedVirtualHosts
ls -l clients.conf
lrwxrwxrwx 1 root root 38 Jun  1 10:05 clients.conf -> ../
ConfiguredVirtualHosts/clients.conf
```

Figure 4-44. Symbolic Link

In order to manage the access for AutoUpdate, there is a `clients_access` file in the `/m1/shared/apache2/conf/AuthorizedUsers` directory. The default user/password to login to the client update feature after installation is `clients/Voyag3R`. This user name and password is added to the `Voyager.ini` on the Voyager client installation on the PC.

```
cd /m1/shared/apache2/conf/AuthorizedUsers
cat clients_access

clients:u4.lpmlq6TPB6
```

Figure 4-45. Authorized Users



IMPORTANT:

It is recommended that you change the clients user password after installation.

User Access for AutoUpdate

See the following to create, change, or delete user access:

- [Adding a Login/Password to the Existing File](#) on [page 4-54](#)
- [Modifying a Login/Password for an Existing User](#) on [page 4-55](#)
- [Deleting a Login/Password for an Existing User](#) on [page 4-55](#)

Adding a Login/Password to the Existing File

The procedure for adding a login/password to the existing file is shown in [Procedure 4-5, Adding a Login/Password to the Existing File](#), on [page 4-55](#).



Procedure 4-5. Adding a Login/Password to the Existing File

1. Login as `root` and enter:

```
cd /m1/shared/apache2/bin
```

2. At the prompt, enter the following where `newname` is the new user:

```
./htpasswd /m1/shared/apache2/conf/ AuthorizedUsers/  
clients_access newname
```

3. When the system prompts you, enter the password for `newname` two times.

Result: An additional login and password is created.

Modifying a Login/Password for an Existing User

The procedure for modifying a login/password to the existing file is shown in [Procedure 4-6, Modifying a Login/Password for an Existing User](#), on page [4-55](#).



Procedure 4-6. Modifying a Login/Password for an Existing User

1. Login as `root`, move to the `/bin` directory, and enter:

```
cd /m1/shared/apache2/bin
```

2. At the `bin` prompt, enter the following where `clients` is the user name:

```
./htpasswd /m1/shared/apache2/conf/AuthorizedUsers/  
clients_access clients
```

3. When the system prompts, enter that user's password two times.

Result: A modified login and password is created.

Deleting a Login/Password for an Existing User

The procedure for deleting a login/password to the existing file is shown in [Procedure 4-7, Deleting a Login/Password for an Existing User](#), on page [4-56](#).



Procedure 4-7. Deleting a Login/Password for an Existing User

1. Login as `root` and enter:

```
cd /ml/shared/apache2/conf/AuthorizedUsers
```

2. Open the `clients_access` file using `vi` or another server text editor.
3. Delete the row for the specific user you want to delete.
4. Save the `clients_access` file.

Result: The login and password are deleted.

Test AutoUpdate

Use Telnet on the Windows PC to test AutoUpdate.

The procedure for testing AutoUpdate is shown in [Procedure 4-8, Testing AutoUpdate](#), on page [4-56](#).



Procedure 4-8. Testing AutoUpdate

1. Click **Start** on the computer where you are installing the clients.
2. Select **Run** from the **Start** menu.

Result: The **Run** dialog box displays, and the cursor defaults to the **Open** field.

3. Enter `cmd`.
4. Enter the following command:

```
telnet <ip | dns> 7099
```

Result: You should receive an empty screen with a blinking cursor. If you do not receive this screen, the following may be causing the problem(s):

- Apache is not running on the Voyager server to which you are trying to connect
 - Network traffic is not permitted to the server and port 7099
-

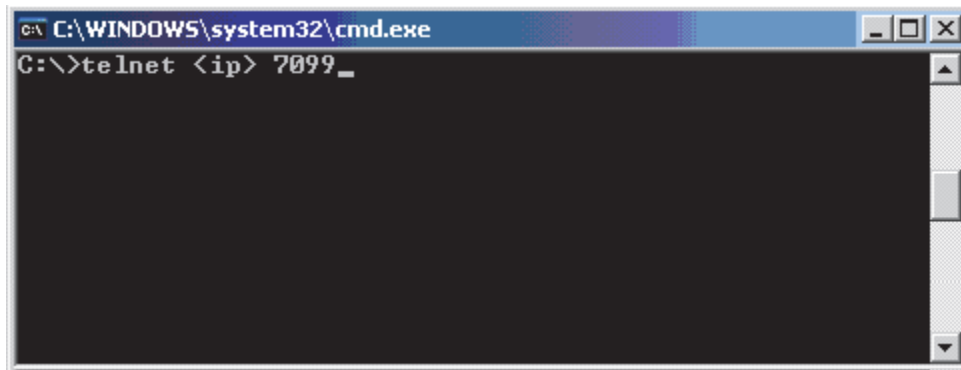


Figure 4-46. Telnet

5. To exit, click the Windows **Close** button (X).

6. Open a browser.

7. Enter the following URL:

`http://<ip | dns>:7099`

8. Enter the following user name and password:

User: clients

Password: Voyag3R

Result: You should see a directory listing of the files on the server in the `/ml/voyager/clients` directory.

Index of /

Icon	Name	Last modified	Size	Description
[DIR]	2008.1.0/	08-Jun-2011 10:24	-	
[]	VoyagerInstall.exe	13-May-2011 09:09	20M	
[]	VoyagerInstall.msi	13-May-2011 09:09	18M	
[DIR]	autoupdate/	01-Jun-2011 09:37	-	
[DIR]	conf/	01-Jun-2011 09:28	-	

Figure 4-47. Directory Listing

NOTE:

Common issues that arise are a result of port 7099 not being accessible on the network.

Client Setup

In order to use this feature, you must first install the clients for Voyager 8.0 or later (see [Client Installation Options](#) on [page 4-2](#)). If you install the clients that match the version of Voyager server software you are running, this feature is not needed.

To ensure a successful client AutoUpdate, you need to make sure that port 7099 on your Voyager application server is open on your network and ensure that firewalls are configured appropriately.

Voyager.ini Setup

The `Voyager.ini` file is an initialization file that contains important connection information that enables your Voyager clients to use the AutoUpdate server. The `[Upgrade]` stanza has been repurposed and now must be configured to only point at the AutoUpdate server (see [\[Upgrade\] Stanza - Voyager 8 and Later](#) on [page 4-44](#)).

Customize AutoUpdate Directories

When it has been confirmed that AutoUpdate is configured and working, there may be reasons to have customized AutoUpdate directories. Different clients may be configured for different staff in your institution. For example, catalogers may need to have different tag tables depending on their responsibilities.

AutoUpdate has the ability to be customized to update clients differently according to your needs. Customization requires that you:

- Create a new directory
- Upload the file(s) to the correct place on the server
- Add or delete files from the `update.txt` file if necessary

Scenario 1 - Deploy Custom Bmarc9xx.cfg for Marc21 Tag Table

For this scenario, you have a Cataloging client installation that has a custom `Bmarc9xx.cfg` file that is different from your primary Cataloging installations.

It is assumed that you already have a custom-created `Bmarc9xx.cfg` file and that you know how to place files on your server with FTP, SFTP, or SCP.

It is also assumed that the clients on the PC are installed in the `C:\Voyager` directory.



Procedure 4-9. Customizing Scenario 1

1. Set up a new AutoUpdate directory.
 - a. Login to the server as the `voyager` user.
 - b. `cd /m1/voyager/clients`
 - c. `cp -pr autoupdate autoupdate-custom`
 - d. FTP, SCP, or SFTP the new `Bmarc9xx.cfg` to `/m1/voyager/clients/autoupdate/voyager/Catalog/TagTable/Marc21`.

2. Update the C:\Voyager\voyager.ini file

- a. Open a window to C:\Voyager\voyager.ini.
- b. Update the [Upgrade] stanza to point to the new AutoUpdate repository.

```
# This stanza replaces the per-stanza NewVersion= in Voyager 7.0
[Upgrade]
Voyager=http://clients:Voyag3R@<IP|DNS>:7099/autoupdate-custom/voyager/
Media=http://clients:Voyag3R@<IP|DNS>:7099/autoupdate/media/
```

Figure 4-48. Scenario 1 [Upgrade] Stanza

Scenario 2 - Deploy Prelinked Institution-Specific reports.mdb

For this scenario, your institution has created a prelinked `reports.mdb` with predefined reports that need to be deployed. You want to avoid overwriting the existing `reports.mdb` in the installation since it might have reports the staff use regularly and want to preserve. The name of the `.mdb` file is `Reports-institution.mdb` and resides in the same directory as the existing `Reports.mdb`.

It is assumed that you already have a custom-created `Reports-institution.mdb` and that you know how to place files on your server with FTP, SFTP, or SCP.

It is also assumed that the clients on the PC are installed in the C:\Voyager directory.



Procedure 4-10. Customizing Scenario 2

1. Set up a new AutoUpdate directory.
 - a. Login to the server as the `voyager` user.
 - b. `cd /ml/voyager/clients`
 - c. `cp -pr autoupdate autoupdate-custom-reports`

- d. FTP, SCP, or SFTP the new Reports-institution.mdb to the /m1/voyager/clients/autoupdate/voyager/Access Reports directory.
- e. Add the following line to /m1/voyager/clients/autoupdate/voyager/update.txt.

```
files=\Access Reports\Reports.mdb,
\Access Reports\Reports-institution.mdb,
...
```

Figure 4-49. Scenario 1 update.txt Change

2. Update the C:\Voyager\voyager.ini file.
 - a. Open a window to the C:\Voyager\voyager.ini file.
 - b. Update the [Upgrade] stanza to look at the new AutoUpdate repository.

```
# This stanza replaces the per-stanza NewVersion= in Voyager 7.0
[Upgrade]
Voyager=http://clients:Voyag3R@<IP|DNS>:7099/autoupdate-custom-reports/voyager/
Media=http://clients:Voyag3R@<IP|DNS>:7099/autoupdate/media/
```

Figure 4-50. Scenario 2 [Upgrade] Stanza

File Transfer Protocol (FTP) - Voyager Client Installations

Use FTP (File Transfer Protocol) to retrieve the Voyager client installation file to your local PC. Check with your system administrator or IT staff to confirm that FTP is an allowable protocol for file downloads.



Procedure 4-11. Downloading a Voyager Client Installation File Using FTP

1. Use the following URL in your browser to begin the FTP process.

```
ftp://downloads.exlibrisgroup.com
```

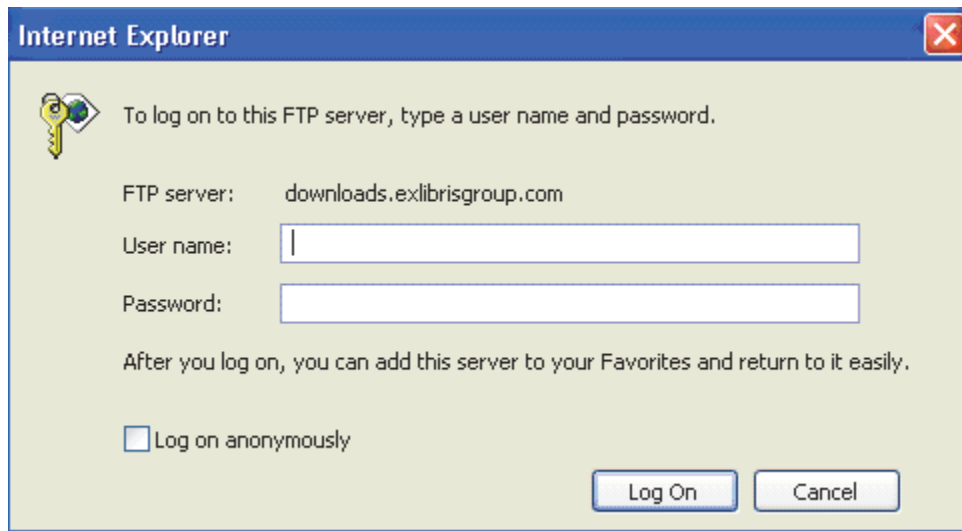


Figure 4-51. FTP Login

NOTE:

The figures in this procedure use Internet Explorer. However, you may use Mozilla Firefox.

2. Enter the user name `voyager`, its password, and click OK.

NOTE:

Refer to the **FTP Passwords for voyager, odbc, preview, and patch users** article in the Knowledge Center for the password for the `voyager` user.

3. Click the `clients` directory from the list of directories.
4. Click the link to the directory of the Voyager version that you are running.

Result: The `VoyagerInstall.exe` and `VoyagerInstall.msi` files display. See [Downloading Voyager Client Files](#) on [page 4-3](#) for more information regarding these installation files.

5. Click the installation file that matches your preferred installation method (`.exe` or `.msi`).

Result: The FTP download prompt displays.

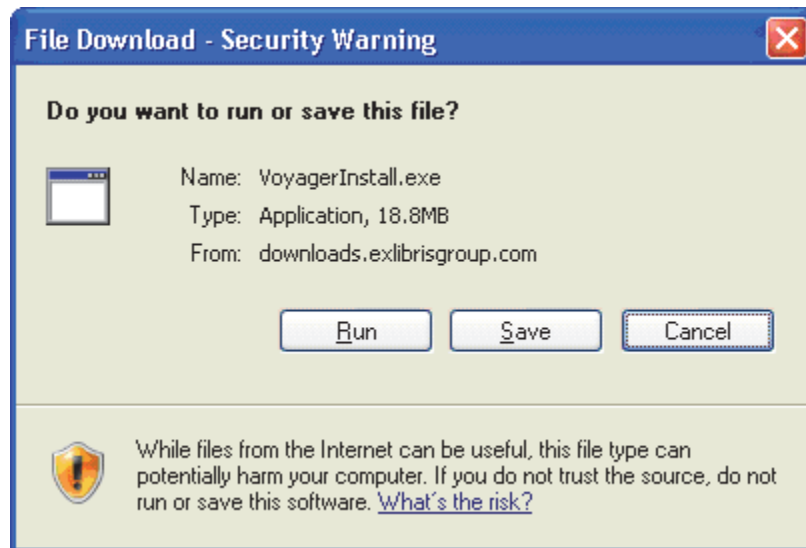


Figure 4-52. FTP Download Prompt

6. Click **Run** or **Save**.

NOTE:

Saving the installation file to your PC or network drive provides you the flexibility of running the installation at a different date/time.

Result: You have completed the FTP of the Voyager client installation file, and you are ready to complete your client software installation following the instructions provided in [Installing Voyager Clients Using the Manual Method](#) on [page 4-4](#).

Overview

Starting with Voyager version 8.2, a combination of patron batch jobs (using XML file input/output) and web services have been designed for making patron changes to the Voyager database. With this development, the following is being implemented:

- A patron XML input/output file standard and patron XSD schema
- Additional UNIX shell scripts for batch processing
- Additional web-based services

These additional facilities are available for importing, exporting, and purging patron data. They use the same XML file structure for input and output.

NOTE:

The existing Patron SIF (Standard Interface File) method continues to be available for making patron record changes. See [Patron Extract \(Patron SIF Method\)](#) on [page 6-1](#) and [Patron Update \(Patron SIF Method\)](#) on [page 7-1](#).

The patron XSD, that describes the structure of the XML, is available in the Developer Network at <https://developers.exlibrisgroup.com/voyager/apis/xmloverhttpwebservices/patronimportservice/>. This schema is used for the patron import, export, and purge XML documents.

For XML examples for patron import, refer to the following path in EL Commons:

EL Commons > CodeShare > Voyager Open Interfaces > XML Over
HTTP Web Services > Patron Import Service

For XML examples for patron export, refer to the following path in EL Commons:

EL Commons > CodeShare > Voyager Open Interfaces > XML Over
HTTP Web Services > Patron Export Service

Batch Processing - UNIX Shell Scripts

For batch processing with the XML interface, use the following shell scripts:

- PpatronImpX (see [PpatronImpX \(Patron Import\)](#) on [page 5-2](#))
- PpatronExpX (see [PpatronExpX \(Patron Export\)](#) on [page 5-8](#))
- PpatronPrg (see [PpatronPrg \(Patron Purge\)](#) on [page 5-11](#))

PpatronImpX (Patron Import)

The patron import shell script is run from the `/m1/voyager/xxxdb/sbin` directory.

As input, the script requires an XML file that you provide that contains the patron information to be imported. The XML input file must be properly formed according to the patron XSD. Refer to EL Commons for examples and the patron XSD schema.

See [Table 5-1](#) for the list of parameters provided with PpatronImpX. All parameters are optional except where identified as mandatory.

Table 5-1. PpatronImpX Parameters

Parameter	Description	Default
-p	Use this parameter to specify the name of the input file. This parameter is mandatory.	The program looks for the input file in the <code>/m1/voyager/xxxdb/local</code> directory if a complete path is not specified.
-d	Use this parameter to specify the Voyager database name.	When no database is specified, the program uses the database name identified in the <code>/m1/voyager/xxxdb/ini/voyager.env</code> file.

Table 5-1. PpatronImpX Parameters

Parameter	Description	Default
-u	Use this parameter to specify the user name and password for the database.	When no user name and password is specified, the program uses the user name and password identified in the /m1/voyager/xxxdb/ini/voyager.env file.
-g	Use this parameter to specify the database host.	When no database host is specified, the program uses the database host identified in the /m1/voyager/xxxdb/ini/voyager.env file.
-y	Use this parameter to specify the database port.	When no database port is specified, the program uses the database port identified in the /m1/voyager/xxxdb/ini/voyager.env file.
-z	Use this parameter to specify the database SID.	When no database SID is specified, the program uses the database SID identified in the /m1/voyager/xxxdb/ini/voyager.env file.
-a	Use this parameter to specify the audit file name.	If no audit file name is specified, the program uses /m1/voyager/xxxdb/rpt/audit.pupdx.YYYYMMDD.HHMM.
-e	Use this parameter to specify the error file name.	If no error file name is specified, the program uses /m1/voyager/xxxdb/rpt/err.pupdx.YYYYMMDD.HHMM.
-m	Use this parameter to specify the number of records for monitoring progress in the audit file (optional).	When nothing is specified, a note is written to the audit file every 100 records/patrons by default.
-t	Use this parameter to indicate that the batch job should run in test mode. In this mode, the program verifies the XML input and completes all validation checks but does not add or update any records in the database.	

Table 5-1. PpatronImpX Parameters

Parameter	Description	Default
-x	Use this parameter to specify the number of errors after which the program stops. NOTE: This refers to patron content errors not XML errors. The XML is validated as being well formed during preprocessing.	If no number is specified, the program stops after 100 errors.
-o	Use this parameter to specify the operator. The text entered for this parameter is put in the OPID for all patron records that are added or modified for this import/update.	If not present, XMLLOAD is used as the OPID.
-c	Use this parameter to specify load transaction counters (if present in the input data). If this parameter is not used, transaction counters are not loaded.	
-h	Use this parameter to request the help information.	

Processing

When `PpatronImpX` is executed, the input XML document is validated to insure that it contains well-formed XML; and the individual patron records are parsed from the source XML document.

NOTE:

If the validation process (of the input XML document) determines that the XML is not well formed, no individual patron records are processed.

During import, each patron record is checked for valid content. For example, location is not required in the input file; but when it is, there should be a match to an existing location code in the database.

Depending on the action specified, the patron record may need to be matched by one of the following to an existing record in the database:

- Institution ID
- Voyager ID (patron ID)
- Barcode
- Social Security number (SSN)

<action> Element

For each patron record in the XML file (specified as the file to import), there needs to be an <action> element (refer to EL Commons for XML file examples). This element is used to specify Insert, Update, or Add. See [Table 5-2](#) for a description of each option.

Table 5-2. <action> Element

Option	Description
Insert	With this option a new patron record (and related records) is created in the database. With this option, records are added to the database; and no existing records are modified. If any existing records are found, an error is reported.
Add	With this option, the action taken may be either insert or update depending on whether a matching record is found and other factors described in <overlayAction> Element on page 5-5.
Update	With this option, the patron record may be modified; and related records may be added, updated, or deleted depending on different factors described in <overlayAction> Element on page 5-5. NOTE: When updating, if a record is not found to update, an error is reported.

<overlayAction> Element

With the Add and Update options in the <action> element, the action performed with these options depends on:

- whether the <overlayAction> element exists and its specified value is either Add or Overlay for the following individual elements:
 - <tempAddressList>
 - <emailList>

- <noteList>
- <patronStatList>
- whether there is nil or null data for nonrequired elements or data types not having an <overlayAction> element specified

Setting the nil attribute to true means that a null value can be assigned to the element. Leaving the value of an element empty or not including it is distinctly different. Example:

If you want to remove Dr. from a patron's title, you need to specify the following in the imported record so that a null value can be specified in the patron's title:

```
<patron>
<action>Update</action>
...
<title xsi:nil="true"/>
...
</patron>
```

Figure 5-1. Example Part 1

Not including anything for title (example below) indicates don't change what is in the title. As a result, no null value is set for title; and the title is not changed.

```
<patron>
<action>Update</action>
...
</patron>
```

Figure 5-2. Example Part 2

Audit File

The default audit file name is `audit.pupdx.YYYYMMDD.HHMM`. The default location is the `/ml/voyager/xxxdb/rpt` directory.

The audit file contains:

- The command line parameters used to run the job
- Location and names of the error and audit files

- A line for every record not added or the problem found
- The following for each record in the file if the load is in test mode:
 - Input match ID: (from input file)
 - Input match ID type: (barcode, institution ID or Voyager ID)
 - Update type: (add, insert, update)
 - Match: (match found, match not found, or multiple matches found)
 - Matched ID: (Voyager record ID; for multiple, all are listed)
 - Addresses added: (number)
 - Notes added: (number)
 - Barcodes added: (number)
 - Statistics added: (number)
 - Errors encountered during potential update or insert
 - List of all potential errors

If none are found, None is specified in the output.
- Progress indicators in groups of 100 (unless a different value was specified with the `-m` flag)
- A summary at the end of the file detailing:
 - Start time, end time, and total run time
 - Input records received
 - Patron records added
 - Patron records updated
 - Patron records written to the exception file

Error File

The error file is an XML document with the default file name of `err.pupdx.YYYYMMDD.HHMM`. The default location for the error file is the `/m1/voyager/xxxdb/rpt` directory.

Whenever a patron record from the input file cannot be loaded into the database, the record is written to the error file. An `<info>` element is added to the record, that contains a `<reply>` element with the error code and message and, if possible, a `<match>` and `<updated>` element containing the relevant data.

If the load is run in test mode, the patron records that would have failed to load are written to the error file.

NOTE:

If the input file is not well formed according to the schema, no data is written to the error file.

Refer to EL Commons as described in the [Overview](#) on [page 5-1](#) for a list/description of error codes.

Web Service for Patron Import

A single patron record can be updated using the POST operation and an XML patron record document. Use the following URL format/syntax:

```
http://server:port/vxws/patronImport?test=no&opid=XMLLOAD&help=no
```

Success or failure is returned in the <reply> stanza of the patron XML. See [Figure 5-3](#) and [Figure 5-4](#) for examples.

```
<reply>
  <reply-text>OK</reply-text>
  <reply-code>00</reply-code>
</reply>
```

Figure 5-3. <reply> Example with No Errors

```
<reply>
  <reply-text>Action = Insert, but match found in database.
    Record not processed.</reply-text>
  <reply-code>12</reply-code>
</reply>
```

Figure 5-4. <reply> Example with Errors

Refer to EL Commons as described in the [Overview](#) on [page 5-1](#) for more information.

PpatronExpX (Patron Export)

The patron export shell script is run from the /m1/voyager/xxxdb/sbin directory.

Input to the export can be one of the following:

- A file of patron record IDs (one ID per line/row)
- A range of records (nn-nn)
- Both (range from within a file)
- No input (indicates export all the records)

See [Table 5-3](#) for the list of parameters provided with `PpatronExpX`. All parameters are optional.

Table 5-3. PpatronExpX Parameters

Parameter	Description	Default
-p	Use this parameter to specify the name of the output file.	When nothing is specified for the output file name, the default is /m1/voyager/xxxdb/rpt/xml.pxtrx.YYYYMMDD.HHMM.
-d	Use this parameter to specify the Voyager database name.	When no database is specified, the program uses the database name identified in the /m1/voyager/xxxdb/ini/voyager.env file.
-u	Use this parameter to specify the user name and password for the database.	When no user name and password is specified, the program uses the user name and password identified in the /m1/voyager/xxxdb/ini/voyager.env file.
-g	Use this parameter to specify the database host.	When no database host is specified, the program uses the database host identified in the /m1/voyager/xxxdb/ini/voyager.env file.
-a	Use this parameter to specify the audit file name.	When nothing is specified for the audit file name, the default is /m1/voyager/xxxdb/rpt/audit.pxtrx.YYYYMMDD.HHMM.
-e	Use this parameter to specify the error file name.	When nothing is specified for the audit file name, the default is /m1/voyager/xxxdb/rpt/err.pxtrx.YYYYMMDD.HHMM.

Table 5-3. PpatronExpX Parameters

Parameter	Description	Default
-m	Use this parameter to specify the monitor progress (optional).	When nothing is specified, a note is written to the audit file every 100 records/patrons by default.
-b	Use this parameter to specify one of the following actions: • Add • Insert • Update This sets the content of the <action> element when the export script is run.	When nothing is specified, Add is the default action.
-o	Use this parameter to specify add or overlay.	When nothing is specified, the default option is overlay.
-r	Use this parameter to specify the range of records to process.	If nothing is specified for -r or -f, all the records are exported.
-f	Use this parameter to specify the name of the file in the /ml/voyager/xxxxdb/local directory that contains a list of Voyager record IDs (one per line/row).	If nothing is specified for -r or -f, all the records are exported.
-i	Use this parameter to specify one of the following to match for the <matchId> section (in the XML): • Voyager ID (patron ID) • Institution ID • Barcode • SSN	When nothing is specified, the Voyager ID is the default used for the <matchId> section.
-h	Use this parameter to request the help information.	

NOTE:

When minor errors are found in a record (such as no expiration date) during export, these are listed in the <reply> stanza in the exception file.

Web Service for Patron Export

A single patron's record can be exported using the web service and an XML document. Use the following URL format/syntax:

`http://server:port/vxws/patronExport?help=no`

Input can be provided with an XML document that contains the patron record ID and ID type (barcode, institution ID, or Voyager patron ID). See [Figure 5-5](#) on [page 5-11](#)

```
<exportParams>
  xmlns="http://www.exlibris.com/Voyager/patron">
    <idNumber>ABC549</idNumber>
    <idType>Barcode</idType>
  </exportParams>
```

Figure 5-5. <exportParams> Example

Input can also be provided by using the following parameters in the URL to specify patron ID and ID type:

- `Id=<insert patron ID>`
- `IdType=<Barcode, InstitutionID, or VoyagerID>`

Refer to EL Commons as described in the [Overview](#) on [page 5-1](#) for more information.

PpatronPrg (Patron Purge)

The patron purge shell script is run from the `/m1/voyager/xxxdb/sbin` directory.

See [Table 5-4](#) for the list of parameters provided with PpatronPrg. All parameters are optional except where identified as required.

Table 5-4. PpatronPrg Parameters

Parameter	Description	Default
-d	Use this parameter to specify the Voyager database name.	When no database is specified, the program uses the database name identified in the <code>/m1/voyager/xxxdb/ini/voyager.env</code> file.

Table 5-4. PpatronPrg Parameters

Parameter	Description	Default
-u	Use this parameter to specify the user name and password for the database.	When no user name and password is specified, the program uses the user name and password identified in the /m1/voyager/xxxdb/ini/voyager.env file.
-g	Use this parameter to specify the database host. NOTE: This must be an IP address.	When no database host is specified, the program uses the database host identified in the /m1/voyager/xxxdb/ini/voyager.env file.
-a	Use this parameter to specify the audit file name.	If no audit file name is specified, the program uses /m1/voyager/xxxdb/local/audit.patpurge.YYYYMMDD.HHMMSS.
-f	Use this parameter to specify the location to which exception files should be written. Refer to Table 5-5 for a list of exceptions.	When no path is specified, the exception files are written to the /m1/voyager/xxxdb/rpt directory.

Table 5-4. PpatronPrg Parameters

Parameter	Description	Default
-p	Use this option to specify that patron records should be purged based on the patron purge date. Optionally, you can specify a date in the format YYYY-MM-DD to override the system date. NOTE: Either -p or -e is required except for when -i is used. Both can be specified. When both are specified, the patron record must meet both criteria.  IMPORTANT: <i>The date comparison logic used for this parameter is less than or equal to. As a result, patron records are removed from the database when the date in the database is equal to or less than the date specified in the -p parameter.</i>	If this parameter is present, the date must be specified.

Table 5-4. PpatronPrg Parameters

Parameter	Description	Default
-e	Use this option to specify that patron records should be purged based on the expiration date. Optionally, you can specify a date in the format YYYY-MM-DD to override the system date. NOTE: Either <code>-p</code> or <code>-e</code> is required except for when <code>-i</code> is used. Both can be specified. When both are specified, the patron record must meet both criteria.  IMPORTANT: <i>The date comparison logic used for this parameter is less than. As a result, patron records are removed from the database when the date in the database is less than the date specified in the <code>-e</code> parameter.</i>	If this parameter is present, the date must be specified.
-i	Use this parameter to specify the name of a file that contains patron IDs that are to be used to identify the patron records to be deleted. The file specified should be a flat file with one patron ID per row. The <code>-i</code> parameter is mutually exclusive from <code>-p</code> and <code>-e</code>. One of the three (<code>-i</code>, <code>-p</code>, or <code>-e</code>) is required. For the patron records to be removed from the database, you must also use the <code>-z</code> parameter.	

Table 5-4. PpatronPrg Parameters

Parameter	Description	Default
-z	Use this parameter to indicate that patron records should be removed from the database.	If this parameter is not specified, report files are created, but the patron records are not removed from the database. NOTE: By default, this job runs in test mode.
-x	Use this parameter to specify the file name and/or path for the XML file of deleted patrons	When no file name and/or path is specified, the following path and/or file name is used: /ml/voyager/xxxdb/rpt/ xml.patpurge.YYYMMDD. HHMMSS
-h	Use this parameter to request the help information.	

Exceptions

Patrons with the exceptions identified in [Table 5-5](#) cannot be purged/removed from the database.

Table 5-5. Purge Exceptions

Field Checked	Exception
sqlerror	SQL error
itemscharged	Current charged items
holdrecall	Current hold/recall
finestees	Current fines/fees
demerits	Current demerits
exception	A circulation transaction exception has been logged against the patron
bookings	Current bookings
callslip	Current callslip requests
routinglist	Patron is on a routing list
shortloan	Current short loans
proxy	Patron is a proxy for another patron

Table 5-5. Purge Exceptions

Field Checked	Exception
ubcharge	Current UB charges
ubfine	Current UB fines
ubrequest	Current UB requests
ubdemerits	Current UB demerits
stubexists	A stub record exists for the patron
historicalfines	The patron has historical fines or fees. NOTE: When the global parameter <code>Delete Patrons with Historical Fines</code> is set to Y, these patrons can be deleted.

Web Admin

Web Admin has been updated to provide patron record import and export capability with XML. These options can be accessed via the Circulation > Reports and Notices link. Select one of the following (radio button) options:

- Patron Import ([Figure 5-6](#))
- Patron Export ([Figure 5-7](#))

☒ Patron Import	Input File: <i>(required)</i> <i>Default directory is \$VOYAGER/ \$DATABASE/local</i>
	Audit File: <i>Default directory is \$VOYAGER/ \$DATABASE/rpt Default filename is audit.pupdx.YYYYMMDD.HHMM</i>
	Error File: <i>Default directory is \$VOYAGER/ \$DATABASE/rpt Default filename is err.pupdx.YYYYMMDD.HHMM</i>
	Operator ID: <i>Default is XMLLOAD.</i>
	Monitor Progress <i>Default is every 100 patrons.</i>
	Stop After Errors: <i>Default is to terminate job after 100 errors.</i>
	Run In TestMode: ☒ Yes ☐ No <i>Patron records will be validated, but no records will be added or updated.</i>
	Load Transaction Counters: ☐ Yes ☒ No <i>(if present in input data)</i>

Figure 5-6. Web Admin - Patron Import

☐ Patron Export	Output File:
	Audit File:
	Error File:
	Monitor Progress:
	Default every 100
	patrons.
	Range: nnn or xxx-yyy:
Voyager Id file: Place in \$VOYAGER/ \$DATABASE/local. If no range or Voyager Id file, export all records.	
Action: When creating extracted records, default Action for all records Add ▼	
Overlay Action: When creating extracted records, overlay Action for individual sections Overlay ▼	
Match ID: When creating extracted records, ID that will be used in Matchid section VoyagerID ▼	

Figure 5-7. Web Admin - Patron Export

The Web Admin fields for import and export correlate with the script options for the XML interface. Refer to EL Commons as described in the [Overview](#) on [page 5-1](#) for additional information regarding these options.

NOTE:

The XML interface version of patron purge is only executable from the /m1/
voyager/xxxdb/sbin directory.

Introduction

The Patron Extract program allows you to retrieve patron records from the database and save that information in a file.

After running this program an output file of patron records is produced. It also creates an audit and error file.

Purpose of this Chapter

This chapter provides

- Overview of the patron extract program
- Parameters that govern the program and running the program
- Output file specification
- Additional files created

Overview of the Patron Extract Program

The patron extract program draws patron records from the database and stores them in a data file that you specify. To do this you must have read-access to the server where your database is stored.

The program extracts patron records to an output file, and creates an audit, and an error file. The default location for these files is `/ml/voyager/xxxdb/rpt` directory, unless otherwise specified by the operator.

The patron extract program retrieves a maximum of three barcodes per patron and pulls active barcodes first. Therefore, in cases of patron records with four or more barcodes only the first three active barcodes will be imported. Also, the extract program will retrieve only the first ten statistical codes per patron.

The extract program should be run from the `/ml/voyager/xxxdb/sbin` directory on your server. Running `Pptrnext` executes the script which has been configured with your database name, username, and password. Therefore, you will not need to enter that information. If `Pptrnext` is entered without any parameters, the program will query the user for the required information.

If you want to run a server activity using a telnet session, and you do not want to perform it interactively, you can use the following command line to make certain that the activity continues, even if the telnet session is lost,

```
nohup <activity with parms> &
```

where `<activity with parms>` is the command followed by any required and optional parameters. This command will not guarantee that the activity will be completed if the server itself is not available.

Parameters

The following parameters govern the `Pptrnext` patron extract program.

-y Required.

Specify `-y` (yes, really) to run the `Pptrnext` legacy program. For the newer version, refer to `PpatronExpX` (described in Chapter 5) that is located in the `/ml/voyager/xxxdb/sbin` directory.

-d Database name -- not required.

Automatically specified by the `Pptrnext` script. Specifying the database name with this parameter overrides the name in the script.

It is the name of the database that will be accessed. You must have read-access on the server to run the extract program.

-u Username and password -- not required.

Automatically specified by the `Pptrnextr` script. Specifying the username and password with this parameter overrides the name and password in the script.

It is the username and password for access to the specified database. Enter it in this format: `username/password`.

-p Patron output filename -- not required.

The name of the output file containing the patron information.

The default is `sif.pxtr.YYYYMMDD.HHMM`, where `YYYY` stands for the year, `MM` for the month, `DD` for the day, `HH` for the hour, and `MM` for the minutes. It is placed in the `/m1/voyager/xxxdb/rpt` directory.

If you specify *only* a filename it is placed in the `/m1/voyager/xxxdb/local` directory.

However, if you want the file placed in a different directory you must provide the complete path and filename. For example, you could enter

```
-p /m1/voyager/xxxdb/log/patron.dat.
```

-a Audit filename -- not required.

The name of the audit file where the audit information will be saved.

The default is `log.pxtr.YYYYMMDD.HHMM`, where `YYYY` stands for the year, `MM` for the month, `DD` for the day, `HH` for the hour, and `MM` for the minutes. It is placed in the `/m1/voyager/xxxdb/rpt` directory.

If you specify *only* a filename it is placed in the `/m1/voyager/xxxdb/local` directory.

However, if you want the file placed in a different directory you must provide the complete path and a filename. For example, you could enter

```
-a /m1/voyager/xxxdb/log/audit.ptn.
```

-e Error filename -- not required.

The name of the error file where any error records and messages will be saved. If there are no errors an empty file is created.

The default is `err.pxtr.YYYYMMDD.HHMM`, where `YYYY` stands for the year, `MM` for the month, `DD` for the day, `HH` for the hour, and `MM` for the minutes. It is placed in the `/m1/voyager/xxxdb/rpt` directory.

If you specify *only* a filename it is placed in the `/m1/voyager/xxxdb/local` directory.

However, if you want the file placed in a different directory provide the complete path and a filename. For example, you could enter

```
-e /m1/voyager/xxxdb/log/error.ptn.
```

-m Monitor records to process -- not required.

The number of records the program will process before a message is displayed, indicating the program's progress.

The default is 100.

-r Number of records to process -- not required.

The number of records that will be processed. To process all the records, type the number zero (0).

The default is 0 (All).

-s Extract stub and child patron records -- not required.

If your database contains stub or child patron records, the `-s` parameter instructs the system to extract all patron records. If the `-s` parameter is not used, only parent patron records are extracted.

-h Help -- not required.

Provides online help about the `Pptrnextr` function. This flag cannot be used with any other parameters.

Running Patron Extract

To run the patron extract command, enter the `Pptrnextr` command from the `/m1/voyager/xxxdb/sbin` directory. The command may be entered on one line with parameters as follows:

```
Pptrnextr -y -p patron.dat -e error.ptn -a audit.ptn -  
m 10 -r 1
```

This command instructs the system to create a patron output file called `patron.dat`. Any errors will be written to the file named `error.ptn` and any audit information will be written to the file named `audit.ptn`. The program will display a message for every ten records processed and it will process only one input record.

Output File Specification

The output file of the patron extract program uses the Patron Record Standard Interface File (SIF).

For information regarding the specific file format see [Patron Record SIF Format on page 19-2](#).

The default name is `sif.pxtr.YYYYMMDD.HHMM`, where YYYY stands for the year, MM for the month, DD for the day, HH for the hour, and MM for the minutes. It is placed in the `/ml/voyager/xxxdb/rpt` directory.

Additional Files

In addition to the output file of patron records, the patron extract program creates two other files, an audit file and an error file.

Audit File

The audit file contains audit information from running the program.

The default audit filename is `log.pxtr.YYYYMMDD.HHMM`, where YYYY stands for the year, MM for the month, DD for the day, HH for the hour, and MM for the minutes. It is placed in the `/ml/voyager/xxxdb/rpt` directory.

This file includes warning messages and information such as:

GRAND TOTALS

Patron Records Received: 216

Patron Records Added: 216

Records Written To Exception File: 0

DETAIL TOTALS

Patron Barcode Records: 258

Patron Status Records: 13

Patron Address Records: 390

Patron Phone Records: 200

Patron Notes Records: 52

END OF REPORT

Error File (Exception File)

The error file contains error records and messages if there were errors when the program ran.

The default is `err.pxtr.YYYYMMDD.HHMM`, where YYYY stands for the year, MM for the month, DD for the day, HH for the hour, and MM for the minutes. It is placed in the `/ml/voyager/xxxdb/rpt` directory.

The error file contains error messages if there were errors when the program ran.

NOTE:

If there were no errors, a file is created, however it is empty.

Introduction

The patron update program allows you to update patron records at any time.

To do this, sites must create an input file of patron records (usually from information provided by the Registrar's Office or other source).

After running this program, audit and error files are created.

Purpose of this Chapter

This chapter provides

- Overview of the patron update program
- Input file specification
- Parameters that govern the program and running the program
- Additional files created

Overview the Patron Update Program

The patron update program updates existing patron records and adds any new patrons included in the input file. To do this you must have write-access to the server where your database is stored.

The purpose of this program is to update existing patron personal information such as Name, Address, and the like. This program does not update or change Patron barcodes or Patron Groups because of the associated circulation history. However, operators may use the program to add new barcodes or groups. If using Patron Update to create a new barcode or group, be sure to consider expiring the original barcode/group.

The program updates patron records on the server based on the information in the input file. The program will look in the `/m1/voyager/xxxdb/local` directory for the input file if a complete path is not specified.

This program generates audit and error files. The default location for these files is the `/m1/voyager/xxxdb/rpt` directory. However, if a filename is provided by the operator, the default location for the audit and error files is the `/m1/voyager/xxxdb/local` directory.

The program will only input a maximum of three barcodes per patron.

The update program should be run from the `/m1/voyager/xxxdb/sbin` directory on your server. Running `Pptrnupdt` executes the script which has been configured with your database name, username, and password. You will not need to enter that information. If `Pptrnupdt` is entered without any parameters, the program will query the user for the required information.



IMPORTANT:

This batch program should only be run at times of low system use or in the evenings. Your staff may note slow response times if run outside of these suggested times.

If you want to run a server activity using a telnet session, and you do not want to perform it interactively, you can use the following command line to make certain that the activity continues, even if the telnet session is lost,

`nohup <activity with parms> &`

where `<activity with parms>` is the command followed by any required and optional parameters. This command will not guarantee that the activity will be completed if the server itself is not available.

NOTE:

For sites that participate in Voyager's Universal Borrowing or sites that have multiple circulation clusters, the patron update batch job does not update or add any stub patron records. However, the audit file provides information on how many stub records were skipped.

Input File Specification

The input file for the Patron update should be supplied in the same format as required in the Voyager patron load program. It is the Patron Record Standard Interface File (SIF).

For information regarding the specific file format see [Patron Record SIF Format](#) on [page 19-2](#).

The program will look in the `/m1/voyager/xxxdb/local` directory if a path is not specified with the filename.

Parameters

The following parameters govern the patron update program.

-p Patron input filename -- required.

The name of the file that contains the patron information. If you specify the filename but do not specify the full path to the file, the update program will look in the `/m1/voyager/xxxdb/local` directory. Therefore, if the input file is not in the `../local` directory, you must include the complete path information to the file.

-d Database name -- not required.

Automatically specified by the `Pptrnupdt` script. Specifying the database name with this parameter overrides the name in the script.

It is the name of the database that will be updated. You must have write access on the server to run the update program.

-u Username and password -- not required.

Automatically specified by the `Pptrnupdt` script. Specifying the username and password with this parameter overrides the name and password in the script.

It is the username and password for access to the specified database. Enter it in this format: `username/password`

-i Index type -- not required.

Indicates the match point for patron records. Enter the appropriate one-letter code of the index type immediately after the flag `-i`.

S SSAN

This causes the program to update patron records based on matching the Social Security Number (SSAN) whether an Institution ID is present or not.

I Institution ID

This causes the program to update patron records based on matching the Institution ID whether an SSAN is present or not.

D Program selected

Based on the data contained in the first input record, this program chooses whether to use the SSAN or the Institution ID to determine whether to update an existing record or add the input item as a new record. It will check first for a SSAN. If one exists, it will use the SSAN to match records for update. If not, it will use the Institution ID to match records for update. If neither exists, the record will be rejected and a choice will be made based on the contents of the next record.

The default is D (Program selected).

-a Audit filename -- not required.

The name of the file where any audit information is saved.

The default audit filename is `log.pupd.YYYYMMDD.HHMM`, where YYYY stands for the year, MM for the month, DD for the day, HH for the hour, and MM for the minutes. It is placed in the `/m1/voyager/xxxdb/rpt` directory.

If you specify a filename *only*, the file is written to the `/m1/voyager/xxxdb/local` directory.

However, if you want the file placed in a different directory provide the complete path and a filename. For example, you could enter

```
-a /m1/voyager/xxxdb/log/audit.dat.
```

-e Error filename -- not required.

The name of the file where any error messages are saved.

The default error filename is `err.pupd.YYYYMMDD.HHMM`, where YYYY stands for the year, MM for the month, DD for the day, HH for the hour, and MM for the minutes. It is placed in the `/m1/voyager/xxxdb/rpt` directory.

If you specify a filename *only*, the file is written to the `/m1/voyager/xxxdb/local` directory.

However, if you want the file placed in a different directory provide the complete path and a filename. For example, you could enter

```
-e /m1/voyager/xxxdb/log/error.dat.
```

-n Notes type -- not required.

Indicates the type of note that will be created for any that are stored in the input file. Enter the appropriate one-letter code of the note type immediately after the flag -n.

G General

This will cause all notes records to be created with the type general.

P Popup

This will cause all notes records to be created with the type pop-up.

The default is G (General).

-o Overlay previous notes -- not required.

Determines how notes in the input file are handled.

A Add new notes

All notes in the input record are added to the database as additional note(s) for this patron.

R Replace existing notes

If notes exist in the input file, all existing notes for this patron will be removed from the database and the notes from the input record will be added to the database as the notes for this patron. If notes for this patron do not exist in the input file, existing notes will not be removed.

I Ignore notes processing

Does not process any notes (new or existing).

The default is A (Add new notes).

-x Extended notes processing -- not required.

Allows multiple notes to be included in the Notes segment of the input record.

These notes are then added to the database as separate notes for the specified patron.

In order for the -x parameter to be able to process multiple notes for a patron, each note must be preceded by a tab character in the input record, including the first note.

-m Monitor records to process -- not required.

The number of records the program processes before a message displays, indicating the program's progress.

The default is 100.

-r Number of records to process -- not required.

The number of records that are processed. To process all the records, enter the number 0.

The default is 0 (All).

-h Help -- not required.

Provides online help about the `Pptrnupdt` function. This flag cannot be used with any other parameters.

Running Patron Update

Before you can run the patron update program you must first create the patron update input file, see [Patron Record SIF Format](#) on [page 19-2](#) in this user's guide.

Users can enter the `Pptrnupdt` command without any parameters. This prompts the program to ask for the parameters interactively from the user.

At the `/ml/voyager/xxxdb/sbin>` enter **`Pptrnupdt`**, the system prompts for input of any required parameters.

For example, if at `/ml/voyager/xxxdb/sbin>` you entered **`Pptrnupdt`**, the system would respond with the following prompts: (possible responses in bold)

```
Enter Input Patron File Name:patron.dat
Enter Audit Report File Name:audit.dat
Enter Error Record File Name:error.dat
Enter Index Type:  S=SSAN, I=Inst.ID, D=Default:D
Enter Default Notes Type -- G=General, P=Pop-up:G
Enter Notes Overlay Type -- A = Add new notes to old
ones, -- R = Replace old notes with new ones, -- I =
Ignore New Notes:I
Enter Extended Notes Processing: -- Y=Yes, N=No:n
Monitor progress after every ### records (enter ###):0
```

Maximum # of records to process, 0 = ALL (enter ###):0

If you do not want to enter the command interactively, an example of the `Pptrnupdt` command with parameters might be entered (on one line) as follows:

```
Pptrnupdt -p patron.dat -i s -e error.dat -a audit.dat  
-n p -m 1 -r 10
```

This command instructs the system to use the input file named `patron.dat`, match the patron records the SSAN, to create an error file named `error.dat`, to create an audit file named `audit.dat`, and to create pop up notes for any notes in the input file, to display a message for every one record processed and to process ten input records.

Additional Files

The patron update program creates an audit (log) file and an error file.

Audit File

The audit file contains audit information from running the program.

The default audit filename is `log.pupd.YYYYMMDD.HHMM`, where YYYY stands for the year, MM for the month, DD for the day, HH for the hour, and MM for the minutes. It is placed in the `/ml/voyager/xxxdb/rpt` directory.

This file includes warning messages and information such as:

```
INVALID RECORD DATA!

Attempt to update LOST/STOLEN barcode
Barcode: 2602

This barcode has NOT been updated for
SSAN:

Record written to exception file.
Patron ID: 3

GRAND TOTALS

Input Patron Records Received:      10
New Patrons Added:                  0
Existing Patrons Updated:            10
Records Written To Exception File:   1

DETAIL TOTALS
```

<i>New Patron Barcodes Added:</i>	<i>0</i>
<i>Existing Patron Barcodes Updated:</i>	<i>0</i>
<i>New Patron Statuses Added:</i>	<i>0</i>
<i>Existing Patron Statuses Updated:</i>	<i>0</i>
<i>New Patron Addresses Added:</i>	<i>3</i>
<i>Existing Patron Addresses Updated:</i>	<i>10</i>
<i>New Patron Phone records Added:</i>	<i>10</i>
<i>Existing Patron Phone Records Updated:</i>	<i>0</i>
<i>New Patron Notes records Added:</i>	<i>0</i>
<i>Existing Patron Notes Records Updated:</i>	<i>0</i>
<i>Existing Stub Records Skipped:</i>	<i>0</i>

END OF REPORT

Error File (Exception File)

The error file contains the records where there were errors when the program ran.

The default error filename is `err.pupd.YYYYMMDD.HHMM`, where YYYY stands for the year, MM for the month, DD for the day, HH for the hour, and MM for the minutes and it is placed in the `/ml/voyager/xxxdb/rpt` directory.

NOTE:

If there were no errors, a file is created, however it is empty.

Introduction

The bursar transfer system allows users to export either the total patron balances, or itemize patron fines/fees information to another system such as a billing agency.

To do this sites must create a configuration file before running the bursar transfer program.

After running this program an output file of patron balances or itemized fines/fees is produced. It also creates audit and error files.

At the completion of the program, in the Voyager database the fine or fee will be considered paid.

NOTE:

This program does not extract accrued fines, demerit points, or accrued demerit points. Also, it does not extract fines or fees from stub or child patron records.

Purpose of this Chapter

This chapter discusses

- Overview of the bursar transfer system
- General steps of the bursar transfer system

- Supported one-way transfers
- Configuration file creation
- Parameters and an example
- Output file specification
- Additional files
- Bursar transfers in the Circulation module

Overview of the Bursar Transfer System

The bursar transfer allows two types of one-way transfers of patron data. You can transfer the total patron balance or transfer itemized patron fines/fees. To do this you must have read-access to the server where your database is stored .

If transferring the total patron balance, transactions are cumulated to calculate the total balance (debit or credit). Only patron information and the patron's current balance is transferred to the billing agency.

If transferring itemized fines/fees, if beyond a specific amount, individual fine/fee information records and patron information is transferred to the billing agency.

Sites must create a configuration file to govern the transfer of fines. This file allows the linkage between locations and patron groups. The bursar system transfers balances or specific fines/fees that are of the circulation locations designated in the configuration file *and* belonging to patrons that are ACTIVE members of patron groups designated in the configuration file. If a patron has fines but is not an active member of a patron group listed in the configuration file, the patron's fines will not be transferred.

The patron must have a Social Security Number or Institution ID to be transferred.

NOTE:

The system will not extract any fines or fees information from any stub (or child) patrons. The system looks for patron records in which the patron ID indicates it is a stub patron record with associated inter-cluster circulation transactions. If found, that record is skipped and its information not transferred.

The bursar program exports information to an output file, and creates an audit and an error file. The default location for these files is
`/m1/voyager/xxxdb/rpt` directory.

NOTE:

For this batch job, the user cannot provide filenames for the output, audit or error files. They are always the default filenames of `sif.burs.yyyymmdd.hhmm`, `log.burs.yyyymmdd.hhmm`, and `err.burs.yyyymmdd.hhmm`.

The export program should be run from the `/ml/voyager/xxxdb/sbin` directory on your server. Running `Pbursar` executes the script which has been configured with your database name, username, and password. Therefore, you will not need to enter that information. If `Pbursar` is entered without any parameters, the program will query the user for the required information.

If you want to run a server activity using a telnet session, and you do not want to perform it interactively, you can use the following command line to make certain that the activity continues, even if the telnet session is lost,

nohup <activity with parms> &

where `<activity with parms>` is the command followed by any required and optional parameters. This command will not guarantee that the activity will be completed if the server itself is not available.

In the Voyager database, once the information is transferred, the fine or fee is considered paid. The fine or fee may be corrected (reduced or waived) after transfer to the billing agency by making a bursar refund debit/error credit. The bursar refund credit will be exported at the next running of the bursar program.

**IMPORTANT:**

It is STRONGLY RECOMMENDED that this program be run when the library is closed or circulation desk activity is extremely slow or conflicts may occur (that is, fines being paid or posted against while the information is being transferred to the SIF).

NOTE:

This server job can be run using WebAdmin, see [Bursar Transfer](#) on [page 31-30](#).

Steps of the Bursar Transfer System

The following steps need to be taken to successfully run the bursar transfer program.

1. Determine the type of transfer, total balances, or itemized fines/fees.

NOTE:

The program defaults to itemizing the fines/fees unless the operator includes the -m (Patron mode) parameter to transfer just patron total balances.

2. Create the configuration file. The `Pbursar` batch program transfers information only for the patron groups belonging to the Circulation locations specified in the configuration file.
3. Run the `Pbursar` batch program, using the required parameters and any optional parameters for your sites specific needs. Information on parameters follows.
4. When completed the batch program creates the output file, an audit file summarizing the job, and an error file.
5. Your library will have to create a batch program that will transfer the data in the SIF into your accounting system.
6. If you need to refund the money to patrons, you can use the Circulation module to post an amount to the patron(s) by selecting the **Fine/Fee** window and posting an amount for refund. If money is refunded, run the Bursar batch program again.

Supported One-Way Transfer Types

You can transfer fines by patron group and location, as defined in the configuration file, see [Configuration File](#) on [page 8-5](#).

There are two types of supported transfers, transferring total patron balances, and transferring itemized patron fines/fees.

Transferring Total Patron Balances

Only patron information and the patron's current balance is transferred to the billing agency. If the patron has a credit with the library, individual credits are transferred. The receiving system gets no information about specific charges.

Transferring Itemized Patron Fines/Fees

Individual fine/fee information records and patron information is transferred to the billing agency. Specific information about the fines/fees are also transferred to the billing agency.

Configuration File

You must create a configuration file in order for the bursar program to execute. The bursar system reads from this file to determine what information to transfer from the database. It is suggested that this file be created using vi or another server text editor.

The configuration file may have the following stanzas:

- [CIRC LOCATIONS]
- [PATRON GROUPS]
- [FINE FEE TYPE]

After the [CIRC LOCATIONS] stanza, list the codes of the circulation locations for which you want to transfer fine and fee information as defined in **System Administration > System > Locations**. See *Locations* in the *Voyager System Administration User's Guide* for more information.

After the [PATRON GROUPS] stanza, list the codes for the patron groups whose fine and fee information you want to transfer as defined in **System Administration > Circulation > Patron Groups**. See *Patron Groups* in the *Voyager System Administration User's Guide* for more information.

After the [FINE FEE TYPE] stanza (optional), list the fine/fee type codes as defined in **System Administration > System > Fines/Fees** to identify the types of fines/fees that you wanted transferred to the bursar system or you can specify ALL (which must be the first entry/row after the stanza). See *Fines/Fees* in the *Voyager System Administration User's Guide* for more information.

NOTE:

The bursar program will only transfer information pertaining to the specified patron groups at those specific locations. The file must contain both circulation location and patron group codes listed in order for information to be extracted. If you want to transfer information for all circulation locations, all patron groups, and all fine/fee types, enter ALL after the appropriate stanza.

[Figure 8-1](#) shows a sample configuration file.

```
[CIRC LOCATIONS]
ALL
[PATRON GROUPS]
UNDERGRAD
GRAD
[FINE FEE TYPE]
F1
```

Figure 8-1. Sample Configuration File

This file instructs the bursar program to transfer the F1 (overdue) fine/fee type information for the undergraduate and graduate patron groups at all circulation locations.

If you specify the optional `[FINE FEE TYPE]` stanza with no entries after it, the system will stop running the bursar transfer program and report an error when you run the Pbursar script.

Since demerits or accrued fine/fee types should not be transferred to the bursar system, these are skipped if specified in the configuration file. If skipping them results in no valid entries to be transferred, the bursar transfer program will stop and log an error.

If you specify any fine/fee types after specifying `ALL` in the `[FINE FEE TYPE]` stanza, the other fine/fee types are ignored and the system processes all fine/fee types.

NOTE:

If the `[FINE FEE TYPE]` stanza is not specified in the configuration file, the system default is to transfer all fines/fees regardless of type for the patron groups specified at the specified locations.

You can name the configuration file whatever you like. Likewise, you can place this configuration file wherever you like on your server. However, in order for the file to be used, you must specify the full path to the file and filename after the `-c` parameter when running the program.

Parameters

The following parameters govern the bursar transfer program.

-d Database name - not required.

Automatically specified by the `Pbursar` script. Specifying the database name with this parameter overrides the name in the script.

It is the name of the database that is accessed. You must have read-access on the server to run the extract program.

-u Username and password -- not required.

Automatically specified by the `Pbursar` script. Specifying the username and password with this parameter overrides the name and password in the script.

It is the username and password for access to the specified database. Enter it in this format: *username/password*

-c Configuration file -- required.

Specify the full path and filename of the bursar configuration file (for example, *-c /m1/voyager/xxxdb/sbin/bursar.cfg*).

A configuration file must exist in order for the bursar program to execute.

-p Patron mode -- not required.

Transfers patron balance totals. The default is to transfer specific fine/fee information, not patron totals.

-m Minimum Fine/Fee Amount -- not required.

Indicates the minimum amount that the fine/fee must be in order to be transferred (in whole numbers).

For example, *-m10* indicates transfer of fines greater than or equal to \$10.00.

If using Patron mode, it indicates the minimum a fine/fee must be in order to be added to the total.

-i Interval for Transfer -- not required.

Indicates the number of days it must be after the fine create date after which the fine may be transferred.

If using Patron mode, it indicates the number of days it must be after the fine create date after which the fine will be added to the total.

-o Operator ID -- not required.

Specifies the operator ID to be used with the bursar postings.

The default is BURSAR

-t Test Mode -- not required.

In test mode, a SIF file is written, but no database updates are made.

-q Quiet Mode -- not required.

Prevents Voyager from prompting for any missing parameters. No hash marks display to indicate the progress of the transfer.

-v Version Information -- not required.

Provides version information about the current `Pbursar` program.

-h Help -- not required.

Provides online help about the bursar transfer system. This flag cannot be used with any other parameters.

Running Bursar Transfer

Users can enter the `Pbursar` command without any parameters. This prompts the program to ask for the parameters interactively from the user.

At the `/m1/voyager/xxxdb/sbin>` enter **Pbursar**, the system prompts for input of any required parameters.

If you do not want to enter the command interactively, an example of the `Pbursar` command with parameters might be entered (on one line) as follows:

```
Pbursar -c/m1/voyager/xxxdb/sbin/bursar.cfg -p -m10 -  
oJane -i5
```

This command directs the system use a configuration file called `bursar.cfg`. To extract from the database patron balances, instead of fine/fee balances. Balances below \$10.00 are not be retrieved. The operator ID is Jane. The balance transferred does not include fines and fees less than five days old.

Output File Specification

The output file of the bursar transfer program is the `Bursar SIF` file. The SIF file is named `sif.burs.yymmdd.hhmm`. It is placed in the `/ml/voyager/xxxdb/rpt` directory.

Bursar SIF Format

[Table 8-1](#) describes the format of the `Bursar SIF` file.

Table 8-1. Bursar SIF Format

	Length	Field	Comment
1	9	Patron Social Security Number (no dashes)	If the Social Security Number and Institution ID is missing, log patron to error file and process next.
10	30	Patron Institution ID	
40	10	Patron Home Location, Location Code	Leave blank if not found.
50	10	Patron Group Code	Use the first active patron group found for a patron that was selected as a run parameter. (A patron may have more than one active patron, we need to pick one that was specified for the run).
60	10	Fine/Fee Type Code	Patron total runs will use hard-coded value of PTRNTOTAL.
70	10	Fine/Fee Create Date	ccyy.mm.dd, patron total runs will use the run date.
80	10	Voyager Location Code	Location where the fine or fee was generated.
90	1	+/-	Credit (-) the patron's billing agency account or debit (+) the account. A plus sign means that the patron account was credited in Voyager. A minus indicates that the fine or fee was corrected after an initial transfer.
91	16	Net due for fine or fee or patron balance.	In Voyager base currency, use decimal separators and decimal places--no thousands separators.

Table 8-1. Bursar SIF Format

	Length	Field	Comment
107	1000	Description of the Fine/ Fee	This string will be right padded with blanks. For patron balance runs, this will always be blank.
1107		Record Length	

NOTE:

This SIF remains in the Latin-1 character set. If elements in the SIF were stored in the database using the Unicode character set, they are converted to Latin-1 before being exported to the SIF.

Additional Files

In addition to the output file of patron total balances or itemized fines/fees information, the bursar transfer program creates two other files, an audit file and an error file.

Audit File

The audit file contains information about the bursar transfer.

It is named `log.bursar.yymmdd.hhmm`, where y is year, m is month, d is day, h is hour, and m is minutes. It is placed in the `/ml/voyager/xxxdb/rpt` directory.

The file includes the current date, time and summary transaction information.

The summary transaction information for patron balance transfers includes the number of patrons credited, total patron credits processed, the number of patron debited, total patron debits processed, total patrons processed, and net debits and credits.

The summary transaction information for individual fine/fee transfer includes the number of patrons credited, total patron credits processed, the number of patrons debited, total patron debits processed, total patrons processed, and net debits and credits, but is also broken apart by fine/fee type code.

[Table 8-2](#) provides an example of the audit file, the fee column displays the fine/fee types and each row contains a summary for each type.

Table 8-2. Audit Log SIF Format

Fee	Credits	Patrons	Debits	Patrons	Net	Patrons
F1	\$101.23	10	\$12.50	1	\$88.73	11
F2	\$50.01	5	\$0	0	\$50.01	5
Total	\$151.24	15	\$12.50	1	\$138.73	16

NOTE:

For this batch job, the user cannot provide a filename for the audit file.

Error File

The error file stores the message regarding any error records that cannot be written to the SIF file.

The file is named `err.burs.yyymmdd.hhmm`, where y is year, m is month, d is day, h is hour, and m is minutes. It is placed in the `/m1/voyager/xxxdb/rpt` directory.

NOTE:

If there were no errors, a file is created, however it is empty.

Bursar Transfers in the Circulation Module

Within the circulation module one can:

- view bursar transfer information
- give a bursar refund

Fines/fees that have been transferred can be viewed in the **Circulation module> Patron record> Fines/Fees History tab**.

The bursar refund credits the charge up to the amount paid by the bursar transfer and the balance due on that transaction becomes a negative amount. This negative amount is displayed as a current charge until the bursar program is run again and can debit this amount and transfer it to the bursar.

To refund an amount,

1. From a selected bursar transfer item on the **Fines/Fees History** tab, click the **Post** button.
2. Select a posting type of **Bursar Refund** and enter the amount.

This amount is considered a Bursar refund in the next run of the batch program. The next Circulation session, the amount displays as a Forgive.

Introduction

Circulation batch processing is used in a variety of ways. This chapter describes maintenance-related batch processing. Separately, another subset of the circjobs can be run that generate data used as input files to Microsoft Access to generate standardized reports and notices. For more information regarding the Microsoft Access reporting options, see the *Voyager Reporter User's Guide*.

Circulation batch jobs can be run from the `/m1/voyager/xxxdb/sbin` directory on the server or by using [WebAdmin](#). For information regarding WebAdmin to run these circulation batch jobs, see [Using WebAdmin](#) on [page 31-7](#).

From the server, the circjobs can be executed from the command line or run as part of a cron job. For repetitive, maintenance-type batch jobs, you may find it more convenient to schedule the work and run the executables as part of a cron job.

Depending on the task being performed, some circjobs are prompted and require your dynamic input when executed.

After running these jobs, an entry in the `circjob.log` file is made.

Purpose of This Chapter

This chapter explains circulation batch jobs that allow institutions to perform actions on multiple records in order to maximize the efficiency of both the user and the software.

For example, batch jobs 34, 35, and 36 can be used to place items on course reserve lists and change their status as follows:

- Active to Inactive.
- Available to On Hold or Recalled.
- Inactive to off the reserve list entirely.

The circulation-related batch jobs discussed in this chapter are as follows:

- Circjob 1: Update Shelving Status. See [Update Shelving Status \(Circjob 1\)](#) on [page 9-3](#).
- Circjob 8: Archive and Expire Call Slip Requests. See [Archive and Expire Call Slip Requests \(Circjob 8\)](#) on [page 9-4](#).
- Circjob 26: Export OPAC Requests. See [Export OPAC Requests \(Circjob 26\)](#) on [page 9-4](#).
- Circjob 27: Archive Short Loans. See [Archive Short Loans \(Circjob 27\)](#) on [page 9-5](#).
- Circjob 29: Purge UB Patron Stub Records. See [Purge Universal Borrowing \(UB\) Patron Stub Records \(Circjob 29\)](#) on [page 9-5](#).
- Circjob 30: Accrued Fines and Demerits. See [Accrued Fines and Demerits \(Circjob 30\)](#) on [page 9-6](#).
- Circjob 31: Patron Suspension. See [Patron Suspension \(Circjob 31\)](#) on [page 9-7](#).
- Circjob 32: Universal Borrowing Request Promotion. See [Universal Borrowing \(UB\) Request Promotion \(Circjob 32\)](#) on [page 9-8](#).
- Circjob 33: Update Remote Circulation Cluster Cache. See [Update Remote Circulation Cluster Cache \(Circjob 33\)](#) on [page 9-9](#).
- Circjob 34: Place Items on Active Course Reserve List. See [Place Items on Active Course Reserve List \(Circjob 34\)](#) on [page 9-10](#).
- Circjob 35: Place Recalls and Holds for Items on Active Course Reserve List. See [Place Recalls and Holds for Items on Active Course Reserve List \(Circjob 35\)](#) on [page 9-12](#).
- Circjob 36: Take Items on Inactive Course Reserve List Off Reserve. See [Take Items on Inactive Course Reserve List Off Reserve \(Circjob 36\)](#) on [page 9-14](#).

- Circjob 37: Forgive Demerits. See [Forgive Demerits \(Circjob 37\)](#) on [page 9-16](#).
- Circjob 38: Retain Patron IDs. See [Retain Patron IDs \(Circjob 38\)](#) on [page 9-17](#).
- Circjob 39: Patron Purge. See [Patron Purge \(Circjob 39\)](#) on [page 9-23](#).
- Circjob 40: Forgive Fines by Patron ID. See [Forgive Fines by Patron ID \(Circjob 40\)](#) on [page 9-27](#).
- Circjob 41: Forgive Fines by Create Date. See [Forgive Fines by Create Date \(Circjob 41\)](#) on [page 9-30](#).
- Circjob 42: Forgive Fines by Patron Group and Expiration Date. See [Forgive Fines by Patron Group and Expire Date \(Circjob 42\)](#) on [page 9-33](#).
- Circjob 43: Synchronize Patron Counters for Universal Borrowing. See [Synchronize Patron Counters for Universal Borrowing \(Circjob 43\)](#) on [page 9-36](#).
- Circjob 44: Automatically Expire Patron Barcode Status. See [Automatically Expire Patron Barcode Status \(Circjob 44\)](#) on [page 9-41](#).
- Circjob 45: Remap Automatically Mapped UB Stub Patrons. See [Remap Automatically Mapped UB Stub Patrons \(Circjob 45\)](#) on [page 9-42](#).

Update Shelving Status (Circjob 1)

This batch job updates the shelving status of items from discharged to not charged, depending on the interval specified in the Voyager System Administration module.

The frequency with which this job should be run depends on your site's shelving policies.

To run the job, at the `sbin>` prompt, enter

```
Pcircjob -j1
```

The `circjob.log` file should include the day, date, time entry, and possibly the following messages.

```
Thu Mar 21 13:18:13 2002 Update Shelving Status...
```

```
Thu Mar 21 13:18:14 2002 ...COMPLETED
```

NOTE:

This job can be run using WebAdmin, see [Circulation Utilities](#) on [page 31-24](#).

Archive and Expire Call Slip Requests (Circjob 8)

This batch job archives any requests with the status Processed, Not Found, or Expired and changes the status of unprocessed requests to Expired. An expire period of zero would indicate that the request should never expire.

It expires unprocessed requests after the interval specified in the Voyager System Administration module has elapsed. These requests will display in WebVoyage until the job archives them.

The frequency with which this job should be run depends on the interval your site has set up in **System Administration> Call Slip Print Groups> Call Slip Definitions**. See the *Voyager System Administration User's Guide*, for more information.

To run the job, at the `sbin>` prompt, enter

```
Pcircjob -j8
```

The `circjob.log` file should include the day, date, time entry, and possibly the following messages.

```
Thu Mar 21 13:18:13 2002 Archive and Expire Callslip  
Requests...
```

```
Thu Mar 21 13:18:14 2002 ...COMPLETED
```

NOTE:

This job can be run using WebAdmin, see [Circulation Utilities](#) on [page 31-24](#).

Export OPAC Requests (Circjob 26)

This batch job writes to a file on the server all OPAC requests that have been routed to the flat file (for example, item-level flat file requests and ARTEmail photocopy and loan requests). This job does not create any notes or notices. It simply writes the request information to a file on your server. The format in which these are written can be customized. For more information see the *Voyager System Administration User's Guide*, regarding request forms.

To run the job, at the `sbin>` prompt, enter

```
Pcircjob -j26
```

The `circjob.log` file should include the day, date, time entry, and possibly the following messages.

Thu Mar 21 13:18:13 2002 Export OPAC Requests...

Thu Mar 21 13:18:14 2002 ...COMPLETED

NOTE:

This job can be run using WebAdmin, see [Circulation Utilities](#) on [page 31-24](#).

Archive Short Loans (Circjob 27)

Updates the status of current short loan requests and archives expired requests in the Short Loans Search List. Short Loans allow your patrons to place a reservation for an item for a specific time period, and then pick it up and charge it out at the scheduled time. See the *Voyager Circulation User's Guide* for more information about short loans.

This batch job should be put into a cron so that it will run automatically.

To run the job, at the `sbin>` prompt, enter

```
Pcircjob -j27
```

The `circjob.log` file should include the day, date, time entry, and possibly the following messages.

Thu Mar 21 13:18:13 2002 Archive Short Loans..

Thu Mar 21 13:18:14 2002 ...COMPLETED

NOTE:

This job can be run using WebAdmin, see [Circulation Utilities](#) on [page 31-24](#).

Purge Universal Borrowing (UB) Patron Stub Records (Circjob 29)

This batch job purges any empty UB patron stub records that exist in the database.

The system looks for patron records in which the Patron ID indicates a UB patron, and then checks for the existence of fines and fees, holds, UB requests, and charged items. The `-f` parameter instructs the system to check for historical fines and fees.

If the system finds an empty stub patron record and you did not include the -f parameter, the stub record is deleted from the database. If the system finds an empty stub record and you did include the -f parameter, the system checks for historical fines and fees. If fines ever existed in the stub patron record, the system skips to the next stub patron record. If they never existed, the record is deleted.

Also, operators may use the -m parameter to retain stub records that were manually mapped.

To run the job without parameters, at the `sbin>` prompt, enter

```
Pcircjob -j29
```

To run the job with both parameters, at the `sbin>` prompt, enter

```
Pcircjob -j29 -f -m
```

NOTE:

This job can be run using WebAdmin, see [Circulation Utilities](#) on [page 31-24](#).

See the *Voyager Circulation Clusters User's Guide* and the *Voyager Universal Borrowing User's Guide* for more information on Purge UB Patron Stubs (Circjob 29).

Accrued Fines and Demerits (Circjob 30)

This circulation batch job calculates accrued fines or demerits.

If **Use Demerits** is not selected in **System Administration> Circulation> Miscellaneous**, then this job calculates the accrued fine. This is the fine that would be assessed if an overdue item were discharged at the time the job is run.

If **Use Demerits** is selected in **System Administration> Circulation> Miscellaneous**, then this job calculates the accrued demerits. This is the number of demerits that would be assessed if an overdue item were discharged at the time the job is run.

These calculations take into account the grace period that has been set up in the **System Administration> Circulation> Policy Definitions> Matrix> Settings** tab for the specific patron group/item type combination.

When the job runs, for each overdue item, it:

- deletes any previous accrued fine or demerits,
- calculates the fine or demerits for the item as if it were discharged immediately, for all patron records where fines/lost item fees apply,

- determines and applies the fine/fee type reason: accrued fine or accrued demerit
- report any accrued demerits or fines for stub patrons back to their home patron records.

NOTE:

There is no mailed notification of accrued fines or demerits. Patrons can check their accrued fines or demerits by asking at a Circulation desk, or by accessing their Patron Information in WebVoyage.

This job should be run daily and can be added to any current crons you may have set up.

To run the job, at the `sbin>` prompt, enter

```
Pcircjob -j30
```

The `circjob.log` file should include the day, date, time entry, and possibly the following messages.

```
Thu Mar 21 13:18:13 2002 Update Accrued  
Fines.Demerits...
```

```
Thu Mar 21 13:18:14 2002 ...COMPLETED
```

NOTE:

This job can be run using WebAdmin, see [Circulation Utilities](#) on [page 31-24](#).

See the *Voyager Circulation User's Guide* for more information about demerits.

Patron Suspension (Circjob 31)

This circulation batch job applies one or more suspensions to a patron's record if they have reached or exceeded the threshold number of demerits (**Max demerits**). The suspension period takes into account exception dates listed in the `exceptdates.cfg` file. This file should be created even if your site does not have any exception dates. See the Demerits appendix in the *Voyager Circulation User's Guide* for more information.

When the job runs, for each patron, it:

- sums the demerits and compares that sum to the threshold number (if the patron belongs to more than one patron group it will use the lowest threshold amount). If the threshold limit is reached or exceeded then,
- applies the suspension, taking into account any exception dates, and

- subtracts the value of the max demerits for the patron record, then
- checks for additional real demerits that reach or exceed the threshold amount. If so, another suspension will be applied with the begin date of that suspension following the end date of the prior suspension. This occurs until there are no longer enough real demerits to invoke a suspension
- if a stub patron is suspended, subtract the value of max demerits on the home patron record.

NOTE:

Suspension dates should be calculated from the system date (current date) or the patron suspension end date, whichever value is greater.

This job should be run daily and can be added to any current crons you may have set up.

To run the job, at the `sbin>` prompt, enter

```
Pcircjob -j31
```

The `circjob.log` file includes information and messages about how this job ran.

NOTE:

This job can be run using WebAdmin, see [Circulation Utilities](#) on [page 31-24](#).

See the *Voyager Circulation User's Guide* for additional information regarding demerits.

Universal Borrowing (UB) Request Promotion (Circjob 32)

If you have configured Universal Borrowing Request Promotion, Universal Borrowing requests (including patron UB eligible local callslips) which cannot be filled at one holding library (database) will automatically be sent to other holding libraries (databases) for fulfillment. This eliminates the need for manual promotion of requests by patrons. This circjob should be run on the patron's home library database.

Request Promotions are performed by a circjob (circjob 32). This circjob promotes requests automatically to other (specified) databases, adhering to all existing validation checks for borrowing requests. It also keeps track of the status of the requests it promotes at the different databases. The callslip status will reflect successful promotions.

Circjob 32 requires a configuration file, the `promoteXXX.cfg` file, where XXX is the call slip print group. Use the `-p` parameter to specify the `promoteXXX.cfg` file you want. See the *Voyager Universal Borrowing User's Guide* for more information on the contents of the `promoteXXX.cfg` file.

This job should be run daily and can be added to any current crons you may have set up.

NOTE:

The Universal Borrowing Request Promotions circjob takes some time, and is resource-intensive, so choose a suitable time to run it (suitable both to you and your fellow UB partners, whose databases you may be searching).

To run the job, at the `sbin>` prompt, enter

```
Pcircjob -j32 -ppromoteXXX.cfg
```

This is assuming that the `promoteXXX.cfg` file is in the `/ml/voyager/xxxdb/ini` directory on your server, if not add the full path.

The `circjob.log` file contains information on how the job ran, including error information. The `circjob.log` file is in the `/ml/voyager/xxxdb/rpt` directory. See the *Voyager Universal Borrowing User's Guide* for more information the possible error messages, along with a description of each.

NOTE:

This job can be run using WebAdmin, see [Circulation Utilities](#) on [page 31-24](#).

See the *Voyager Universal Borrowing User's Guide* for more information about this batch job.

Update Remote Circulation Cluster Cache (Circjob 33)

For sites that participate in Voyager's Universal Borrowing this batch job will contact all remote UB databases and retrieve their Circulation Clusters and patron groups such that they are available for use in the Circulation module or WebVoyage.

Once contacted the tables in the local database are updated with information from the remote databases. The system then checks that no patron groups mappings are orphaned. That is, if a remote site had deleted a patron group the system updates that information in the local database.



IMPORTANT:

This job should be run after your upgrade to have the Circulation Cluster and Patron Group information available locally.

The job can be run as often as the site wants. It can be set up on a cron. Lastly, it should be run any time that a remote database has altered its patron groups if you are aware of this change.

NOTE:

This job can be run using WebAdmin, see [Circulation Utilities](#) on [page 31-24](#).

For additional information regarding Circjob 33, Update Remote Circulation Cluster Cache, see the *Voyager Circulation Clusters User's Guide*.

Place Items on Active Course Reserve List (Circjob 34)

This batch job sets the On Reserve flag for items on active course reserve lists at one time. This job should be run as necessary.

The parameters available when running this job are described in [Table 9-1](#).

NOTE:

Required entries are indicated by an (R) in the Name field. If only one of the below values is present, the system uses the current date as the other. If neither is provided, the system uses the current date as the start date and an open-ended end date.

Table 9-1. Parameters for Circjob 34

Option	Name	Description
-s	Start date (R)	Beginning date used to find active reserve list.
-e	End date (R)	End date used to find active reserve list.

When the job runs, it:

- finds all reserve lists whose effective date is within the start and end dates provided
- finds all items on these lists where the On Reserve flag is No
- changes the On Reserve flag to Yes

- changes the temporary location of the item to the location associated with the reserve list
- changes the temporary item type of the item to the item type associated with the reserve list
- creates the output file.

With the exception of the output file, this is the same as the manual process done in the Circulation module.

NOTE:

If an item is on multiple reserve lists, it is placed on reserve using the temporary location and temporary item type of the first reserve list the job finds.

The output file is named `circjob.onreserve.yyyymmdd.hhmmss`, where yyyy is the year, mm is the month, dd is the day, hh is the hour, mm is the minute, and ss is the seconds that the file was created. This file is located in the `/ml/voyager/xxxdb/rpt` directory.

The format of the output file is described in [Table 9-2](#).

NOTE:

Required entries are indicated by an (R) in the Name field.

Table 9-2. Description of output SIF from Circjob 34

Name	Value	Description
On Reserve (R)	Y or N	Yes (Y) or No (N) is the On Reserve status of the item after it has been processed.
Item Id (R)	Numeric	The Voyager item id of the item in question.

To run the job, at the `sbin>` prompt, enter

```
Pcircjob -j34 -sYYYYMMDD -eYYYYMMDD
```

NOTE:

This job can be run using WebAdmin, see [Circulation Utilities](#) on [page 31-24](#).

For information regarding Course Reserve, see the *Voyager Circulation User's Guide*.

Place Recalls and Holds for Items on Active Course Reserve List (Circjob 35)

This batch job places Holds and Recalls for items on active course reserve lists at once. This job should be run as necessary.

The parameters available when running this job are described in [Table 9-3](#).

NOTE:

Required entries display an (R) in the Name field. If only one of the below start or end date values is present, the system uses the current date as the other. If neither is provided, the system uses the current date as the start date and an open-ended end date.

Table 9-3. Parameters for Circjob 35

Option	Name	Description
-s	Start date	Beginning date used to find active reserve list. NOTE: If only one of these is present, use the current date as the other. If neither are provided the system uses the current date as the start date and an open ended end date.
-e	End date	End date used to find active reserve list. NOTE: If only one of these is present, use the current date as the other. If neither are provided the system uses the current date as the start date and the an open ended end date.
-i	Operator id (R)	Operator id of person running the batch job.
-L	Location code (R)	This is the location code of the circulation desk that is the create and pickup locations for the holds that the job creates. Location codes with spaces must be entered with quotation marks to process successfully as in the following: <code>Pcircjob -j41 -i operator99 -L "Course Reserves" -s 2008-12-29 -e 2008-12-31</code>

Table 9-3. Parameters for Circjob 35

Option	Name	Description
-D	Due Date	This sets the due date for recalled items. If not provided, the system uses the recall rules defined in the System Administration module. See the <i>Voyager System Administration User's Guide</i> for more information.
-P	Print Location Code	This is the print location code used when generating recall notices. It must be a valid print location. If specified, a recall notice is appended to the current <code>crcnotes.*.inp</code> file, where the * is the print location code of the print location given. If not specified, notices are produced after running Circjob 3. See the <i>Voyager Reporter User's Guide</i> for more information about creating input files for circulation.

When the job runs it:

- finds all reserve lists whose effective date is within the start and end dates provided
- finds all items on these lists where the On Reserve flag is No
- if the item is charged to a patron, place an administrative recall request using the information specified in the command line, generate a recall notice
- if the item is not charged, place an administrative hold request using the information specified in the command line
- creates the output file.

NOTE:

If an item has pending holds or recall requests, cancel those requests. Also if the item is On Hold for a patron cancel that as well.

The output file is named `circjob.holdrecall.yyyymmdd.hhmmss`, where yyyy is the year, mm is the month, dd is the day, hh is the hour, mm is the minute, and ss is the seconds that the file was created. This file is located in the `/ml/voyager/xxxdb/rpt` directory.

The format of the output file is described in [Table 9-2](#)

NOTE:

Required entries display an (R) in the Name field.

Table 9-4. Description of Output SIF from Circjob 35

Name	Value	Description
Type of Request (R)	H, R, or N	Hold, Recall, or No request placed.
Item Id (R)	Numeric	The Voyager item id of the item in question.
Patron Id	Numeric	The Voyager patron id for the patron. Required to place request.
Hold/Recall Id	Numeric	The Voyager Hold/Recall id of the newly-placed request. Required to place request.

To run the job, at the `sbin>` prompt, enter

```
pcircjob -j35 -s YYYYMMDD -e YYYYMMDD -i crclerk  
-L RESV -D YYYYMMDD -P RESV
```

NOTE:

This job can be run using WebAdmin. See [Circulation Utilities](#) on [page 31-24](#).

For information regarding Course Reserve, see the *Voyager Circulation User's Guide*.

Take Items on Inactive Course Reserve List Off Reserve (Circjob 36)

This batch job takes inactive course reserve lists off reserve. This job should be run as necessary.

The parameters available when running this job are described in [Table 9-5](#).

NOTE:

Required entries display an (R) in the Name field. If only one of the below values is present, the system uses the current date as the other. If neither is provided, the system uses the current date as the start date and an open-ended end date

Table 9-5. Parameters for Circjob 36

Option	Name	Description
-s	Start date	Beginning date used to find active reserve list.
-e	End date	End date used to find active reserve list.

When the job runs it:

- finds all reserve lists whose expire date is within the start and end dates provided
- finds all items on these lists where the On Reserve flag is Yes
- for each of these items
 - if the item is on another reserve list whose effective dates include the current date, leave it on reserve
 - if not set the On Reserve flag to No and remove the temporary location

The output file is named `circjob.offreserve.yyyymmdd.hhmmss`, where yyyy is the year, mm is the month, dd is the day, hh is the hour, mm is the minute, and ss is the seconds that the file was created. This file is located in the `/ml/voyager/xxxdb/rpt` directory.

The format of the output file is described in [Table 9-6](#).

NOTE:

Required entries display an (R) in the Name field

Table 9-6. Description of output SIF from Circjob 36

Name	Value	Description
On Reserve (R)	Y or N	Yes (Y) or No (N) is the On Reserve status of the item after it has been processed.
Item Id (R)	Numeric	The Voyager item id of the item in question.

To run the job, at the `sbin>` prompt, enter

```
Pcircjob -j36 -sYYYYMMDD -eYYYYMMDD
```

NOTE:

This job can be run using WebAdmin. See [Circulation Utilities](#) on [page 31-24](#).

For information regarding Course Reserve, see the *Voyager Circulation User's Guide*.

Forgive Demerits (Circjob 37)

This batch job forgives an eligible patron's actual demerits. It is run on an entire database, therefore it includes all the Circulation Clusters within that database. Administrators provide the number of demerits, which may be a fraction, that they want to forgive for each eligible patron. Typically this job will be run at the end of semesters and over holidays.



IMPORTANT:

This batch job does not forgive fines or fees.

Patrons are eligible to have their demerits forgiven if they meet this criteria.

- have actual demerits
- the patron's number of total demerits, actual and accrued, does not exceed that patron group demerit threshold, that is the patron is not blocked. When multiple patron groups exist use the patron group with the lowest threshold.
- the patron is not suspended
- the patron has no outstanding fines or fees

NOTE:

There is no record of these demerits, and changes made by this batch job are permanent.

This job should be run after Pccircjob 30, Accrued Demerits, since it establishes eligibility to have demerits forgiven.

For sites participating in Universal Borrowing, this job may be run against databases in which demerit usage is not enabled. Also, reliable network connection must exist among Voyager databases so the home patron record may be updated. If there is a network interruption, the software will rollback changes in the local cluster for the patron currently being processed.

The -q parameter is used to define the number of demerits to be forgiven.

When the job runs it:

- connects to the remote databases
- finds all eligible patrons
- forgives the number of demerits provided.
- updates the patron record
- creates circjob log file

The `circjob.log` file includes the session beginning and end times, it lists any stub patrons whose demerits were not forgiven due to an inability to connect to their home database, and the total number of patron who had demerits forgiven.

To run the job instructing the system to forgive 25 demerits, at the `sbin>` prompt, enter

```
Pcircjob -j37 -q25
```

NOTE:

This job can be run using WebAdmin, see [Circulation Utilities](#) on [page 31-24](#).

For additional information regarding demerits, see the *Voyager Circulation User's Guide*.

Retain Patron IDs (Circjob 38)

When retention is configured in the System Administration module, the Circjob 38 batch job allows libraries to control what identification information is retained for item circulation and item distribution history using the following parameters:

- Number of patron/patron IDs to retain
- Number of days to retain patron IDs

Configuring retention means that the following are selected on the **Systems - Miscellaneous** workspace:

- **Retain Patron Id for Circ History** check box, and
- **Retain Patron Id for Distribution History** check box.

There are three scenarios determining the number of patrons whose information is retained:

- If a **Retain Patron Id...** check box *is selected* and the job *is not run*, all the associated patron information is retained.
- If a **Retain Patron Id...** check box *is selected* and the job *is run*, the specified number of patron IDs are retained or the patron IDs are retained for the specified number of days.

- If a ***Retain Patron Id...*** check box *is not selected* and the job *is run*, no patron information is retained.

When specifying the number of patrons to keep, the system retains the most recent patron IDs. Recency is determined by using the discharge date and the distribution date.

Essentially, for the last x number of patron IDs, the link between the item and the patron's information is retained. Those patron IDs not kept means that the link between the item and the patron's ID is broken and there is no way to retrieve that information.

For example, libraries that want to keep the circulation history of just the last few patrons who charged an item can define the number of patron IDs to retain at three. When that is specified, the last three patrons who charged the item are kept. This information displays in the Circulation module to operators with appropriate authority to view patron information.



IMPORTANT:

To keep a consistent history, administrator's should request the same number of patron IDs to be retained each time the job is run.

This job can be run as necessary. It can also be added to a cron.

The parameters available when running this job are described in [Table 9-7](#). For this job, you must specify one of these parameters; but you cannot specify both. These parameters require that a numeric value be specified as in -Tn or -An. Validation is performed to ensure that a numeric value is entered. When no value or a non-numeric value is entered, the system prompts you to enter a numeric value.

Table 9-7. Parameters for Circjob 38

Parameter	Name	Description
-T	Number of patron IDs to retain	Use this parameter to specify the number of patrons (-Tn) whose identification information is retained for an item's circulation or distribution history. If zero is specified, no patrons are retained. NOTE: To retain all patron IDs, simply select the appropriate box on the System - Miscellaneous workspace in the System Administration module. There is no need to run this job.
-A	Number of days to retain patron IDs	Use this parameter to specify the number of days to retain patron IDs (-An) for an item's circulation or distribution history.

NOTE:

When an item is discharged, an entry is made in the CIRCULATION_TRANSACTIONS_ARCHIVE table and, if retention is configured, patron identification information is also added to this table. Similarly, when an item is distributed and, if retention is configured, patron identification information is added to the DISTRIBUTION_TRANSACTION table.

When Circjob 38 runs with -T specified:

- for each item in the CIRCULATION_TRANSACTIONS_ARCHIVE table, only the most recent x number of patron ID's (where x is the number provided with the -T parameter) retain the link between the item and the patron information. For all other IDs the link is broken.
- for each item in the DISTRIBUTION_TRANSACTIONS table, only the most recent x number of patron ID's (where x is the number provided with the -T parameter) retain the link between the item and the patron information. For all other IDs the link is broken.

There is no output file for this job. However, an entry in the `circjob.log` file is made indicating that the job was run and the number of records updated (Figure 9-1). In the example in Figure 9-1, there were 16 total records, 5 records are retained as the job parameter requests, and 11 records are updated.

```
Thu Apr 8 11:25:09 2004 RetainPatron...
Thu Apr 8 11:25:10 2004 11 Records updated in the table circ_trans_archive
Thu Apr 8 11:25:10 2004 0 Records updated in the table distribution_transaction
Thu Apr 8 11:25:10 2004 ...COMPLETED
```

Figure 9-1. Example of the Message in Circjob.log file after Running Circjob 38

Patron IDs that are retained for circulation history can be viewed in the Circulation module, if the operator has the authority to view Circulation history. See creating, editing, and deleting circulation profiles in the *Voyager System Administration User's Guide* and the *Voyager Circulation User's Guide* for additional information.

Patron IDs that are retained for distribution history are accessible in the DISTRIBUTION_TRANSACTIONS table using SQL only. They are not visible in the circulation module.

This job can be run using a command line (see [Procedure 9-1, Running Circjob 38 from the Command Line](#)), interactively on the server (see [Procedure 9-2, Running Circjob 38 Interactively](#)), or in Webadmin (see [Circulation Utilities](#) on page 31-24).



Procedure 9-1. Running Circjob 38 from the Command Line

Use the following to run Circjob 38 using a command line.

1. Log on to your Voyager server and navigate to the `/m1/voyager/xxxdb/sbin` directory.
2. At the `sbin>` prompt, enter `Pcircjob -j38 -T#` or `Pcircjob -j38 -A#` and press enter.

For `-T`, the `#` is the number of patron IDs you want to retain. For `-A`, the `#` is the number of days you want to retain patron IDs.

Result: The job runs retaining the patron information based on the criteria that you provided. The system returns to the `sbin` prompt.

The information that displays when the job runs is also entered into the `circjob.log` file.



Procedure 9-2. Running Circjob 38 Interactively

Use the following to run Circjob 38 interactively.

1. Log on to your Voyager server and navigate to the `/ml/voyager/xxxxdb/sbin` directory
2. At the `sbin>` prompt enter **Pcircjob**

Result: The screen displays the possible circulation batch jobs to run (see [Figure 9-2](#)).

```

Initializing...
1 -- Update Shelving Status
2 -- Overdue Notices
3 -- Recall Notices
4 -- Fine Fee Notices (new fines/fees only)
5 -- Hold Recall Available Notices
6 -- Hold Recall Cancelled Notices
7 -- Courtesy (Due) Notices
8 -- Archive and Expire Call Slip Requests

12 -- All Daily Jobs (1 - 8)
14 -- Fine Fee General Statements (all patrons with an outstanding balance)

19 -- Hold Shelf Expired Report (date range required)
20 -- Reserved Items Active Report (date range required)
21 -- Reserved Items Expired Report (date range required)
22 -- Missing in Transit Report
23 -- Transaction Statistics Report (date range required)
24 -- Transaction Exceptions Report
25 -- Global Transaction Statistics Report (date range required)

26 -- Export OPAC Requests
27 -- Archive Short Loans
28 -- Automatic Orders for Distribution Items

29 -- Purge UB Patron Stub Records

30 -- Accrued Fines/Demerits
31 -- Apply Suspensions

32 -- UB Request Promotion

33 -- Update Remote Circulation Cluster Cache

34 -- Place Items on Active Course Reserve List
35 -- Place Recalls and Holds for Items on Active Course Reserve List
36 -- Take Items on Inactive Course Reserve Lists Off Reserve

37 -- Forgive Patron Demerits

38 -- Retain Patron IDs (either the number of patron IDs to retain OR the number of days to retain pa
tron IDs is required)

39 -- Patron Purge

40 -- Forgive Fines/Fees by Patron ID (Patron ID file required)
41 -- Forgive Fines/Fees by Date (date range required)
42 -- Forgive Fines/Fees by Patron Group and Expiration Date

43 -- Synchronize Patron Counters for Universal Borrowing

44 -- Expire Patron Barcodes

45 -- Reswap Automatically Mapped UB Stub Patrons after changing mapping

99 -- Quit
  
```

Figure 9-2. Available Circulation Batch Jobs

3. At the Process Job #? prompt enter **38** and press enter.
4. Select one of the following options and press enter:
 - 1 -- Number of patron IDs to retain
 - 2 -- Number of days to retain patron IDs
5. Depending on the option that you selected in the previous step, enter a value when the system prompts you with one of the following:

- Please enter the number of patron IDs to retain:
- Please enter the number of days to retain patron IDs:

Result: After entering a value, the job runs and retains the patron information per your selection and the value entered. The system returns to the `sbin` prompt.

Note that the information displayed on the screen when the job runs is also entered into the `circjob.log` file.

Patron Purge (Circjob 39)

This circulation job allows operators to delete patron records that are marked for deletion from the database. An expiration date and/or a purge date must be set in the patron's record to be considered for deletion (see the *Circulation User's Guide*, Patron Records for information on setting the dates).

Patron records that match the criteria specified on the command line are removed from the database as long as the patron records do not contain any of the exceptions listed in [Table 9-8](#).

Patron Purge Files

When Circjob 39 is run, the system updates the `circjob.log` file with a report and creates many files under the `/ml/voyager/xxxdb/rpt` directory. These files include exception files, a deleted file, and a log file.

For each exception that is encountered while running this job, the system creates a SIF file named `patron.purge.name.yyyymmddhhmmss` where:

- `name` is the exception name from [Table 9-8](#)
- `yyymmdd` is the date (year, month and day)
- `hhmmss` is the time (hours, minutes, seconds)

Each exception file contains the patron records that could not be deleted for that particular exception. For more information on the structure of the patron records, see [Patron Record SIF Format](#) on [page 19-2](#).

If any records are deleted during this job, the system creates a SIF file named `patron.purge.deleted.yyyymmddhhmmss` where `yyymmddhhmmss` is the date in year (`yyyy`), month (`mm`), and day (`dd`). This file contains all patron records that were deleted during the running of this job.

Also, the system creates a file named `patron.purge.log.yyyymmddhhmmss` where `yyyyymmddhhmmss` is the date in year (`yyyy`), month (`mm`), and day (`dd`). This file contains a list of eligible patron record IDs and their deletion status.

The following table lists the reasons for which a patron record *cannot* be purged from the database and the exception file name.

Table 9-8. Patron Purge Exceptions/Names

Name	Exception
sqlerror	SQL Error
itemscharged	current charged items
holdrecall	current hold/recall
finestees	current fees
demerits	current demerits
exception	a circulation transaction exception has been logged against the patron
bookings	current bookings
callslip	current call slip requests
routinglist	patron is on routing list
shortloan	current short loans
proxy	patron is proxy for another patron
ubcharge	current UB charges
ubfine	current UB fines
ubrequest	current UB requests
ubdemerits	current UB demerits
stubexists	a stub record exists for the patron
historicalfines	historical fines or fees NOTE: If global parameter, Delete Patrons with Historical Fines , is set to Y , patron records with historical fines/fees are eligible for deletion. For more information on setting global variables, see the <i>Voyager System Administration User's Guide</i> , Circulation Configuration.

Patron Purge Command Line (Circjob 39)

To run Circjob 39, enter the following command line from the `sbin>` prompt:

```
Pcircjob -j39 -b -a YYYY-MM-DD -z
```

This job allows the operator to set the criteria for purging patrons from the database. Parameters may be entered on the command line to specify the following criteria:

- check purge date (default) or expiration date
- use system date (default) or specified date
- run in test mode (default) or deletion mode

The parameters available when running this job are described in [Table 9-9](#).

Table 9-9. Parameters for Circjob 39

Parameters	Description
-j <job #>	Specify 39 for the patron purge job (required).
-a <YYYY-MM-DD>	Specify a date YYYY-MM-DD to override the system date. NOTE: If this parameter is not specified, the system date is used.
-b	Remove patron records by comparing the patron expiration date with the system date or a specified date. NOTE: If this parameter is not specified, the patron records are removed on the basis of the patron purge date by default.
-z	Must be specified to remove patron records from the database. NOTE: By default, this job runs in test mode. If this parameter is not specified, specification files are created, but the patron records are not removed from the database.

Table 9-9. Parameters for Circjob 39

Parameters	Description
-y	Use this parameter to indicate <i>yes, really</i> use the pre-Voyager 8.2 version of patron purge. NOTE: If you do not use -y, the system prompts you to use the XML interface version of patron purge in the /sbin directory. Refer to PpatronPrg (Patron Purge) on page 5-11 for more information.



Procedure 9-3. Purging Patron Records from the Database (Circjob 39)

This procedure deletes all patron records that have a purge date that occurs before the system date.

1. Log on to your Voyager server and navigate to the /ml/voyager/xxxxdb/sbin directory.
2. At the prompt, type `Pcircjob -j39 -z` and then press enter.

Result: The job updates the `circjob.log` file and creates the following time-stamped files:

- patron purge exception files (a SIF file for each exception that occurs)
- a patron purge log file which lists the activity of the job
- a patron purge deleted file (a SIF file containing all deleted records)

The following information displays ([Figure 9-3](#)):

```
/ml/voyager/xxxxdb/sbin $ Pcircjob -j 39 -z
Initializing...
Patron Purge...
...COMPLETED
```

Figure 9-3. Purging Patron Records from the Database (Circjob 39)

Forgive Fines by Patron ID (Circjob 40)

This batch job forgives fines and fees of patrons based on a supplied list of patron IDs.

This job is typically run when patron records need to be purged from the system and existing fines or fees on patron accounts are keeping them from being deleted.

After the patron ID list is run through the system, the system forgives all the fines and fees for the IDs on the list and generates an error log and audit report (see [Error Logging for Batch Forgive Jobs](#) on [page 9-35](#) and [Audit Report for Batch Forgive Jobs](#) on [page 9-36](#)). Patron IDs can then be purged from the system using Circjob 39.

Circjob 40 can be run from the server using [Procedure , Running circjob 40 from the command line](#) or [Procedure 9-4, Running circjob 40 interactively](#). It can also be run through the WebAdmin client. See [Procedure 31-11, Forgiving fines by patron ID \(circjob 40\)](#), on page [31-24](#).



IMPORTANT:

The file that you direct circjob 40 to process should include a list of carriage-return-delimited patron IDs.

Running circjob 40 from the command line

To run Circjob 40, enter the following command line from the `sbin>` prompt:

```
Pcircjob -j40 -i operatorID -t -p forgive.txt
```

where

- `-i operatorID` = up to 10 alphanumeric characters (required)
- `-t` = run in test mode (optional), and
- `-p forgive.txt` = input file name (required) preceded by location/path if different from `/m1/voyager/xxxdb/local`.



Procedure 9-4. Running circjob 40 interactively

Use the following to run circjob 40 using prompts from the server.



IMPORTANT:

Optional parameters such as running the job in test mode are not available when you use interactive prompts.

1. Log on to your Voyager server and navigate to the `/ml/voyager/xxxxdb/sbin` directory.
2. At the `sbin>` prompt, enter `Pcircjob`

Result: The system displays numbered options for all available circulation batch jobs.
3. At the `Process Job #?` prompt, enter `40`.
4. At the `Operator ID?` prompt, enter your operator ID. (This tags the job with your identification number in the system.)
5. At the `Enter Input Patron File Name:` prompt, enter the file name. If the file is not in the `/ml/voyager/xxxxdb/local` directory, precede the file name with the directory path.

Result: The system runs the batch job and returns the log and audit report.

For information on running the log using the WebAdmin client, see [Procedure 31-11, Forgiving fines by patron ID \(circjob 40\)](#), on page [31-24](#).

Error Logging for Batch Forgive Jobs

Errors for `circjob 40` are logged with other jobs in the comprehensive `circjob.log` as well as in the `err.forgive.YYYYMMDD.HHMM` log. Both can be found in the `/ml/voyager/xxxxdb/rpt` directory.

`circjob.log`

The `circjob.log` records

- the start and end time of the processing and
- any errors that prevent the job from running accurately (such as missing parameters, the inability to locate an input file).

An example of a `circjob.log` entry when no errors are present is:

```
Mon Sep 17 09:42:29 2007 Forgive Fines/Fees...
```


Mon Sep 17 09:42:29 2007 ...COMPLETED

`err.forgive.YYYYMMDD.HHMM`

The error report, `err.forgive.YYYYMMDD.HHMM` includes

- patron ID
- patron last name
- patron first name, and
- error description.

Errors include issues retrieving patron IDs or forgiving eligible fines.

Audit Report for Batch Forgive Jobs

The audit report, `log.forgive.YYYYMMDD.HHMM`, is deposited in the `/ml/voyager/xxxdb/rpt` directory and includes

- patron ID, last name, first name, and amount of forgiven fine(s)
- totals for all patrons combined:
 - number of patrons with fines eligible to be forgiven
 - number of fines successfully forgiven
 - amount of fines successfully forgiven
 - number of patron records with at least one error.

An example of a `log.forgive.YYYYMMDD.HHMM` file:

```
Patron 97: Deer, Beverly forgiven 1 fees totalling 10.00
Patron 98: Fegner, Karen forgiven 1 fees totalling 20.00
Patron 99: Hobat, Matt forgiven 1 fees totalling 30.00
Patron 100: Julley, Eli forgiven 1 fees totalling 40.00
```

```
4 patrons had fines or fees eligible to be forgiven.
4 patrons had fines or fees that were forgiven.
Total amount forgiven: 100.00
0 patrons had at least one error.
```

Forgive Fines by Create Date (Circjob 41)

This batch job forgives fines and fees of patrons based on the date the fines and fees were incurred. An operator can further limit the scope by location.

To run the job directly on the server, use [Procedure , Running circjob 41 from the command line](#) or [Procedure 9-5, Running circjob 41 interactively](#). To run the job using the WebAdmin client, see [Procedure 31-12, Forgiving fines by date created \(circjob 41\)](#), on page [31-26](#).

Running circjob 41 from the command line

To run circjob 41, enter the following command line from the `sbin>` prompt, substituting your own variables for the text in italics.

```
Pcircjob -j41 -i operator99 -r -t -L main -s 1998-09-01  
-e 2002-08-31
```

where

- `-i operatorID` = up to 10 alphanumeric characters (required)
- `-r` = a flag that determines whether fines are forgiven for (Universal Borrowing) stub records (optional).

Enter the `-r` to have stub record fines forgiven.

- `-t` = run in test mode (optional).
- `-L` = forgive fines from these Circulation locations (optional).

The default is **All**.

To specify multiple location codes, enter them separated by commas.

Location codes with spaces must be entered with quotation marks to process successfully as in the following:

```
Pcircjob -j41 -i operator99 -L "Course Reserves" -s  
2008-12-29 -e 2008-12-31
```



IMPORTANT:

The location code must be typed exactly the same (including uppercase/lowercase punctuation) as it is stored in Voyager System Administration. Voyager System Administration permits codes with the same spelling but different uppercase/lowercase usage to be stored concurrently. For example, law, Law, and LAW may all be valid location codes stored concurrently in Voyager System Administration.

- **-s** = fines created after start date (**yyyy-mm-dd**). Required.
- **-e** = fines created before end date (**yyyy-mm-dd**). Required.



Procedure 9-5. Running circjob 41 interactively

Use the following to run circjob 41 using prompts from the server.

**IMPORTANT:**

Optional parameters such as forgiving stub record fines and limiting which Circulation locations to forgive are not available when you use interactive prompts.

1. Log on to your Voyager server and navigate to the `/ml/voyager/xxxxdb/sbin` directory.
2. At the `sbin>` prompt, enter **Pcircjob**.

Result: The system displays numbered options for all available circulation batch jobs.
3. At the `Process Job #?` prompt, enter **41**.
4. At the `Operator ID?` prompt, enter your operator ID. (This tags the job with your identification number in the system.)
5. At the `Start Date?` prompt, enter the start date for forgiving fines (required) in the following format: **yyyy-mm-dd**.
6. At the `End Date?` prompt, enter the end date for forgiving fines (required) in the following format: **yyyy-mm-dd**.

Result: The system runs the job and forgives all fines incurred within the specified dates for the specified groups.

Once the system forgives fines and fees incurred within the specified parameters, you can run batch jobs (such as job 39) that delete patron records from the system.

NOTE:

For information on running circjob 41 using the WebAdmin client, see [Procedure 31-12, Forgiving fines by date created \(circjob 41\)](#), on page [31-26](#).

Error Logging for Batch Forgive Jobs

Errors for circjob 41 are logged with other jobs in the comprehensive `circjob.log` as well as in the `err.forgive.YYYYMMDD.HHMM` log. Both can be found in the `/m1/voyager/xxxdb/rpt` directory.

`circjob.log`

The `circjob.log` records

- the start and end time of the processing and
- any errors that prevent the job from running accurately (such as missing parameters, the inability to locate an input file).

An example of a `circjob.log` entry when no errors are present is:

```
Mon Sep 17 09:42:29 2007 Forgive Fines/Fees...  
Mon Sep 17 09:42:29 2007 ...COMPLETED
```

`err.forgive.YYYYMMDD.HHMM`

The error report, `err.forgive.YYYYMMDD.HHMM` includes

- patron ID
- patron last name
- patron first name, and
- error description.

Errors include issues retrieving patron IDs or forgiving eligible fines.

Audit Report for Batch Forgive Jobs

The audit report, `log.forgive.YYYYMMDD.HHMM`, is deposited in the `/m1/voyager/xxxdb/rpt` directory and includes

- patron ID, last name, first name, and amount of forgiven fine(s)
- totals for all patrons combined:
 - number of patrons with fines eligible to be forgiven

- number of fines successfully forgiven
- amount of fines successfully forgiven
- number of patron records with at least one error.

An example of a `log.forgive.YYYYMMDD.HHMM` file:

```
Patron 97: Deer, Beverly forgiven 1 fees totalling 10.00
Patron 98: Fegner, Karen forgiven 1 fees totalling 20.00
Patron 99: Hobat, Matt forgiven 1 fees totalling 30.00
Patron 100: Julley, Eli forgiven 1 fees totalling 40.00
```

```
4 patrons had fines or fees eligible to be forgiven.
4 patrons had fines or fees that were forgiven.
Total amount forgiven: 100.00
0 patrons had at least one error.
```

Forgive Fines by Patron Group and Expire Date (Circjob 42)

This batch job forgives fines and fees of patrons based on a combination of patron group and patron expiration date.

Running circjob 42 from the command line

To run circjob 42, enter the following command line from the `sbin>` prompt, substituting your own variables for the text in italics.

```
Pcircjob -j42 -i operator99 -r -t -L main, "Course Reserves"
-s 1998-09-01 -e 2002-08-31 -g CAS01 CAS02
```

where

- `-i operatorID` = up to 10 alphanumeric characters (required).
- `-r` = a flag that determines whether fines are forgiven for stub records (optional).

Enter the `-r` to have stub record fines forgiven.

- `-t` = run in test mode (optional).
- `-L` = forgive fines from these Circ locations (optional).

The default is **All**.

To specify multiple location codes, enter them separated by commas.

Location codes with spaces must be entered with quotation marks to process successfully as in the following:

```
Pcircjob -j41 -i operator99 -L "Course Reserves"  
-s 2008-12-29 -e 2008-12-31
```



IMPORTANT:

The location code must be typed exactly the same (including uppercase/lowercase punctuation) as it is stored in Voyager System Administration. Voyager System Administration permits codes with the same spelling but different uppercase/lowercase usage to be stored concurrently. For example, law, Law, and LAW may all be valid location codes stored concurrently in Voyager System Administration.

- -s = patron record expiration date range start (yyyy-mm-dd). Required.
- -e patron record expiration date range end (yyyy-mm-dd). Required.
- -g = patron groups to include in batch write-off.



Procedure 9-6. Running circjob 42 interactively

Use the following to run circjob 42 using prompts from the server.



IMPORTANT:

Optional parameters such as forgiving stub record fines and limiting which Circulation locations to forgive are not available when you use interactive prompts.

1. Log on to your Voyager server and navigate to the /ml/voyager/xxxxdb/sbin directory.

2. At the sbin> prompt, enter **Pcircjob**.

Result: The system displays numbered options for all available circulation batch jobs.

3. At the Process Job #? prompt, enter **42**.
4. At the Operator ID? prompt, enter your operator ID. (This tags the job with your identification number in the system.)

5. At the `Start Date?` prompt, enter the start date for the patron expiration date range for forgiving fines (required) in the following format: `yyyy-mm-dd`.
6. At the `End Date?` prompt, enter the end date for the expiration date range for forgiving fines (required) in the following format: `yyyy-mm-dd`.
7. At the `Forgive fines from these patron groups:` prompt, enter a comma-delimited list of patron group codes to forgive or enter **ALL**.

Result: The system runs the job and forgives all fines incurred within the specified dates for the specified groups.

Once the system forgives fines and fees incurred within the specified parameters, you can run batch jobs (such as job 39) that delete patron records from the system.

NOTE:

For information on running Circjob 42 using the WebAdmin client, see [Procedure 31-13, Forgiving fines by patron group and expiration date \(circjob 42\)](#), on page [31-27](#).

Error Logging for Batch Forgive Jobs

Errors for circjob 42 are logged with other jobs in the comprehensive `circjob.log` as well as in the `err.forgive.YYYYMMDD.HHMM` log. Both can be found in the `/ml/voyager/xxxdb/rpt` directory.

`circjob.log`

The `circjob.log` records

- the start and end time of the processing and
- any errors that prevent the job from running accurately (such as missing parameters, the inability to locate an input file).

An example of a `circjob.log` entry when no errors are present is:

```
Mon Sep 17 09:42:29 2007 Forgive Fines/Fees...
Mon Sep 17 09:42:29 2007 ...COMPLETED
```

`err.forgive.YYYYMMDD.HHMM`

The error report, `err.forgive.YYYYMMDD.HHMM` includes

- patron ID

- patron last name
- patron first name, and
- error description.

Errors include issues retrieving patron IDs or forgiving eligible fines.

Audit Report for Batch Forgive Jobs

The audit report, `log.forgive.YYYYMMDD.HHMM`, is deposited in the `/ml/voyager/xxxdb/rpt` directory and includes

- patron ID, last name, first name, and amount of forgiven fine(s)
- totals for all patrons combined:
 - number of patrons with fines eligible to be forgiven
 - number of fines successfully forgiven
 - amount of fines successfully forgiven
 - number of patron records with at least one error.

An example of a `log.forgive.YYYYMMDD.HHMM` file:

```
Patron 97: Deer, Beverly forgiven 1 fees totalling 10.00
Patron 98: Fegner, Karen forgiven 1 fees totalling 20.00
Patron 99: Hobat, Matt forgiven 1 fees totalling 30.00
Patron 100: Julley, Eli forgiven 1 fees totalling 40.00
```

```
4 patrons had fines or fees eligible to be forgiven.
4 patrons had fines or fees that were forgiven.
Total amount forgiven: 100.00
0 patrons had at least one error.
```

Synchronize Patron Counters for Universal Borrowing (Circjob 43)

Circjob 43 (-j43) is designed to address patron transaction data discrepancies that occur in a Universal Borrowing (UB) environment. Running circjob 43 ensures that Universal Borrowing (UB) data in the home cluster is the most up-to-date by reading the currently available data from remote clusters. It cycles through all the UB-eligible, non-stub patron records in the cluster, connects to each remote cluster, and updates appropriate tables at the home cluster using data from the remote clusters.

In a Universal Borrowing environment you may notice that UB counters in the patron table do not match the real transaction activity throughout the environment. Local and remote tables are intended to be updated as transactions occur.

However, it is possible for transaction activity to occur when a remote location is unavailable that results in patron transaction counters becoming out of sync with reality. This may lead to patrons encountering some circulation block behavior. Your site's experience with these symptoms determines your use of circjob 43.

When run, circjob43 cycles through all the UB-eligible, non-stub patron records in the cluster and connects to each remote cluster; and the appropriate tables are updated at the home cluster using data from remote clusters.

Frequency of Use

Circjob 43 may be run at any time and as frequently as your site decides it is appropriate.

The level of Universal Borrowing activity may be a determining factor.

Remote databases need to be available when synchronizing with those databases using circjob 43.

How to Access

Circjob 43 may be run/accessed in the following manner:

- From the server menu list (see [Server Access](#) on [page 9-37](#)).
- In command line mode on the server (see [Command Line Access](#) on [page 9-39](#)).
- From WebAdmin (see [WebAdmin Access](#) on [page 9-40](#)).

Server Access

To access the circjob 43 option (Synchronize Patron counters for Universal Borrowing) from the server menu list, run `Pcircjob` which is located in `/m1/voyager/xxxdb/sbin/` where `xxxdb` is your database name. This displays the list of available circulation batch jobs. See [Figure 9-4](#).

```
37 -- Forgive Patron Demerits
38 -- Retain Patron IDs (the number of patron IDs to retain required)
39 -- Patron Purge
40 -- Forgive Fines/Fees by Patron ID (Patron ID file required)
41 -- Forgive Fines/Fees by Date (date range required)
42 -- Forgive Fines/Fees by Patron Group and Expiration Date
43 -- Synchronize Patron Counters for Universal Borrowing
; 99 -- Quit

Process Job #? █
```

Figure 9-4. Circjob 43 option on the server

At the `Process Job #?` prompt, enter 43 and any additional parameters. See [Table 9-10](#) for a list of parameters.

Table 9-10. Circjob 43 Parameters

Parameter	Description
-t	Test mode parameter. All changes to the database are rolled back after the reporting is complete for each patron. Nothing actually changes. The test mode report should be the same as the report from a normal run except for the possible discrepancy in the order in which patrons are processed. NOTE: Test mode must be a selected parameter. Circjob 43 does not prompt for it.
-C	Cluster parameter. Specify -C followed by the circulation cluster code. Use this parameter to select a single cluster from a multiple-cluster environment to process synchronization. If you do not specify -C in a multiple-cluster environment, the circjob prompts for a cluster code. If the local database does not have circulation clusters enabled, no other parameters are required. The -C should not be specified. However if it is specified, you must supply the correct circulation cluster code for that cluster.

**TIP:**

A complete list of parameters for all circulation batch jobs can be displayed by running `Pcircjob -h` from the `sbin` directory.

Command Line Access

Circjob 43 can be run/entered in command line mode.

For example, enter the following at the `/ml/voyager/xxxdb/sbin/` prompt (where `xxxdb` is your database name) to run circjob 43 in test mode:

```
Pcircjob -j43 -t
```

Or using the cluster code, see the following example:

```
Pcircjob -j43 -C C01DB
```

WebAdmin Access

Circjob 43 can be accessed from WebAdmin.

Select **Reports and Notices** from the Circulation menu list in WebAdmin to display the circjob 43 option. See [Figure 9-5](#).

 Synchronize UB Information	Circ Cluster Code (for multi-cluster DB only) Run In Test Mode? ☐

Figure 9-5. Circjob 43 option from WebAdmin

[Table 9-10](#) on [page 9-39](#) describes the cluster parameter and the test mode parameter.

For more information regarding WebAdmin, see Chapter [31](#) titled "[WebAdmin](#)."

Circjob Log

The `log.ubsynch.YYYYMMDD.HHMM` is the primary log file for circjob 43. It reports details on every stub analyzed.

Consistent with other circjob processing, general status reports and error messages are appended to the `circjob.log` located in `/ml/voyager/xxxdb/rpt/` where `xxxdb` is your database name.

Error Conditions

The system registers an error condition in the following situations:

- Invalid Cluster Code option specified for `-C`.
- Parameter error preventing access to the local database.
- Local database has no clusters.
- Local database not enabled for Universal Borrowing.
- Failure to connect to a remote database.
- Failure to retrieve data from a remote library.
- Failure to update local data for a patron.

Automatically Expire Patron Barcode Status (Circjob 44)

This batch job is designed to enable you to change the barcode status from Active to Expired for selected groups of patrons whose expiration date is on or before a certain date.

The frequency with which this job should be run depends on your site's patron barcode expiration policies.

To run the job, at the `sbin>` prompt, enter

```
Pcircjob -j44
```

This job prompts you for:

- Patron Group

This setting is required.

You can specify one or more patron groups (such as FAC, GRAD, and/or STU, for example) or ALL. ALL is the default.

At the prompt, press Enter (leave blank) to accept the default.

- Expiration Date

This setting is required.

For this prompt, specify the expiration date. Patrons whose records expire on or before this date have their barcode expiration status changed from Active to Expired.

Enter the date in the YYYYMMDD format. Today's date is the default.

At the prompt, press Enter (leave blank) to accept the default.

Optionally, you may specify a modifying operator ID with the `-i` option followed by the operator ID. See the example below.

```
Pcircjob -j44 -i 12345
```

Remap Automatically Mapped UB Stub Patrons (Circjob 45)

The Circjob 45 batch job performs a remapping of the automatically mapped Universal Borrowing (UB) stub patrons after the patron group mapping is changed in Voyager System Administration.

**IMPORTANT:**

This job does not remap manually mapped stub patrons.

The Voyager System Administration operation is done in Circulation - Patron Group Mapping. Patron Group Mapping defines which patron groups from another (remote) circulation cluster are mapped into your active (local) circulation cluster. Refer to the *Voyager Universal Borrowing User's Guide* for more information about this circulation configuration.

The Circjob 45 should only be run after the Circulation - Patron Group Mapping change in Voyager System Administration is complete.

**TIP:**

Keep your Voyager System Administration client open while running this job. It will be helpful in choosing the database code and checking the log file after the job is complete.

The parameters available when running this job are described in [Table 9-11](#).

Table 9-11. Circjob 45 Parameters

Parameter	Description
-R <database code>	Use to specify the required database code to process. Use ALL to specify processing all of the database codes.
-U	Perform the remap update.

When the job runs, it does the following:

- Finds all of the UB stub patrons for the database specified or all databases if requested
- Checks to see if the mapped patron group has changed for the stub patron

- Reports which stub patrons need to be updated
- Updates the stub patrons with the new patron group if the -U parameter is included

To run this job in test mode, enter the following at the `sbin>` prompt:

```
Pcircjob -j45 -R <DB_CODE>
```

To run this job and perform the update, enter the following at the `sbin>` prompt:

```
Pcircjob -j45 -R <DB_CODE> -U
```

To run this job for all database codes, enter the following at the `sbin>` prompt:

```
Pcircjob -j45 -R ALL -U
```

NOTE:

This job only runs on a UB-enabled database. It can also be run using WebAdmin.

Introduction

The media scheduling batch jobs discussed here are jobs that do not get processed using the reporter module. They do not produce a report or notice, but are still important for media services.

NOTE:

Those media scheduling batch jobs that produce an input file (Mediajobs 1 - 4) and are consequently run through the reporter module to create a report or notice and are discussed in the *Voyager Reporter User's Guide*.

Purpose of this Chapter

This chapter discusses the media scheduling batch job, Mediajob 5: Retain Patron IDs.

Retain Patron IDs (Mediajob 5)

This media scheduling batch job allows libraries to define the number of patrons (patron IDs) whose identification information is retained (or not retained) for media booking history providing retention is configured in the System Administration module.

Configuring retention means that the **Retain Patron Id for Media Booking History** check box on the **Systems - Miscellaneous** work space is checked.

There are three scenarios determining the number of patrons whose information is retained:

- If the **Retain Patron Id for Media Booking History** check box is checked and the job is not run, all the associated patron information is retained.
- If the **Retain Patron Id for Media Booking History** check box is checked and the job is run, the specified number of patron's information is retained.
- If the **Retain Patron Id for Media Booking History** check box is not checked and the job is run, no patron information is retained.

It may be easier to think of this job as when media bookings are *not* retained. Since media bookings can contain multiple resources, each element of the booking is evaluated for retention. The link between the patron's information and the booking is broken only if every resource in the booking has been charged/scheduled more times than the number of IDs (parameter -n) provided in the job.

For example, a booking might include a videotape, a VCR, and a scheduled room to view the tape. Also in this example assume that the video tape has been charged out ten times, the VCR has been charged out five times, and the room has been scheduled two times.

- If this job is run asking to retain four IDs, the above booking is retained because the room has only been scheduled two times.
- If, however, this job is run asking to retain one ID, this booking is not retained because each element, the tape, VCR, and room have been charged more than one time.

NOTE:

To keep a consistent history, administrator's should request the same number of patron IDs to be retained each time the job is run.

This job can be run as necessary. It can also be added to a cron.

The parameter available when running this job is described in [Table 10-1](#). An (R) designates a required parameter.

Table 10-1. Parameter for Mediajob 5

Parameter	Name	Description
-n (R)	Number of IDs to retain	Number of patrons whose identification information is retained for a media booking history. If zero, no patrons are retained. NOTE: If you want to retain all patron IDs, simply check the Retain Patron Id for Media Booking History check box on the System - Miscellaneous work space in the System Administration module. There is no need to run this job.

NOTE:

When a booking is archived, an entry is made in the MEDIA_BOOKING_ARCHIVE table, and if retention is configured, patron identification information is also added to this table.

When Mediajob 5 runs:

- for each item in the MEDIA_BOOKING_ARCHIVE table, those bookings where each element of the booking have been charged more times than the number provided with the -n parameter no longer retain the link to the patron information. For all other bookings the link to the patron information remains.

There is no output file for this job. However, an entry in the `mediajob.log` file is made indicating that the job was run and indicates the number of records updated ([Figure 10-1](#)).

```
Tue Nov 16 12:31:13 2004 Media RetainPatron...
Tue Nov 16 12:31:13 2004 1 Records Updated
Tue Nov 16 12:31:13 2004 ...COMPLETED
```

Figure 10-1. Mediajob.log File after Running Mediajob 5

Patron IDs that are retained for media booking history can be viewed in the Media Scheduling module, if the operator has the authority to view Media Booking history. See the *Voyager Media Scheduling System Administration User's Guide* and the *Voyager Media Scheduling User's Guide* for additional information.

This job can be run using a command line, see [Procedure 10-1, Running Mediajob 5 from the Command Line](#), interactively on the server, see [Procedure 10-2, Running Mediajob 5 Interactively](#), or in Webadmin see [Procedure 31-15, Running Retain Patron IDs Media Scheduling, Mediajob 5](#).



Procedure 10-1. Running Mediajob 5 from the Command Line

Use the following to run Mediajob 5 using a command line.

1. Log on to your Voyager server and navigate to the `/ml/voyager/xxxxdb/sbin` directory
2. At the `sbin>` prompt, enter `Pmediajob -j5 -n#`

where the `#` is the number of patron IDs you wish to retain (see [Figure 10-2](#)) and press enter. In this example three IDs should be retained.

```
/ml/voyager/xxxxdb/sbin $ Pmediajob -j5 -n3

Initializing...

Media RetainPatron...
number of retain patron: 3
1 Records Updated
...COMPLETED
```

Figure 10-2. Running Media Scheduling Batch Job 5 Using the Command Line

Result: The job runs retaining the specified number of patron's information. The system returns to the `sbin` prompt.

The information displayed on the screen when the job runs is also entered into the `mediajob.log` file ([Figure 10-3](#)).

```
Tue Nov 16 12:31:13 2004 Media RetainPatron...
Tue Nov 16 12:31:13 2004 1 Records Updated
Tue Nov 16 12:31:13 2004 ...COMPLETED
```

Figure 10-3. Mediajob.log File after Running Mediajob 5



Procedure 10-2. Running Mediajob 5 Interactively

Use the following to run Mediajob 5 interactively.

1. Log on to your Voyager server and navigate to the `/ml/voyager/xxxxdb/sbin` directory
2. At the `sbin>` prompt enter `Pmediajob`

Result: The screen displays the possible media batch jobs to run (see [Figure 10-4](#)).

```
/ml/voyager/xxxxdb/sbin $ Pmediajob
Initializing...
```

Media Scheduling Jobs

```
1 -- Overdue Notices
2 -- Inventory Reports
3 -- Booking Statistics
4 -- Charge Statistics

5 -- Retain Patron IDs

99 -- Quit
```

```
Process Job #?
```

Figure 10-4. Available Media Batch Jobs

3. At the `Process Job #` prompt enter 5 and press enter (see [Figure 10-5](#)).

```
/m1/voyager/xxxdb/sbin $ Pmediajob
```

```
Initializing...
```

```
Media Scheduling Jobs
```

```
1 -- Overdue Notices
2 -- Inventory Reports
3 -- Booking Statistics
4 -- Charge Statistics

5 -- Retain Patron IDs

99 -- Quit
```

```
Process Job #? 5
```

Figure 10-5. Providing the Media Scheduling Batch Job Number

4. At the `Please enter the number of patron IDs to retain:` prompt enter the number of IDs you want to retain and press enter. In this example three IDs are retained (see [Figure 10-6](#)).

```
Process Job #? 5
Please enter the number of patron IDs to retain : 3
```

Figure 10-6. Example of Retaining Three Patron IDs

Result: The job runs, retaining the specified number of patron's information ([Figure 10-7](#)). The system returns to the `sbin` prompt.

```
Media RetainPatron...  
number of retain patron: 3  
1 Records Updated  
...COMPLETED
```

Figure 10-7. On Screen Display when Running Media Scheduling Batch Job 5

The information displayed on the screen when the job runs is also entered into the `mediajob.log` file ([Figure 10-8](#)).

```
Tue Nov 16 12:31:13 2004 Media RetainPatron...  
Tue Nov 16 12:31:13 2004 1 Records Updated  
Tue Nov 16 12:31:13 2004 ...COMPLETED
```

Figure 10-8. Mediajob.log file after Running Mediajob 5

Introduction

The bulk export of MARC records program (marcexport) allows the export of many MARC records at one time. You can export records to:

- Send records to an outside source
- Share records between databases
- Delete bibliographic and MFHD records from your database
- Delete duplicate bibliographic records from your database

You can specify the records you want to export using a variety of different criteria such as record create dates or ISBN.

The program produces:

- An output (export) file of records
- An audit file (that also contains any export error messages)

Purpose of this Chapter

This chapter provides the following:

- Overview of the bulk export of MARC records program
- Parameters that govern the running of the marcexport program

- Output file specification
- Additional files created

Overview of Bulk Export

The Bulk export of MARC records program allows you to export many MARC records at one time.

Generally exports are one type of record at a time, for example, bibliographic, authority, or holdings records, but you can also export bibliographic holdings (bib-MFHD) groups.

Since all MARC records in the Voyager database are in the UTF-8 character set, records exported are output in the Unicode character set unless otherwise stated. See the -a parameter in the parameters section of this chapter.

Also, sites may run the UseMARCON API which converts records from one MARC format to another. See [UseMARCON Configuration for Use with Voyager](#) on [page A-1](#).

You can specify the records you want to export using a variety of different criteria, for example, record create dates or ISBN; see the parameters section of this chapter.

Some parameters require the creation of an input file; see the parameters section of this chapter for more information.

The program exports records into an output (export) file, and creates an audit file. The default location for these files is the `/m1/voyager/xxxdb/rpt` directory, unless otherwise specified by the operator.

The `marcexport` program runs from the `/m1/voyager/xxxdb/sbin` directory on your server. Running `Pmarcexport` executes the script which has been configured with your database name, username and password. Therefore, you will not need to enter that information. If `Pmarcexport` is entered without any parameters, the program will prompt the user for the required information.

If you want to run a server activity using a telnet session, and you do not want to perform it interactively, you can use the following command line to make certain that the activity continues, even if the telnet session is lost,

`nohup <activity with parms> &`

where `<activity with parms>` is the command followed by any required and optional parameters. This command will not guarantee that the activity will be completed if the server itself is not available.

NOTE:

This server job can be run using WebAdmin, see [Bulk Export MARC Records](#) on [page 31-17](#).

Parameters

The following parameters govern the marcexport program.

-o USMARC output filename -- not required.

The name of the file into which the exported records will be placed.

The default is `marc.exp.yyyymmdd.hhmm`, where y is the year, m is the month, d is the day, h is the hour, and m is the minute. It is placed in the `/ml/voyager/xxxdb/rpt` directory.

NOTE:

If the output file exceeds 2³⁰ (1073741824) bytes, the system appends a 0, 1, 2, etc. in sequence to differentiate the output files (i.e.

`marc.exp.yyyymmdd.hhmm.0`).

-r Record type -- required.

Specify a record type to be exported see [Table 11-1](#)

Table 11-1. Record Type and Corresponding Parameter Value

Code	Record Type	Code	Record Type
A	Authority records	H	Holdings
B	Bibliographic records	M	Main or Added Entry Authority records
E	Series Authority records	S	Subject Authority records
G	Bib-MFHD record groups		

NOTE:

For bib-MFHD record groups or holdings:

- The code for bib-MFHD groups (G) results in writing to the export file the specified bib records followed by their associated MFHDs.
- The bibliographic record ID (001 field of the bibliographic record) is exported to the 004 field of the holdings record.

- Also, the following parameters are not supported when you select bib-MFHD groups:
-l (Library), -O (OCLC only), and -X (exclude file).
- If the MFHD is linked to a serial component with uncollapsed issues set to Display in OPAC, additional fields are appended to the exported MFHD for these issues.

-n Create control number (001) from LCCN (010a) -- not required.

Must use with the System control identifier (option -c).

This parameter takes the number located in the 010a subfield and creates an 001 field if one does not exist. The 010 LCCN is left intact. If there is no 010 field, the record is logged, an error message is output, and the record is not exported.

-c <code>System control identifier -- not required.

Must use with the Create control number (option -n).

This parameter creates an 003 field using the *code* specified on the command-line. Also, it creates an 035 field using the *code* and the bib id number.

-s Modifying Agency -- not required.

Sets the 040d \$d (Modifying Agency) of each exported bibliographic and authority record. Type the code to be used immediately after the -s flag. The code may be a maximum of 15 characters.

-m Export mode -- required.

Indicates the method used to determine which records to export. Enter only one of the following one-letter codes immediately after the flag -m.

M	MARC ID input file
I	ISBN input file
R	MARC ID number range
C	Create dates only
U	Update dates only
B	Both create and update dates
K	OK to export file
S	Suppressed

-t Export target -- required.

Indicates which records will be exported. Enter the appropriate information immediately after the flag -t.

The syntax to enter date information on the command line after the -t parameter is:

-t~~yyyy-mm-dd~~:~~yyyy-mm-dd~~

If the export mode is:

M MARC ID input filename

This exports any MARC records having MARC IDs (Voyager record IDs) given in the input file.

Specify the name of the input file that contains a list of MARC ID numbers to be exported.

To create the input file containing the MARC IDs use vi or another server side text editor . Each MARC ID in the file must be on a separate line. You can name the file whatever you want. Also, it can be located wherever you want. If the file is not in the same directory as `Pmarcexport`, that is the `/sbin` directory, include the full path to the file.

I ISBN input filename

This exports any MARC records having ISBN numbers given in the input file.

Specify the name of the input file that contains a list of ISBN numbers of bibliographic records to be exported.

To create the input file containing the ISBN numbers use vi or another server side text editor. Each ISBN number in the file must be on a separate line. You can name the file whatever you want. Also, it can be located wherever you want. If the file is not in the same directory as `Pmarcexport`, that is the `/sbin` directory, include the full path to the file.

R MARC ID number range

This exports any MARC records having MARC IDs (Voyager record IDs) within the range in the input file.

Specify a range of MARC record ID numbers to be exported. Must be either: ALL, for all records, or in the format NNNNN-NNNNN (where the N is replaced by the appropriate number). This includes records matching the upper and lower extent of the range.

C Create date range

This exports any MARC records of the specified type that have a creation date within the specified range.

Specify a date range. The range must be in the format
yyyy-mm-dd hh mi ss:yyyy-mm-dd hh mi ss (hh, mi, and ss, corresponding
to hour, minute and second, are optional), or
today -n (where n=number of days).

U Update date range

This exports any MARC records of the specified type that have a update
date within the specified range.

Specify a date range. The range must be in the format
yyyy-mm-dd hh mi ss:yyyy-mm-dd hh mi ss (hh, mi, and ss, corresponding
to hour, minute and second, are optional), or
today -n (where n=number of days).

B Both create and update date range

This exports any MARC records of the specified type that have either a
creation or update date within the specified range.

Specify a date range. The range must be in the format
yyyy-mm-dd hh mi ss:yyyy-mm-dd hh mi ss (hh, mi, and ss, corresponding
to hour, minute and second, are optional) or
today -n (where n=number of days).

K OK to export file

This variable locates the records that are marked OK to export on a
specified date or within a date range. The date is either set when the
records are imported through Bulk Import, or when the **OK to export** check
box is set manually in the Cataloging module.

Specify a date range. The range must be in the format
yyyy-mm-dd hh mi ss:yyyy-mm-dd hh mi ss (hh, mi, and ss, corresponding
to hour, minute and second, are optional) or
today -n (where n=number of days).

All the records that are deemed OK to export in Cataloging module and
that contain the date(s) you enter here are the records that are exported.

S Suppressed records.

This exports any MARC records that have been suppressed between the
specified dates will be exported.

Specify a date range. The range must be in the format
yyyy-mm-dd hh mi ss:yyyy-mm-dd hh mi ss (hh, mi, and ss, corresponding
to hour, minute and second, are optional) or
today -n (where n=number of days).

-X Exclude -- not required.

Used only with export modes C, U, or B. This option is not supported with
bib-MFHD groups (-rG).

Excludes records from being exported based on certain cataloging locations. The cataloging location is location at which the record was created or most recently modified, rather than the MFHD location, from being exported. The system looks for the most recent based on update date or create date.

The `-X` parameter does not look at any cataloging location associated to any action besides 'CREATE' or 'UPDATE'. Therefore, it will not exclude based on a cataloging location associated with a 'REPLACE' action.

These cataloging locations must appear in the list contained in the user-created exclusion file.

The exclude file should include valid location codes as found in the System Administration module. Also, these location codes are case sensitive, that is if the location code is in upper case in the System Administration module, make sure that it is in upper case in the exclude file.

If the case does not match you may get errors such as `Could not find location id for location XXXX`, or `Failure to retrieve id list of records to export`.

For export mode U only the entry with the latest date will be considered.

To create the exclusion containing the cataloging locations that are not to be exported file use vi or another server side text editor. Each location code must be listed, on a separate line in the file. The location codes are found in the System Administration module. Also, location codes are case-sensitive (if the location is listed in System Administration as Main, you must enter Main, not MAIN or main).

You can name the file whatever you want. Also, it can be located wherever you want. If the file is not in the same directory as Pmarcexport, that is the `/sbin` directory, include the full path to the file.

An example might be `-X exclude.dat`, where `exclude.dat` is the name of the exclusion file that contains a list of cataloging happening location codes where records were created or updated that should not be exported.

-i Ignore Suppressed Bibliographic Records -- not required.

Indicates that suppressed records should be prevented from being exported. This option overrides the export of suppressed records (`-mS`).

-l Library -- not required.

Used only for bibliographic record type. This option is not supported with bib-MFHD groups (`-rG`).

Indicates that only records with the NUC code, as typed immediately after the flag -l, should be exported from the specified export mode (-m).

-O Only OCLC records -- not required.

Used only for bibliographic record type. This option is not supported with bib-MFHD groups (-rG).

Specifies that only OCLC records, as identified by the presence of ocm, **oc17**, **OCoLC**, or, (OCoLC) ocm in an 035 \$a or \$9, should be exported from the specified export mode (-m).

-w MFHD 852\$a updating -- not required.

When MFHDs are exported (including bib-MFHD groups), the string following the -w switch will replace the contents of any 852\$a in all exported MFHDs.

-a Assign Character mapping -- not required.

When this switch is used, -a<CODE>, the exported MARC records are converted from the default UTF-8 character encoding to the specified non-Unicode character set indicated by the CODE.

[Table 11-2](#) provides the supported character sets and their corresponding codes.

Table 11-2. Character Mapping Codes

Code	Record Type	Code	Record Type
V	Voyager Legacy	L	Latin-1
R	RLIN Legacy Encoding (this is RLIN's old character set mapping)	M	MARC21 MARC-8
O	OCLC		

For example, if you want to convert records from UTF-8 to the OCLC character set, you would use -aO on the command line. Not specifying any character mapping information (not using the -a switch) causes the program to export the records as they currently exist in UTF-8.



IMPORTANT:

When a UTF-8 character is not defined in the character set specified, the character is decomposed (assuming it is precomposed) and then a conversion is attempted, if there is no mapping for the decomposed characters the system substitutes an XML entity reference. The XML entity reference is in the format:

where 1234 is the hexadecimal representation of the Unicode (UTF-8) character.

Provides version information about the current `Pmarcexport` program

Provides online help about the `Pmarcexport` program. This flag cannot be used with any other parameters.

Prevents Voyager from prompting for any missing parameters. No hash marks display to indicate the progress of the transfer.

Excludes serials checkin details when holdings records or bibliographic-holdings record groups are exported from Voyager. The default is to include serials checkin data when exporting holdings records.

-b Copy OCLC Number from Bib to Holdings -- not required.

Use -b to copy the OCLC number (including the OCoLC string) from the bibliographic record's 035 \$a to the holdings record upon export. (This does not change what is stored in the database.)

The -b is ignored if the value for the -r is anything other than H (as in -rH).

NOTE:

In the case of bound-with bibliographic records, the first OCLC record number from each bibliographic record that is attached to the holdings record is included.

-L Export only records with holdings for this location code.

Use -L to specify a location code. With -L, you can only specify record types B (bibliographic records) or G (Bibliographic-MFHD record groups). See [Table 11-1](#) on [page 11-3](#). A date/date range must be included. See the following for a command line example:

```
Pmarcexport -rB -mU -ttoday -o/tmp/test.mrc -L cl
```

When you specify -rG, the holdings from the location specified in -L are exported and additionally any other holdings for the bibliographic record are exported.

Running the Marcexport Program

Users can enter the `Pmarcexport` command without any parameters. This prompts the program to ask for the parameters interactively from the user.

At the `/m1/voyager/xxxdb/sbin>` enter **Pmarcexport**, the system prompts for input of any required parameters.

If you do not want to enter the command interactively, an example of the `Pmarcexport` command with parameters might be entered (on one line) as follows:

```
Pmarcexport -rB -mU -t1998-10-25:1998-10-27
```

This command would export all updated bibliographic files within the specified date range of 10/25/98-10/27/98 to
`/m1/voyager/xxxdb/rpt/marc.exp.date.time.`

Output File Specification

The output file contains a file of MARC format records. The default filename is `marc.exp.yyyymmdd.hhmm`, where y is the year, m is the month, d is the day, h is the hour, and m is the minute. It is placed in the `/m1/voyager/xxxdb/rpt` directory.

NOTE:

If the output file exceeds 2³⁰ (1073741824) bytes, the system appends a 0, 1, 2, and so forth in sequence to differentiate the output files such as `marc.exp.yyyymmdd.hhmm.0`.

The `marc.exp` file that resides in `/m1/voyager/xxxdb/rpt` needs to be sent to an outside source via FTP or other file transfer method.

Additional Files

The `marcexport` program creates an audit file. It does not create an error file.

Audit File

The audit file contains information about the `marcexport`, including error information.

It is named `log.exp.yyyymmdd.hhmm`, where y is the year, m is the month, d is the day, h is the hour, and m is the minute. It is placed in the `/m1/voyager/xxxdb/rpt` directory.

For this batch job, the user cannot specify a filename for the audit file.

If the program cannot convert from UTF-8 to the specified character set the audit file contains a message such as:

```
Bib/mfhd/auth record: XXXX failed UTF8 translation
```

where XXXX is the record id.

Introduction

The Prebulk program

- checks your bibliographic records for errors before import. Prebulk can create a file of good bibliographic records that can be imported without errors. Bibliographic records containing errors are written to an error file and can be dealt with separately without disturbing the main Bulk Import process
- strips out one or many tags or subfields from bibliographic records. The resulting file can be either a file of only stripped bibliographic records or an interleaved bib/MFHD file (a file of stripped bibliographic records and MFHDs)
- creates an interleaved file of bibliographic records and MFHDs (holdings records) to be loaded into a Voyager database using Bulk Import. (Universal Catalog databases require a interleaved bib/MFHD file when loading records into the UC database.)

To do this sites must create an input file of records and a configuration file.

After running this program an output file of records is produced. It also creates a log (audit) file and an error file. The prebulk output file is UTF-8 encoded.

Purpose of this Chapter

This chapter provides

- Overview of Prebulk processing
- Input file
- Configuration file
- Parameters and an example
- Output file specification
- Additional Files

Overview of Prebulk Processing

The Prebulk program is designed to pre-process bibliographic records and create an output file that may be imported into a Voyager database.

The manner in which the file is processed can be customized. This is done by creating a configuration file that will govern the Prebulk process.

Therefore, sites must create a configuration file directing the processing, and an input file of records to be processed.

Prebulk can be used for three main purposes: to check records, strip tags from records and/or create an interleaved file of bibliographic and MFHD records. In the bib-MFHD file the newly-created holdings records are created based on the configuration file.

The Prebulk program looks to the `/m1/voyager/xxxdb/local` directory for the input file.

It creates an output file of processed records that are UTF-8 encoded. The default name is `out.prebulk.yyyymmdd.hhmm`, where `y` is the year, `m` is the month, `d` is the day, `h` is the hour, and `m` is the minute. The file is placed in the `/m1/voyager/xxxdb/rpt` directory.

The prebulk program also creates error and log files. The default filenames are, `err.prebulk.yyyymmdd.hhmm`, and `log.prebulk.yyyymmdd.hhmm`, where `y` is the year, `m` is the month, `d` is the day, `h` is the hour, and `m` is the minute. These files are placed in the `/m1/voyager/xxxdb.rpt` directory.

If you specify an output, audit or error filename only, they are placed in the `/m1/voyager/xxxdb/sbin` directory.

If you want any of the files to be read from or created in a different directory than the default, enter the full path and the filename.

The Prebulk program should be run from the `/m1/voyager/xxxdb/sbin` directory on your server.

Once the files have been generated by Prebulk, they can be imported into the database using the Bulk Import program. When importing Prebulk-generated files into the database, the results of the Bulk Import process will depend on the type of file being imported: an interleaved file or not.

See [Bulk Import, Replace, and Merge of MARC Records](#) on [page 13-1](#), and *Bulk Import Rules* in the *Voyager System Administration User's Guide* for more information regarding Bulk Import Rules.

Using Prebulk to Check the Records

Prebulk can be used to check records in your bibliographic import files and produce a new file with only the good bibliographic records.

To remove erroneous records, set **CREATEMFHD=NO** in the `[OVERRIDES]` stanza of the configuration file to create a bibliographic file. This file can then be imported with the bulk import feature to create one holdings record and one item per bibliographic record. For more information on the configuration file, see [Creating the Configuration File](#) on [page 12-5](#).

NOTE:

This requires that you do not strip the fields containing barcodes, location, or other information necessary to the creation of holdings records and items. For more information on stripping fields, see [Using Prebulk to Strip Fields](#) on [page 12-3](#).

Using Prebulk to Strip Fields

Prebulk can strip fields out of the bibliographic records before you import them.

Stripping fields is done by listing the field to be stripped after the `[Strip]` stanza in the configuration file, see [Strip Stanza](#) on [page 12-9](#) for more information.

If you use Prebulk only to strip fields from your records you can import the files into the database per normal Bulk Import procedures. Importing through Bulk Import, you can create one holdings record and one item per bibliographic record.

However, this requires that you not strip the fields containing barcodes, location, or other information necessary to the creation of holdings records and items. See the *Voyager System Administration User's Guide* for more information.

Using Prebulk to Create an Interleaved File

The Prebulk program can process bibliographic records with holdings information stored in a particular field in the bibliographic record and create an interleaved file of bibliographic and MFHD records. Also, some tags and/or subfields can be stripped.

Prebulk can create multiple (duplicate) MFHDs when creating an interleaved bib/MFHD file. However, doing this type of processing also prevents the creation of any items in the database, which Bulk Import allows you to do without performing this type of processing.

When processing records, Prebulk looks in the bibliographic record for the field listed in the [MFHDTAG] stanza in the configuration file and, for every instance of that tag contained in the bibliographic record, Prebulk generates a new holdings record. The newly-created holdings records are created based on a configuration file that you can create.

Bulk Importing of the Interleaved File Generated by Prebulk

When the records are being imported, the holdings records from the Prebulk file will load. When importing this type of file (and only when importing this type of file), Bulk Import will not automatically create any other holdings records or any item records. The bibliographic records are loaded according to the settings in the Bulk Import profile in the System Administration module. See *Bulk Import Rules* in the *Voyager System Administration User's Guide* for more information.

If the location of the MFHD(s) in the interleaved bib/MFHD file matches the location of the existing MFHD(s) in the database, Bulk Import will delete all existing MFHD(s) in the database with that matching location code. After the existing MFHD(s) are deleted, the system adds a new MFHD for each location listed in the interleaved bib/MFHD file.

If the location of the MFHD(s) in the interleaved bib/MFHD file matches the location of a MFHD in the database that has an item attached, then the MFHD in the database will *not* be deleted. The incoming MFHD will then be added to the database at that location.

NOTE:

This is the only situation where two MFHDs will be allowed at the same location.

If the interleaved bib/MFHD file contains more than one MFHD with the same location code, only one MFHD will be created in the database upon bulk import. This means that if you want to create more than one MFHD for each bib, each MFHD must have a unique location code. This is in order to prevent Universal Catalog databases from having more than one MFHD for each location.

Holdings Created from the Interleaved File

When Prebulk creates the new holdings information for a bibliographic record, a standard Voyager MFHD is created. The holding records are created by reading the records stored in the input file and processing each record based on the settings in the Prebulk configuration file. Records processed are written to the output file, `out.prebulk.yyyymmdd.hhmm`.

The leader is the standard MFHD default.

The **Record Type** in the MFHD leader will be set to serial item holdings (y) if the bibliographic record has s in offset 7 of its leader. Otherwise the **Record Type** of the MFHD leader will be set to single-part item holdings (x).

Input File

Your site creates an input file of bibliographic records to be processed using the Prebulk program. The program by default looks for this file in the `/ml/voyager/xxxdb/local` directory, unless otherwise specified.

Creating the Configuration File

You must create a configuration file that governs the Prebulk program execution. The Prebulk program reads from this file to determine how records are to be processed, that is what is to be done with each record.

The configuration file must be created using the server side text editor, `vi`. To separate pieces of data in the configuration file, Prebulk requires the tab stops that only `vi` can produce.



IMPORTANT:

Configuration files created in any other editing programs, such as Notepad, and then FTP-ed to the server will cause Prebulk to fail. Use only vi to create your configuration files.

The configuration file must be tab-delimited. A tab-delimited file means that for any entries in the file where multiple pieces of information must be specified on the same line, you must separate each piece of information with a tab. The Prebulk program looks for the tab (the tab character) as a separator, it will not recognize spaces as separators between pieces of information.

For example, your [LOCATIONS] stanza might look like [Figure 12-1](#)

[LOCATIONS]	
MAIN ->	MAORI-U ->099,090,050
REFERENCE ->MAORI-R ->	949,099(1),090,050

Figure 12-1. Sample LOCATIONS Stanza

There are three pieces of information in the first line: MAIN, MAORI-U, and 099,090,050. Each of these is separated by a tab. (In this document, we will use -> to indicate a tab.) Your file should use the normal tab character (the tab key). The tab character is counted (treated as a regular character) when determining the length of strings.

For example, the last line in the previous example would appear in the file as:

REFERENCE [tab]MAORI-R [tab] 949,099 (1) , 090,050

NOTE:

When the # character appears in the first position on a line, that tells the Prebulk program that the line should be ignored. For example, when you want to enter a comment onto a line, preceding the comment with a # will cause Prebulk to not try and process that line as valid information.

Default Location of the Configuration File

The default location for the configuration file is the /ml/voyager/xxxdb/local directory.

Name of the Configuration File

The name of the configuration file can be whatever you want. Prebulk has no default name for the configuration file. Prebulk.cfg is a suggested name, but not required or necessary.

Stanza Types

There are several stanzas that can be included in the configuration file.

- Overrides
- Strip
- MFHDtag
- Calltypes
- Locations
- Mapping
- 008

Using the Default Call Number

To force the Prebulk program to use the default call number for a particular record you must provide the `[LOCATIONS]` stanza with a reference to a field that contains no default call number.

To do this, you must specify the call number location in the third field of the `[LOCATIONS]` stanza, see [LOCATIONS Stanza](#) on [page 12-12](#). The location must be a field and override indicator combination, and must be a call number location that does not exist, that is, one that does not contain a call number. The subfield indicator is required in order to perform this action properly.

Using a Call Number Stored in the Record

Use the following to force Prebulk to use a call number specified in the record. You must set this in the configuration file:

In the `[MFHDTAG]` stanza you must list the field and subfield where the call number is stored. The `[MFHDTAG]` stanza identifies the MARC tag which contains the data that is to be used to create the MFHD. See [MFHDTAG Stanza](#) on [page 12-11](#) for more information.

You must have the field and indicator listed in the third column of the `[LOCATIONS]` stanza. The location must be a field and override indicator combination. For example, if the call number was stored in the 949 field, the field and indicator would read 949(1). See [LOCATIONS Stanza](#) on [page 12-12](#) for more information.

The `[CALLTYPES]` stanza must be empty. This is because Prebulk will automatically look in the subfield 'a' of the fields listed in the `[CALLTYPES]` stanza. See [CALLTYPES Stanza](#) on [page 12-12](#) for more information.

In the [MAPPING] stanza, the field and subfield of the call number must be mapped to the appropriate field and subfield in the new MFHD. See [MAPPING Stanza](#) on [page 12-15](#) for more information.

Overrides Stanza

[Figure 12-2](#) provides an example of the [OVERRIDES] stanza.

[OVERRIDES]
CREATEMFHD=yes
DEFAULTCALLNO=PZ3.L129
DEFAULTCALLIND=0
USE001FOR014=no
uses 003 for location only, in addition
another tag may be used for other mapping
(or not) [MAPPING] stanza may be blank if
003 is used
USE003FORLOC=no

Figure 12-2. OVERRIDES Stanza Example

CREATEMFHD

This variable indicates whether or not MFHDs (Holdings records) are to be created. Entering YES indicates that holdings records will be created. If YES is entered, the resulting file will be in interleaved bib/MFHD format. Entering NO indicates no holdings records are to be created and that the program is being run only to strip tags or check records. If NO is entered, a standard bibliographic file will be created. YES and NO are not case sensitive.

DEFAULTCALLNO

If no call number is found in the [MAPPING] stanza (see [MAPPING Stanza](#) on [page 12-15](#) for more information), this call number is placed in the 852 \$h of the new MFHD. If no call number is found and this field is blank, no call number is created. The value can be blank or any string. The Default Call Number must be blank if the tag specified in the [MFHD TAG] stanza is listed for any of the locations in the [Locations] stanza.

DEFAULTCALLIND

The value for indicator one of the 852 if the **DEFAULTCALLNO** is used. This should be a number from 0 through 8. Pipes and blanks are not acceptable.

USE001FOR014

Entering Yes indicates that an 014 will be created with a \$a containing the value of the 001 tag. The indicators for the 014 are 1 and blank. Entering No indicates the 001 will not be used for the 014. YES and NO are not case sensitive.

USE003FORLOC

Entering Yes indicates that the incoming 003 tag will be the location used in the 852 \$b. This is the default used if no location is specified in the field listed in the [MFHDTAG] stanza. Entering No means that it will not. YES and NO are not case sensitive.

Strip Stanza

The [STRIP] stanza is a list of tags and/or subfields that you want to remove from the incoming bibliographic record (see [Figure 12-3](#)).

[STRIP]
Consists of a tab-delimited list of tags
and/or subfields
to be stripped from the incoming bib record
No subfields listed means the entire tag is
stripped. 001,008,245 tags may not be
completely stripped.
650-> 04-> w
700-> X5-> asd

Figure 12-3. The STRIP Stanza

- Listing a tag alone will strip the tag entirely. If tag and indicators are listed, then only tags with those indicators are removed. If tag, indicators, and subfields are listed, then all occurrences of the subfield within the matching tag and indicators are removed.
- X denotes a wild card and any indicator value will match.

- | indicates that the indicator is blank.
- If no data is left in the tag after stripping subfields, then the tag is removed.
- The 245 tag may not be removed entirely but subfields can be stripped.
- The 003 tag is not automatically stripped; it must be included in the [STRIP] stanza in order for it to be stripped.
- If no data is left in the 245 after stripping subfields, a log message exception is generated and the entire 245 is maintained.
- If you put the tag specified in the [MFHDTAG] stanza in the list to be stripped, Prebulk will build any holdings information you specify in the configuration file before stripping the tag.
- 001 and 008 tags may not be stripped.

Non-Standard 035 Field Elimination

With the migration of bibliographic records from some library management systems into Voyager a non-standard 035a subfield may be left in the bibliographic record. The MARC standard dictates that the system control number field be prefixed by the organization of the original system control number. Therefore, a standard 035a would contain an alphabetic code for an organization and its system control number. For example, \$035a Ex Libris 12345. A non-standard 035a contains either only alphabetical data, or only numeric data.

With prebulk processing, administrators can specify that non-standard 035 fields be stripped from the record. This eliminates bulk import de-duplication problems.

Users can choose to strip out those 035 fields where the subfield a contains only letters or only numbers.

System Administrators use the prebulk configuration file's [STRIP] stanza to indicate what they want to strip.

NOTE:

As with any field you want to strip, indicators or wildcards must be included.

Examples of the entry in the [STRIP] stanzas are as follows:

To strip the subfield a of the 035 fields enter this line in the [STRIP] stanza:

```
035 XX a
```

To strip non-standard 035 fields, enter one of these lines in the [STRIP] stanza:

```
035 XX a* where XX is the indicator wildcard, or
```

```
035 | | a* where | means the indicator is blank.
```

An *(asterisk) after the 'a' is the function to strip the non-standard fields.

When stripping non-standard fields the steps of this process are:

1. The system administrator runs the prebulk program.
2. The program reads in the configuration file.
3. If the program identifies that a record has a non-standard 035a, that is only numbers or only letters (punctuation and spaces in the 035a are ignored), it strips that entire field from the record.

MFHDTAG Stanza

The [MFHDTAG] Stanza identifies the MARC tag which contains the data that is to be used to create the MFHD, such as the holdings location, call number, barcode, operator and item type (see [Figure 12-4](#)).

[MFHDTAG]
nothing below here is required if only using
program to strip tags
'XXX' means no tag, use default location
949

Figure 12-4. The MFHDTAG Stanza

- Values must be from 010 through 999 or XXX, where XXX means that no tag contains holdings information and the default location should be used in the 852 \$b.
- Only one tag may be used as the MFHD tag.
- Each occurrence of the tag in a bibliographic record will create a separate MFHD record.

CALLTYPES Stanza

The [CALLTYPES] stanza specifies what indicator is to be used when creating an 852 in the MFHD. The indicator may be different for each call number type listed. When the appropriate call number is determined by the [LOCATIONS] stanza its field number will be checked against this list and the call number will be placed in the 852 in the new MFHD with the subfield specified in this list (see [Figure 12-5](#)).

[CALLTYPES]
assign these call number types to tags
unless otherwise noted
099 -> 8
050 -> 0
086 -> 3

Figure 12-5. The CALLTYPES Stanza

- There must be at least one location listed.
- The first field can be 010-099.
- The second field can be 0-8 or c for CODOC numbers.
- The value in the second field is placed in the first indicator of the 852 field in the MFHD.
- Each row of the stanza have a tab separating the two fields.

NOTE:

Any fields listed in the [LOCATIONS] stanza that do not have the indicator specified immediately after must be listed in the [CALLTYPES] stanza. If fields appear without indicators and are not listed in the [CALLTYPES] stanza, the Prebulk program will fail.

LOCATIONS Stanza

The [LOCATIONS] stanza determines what location is to be used in the MFHD (see [Figure 12-6](#)).

[LOCATIONS]
tab delimited by location. the first line is
the default setting. no match means no call
number and a blank indicator will be used
unless override option.
T means use 245 \$a in \$1 with 7 indicator
#
#Incoming Loc New Loc Tag(Override)Hierarchy
MAIN -> MAORI-U -> 099,090,050
REFERENCE -> MAORI-R -> 949(1),099(1),090,050
PERIODICAL -> MAORI-P -> T
entries not listed default to the first
entry with no normalization

Figure 12-6. The LOCATIONS Stanza

- You must specify a list of locations that may be found in the bibliographic record. For each location, must specify a name to be given (that is, whether you want to rename the locations or not), and create a list of tags that specifies which tags are preferred over others. The name in the second field must be a location code (of not more than ten characters) as assigned in the System Administration module.

NOTE:

Entering an invalid location code will not cause problems within the Prebulk program, but it will cause problems when attempting to import the records with the Bulk import program. You must be sure that the location codes you enter are identical to the location codes listed in the System Administration module. See *Locations* in the *Voyager System Administration User's Guide* for more information.

If a record does not match any of the locations, the first row will be used as a default. Each row of the [LOCATIONS] stanza contains three different fields.

- The first field is an incoming location found in the field referenced in the [MFHDTAG] stanza. The location entered in this field must exactly match the location listed for the new MFHD (the location listed in the field in the [MFHDTAG] stanza). See [MFHDTAG Stanza](#) on [page 12-11](#) for more information.

- The second field is the location code to be placed in the 852 \$b. It may be the same as the incoming location name or a variation.
- The third field is the tag hierarchy to be used for that location to create a call number in the 852 \$h and \$i. For that location, you can specify which tags are to be used above others. For tags from 100 through 999, you must specify an indicator after the tag (for example, 949(2)). For tags from 010 through 099, an indicator is optional, but if it is not included, the tag must be listed in the [CALLTYPES] stanza as well.
- Each row of the stanza must have a tab separating each of the three fields.

For example, if the Archives department (which has a location code of Archives, and to be renamed Archv) puts the call number in the 099 and the rest of your locations (such as the main cataloging area, for which the location code is Cat, and not to be renamed) put the call number in the 050, you would want to create two locations in your list. The Archv location should have the 099 tag listed first so that if there is both an 050 and an 099 in the record, the 099 will be chosen over the 050. To handle all of the other locations, you could create a separate row for each one, but you could also just create one row for one of the locations and place it first so that if a record matches neither record, it will choose the default row (which is the first row).

Your [LOCATIONS] stanza might look like [Figure 12-7](#).

[LOCATIONS]
Cat -> Cat -> 050,099,090
Archives -> Archv -> 099,050,090

Figure 12-7. Sample LOCATIONS Stanza

NOTE:

The first indicator of the new 852 will be determined by the values in the [CALLTYPES] stanza. You can change this for a particular tag by entering a value in parentheses following the tag in the hierarchy.

If a T is found in the call number hierarchy, then no subfield h and i will be created. Instead a \$I (eI) will be created and the first indicator will be 7. The first field of the call number tag will be used as the \$h and all remaining subfields are subfield i.

The first field can be any string. The second field can be any string up to ten characters. The third field can be a number from 010 through 999 or T. Values in parentheses can be 0-8. No normalization of strings is done other than stripping preceding and trailing blanks.

If incoming data does not match any field in the `[LOCATIONS]` stanza, the first entry is used as a default and an exception with 001 value is placed in the log file.

This means that if you want a location name to remain what it was originally, it must be listed in the `[LOCATIONS]` stanza and must have the same location specified in the first and second fields.

MAPPING Stanza

The `[MAPPING]` stanza indicates the subfields in the tag (specified in the `[MFHDTAG]` stanza) which contain specific data, and where that data goes in the MFHD to be created. All of the subfields indicated in the list must be in the tag listed in the `[MFHDTAG]` stanza (see [Figure 12-8](#)).

[MAPPING]
Incoming subfield and outgoing
tag/subfield. Only one of each tag
will be created if using XXX, then no
mapping only 852 b.
852 b is required. Multiple subfields are in
order listed except 852 k is before \$h and
852 m is after \$h and \$i.
Multiple subfields are concatenated with
blanks.
l -> 852b
vnk -> 866a
\$(If an 866 a is created, a \$80 is also
created
p -> 852k
s -> 852m
c -> 852t
h -> 852h
\$(Illegal if no match in [LOCATIONS])
i -> 852i
\$(Also illegal if no match in [LOCATIONS])

Figure 12-8. The MAPPING Stanza

- The first field in each row is the subfield of the tag specified in the [MFHDTAG] stanza where a particular piece of information is stored. The second field is the field and subfield where that piece of information is to go in the new MFHD.
- One 852 and one 866 is created. The indicators of the 852 are the call number indicator and blank. All other tags have blank indicators. If an 866 tag is created, a preceding \$8 with the value of 0 is also created. Only one tag of each type will be created.

NOTE:

The value mapped to the 852 b will be used as the incoming location for the [LOCATIONS] stanza.

If the tag specified in the [MFHDTAG] stanza is not listed in the [LOCATIONS] stanza, the tags will be in numerical order. Subfields will be in the order found in the [MAPPING] stanza with the exception of the call number sequence in the 852 which will always have the following subfields in the order k, h, i, and then m.

If the tag specified in the [MFHDTAG] stanza is listed in the [LOCATIONS] stanza, the tags will be placed in the new MFHD in exactly the order that they appear in the [MAPPING] stanza. The tags will not be sorted.

If multiple occurrences of an incoming subfield appear in one field of the first row, the information will be concatenated with a single space between incoming subfields.

008 Stanza

The [008] stanza specifies how to create the 008 in the new MFHD (see [Figure 12-9](#)).

[008]
#use the 008 here based on the value of
#offsets 6 and 7 of the leader
top is default, length must be correct
#(YYMMDD means date of run), spaces count
000-> 6-> am-> YMMDDetc
000-> 6-> as-> YMMDDetc
000-> 6-> gm-> 9812160p 8 4001aueng00000000
#overrides the information in the leader
007-> 0-> p-> YYMMDDetc

Figure 12-9. The 008 Stanza

- You can specify a location in a field in the bibliographic record and if it matches a string that you specify, you can designate a string that is to become the 008.
- Each row contains four fields. The first field is the fixed length field to be examined (000 indicates the leader). The second field is the offset to examine. The third field is the value to match. The final field is the 008 to be created if the value is matched.

NOTE:

The first entry will be the default if no value is matched. If multiple values are matched, the final matching entry in the map overrides all other matches. Specifying YYMMDD in the first six entries in the fourth field indicate that the program run date is to be used to create this field.

NOTE:

The first field may be 000-009. The second field can be any 3 digits. The third field can be any string. The final field (the 008) must be 32 characters in length including alphanumeric characters, tabs, pipes, and spaces.

[Figure 12-10](#) provides a sample configuration file.

```
[OVERRIDES]
CREATEMFHD=YES
DEFAULTCALLNO=PS123.A2 1999
DEFAULTCALLIND=8
USE001FOR014=YES
USE003FORLOC=NO

[STRIP]
015-> ||-> a
020-> ||-> a
043-> ||-> a
082-> ||-> a
260-> ||-> abc
300-> ||-> a
400-> XX-> abt
440-> XX-> abnv
490-> 0|-> a
500-> XX-> a
504-> ||-> a
600-> 1|-> a
651-> |0-> axy

[MFHDTAG]
949

[CALLTYPES]
090-> 0
#050-> 0
099-> 0

[LOCATIONS]
#main->-> main-> 050,090(1),099
#serials->-> serials-> T
#Reference->-> Reference-> 050,090(1),099
main->->main-> 092(1)

[MAPPING]
b-> 852b
i-> 852i
h-> 852h

[008]
#->->->000000000111111111222222222333
#->->->12345678901234567890123456789012
000-> 6-> am-> YYMMDD
```

Figure 12-10. Sample Configuration File

Parameters

The following are the valid parameters for Prebulk.

-i Input file -- required.

The name of the file where the bibliographic records to be processed are to be found. The default location for the input file is the `/m1/voyager/xxxdb/local` directory. Use the entire path to the file in the command line.

-o Output file -- not required.

The name of the file where the bibliographic and holdings records are to be written after being processed. The output file created is called `out.prebulk.YYYYMMDD.HHMM`, where y is the year, m is the month, d is the day, h is the hour, and m is the minute. It is placed in the `/m1/voyager/xxxdb/rpt` directory. If you specify a filename, but no path, the file is placed in the `/m1/voyager/xxxdb/sbin` directory. If the name is that of an existing file, Prebulk will not run unless you specify `-f` to overwrite the existing file.

NOTE:

The system displays an error message if you specify an output file that has the same name as the input file.

-c Configuration file -- required.

The name of the file where the Prebulk configuration information is stored. There is no default name for the file. Specify the full path and file name for the configuration file.

-e Error file -- not required.

The name of the file where error information is to be written. Prebulk will write any records that cannot be processed to this file. The error file created is called `err.prebulk.YYYYMMDD.HHMM`, where y is the year, m is the month, d is the day, h is the hour, and m is the minute. It is placed in the `/m1/voyager/xxxdb/rpt` directory. If you specify a filename, but no path, the file is placed in the `/m1/voyager/xxxdb/sbin` directory. If the name is that of an existing file, Prebulk will not run unless you specify `-f` to overwrite the existing file.

-p Parameter file -- not required.

The name of the file that contains a list of parameters commonly entered when running Prebulk. This allows you to create a file containing the parameters that you enter each time onto a single line. A sample parameter file might be:

```
-i Input.bib
-o Output.bib
-c prebulk.cfg
```

Therefore, instead of entering parameters repeatedly, you simply need to enter **-p** and the filename.

-a Append tag for log file -- not required.

Adds the log information to the end of the previously generated log file.

-f Force overwrite of output files switch -- not required.

Tells Prebulk to overwrite any existing output file of the name specified by the **-o** switch.

-l Prebulk log file -- not required.

The log file for processing of the record. Contains the number of bibliographic records processed (not including erroneous records) and the number of holdings records created. The log file created is called `log.prebulk.YYYYMMDD.HHMM`, where y is the year, m is the month, d is the day, h is the hour, and m is the minute. It is placed in the `/m1/voyager/xxxdb/rpt` directory. If you specify a filename, but no path, the file is placed in the `/m1/voyager/xxxdb/sbin` directory. If the name is that of an existing file, Prebulk will not run unless you specify **-a** to append the new information to the existing file.

Running the Prebulk Program

Before you can run the prebulk program you must first create the input file and configuration file.

Users can enter the `Pprebulk` command without any parameters. This prompts the program to ask for the parameters interactively from the user.

At the `/m1/voyager/xxxdb/sbin>` enter **Pprebulk** and the required parameters after the command. All of the parameters must be on the same line.

For example:

```
Pprebulk -i /m1/voyager/xxxdb/local/input.bib -o
output.bib -c prebulk.cfg
```


NOTE:

Use the full path to the input file with the `-i` parameter.

This command instructs the system to use the input file named `input.bib`, create an output file named `output.bib` and use the configuration file named `prebulk.cfg` to direct the processing of the records.

NOTE:

Running more than one prebulk import session at the same time should be avoided because simultaneous updating can cause unintended results. Ideally, you should run the prebulk program at low usage times such as when staff is not making updates/deletes to records in Cataloging.

Output File Specification

The output file is the file where the bibliographic and holdings records are to be written after being processed. It is in MARC format and it is UTF-8 encoded.

The default filename is `out.prebulk.YYYYMMDD.HHMM`, where `y` is the year, `m` is the month, `d` is the day, `h` is the hour and `m` is the minute, unless a name is specified with the `-o` parameter. The file is placed in the `/m1/voyager/xxxdb/rpt` directory. If you specify a filename, but no path, the file is placed in the `/m1/voyager/xxxdb/sbin` directory.

If the name is that of an existing file, Prebulk will not run unless you specify `-f` to overwrite the existing file.

Additional Files

In addition to the output file of records, the Prebulk program creates an error and a log file.

Log (Audit) File

The log file contains the number of bibliographic records processed (not including erroneous records) and the number of holdings records created.

The default filename is `log.prebulk.YYYYMMDD.HHMM`, where `y` is the year, `m` is the month, `d` is the day, `h` is the hour and `m` is the minute. It is placed in the `/m1/voyager/xxxdb/rpt` directory, unless specified by the `-l` parameter.

If the name is that of an existing file, Prebulk will not run unless you specify -a to append the new information to the existing file.

Error File

The error file is where Prebulk will write any records that cannot be processed.

The default filename is `err.prebulk.YYYYMMDD.HHMM`, where y is the year, m is the month, d is the day, h is the hour and m is the minute. It is placed in the `/ml/voyager/xxxdb/rpt` directory, unless specified by the -e parameter.

If the name is that of an existing file, Prebulk will not run.

If there were no errors, a file is created, however it is empty.

Introduction

The Bulk Import, Replace, and Merge (or Bulk Import) program:

- Imports, replaces, or merges many authority or bibliographic records (for example, USMARC, RLIN, and MARCIVE records) at one time
- Allows for the automatic creation of MARC holdings records and Voyager item records based on an import profile established in the System Administration module
- Can also load holdings records (MFHDs), item records, and purchase orders (POs)
- Indexes the records completely

For example, you might want to load a file of bibliographic records that were supplied by a bibliographic utility.

To do this, sites must have a file of records (in MARC format) to import. Sites may run the UseMARCON API which converts records from one MARC format to another. See [UseMARCON Configuration for Use with Voyager](#) on [page A-1](#).

Also sites must create a bulk import rule and a duplicate detection profile in the System Administration module to govern the import process.

After running the program, the system creates an audit and an error log. Other files are created also depending on the Add or Replace duplicate handling option selected in the duplicate detection profile in the System Administration module.

Purpose of this Chapter

This chapter discusses:

- Overview of the Bulk Import program
- Input file specification
- Parameters that govern the program and running the program
- Creation of holdings and item records
- Additional files

Overview of the Bulk Import, Replace, Merge MARC Records Program

Bulk Import can be used to load the following types of information

- Bibliographic record load only
- MFHDs only
- MFHDs and POs
- MFHDs and Items
- MFHDs, Items and PO

Prior to running the Bulk Import program, the user must set up Bulk Import Rules and Duplicate detection profiles in the System Administration module (see *Bulk Import Rules* and *Bibliographic Duplicate Detection* in the *Voyager System Administration User's Guide* for more information), and have an input file of records to load.

The program looks in the `/m1/voyager/xxxdb/sbin` directory for the input file if a complete path is not specified.

This program generates audit and error files. The default location for these files is the `/m1/voyager/xxxdb/rpt` directory. The default filename for the log file is `log.imp.YYYYMMDD.HHMM`, where y is year, m is month, d is day, h is hour, and m is minute. The default filename for the error file is `err.imp.YYYYMMDD.HHMM`, where y is year, m is month, d is day, h is hour, and m is minute.

The additional files created are based on the type of bulk import being done. Refer to [Additional Files](#) on [page 13-9](#) for an description of these additional files and their names. These files are placed in the `/m1/voyager/xxxdb/rpt` directory.

The bulk import program runs from the `/ml/voyager/xxxdb/sbin` directory on your server. Running the `Pbulkimport` script sets the environment variables, tells the server the pieces it needs, executes the Bulk Import program, and writes to a log and an error file, as well as creating files of records based on the type of import.

For optimum importing performance, import 10,000 records (or less) at one time. If your record file is larger than 10,000 records, it should be broken into smaller sets of records (using the `-b` and `-e` parameters) and then imported one after the other.

NOTE:

When a new record (record status in the bibliographic leader is set to `n`) in the database is replaced or merged with another record, the record status of the record in the database is automatically changed to corrected (`c` in the leader).

If you want to run a server activity using a telnet session, and you do not want to perform it interactively, you can use the following command line to make certain that the activity continues, even if the telnet session is lost:

```
nohup <activity with parms> &
```

where `<activity with parms>` is the command followed by any required and optional parameters. This command does not guarantee that the activity will be completed if the server itself is not available.

NOTE:

This server job can be run using WebAdmin, see [Bulk Import MARC Records on page 31-12](#).

Generating Keyword Index Files

By default, when you run Bulk Import, keyword index files are not regenerated. To indicate that you want keyword index files regenerated during Bulk Import, you need to use the `-KADDKEY` parameter (see [Parameters on page 13-4](#)).

When importing bibliographic records with `-KADDKEY` specified, Bulk Import simultaneously updates the Oracle tables as well as, the keyword index files. If you import a large number of records, performance may be negatively affected.

When Bulk Import is run (without the `-KADDKEY` parameter) bibliographic records are added to the database but no keyword index files are updated. The result of running Bulk Import without the `-KADDKEY` parameter is that Voyager Keyword searching will not find these newly-imported records until a Keyword Index Regen is run.



TIP:

To run a keyword regen, use the Voyager UTIL Menu or, if you prefer, create a case for Support in the Salesforce Support Portal to schedule Voyager Support to run the regen for you at no charge.

Patrons and staff can run keyword searches, without error, while Bulk Import is running.

Parameters

The following parameters govern the bulk import program.

-f Filename -- required.

The filename containing the records you are importing. The default location of the file is the `/ml/voyager/xxxdb/sbin` directory. If the file is in a different directory, use the complete path.

-i Import code -- required.

The Bulk Import Rule code. This is the code specified in the **Code** field, located on the **Rule Name** tab in the **Cataloging - Bulk Import Rules** section in the System Administration module. It instructs the system to use the Bulk Import Rule associated with the code specified and follow all of the rules defined therein.

This is also where you specify whether all loaded records should be suppressed from the OPAC and whether MARC holdings (MFHDs) and Voyager item records should automatically be created. MFHDs may be in different locations.

If the profile you select performs duplicate detection, note that if any single index listed in the duplication hierarchy in System Administration matches with more than 1000 records in the database, all duplicate detection will stop.

Only the first 100 records above the matching threshold will be returned to the client. There is no limit to the number of indexes that can be put in the hierarchy to check. However, this will hurt the accuracy of the matching being performed.

Also if the profile that you select has no indexes selected, the records are added unconditionally to the database. See *Bulk Import Rules* in the *Voyager System Administration User's Guide* for more information.

-o Operator name -- not required.

The name of the operator importing the records. This information is recorded and used in Voyager to identify who last modified the record.

-l Location Code -- not required.

The code for the cataloging happening location (as defined in System Administration module) that will be used in Voyager to identify the location from which the record was last modified.

-b Begin record -- not required.

The first record in the file to be imported. For example, specifying the number 5 would instruct the program to begin importing from the fifth record in the file. This parameter is used with the -e parameter when importing fewer records than the entire file.

-e End record -- not required.

The last record in the file to be imported. For example, specifying the number 10 would instruct the program to stop importing after the tenth imported record. This parameter is used with the -b parameter when importing fewer records than the entire file.

-m Load MFHDs -- not required. Must have interleaved file.

Load MFHDs with bibliographic records from a single interleaved bib-MFHD file.

-a MFHD location code -- not required. Must have interleaved file.

After the -a parameter enter a location code, for example **-aCIRC**. Then for all incoming MFHDs, the location code listed will be used in the MFHDs 852 field, subfield b. The location codes are defined in the Locations section of System Administration. The location code must match the code in System Administration exactly (it is case sensitive). Therefore, if you use **-aCIRC**, as an example, the MFHDs will have the location CIRC.

If used in conjunction with the -m parameter, it will act as a match point, and add a new MFHD with the location specified.

-r Delete MFHDs -- not required.

Use this variable to delete specified MFHDs from your database.

The import file must be an interleaved file of bibliographic records and MFHDs. You cannot import a file of just MFHD records in an attempt to delete matching MFHDs from the database.

First, the incoming bibliographic records are matched with the bibliographic records on the database. For all matching bibliographic records, holdings records in the database with the same location code as the holdings records in the incoming file will be deleted. (The variable does not import records.)

NOTE:

You cannot delete records that are linked to an item record or purchase order.

-x Delete bibliographic records -- not required.

This option is used only with Delete MFHDs (`-r`). To delete bibliographic records, both `-x` and `-r` should be entered as part of the same `Pbulkimport` command.

Use this variable to delete bibliographic records in your database that match the records in the data file.

The import file must be an interleaved file of bibliographic records and MFHDs.

This option does not import records. Records are only deleted from the database. The records in the file are matched with the records in the database.

This means that the location in the record in the data file must match the owning library of the bibliographic record in the database. If any of the matching bibliographic records in the database do not have any MFHDs attached to them (after having been deleted using the `-r` command), the bibliographic record in the database will be deleted.

If the import file contains only bibliographic records, it will delete those matching bibliographic records in the database if there is no linked MFHD and if the record is not linked to a purchase order.

This option is generally used by Universal Catalog databases to allow the local libraries to create files containing any records that they have suppressed or deleted from their database so that they can be removed from the Universal Catalog database as well. For more information see the *Voyager Universal Catalog User's Guide*.

-k OK to export -- not required.

Use this variable to select the OK to export check box on the **System** tab of bibliographic, authority, and holdings records on view in the Cataloging module. The date on which each MARC record was last marked OK to export displays in the Cataloging module, **History** tab of the record on view in the Cataloging module.

-K ADDKEY Generate keyword index -- not required.

This parameter and argument can be used when importing bibliographic records to generate keyword indexes.

-h Help -- not required.

This parameter provides online help about the `Pbulkimport` function. This flag cannot be used with any other parameters.

-p Add Copy Number For New Item Records -- not required.

This parameter specifies a single copy number to all newly created item records when it is used by itself. A number from 0 through 99999 (up to five places) may be specified consistent with the Cataloging client.

When there are multiple copies/items being created, the number specified with `-p` represents the number to start with. For example, if there are four copies/items being created and `-p` specifies 64, the copies will be numbered 64, 65, 66, and 67.

NOTE:

If `-p` is not used, the copy number defaults to zero when one copy/item is created. When multiple copies/items are created and `-p` is not used, the copy number defaults to starting with the number one as in copies numbered 1, 2, and 3 for three new copies.

NOTE:

When the `-p` command line option is provided, the Cataloging bulk import profile configuration settings (see [Figure 13-1](#)) in Voyager System Administration are ignored.

Cataloging Bulk Import Copy Numbering Settings

Use Copy Number From:

Field: Subfield:

Ind 1: Ind 2:

If Copy Number Not Present in Incoming Record or Not Specified Above

☐ Single Copy = 0; Multiple Copy = 1, 2, 3...

☐ Sequentially number copies starting with

☒ Number all copies

OK Cancel

Figure 13-1. Cataloging Bulk Import Profile Configuration Example

-C Do NOT create 035 -- not required.

Use the -C parameter to specify that the 035 field is not to be created when replacing or creating bibliographic records during bulk import. The bulk import default behavior is to create the 035 from the 001 of the imported bibliographic record.

NOTE:

-C is intended to prevent duplication. If you are importing a record with -C and the incoming bibliographic record does not contain a 035 field, one will be created.

-M Allow multiple bulk import processes -- not required.

Use the -M parameter to specify your intention to run multiple concurrent bulkimport sessions. This can be done safely if there is no data overlap between runs.



IMPORTANT:

Using -M assumes that you have confirmed that there is no data overlap that might be an issue for the multiple concurrent bulkimport sessions you are running with -M.

When `-M` is not specified as a parameter for `Pbulkimport`, the `Pbulkimport` script checks for a running bulkimport process for the current database. If it finds a running bulkimport process, it terminates and displays a message indicating that there is a bulkimport process that is already running.

Running Bulk Import

An example of a `Pbulkimport` command might be entered (on one line) as follows.

```
Pbulkimport -fmarc.rec -odeb -iADDCOND -b1 -e1200  
-lMain
```

This command instructs the system to import the records contained in the file named `marc.rec` numbered from 1 to 1200, following the ADDCOND bulk import rule as specified in the System Administration module, and give the operator name of deb and cataloging happening location of Main.

Also, if MFHDs and items are created the Call Number hierarchy set up in System Administration for the ADDCOND import/replace profile and the Item Type hierarchy will be used to create the appropriate Voyager item records for all MFHDs created.

MFHD matching is coded in the software to match on 852 \$b (the MFHD location) for associated bibliographic records. If a match is found, the existing MFHD is deleted and replaced by the incoming MFHD record.

NOTE:

Running more than one bulk import session at the same time should be avoided because simultaneous updating can cause unintended results. Ideally, you should run the bulk import program at low usage times such as when staff is not making updates/deletes to records in Cataloging.

Input File Specification

The input file is a file of Bibliographic or Authority records, or an interleaved file of Bibliographic/Holdings records in MARC format.

Additional Files

The bulk import program creates an error and log file, as well as other files depending on the Add or Replace duplicate handling option selected in the duplicate detection profile in the System Administration module.

Log File

After the `Pbulkimport` job is complete, an activity summary is written to a log file. This file contains a summary, which is broken down between Bibliographic and Authority records in the first group and Holdings records in the second group. The log file includes the number of:

- Records processed
- Records added (to the database)
- Records discarded (to the `discard.imp` file)
- Records rejected (written to the `reject.imp` file, does not apply to holdings records)
- Records that caused errors (written to the `err.imp` file)
- Records replaced or merged (written to the `replace.imp` file)
- Records deleted (see [Delete](#) on [page 13-10](#))

This log file also includes information about any records that fail conversion in the bulk import process. See [Messages in Log and Error Files](#) on [page 13-12](#).

The default filename is formatted as `log.imp.PID.date.time`, and it is placed in the `/ml/voyager.xxxdb/rpt` directory.

Delete, Discard, Replace, and Reject Files

The bulk import program creates delete, discard, replace, and reject files depending on the Add or Replace option selected in the Import/Replace profile in the System Administration module.

NOTE:

The files are named based on the time and date that `Pbulkimport` was run. Since the files are named with the minute and not the seconds, if bulk import is run more than once per minute, the files will be overwritten with the information from the latest run.

Delete

An incoming record will be put in the delete file if it has a delete symbol in the record.

The following are the delete log filenames where `cc` is the century, `yy` is the year, `mm` (first occurrence) is the month, `dd` is the day, `hh` is the hour, and `mm` (second occurrence) is the minute:

- `delete.bib.imp.ccyymmdd.hhmm`

This file contains the bibliographic records that are marked as cancelled.

- `delete.auth.imp.ccyymmdd.hhmm`

This file contains authority records marked for deletion as indicated by D, S, or X in the leader (position 5).

- `deleted.mfhd.marc`

This file contains the holdings (MFHD) records that are deleted when using bulk import.

These files are placed in the `/m1/voyager/xxxdb/rpt` directory.

Discard

An incoming record will be put in this file if it could not be added, merged, or replaced because there are multiple records above the replace threshold or the warning threshold.

The filename is `discard.imp.ccyymmdd.hhmm`, where c is the century, y is the year, m is the month, d is the day, h is the hour, and m is the minute, and it is placed in the `/m1/voyager/xxxdb/rpt` directory.

Replace

An existing database record will be put in this file if it is replaced by or merged with an incoming record.

The filename is `replace.imp.ccyymmdd.hhmm`, where c is the century, y is the year, m is the month, d is the day, h is the hour, and m is the minute, and it is placed in the `/m1/voyager/xxxdb/rpt` directory.

Reject

An incoming record will be put in this file if it could not be added, merged, or replaced because it has a lower quality rating than the existing record.

The filename is `reject.imp.ccyymmdd.hhmm`, where c is the century, y is the year, m is the month, d is the day, h is the hour, and m is the minute, and it is placed in the `/m1/voyager/xxxdb/rpt` directory.

Error File

When Bulk Import encounters records during import that cannot be processed, they are written to the error file.

The default filename is `err.imp.yyyymmdd.hhmm`, where y is the year, m is the month, d is the day, h is the hour, and m is the minute, and it is placed in the `/ml/voyager.xxxdb/rpt` directory.

This file includes any records that were incorrectly formatted MARC records found during the `Pbulkimport` job. These errors can include the messages listed below.

Messages in Log and Error Files

There are many messages that may display in the log file after running bulk import.

Authority record has a status of D, S, or X written to reject.imp file

An authority record with this status cannot be imported.

Could not delete Mfhd NNNNN. There are attachments.

A MFHD record with attached items or purchase orders cannot be deleted or replaced.

DB Bib Record cancelled/deleted by Import Record!

The Import/Replace profile dictated that an existing record in the database should be removed and replaced by the imported record, based on the Cancellation code selected. The record from the database is written to the delete file.

Delete Failed - Items Attached

A bibliographic record with attached items cannot be deleted.

Delete Failed - MFHDs Attached

A bibliographic record with attached holdings records cannot be deleted.

Delete Failure

If you receive this message, contact Ex Libris Customer Support.

Duplicate detected. Discarding dupe import record!

The Import/Replace profile dictated that an imported record should never overwrite a matching record in the database. The imported record is written to the discard file.

Duplicate detection failure!

If you receive this message, contact Customer Support.

Duplicates above replace threshold. Adding anyway

The Import/Replace profile dictated that any duplicate records will be ignored and that the imported record will always be added to the database (Add-Unconditional).

Duplicates above thresholds. Cannot resolve!

The Import/Replace profile dictated that if more than one duplicate above the replace threshold is detected, the imported record should not be saved to the database. The record is sent to the discard file.

Existing database record replaced by imported

The Import/Replace profile dictated that an imported record will always overwrite a single matching record in the database (Replace) and the record is overwritten.

Failed to load converter for <character set>

Mapping characters from the listed character set to UTF-8 is not working. If you receive this message, contact Ex Libris Customer Support.

No Matches found for Input Cancel Record. Import Record Discarded!

The Import/Replace profile dictated that an existing record in the database should be found which matches the imported record and be then replaced, based on the Cancellation code selected. The imported record is written to the discard file.

No item option in use - No item records created

No items will be created during this run of Bulk Import because Load Bib/MFHD is the option specified in the Bulk Import Rules.

Non-Bibliographic Record detected! Holdings record written to error file

The record is not a bibliographic record type. The record is written to the error file. If you are trying to load or delete MFHDs, make sure the appropriate switch is included on the command line.

Record *nnn* discarded -- no matching records in the database

Because a duplicate detection profile which had the discard incoming records option checked, record *nnn* was discarded. For more information on the discard incoming records option, see *Bulk Import Rules* and Bibliographic Duplicate Detection in the *Voyager System Administration User's Guide*.

Record does not match format for '<character set>' change your import rule

The records do not match the input rule character set, if this is the first record then it will exit, otherwise it will just reject the record.

Record Retrieval Failure

If you receive this message, contact Ex Libris Customer Support.

Record Parse Failure

If you receive this message, contact Ex Libris Customer Support.

Unparseable record written to error file

The `Pbulkimport` command was unable to determine what was being imported. The unrecognizable record is written to the delete file.

If the error involved the conversion, the error messages are similar to the output logs from the conversion process. The differences are that bulk import does not give the field index and loose translations are errors for bulk import.

Introduction

Global Headings Change (GHC) is a way to update all the name, title, name/title, or subject fields of Bibliographic records, based on a change in an Authority record.

Purpose of This Chapter

This chapter provides

- Overview of the Global Headings Change process
- Parameters used to govern the GHC jobs
- Step 1: Process Global Heading Change Queue- Catjob 11
- Step 2: Process Global Heading Change Queue- Catjob 12
- Step 3: Process Global Heading Change Fields- Catjob 13

Overview of GHC

Global Headings Change (GHC) updates the name, title, name/title, or subject fields of Bibliographic records, based on a change in an Authority record.

When a change is made to an authority record the Global Change Queue (in the Cataloging module) displays the change as well as the associated records. This allows the user to decide which changes to implement.

GHC is basically a three step process.

- the bibliographic records that contain the old heading to be changed are found,
- a preview of the change is available,
- the change is actually made.

See the *Voyager Cataloging User's Guide* for information about Global Headings Change and Authority Control.

The GHC process consists of running a job on the server, then doing some activity in the cataloging module.

The catjobs are run from the `/ml/voyager/xxxdb/sbin` directory on your server.

After running each job, entries are made in the `catjob.log` file, located in the `/ml/voyager/xxxdb/rpt` directory.



IMPORTANT:

Jobs 11, 12, and 13 MUST BE RUN IN THAT ORDER for a specific heading change to be complete.

Any records that are to be processed through the GHC queue must not be manually changed between the steps of the batch jobs. If any changes are made to a record it will be removed from the queue.

NOTE:

These server jobs can be run using WebAdmin, see [Cataloging Utilities](#) on [page 31-12](#)".

Parameters

The following parameters govern the Global Headings Change batch jobs.

-d Database name -- not required.

Automatically specified by the script

It is the name of the database that will be updated. You must have write access on the server to run the GHC program.

-u Username and password -- not required.

Automatically specified by the script. Specifying the username and password with this parameter overrides the name and password in the script.

It is the username and password for access to the specified database. Enter it in this format: username/password

-j Job number -- required.

Number of the job that you want to run.

-L Location code -- not required.

Used for job 13. It is the cataloging location code for the cataloging happening location. This updates the record's location code in the **History** tab of the bibliographic record. If prompted for a code, press the **Enter** key if you do not want a location associated with the update.

NOTE:

The implication of not using a location when running Catjob 13 is:

- if this site contributes to a Universal Catalog (UC), *and*
- when importing into that UC you've set the Catjob parameter such that the system only looks at the last (update) location, *and*
- there are specified locations from which records should be excluded, that is not imported into the UC, *then* this record would be exported.

-o Operator -- not required.

Used for job 13. This updates the record's update operator in the **Record History** dialog.

-l List job options -- not required.

List of job options.

-v Version -- not required.

Display version information.

-h Help -- not required.

Display help and usage statement.

Process Global Heading Change Queue Step 1 (Catjob 11)

This batch job begins the Global Headings Change process by finding all of the possible headings to be changed. The new heading will appear in the GHC queue (in the Cataloging module) as it would appear in all Bibliographic records with that heading, and it will show how many Bibliographic records are associated with the heading.

This job should be run from the `/ml/voyager/xxxdb/sbin` directory on your server. At the `/sbin>` enter

```
Pcatjob -j11
```

After running this job the operator should look at the GHC queue (in the Cataloging module), if you click the plus sign you will see the new heading.

Select the box marked **Process**.

It writes to the `catjob.log` file that is placed in the `/ml/voyager/xxxdb/rpt` directory.

The `catjob.log` should include the day, date, time entry, and the messages:

```
Fri Dec 28 09:50:32 2001 Job execution begun.
Fri Dec 28 09:50:34 2001 Connection to Voyager Database
successful...
Fri Dec 28 09:51:21 2001 process global heading change
queue
Fri Dec 28 09:51:21 2001 Starting heading change queue
processing...
Fri Dec 28 09:51:22 2001 Completed heading change queue
processing.
Fri Dec 28 09:51:58 2001 Job execution complete, or
Fri Dec 28 09:51:28 2001 No heading queue entries to
process at this time!
```

Now you are ready to run Catjob 12.

Process Global Heading Change Queue Step 2 (Catjob 12)

This batch job prepares the preview of all fields in the bibliographic and authority records to be changed.

This job should be run from the `/ml/voyager/xxxdb/sbin` directory on your server. At the `/sbin>` enter

```
Pcatjob -j12
```

After running the job the operator should look in the GHC queue (cataloging module) to see the proposed changes.

Click the **Preview** button to look at the bibliographic records to be changed. At this point you can delete any records you want from this list.

Select the **Process** box on the headings in which you would like the change to be reflected.

It writes to the `catjob.log` file that is placed in the `/ml/voyager/xxxdb/rpt` directory.

The `catjob.log` should include the day, date, time entry, and the messages:

```
Fri Dec 28 10:17:08 2001 Job execution begun.  
Fri Dec 28 10:17:09 2001 Connection to Voyager Database  
successful...  
Fri Dec 28 10:17:11 2001 process global heading changes  
Fri Dec 28 10:17:11 2001 Starting heading change  
processing...  
Fri Dec 28 10:17:12 2001 Total Bibs processed: 1  
Fri Dec 28 10:17:12 2001 Total Auths processed: 0  
Fri Dec 28 10:17:12 2001 Completed heading change  
processing.  
Fri Dec 28 10:17:15 2001 Job execution complete.
```

Now you are ready to run Catjob 13

Process Global Heading Change Fields Step 3 (Catjob 13)

This batch job updates the bibliographic and authority records if the heading was marked for processing in the cataloging module's global headings change queue.

This job should be run from the `/m1/voyager/xxxdb/sbin` directory on your server. At the `/sbin>` enter

```
Pcatjob -j13 -L(location code) -o(Operator)
```

Neither the `-L` parameter, **Location code** nor the `-O` parameters are required. If used, the location and operator are logged in the **History** tab of the Bibliographic record.

NOTE:

If running the job from the menu, the system prompts for a Location and Operator code even though they are not required, operators should press the **Enter** key to indicate that no location or operator should be associated with the update.



IMPORTANT:

This step is very machine intensive. The time it takes to run is site-specific, and depends on the time of day and usage of the system. Run this job during slow use periods.

After the job has run, go back into the Cataloging module and look at the GHC queue, notice that they are gone as the changes have been made. You can also search for a bibliographic record that was to change and verify that it has changed appropriately.

Any headings still listed in the queue have not been processed. This means that the system recognizes that a change to the authority record has happened but the changes have not been made to the corresponding bibliographic records.

It writes to the `catjob.log` file that is placed in the `/m1/voyager/xxxdb/rpt` directory.

The `catjob.log` should include the day, date, time entry, and the messages:

```
Fri Dec 28 10:20:48 2001 Job execution begun.
Fri Dec 28 10:20:49 2001 Connection to Voyager Database
successful...
Fri Dec 28 10:20:51 2001 process global heading change
fields
```

Fri Dec 28 10:20:56 2001 Starting heading change fields
processing...

Fri Dec 28 10:21:01 2001 Total records processed:
3

Fri Dec 28 10:21:01 2001 Total records left
unprocessed: 1

Fri Dec 28 10:21:01 2001 Completed heading change field
processing.

Fri Dec 28 10:21:08 2001 Job execution complete.

Storage Barcode Verify (Pstrgvfy) Program

15

Introduction

The Storage Barcode Verify (Pstrgvfy) program serves three purposes

- To change the location codes of item records or item records and MFHDs.
- To populate the Operator or Location columns of a MFHD's **History** tab in the Cataloging module (if the location codes of MFHDs are changed).
- To verify the barcodes of item records or item records and MFHDs.

When you run the change location command, the program also checks for duplicate barcodes. When you run the verify barcode command, the program checks for duplicate barcodes as well as verifies the location code and status.

Sites must create an input file consisting of the barcodes associated with records where you want to change the location code, populate the **History** tab, or verify the barcode.

NOTE:

Because of Pstrgvfy's ability to change location codes and verify barcodes, it is used to load and verify barcodes in the Automated Retrieval System (ARS). The Automated Retrieval System (ARS) is an extension module used to handle storage and accessibility to library materials that are warehoused at remote storage facilities. For detailed information, see the *Voyager Automated Retrieval System (ARS) User's Guide*.

Purpose of This Chapter

This chapter discusses

- Changing location codes of a item records or MFHDs
- Populating the operator or location columns of a MFHD's **History** tab
- Verifying barcodes of item records or MFHDs
- Input file
- Command line parameters
- Command file
- Log file
- Sample command line

Changing Location Codes of Item Records or MFHDs

`Pstrgvfy` allows you to automatically change the location codes of item records or item records and MFHDs (852|b field).



IMPORTANT:

Although you can change the location codes of item records alone, you cannot change MFHD location codes alone. MFHD location codes must be changed with item record location codes simultaneously.

The results of the location code change are reported in the log file (if a duplicate barcode is found, the log file will include information to that effect as well as the barcode duplicated).

You specify the item records or item records and MFHDs for which you want the location codes changed in an input file. This input file contains a list of barcodes that are associated with the item records or item records and MFHDs. The location code to which you want the item records or item records and MFHDs changed is specified at the `Pstrgvfy` command line (using the `-m` parameter).

For detailed information on locations and location codes, see the *Voyager System Administration User's Guide*.

Changing the location code of item records or item records and MFHDs using `Pstrgvfy` is particularly useful if you are moving many materials from one physical location to another. For example, if you are pulling many items from your

main stacks (with a location code of Main) and placing them in a special collection (with a location code of SpColl2), `Pstrgvfy` allows you to change the location of the associated item records or item records and MFHDs (852|b field) from Main to SpColl2 without having to manually wand in each item and change the location yourself. In this way, `Pstrgvfy` saves a significant amount of time and effort.

NOTE:

`Pstrgvfy` does not change the location for a line item on a purchase order. The line item location, which contains the serials history, populates the MFHD location when it is created. If `Pstrgvfy` is used to change the location of the MFHD/item record, then it will not match the line item location. When the line item location and the MFHD location are different, WebVoyage displays the holdings information based on the MFHD location. Therefore, if you change the MFHD/item record location using `Pstrgvfy`, the serials history will not display. You can change the line item location using the Acquisitions module. When the locations are the same, both the holdings information and the acquisitions serials history will display in WebVoyage. See the *Voyager Acquisitions User's Guide* for more information.

Populating the Operator or Location Columns of a MFHD's History Tab

If you change the location codes of MFHDs using `Pstrgvfy`, you can opt to specify the operator name or Cataloging location code that will populate the Operator or Cataloging Location columns on the **History** tab of MFHDs in the Cataloging module. See the *Voyager Cataloging User's Guide* for details on the MFHD **History** tab. This is useful for record keeping purposes, as it allows you to keep track of which operators have made changes to MFHDs, and at what location.

NOTE:

This does not apply to item records, as item records do not have a **History** tab. In addition, it does not apply if you have verified barcodes related to MFHDs using `Pstrgvfy`.

You specify the operator name or Cataloging location code at the command line following the `-o` or `-g` parameters respectively. If you do not specify an operator name, the default (SYSTEM) will populate the Operator column. If you do not specify a Cataloging location code, the Cataloging Location column will be empty.

Verifying Barcodes of Item Records or MFHDs

`Pstrgvfy` allows you to compare and verify barcodes related to item records, or item records and MFHDs, at specific locations. The verification process includes

- Checking for duplicate barcodes
- Ensuring that the related item records, or item records and MFHDs, have the correct location code
- Ensuring the barcodes do not have an inactive status

The results of the verification are reported in the `Pstrgvfy` log file (if a duplicate barcode is found or if the barcode is at the wrong location, the log file will include information to that effect as well as the barcode duplicated).

You specify which barcodes you want verified using an input file. The barcodes in the input file are compared with barcodes related to item records, or item records and MFHDs, with the location code you specify at the command line (via the `-m` parameter).

If you want to compare the barcodes in the input file with barcodes related to item records, or item records and MFHDs at multiple locations, you can create a command file (as opposed to specifying multiple `-m` parameters at the command line). This command file contains a list of `-m` parameters and location codes.

Pstrgvfy Input File

You must create an input file to use `Pstrgvfy`. This input file is an ASCII flat file that you can create in any text editor, such as Microsoft® Notepad.

It consists of a list of barcodes that you want in or enter by some other means. No other information should be included in the input file.

Depending on what you want `Pstrgvfy` to do (and of course, which command line parameters you use), `Pstrgvfy` will take the list of barcodes in the input file and either change the location code of the associated item records or item records and MFHDs, or verify the barcodes in the file against those associated with item records, or item records and MFHDs.

If you are changing location codes you can also command `Pstrgvfy` to populate the Operator or Location columns of the MFHD's **History** tab in the Cataloging module.

**IMPORTANT:**

Each barcode in the input file must be placed on its own line.

By default, the input file will be taken from the `/m1/voyager/xxxdb/rpt` directory. You can place the input file in a different directory, in which case you need to specify the directory at the command line (with the `-i` parameter).

Pstrgvfy Parameters

The following parameters govern the Pstrgvfy program.

-d Database name -- not required.

Automatically specified by the `Pstrgvfy` script from the `voyager.env` file. It is the name of the database that will be accessed. You must have read-access on the server to run the extract program.

-u Username and password -- not required.

Automatically specified by the `Pstrgvfy` script from the `voyager.env` file. It is the username and password for access to the specified database. Enter it in this format: `username/password`.

-i Input file -- required.

If the input file is located in a directory other than the default (`/m1/voyager/xxxdb/rpt`), the `-i` is used to specify its path and filename with extension.

-m Location -- required.

This is only valid with the `-a` or the `-v`.

When combined with the `-a`, `-m` is used to specify a single location code to which `Pstrgvfy` will set item records, or item records and MFHDs, related to the barcodes in the input file. Only one `-m` can be used with the `-a`. If you use more than one, `Pstrgvfy` will only apply the first one.

When used with the `-v`, `-m` will verify the barcodes listed in the input file against those associated with item records or MFHDs for the location you specify. You can specify multiple locations, in which case you should create a command file (as opposed to using multiple `-m` parameters at the command line).

NOTE:

Location codes are case sensitive and must appear exactly as they do in System Administration.



IMPORTANT:

The -m parameter can be used with either the -a or the -v parameters but it cannot be used with both in the same command.

-a Alter/update mode flag -- either -a or -v is required.

This is only valid with the -m; cannot be used with the -v.

When used with the -m, this stand-alone parameter (no qualifying data needed) is used to change the location codes of item records, or item records and MFHDs, related to the barcodes listed in the input file. You must also use the -b parameter if you want to change the location code of MFHDs. If you do not, only the location code of item records will be changed.

If a duplicate barcode is found, it will be counted as bad in the log file, and its location will not be changed or added.

-b Update of MFHD 852b -- not required.

This is only valid with the -a; cannot be used with the -v.

When used with the -a and the -m, this stand-alone parameter (no qualifying data needed) is used to change the location code of MFHDs (in addition to item records) that are related to barcodes in the input file. `Pstrgvfy` will set the 852[b field of the MFHDs to the location code specified using the -m parameter. If the location code in the 852[b field matches the one specified via the -m, no change will be made. `Pstrgvfy` will acknowledge that no change was made in the log file.

If you do not use this parameter, only the location codes of item records will be changed.

-v Verify mode flag -- either -v or -a is required.

This is only valid with the -m; cannot be used with the -a.

This stand-alone parameter (no qualifying data needed) is used to verify barcodes in the input file against barcodes associated with item records or item records and MFHDs with the location code you specify with the -m, (or with multiple location codes you specify in a command file). Using the -v, `Pstrgvfy` will check for

duplicate barcodes, ensure that the associated item records or item records and MFHDs have the location code you specify, and ensure that the barcodes do not have an inactive status.

-o Operator ID -- not required.

This is only valid with the -b.

The -o is used to specify an operator name that will populate the Operator column on the MFHD's **History** tab, (indicating which operator altered the MFHD). This does not apply to item records, as item records do not have a **History** tab. You define operator names in the System Administration module. See the *Voyager System Administration User's Guide* for more information.

If you do not specify an operator name with the -o, or if the operator name is invalid, the default (SYSTEM) will be used.

-g Cataloging Location -- not required.

This is only valid with the -b.

With the -g you can specify a Cataloging location code (case sensitive) that will populate the Cataloging Location column on the MFHD's **History** tab in the Cataloging module. This does not apply to item records, as item records do not have a **History** tab.

The location specified with the -g must be a valid location code as defined in System Administration module. If it is invalid (for example if you misspell it at the command line), Pstrgvfy will stop. Pstrgvfy will acknowledge that the location code is invalid in the log file.

If the location is not a valid Cataloging happening location, Pstrgvfy will process the input file of barcodes but will not populate the Cataloging Location column on the **History** tab of the MFHD. Pstrgvfy will acknowledge that the location is not a valid Cataloging happening location in the log file. You define locations as Cataloging happening locations in the Cataloging Policy Groups portion of the System Administration module. See the *Voyager System Administration User's Guide*.

If you do not specify a Cataloging location with the -g, the Cataloging Location column of the MFHD will be empty.

-c Command filename -- not required.

This is only used with the -v.

With the `-c` you specify the path and filename (with extension) of a command file to be referenced by `Pstrgvfy`. This command file contains a list of `-m` parameters with location codes from which you want barcodes in the input file to be compared and verified.

The `-c` is only used if barcodes of item records or MFHDs are being verified (with the `-v` parameter); not added/changed.

-l Log filename -- not required.

If you do not want the `Pstrgvfy` log file written to the default directory (`/m1/voyager/xxxdb/rpt`), you can specify a path and filename (with extension) using the `-l`.

-h Help -- not required.

The `-h` provides a listing and brief description of all the valid parameters for the `Pstrgvfy` script.

Pstrgvfy Command File

If you want to compare and verify barcodes in the input file against those related to item records or item records and MFHDs at more than one location, you can create a `Pstrgvfy` command file. That way, you do not have to use multiple `-m` parameters at the command line followed by multiple location codes. The command file will only be referenced by `Pstrgvfy` if the `-c` parameter is used at the command line.

You can create the command file in a text editor such as Notepad. The command file has the `.com` file extension (`strgvfy.com`), and by default, resides in the `/m1/voyager/xxxdb/local` directory. You can place the command file in a directory other than the default, in which case you must specify a full path name to the file (with extension) at the command line using the `-c` parameter. For example:

```
-c/m1/voyager/xxxdb/temp/strgvfy.com
```

where `strgvfy.com` is the name of the command file.

NOTE:

The command file should not reside in the `/m1/voyager/xxxdb/sbin` directory as that directory is used primarily for the `Pscripts`.

The command file contains a list of -m parameters, each followed by a single location code. Each -m and location code appears on its own, separate line exactly as it does in the System Administration module, as in the following example:

```
-m Main
-m SpColl2
-m SpColl3
```

Pstrgvfy Log File

A log file called `log.strgvfy.date.time`, is automatically created by `Pstrgvfy`. It includes the following information:

- A count of all the barcodes changed or verified for the locations specified
- The date and time the run started and ended
- (Depending on whether or not you changed location codes or verified barcodes) information about problems `Pstrgvfy` encountered during its run (for example, invalid location codes, duplicate barcodes, barcodes at wrong locations.)

By default, `Pstrgvfy` places the log file in the `/ml/voyager/xxxdb/rpt` directory. You can specify another filename and path to which you want the log file written using the `-l` parameter at the command line.

NOTE:

If you specify another filename and path using the `-l` command line parameter, the date and time will not be included as a part of the `Pstrgvfy` log filename.

An example of the contents of a `Pstrgvfy` log file is:

```
.Barcode Alter.Verify Processing: Start Time: Mon June
20 08:58:00 20 00

Error: Location Id Lookup| <SpColl1>
Error: Invalid location code.| <SpColl4>
Error: Barcode at Wrong location.| <CIRC>
Error: Barcode is duplicated.| <39550000123456>
Error: Barcode at Wrong location.| <CIRC>
Error: Barcode at Wrong location.| <CIRC>
Error: Barcode at Wrong location.| <CIRC>
Error: Barcode at Wrong location.| <CIRC>
```

```
Final counts:  Read: 6|Valid: 0|Bad: 6|  
Barcode Alter/Verify Processing: End Time: Mon Jun 20  
08:58:01 2000
```

This sample log file pertains to a run that changed location codes.

Sample Pstrgvfy Command Line

Pstrgvfy is run from the /m1/voyager/xxxdb/sbin directory on the Voyager server.

An example command line might be, at /m1/voyager/xxxdb/sbin> enter

```
Pstrgvfy -i/m1/voyager/testdb/local/  
out.strg.2000405.1018 -mSpColl2 -a -b -gCatOp -oSpColl2
```

This instructs Pstrgvfy to reference an input file located in the /m1/voyager/testdb/local directory, called out.strg.2000405.1018, change the location code of both item records and MFHDs associated with barcodes in the input file to SpColl2, populate the operator name and Cataloging location code on the MFHD's **History** tab with CatOp and SpColl2 respectively.

Introduction

Included with Voyager is the `Popacjob` program which is comprised of three parts.

- The first part, OPAC Log Export (formerly *Popaclogexp*), pulls the specified information from the `OPAC_search_log` table in the database into a comma-delimited text file. The information in the comma-delimited text file is presented in [Table 16-1](#) on [page 16-2](#).
- The second part, SDI Searches, runs search queries stored by patrons at the appropriate intervals. SDI allows patrons to choose intervals for each saved search query to be run automatically (on the local database) and the results e-mailed to them in the form of a URL.
- The third part, OPAC Bib Usage Log Export (formerly performed using the `-b` switch when running *Popaclogexp*), pulls the specified information from the `bib_usage_table` in the database into a comma-delimited text file. The information in the comma-delimited text file is presented in [Table 16-2](#) on [page 16-3](#).

Purpose of This Chapter

The purpose of this chapter is to detail the three components of `Popacjob`, including configuration and running the jobs as crons.

OPAC_Search_Log Table

For OPAC search logging, the following information is stored in the OPAC_search_log table in the database for every search (see [Table 16-1](#)).

Table 16-1. OPAC_search_log Table

FIELD	DESCRIPTION
Search_date	Search date and time.
Stat_string	The requesting client's 10-character ID number from the OPAC.ini file (see step 2 of "Setting Up OPAC Search Logging" on 16-4).
Session_id	String generated by OPAC client when started or reset (yyyymmddhhmmss)
Search_type	Type of search performed (Author, Subject.).
Search_string	Query entered as the criteria for the search.
Limit_flag	Were limits in affect? Y/N
Limit_string	Details on limits.
Index_type	A=authority, B=browse, K=keyword or L=left-anchored.
Relevance	Were the results relevance ranked? Y/N
Hyperlink	Was the search the result of clicking a hyperlink? Y/N
Hits	The number of hits returned by the search. Note: Hit count will not apply to Heading Subject and Heading Call Number searches which are browse type searches of an entire index. For these searches, a "-1" will be recorded.
Search_tab	0-5 search tab number.
Client_type	W=Web OPAC O=Tomcat WebVoy��ge OPAC A=ASCII OPAC Z=Z39.50
Client_ip	The IP (Internet Protocol) address of the requesting client, for example, 128.218.1.38.
dbkey	Unique identifier of the database which the user performed the search against.
Redirect Flag	Y= Search performed was redirected. N= Search performed was not redirected.

Bib_usage_log Table

For bibliographic usage logging, the following information is stored in the bib_usage_log table in the database (see [Table 16-2](#)).

Table 16-2. Bib_usage_log Table

FIELD	DESCRIPTION
Client_type	C=Cataloging Z=Z39.50 W=Web OPAC A=ASCII OPAC G=Other
Use_date	The date and time on which the usage of the bibliographic record occurred.
Operator_id	The ID of the operator who instigated the transaction of the bibliographic record. Note: This information will be recorded only for Cataloging clients, not for OPAC clients.
Location_id	The id of the location from which the transaction of the bibliographic record was done. Note: This information will be recorded only for Cataloging clients, not for OPAC clients.
Session_id	String generated by OPAC client when started or reset (yyyymmddhhmmss).
Stat_string	The requesting client's ten-character ID number from the opac.ini file (see step 2 of "Setting Up OPAC Search Logging" on 16-4). Note: This information will be recorded only for OPAC clients, not for Cataloging clients.
Client_ip	The IP (Internet Protocol) address of the requesting client, for example, 128.218.1.38. Note: This information will be recorded only for OPAC clients, not for Cataloging clients.
Bib_id	The id of the bibliographic record involved in the transaction.

Table 16-2. Bib_usage_log Table

FIELD	DESCRIPTION
Use_type	M=save, MARC format D=display S=save, text format E=e-mail P=print Z=Z39.50 retrieval Note: S and E will be recorded only for OPAC clients, not for Cataloging clients, because a bibliographic record cannot be saved in text format or e-mailed from Cataloging. Also, P will not be recorded for Web OPAC because printing is done by the web browser.

The information stored in the above mentioned tables can be accessed in one of two ways by querying the database using MS Access (through linking tables) or with Popacjobs 1 and 3 (see [Accessing Logged Information](#) on [page 16-6](#)). Any SQL software can query the database for the appropriate information.

NOTE:

Search logging takes up a lot of drive space. A single record is around 600 characters in length; saving 1,000 searches to the database requires 73 megabytes of hard-drive space. If you plan on keeping Search Logging activated for any length of time, you should plan on regularly extracting information and purging it from the database (by specifying option -p).

Setting Up OPAC Search Logging

Use the following to set up OPAC search logging.

1. Turn the Search Logging feature on.

In the `/ini` directory on the server (`/ml/voyager/xxxdb`), the `voyager.ini` file contains a line reading

LOGSEARCH=

2. Place a `y` after the equals sign to turn search logging on; or an `n` after the equals sign to turn search logging off.

At the time of installation, search logging is turned off.

Optionally, you can set the identifier string for computers that will be running OPAC. This string is in the `opac.ini` file of each WebVoyage display directory (`vcit`, `zcit`, `z3950`, and so on) and its format is as follows:

StatString=

Any text after the equals sign will be recorded in the `OPAC_search_log` table or `bib_usage_log` table as the requesting OPAC clients's ID number. It may be a maximum of ten characters (including spaces). If left blank, the field will display as `WebOpac`.

The string may be unique for each WebVoyage display directory containing its own `opac.ini` file (`vcit`, `zcit`, `z3950`, etc...), or you can assign the same string to any number of display directories.

If OPAC is running on terminals from a server (WebVoyage), all terminals connecting to that server will return the same ID string.

In addition, regardless of whether the `StatString=` feature is in use, the IP address of each requesting OPAC client will be recorded in the `Client_ip` field of the `OPAC_search_log` table.

Setting Up Bibliographic Usage Logging

To set up bibliographic usage logging, you must do the following:

Turn the bibliographic usage logging feature on.

In the `/ini` directory on the server (`/ml/voyager/xxxdb`), the `voyager.ini` file contains the following variable:

LOGBIBUSAGE=

Place a **Y** after the equals sign to turn bibliographic usage logging on; or an **N** after the equals sign to turn bibliographic usage logging off.

At the time of installation, bibliographic usage logging is turned off.

Optionally, you can set the identifier string for computers that will be running OPAC. This string is in the `opac.ini` file of each WebVoyage display directory (`vcit`, `zcit`, `z3950`, etc...) and its format is as follows:

StatString=

Any text after the equals sign will be recorded in the OPAC_search_log table or bib_usage_log table as the requesting OPAC clients's ID number. It may be a maximum of ten characters (including spaces). If left blank, the field will display as WebOpac.

The example in [Figure 16-1](#) shows information in the opsrchlgexport.log file from WebVoyage sessions on a database using SDirect as its StatString.

Figure 16-1. Sample StatString Results

The string may be unique for each WebVoyage display directory containing its own opac.ini file (vcit, zcit, z3950, etc...), or you can assign the same string to any number of display directories.

If OPAC is running on terminals from a server (WebVoyage), all terminals connecting to that server will return the same ID string.

In addition, regardless of whether the StatString= feature is in use, the IP address of each requesting OPAC client will be recorded in the Client_ip field of the bib_usage_log table.

Setting up Selective Dissemination of Information (SDI)

For information on enabling and configuring SDI, see [SDI Searches Program \(Classic WebVoyage\)](#) on [page 16-15](#).

Accessing Logged Information

OPAC Search Log Export program

Included with Voyager is the Popacjob program, which is comprised of three parts. The first, OPAC Log Export (formerly Popaclogexp), pulls the specified information from the OPAC_search_log table in the database into a comma-delimited text file. The information in the comma-delimited text file is presented in [Table 16-1](#) on [page 16-2](#).

You can run the OPAC Search Log Export and Opac Bib Usage Log Export programs as separate jobs to access both search logging information and bibliographic usage information.

OPAC Search Log Export may be run interactively, or set to run automatically as a cron. Also, for information about running the job in WebAdmin see [Popacjob](#) on [page 31-36](#).

Interactive Method

Use the following steps to run OPAC Search Log Export interactively:

Switch to the `/sbin` directory (`/ml/voyager/xxxdb`)

Type **Popacjob**.

Press **Enter**.

A menu containing the following four options displays (see [Figure 16-2](#)):

- 1 -- Opac Search Log Export
- 2 -- SDI Searches
- 3 -- Opac Bib Usage Log Export
- 99 -- Quit

At the Process Job # prompt, type 1.

Press **Enter**.

Figure 16-2. OPAC Job Options Menu

The program will prompt you for a date range (see [Figure 16-3](#)). Enter the desired date range and press **Enter**.

The range formats, *yyyy-mm-dd:yyyy-mm-dd* or *today-n* allow you to do one of the following:

- *yyyy-mm-dd:yyyy-mm-dd* format: Specify dates from which Voyager will extract entries from the applicable table in the database (for example, 1997-01-01:1998-03-15 will extract every entry from 1/1/97 through 3/15/98).
 - *today-n* format: Specify the number of days before the current date for which Voyager will extract entries (for example, today-7 will extract entries for the current date and the 7 days previous). Excluding the *-n* switch will set the date to the current day.
-

Figure 16-3. OPAC Search Log Export Job Date Range Prompt

NOTE:

When running this job from the menu, the *-p* parameter, that is available when running the job as a cron or from the command line, is not available.

Cron Method

You can run OPAC Search Log Export as a cron, using the following options:

- j Specifies which Popacjob program to run.
 - 1 will run OPAC Search Log Export
 - 2 will run SDI Searches
 - 3 will run OPAC Bib Log Export

Set this to 1 to run OPAC Search Log Export.

-o<filename>Creates an output file with the specified filename. The default filename is opsrchlgexport.dat.

The range formats, `-r<yyyy-mm-dd:yyyy-mm-dd>` OR `-r<today-n>` allow you to do one of the following:

- `-r<yyyy-mm-dd:yyyy-mm-dd>` format: Specify dates from which Voyager will extract entries from the applicable table in the database (for example, `-r1997-01-01:1998-03-15` will extract every entry from 1/1/97 through 3/15/98).
- `-r<today-n>` format: Specify the number of days before the current date for which Voyager will extract entries (for example, `-rtoday-7` will extract entries for the current date and the 7 days previous). Excluding the `-n` switch will set the date to the current day.

`-p` Purges the applicable table in the database of the specified entries after saving them to the export file. (Recommended)

`-q` Quiet mode - will not prompt for any input.

`-v` Provides version information.

`-h` Provides help for the program.

After running the program:

Data extracted from the `OPAC_search_log` table is placed in the `/rpt` directory in the file `opsrchlgexport.dat` or in the file specified after the `-o` parameter. The log file for this operation is named `opsrchlgexport.log` or uses the filename specified after the `-o` parameter with the file extension of `.log`. A file in the `/log` directory called `log.voyager` gets updated with a record of the extraction operation.

SDI Searches Program (Tomcat WebVoyage)

Included with Voyager is the `Popacjob` program that is comprised of three parts. The second part, SDI (Selective Dissemination of Information) Searches, runs search queries stored by patrons at the appropriate intervals. This section describes the setup for the Tomcat WebVoyage OPAC.

SDI allows patrons to choose intervals for each saved search query to be run automatically using the IP address and port pulled from the `database_address` table in the LOCAL database and , subsequently, the results e-mailed to them in the form of a URL.

Saved Searches must be enabled to use SDI (see [Enabling SDI](#) on [page 16-10](#)). SDI Searches may be run interactively (see [Run SDI - Interactive Method](#) on [page 16-13](#)) or set to run automatically as a cron (see [Run SID - Cron Method](#) on

[page 16-14](#)). Your library must also configure the `sdiemail.ini` file before running the SDI Searches program (see [Customizing the Search Results URL Page Sent to Patrons by SDI](#) on [page 16-18](#)).

Enabling SDI

Use the following procedure to enable SDI and customize the labels for the SDI options.

NOTE:

Directory path references to `xxxdb` implies that you need to substitute your database path name; and where `[skin]` is referenced, substitute the path name that is used at your site. The default skin path provided is `en_US` as in the following:

```
/ml/voyager/xxxdb/tomcat/vwebv/context/vwebv/ui/en_US/
```



Procedure 16-1. Enabling SDI and Customizing Labels

1. Make a backup copy of the `webvoyage.properties` file that is located in `/ml/voyager/xxxdb/tomcat/vwebv/context/vwebv/ui/[skin]/`.
2. Enable the SDI option in `webvoyage.properties`.

Search for `option.SDI` (see [Figure 16-4](#)).

```
#=====#
# Option to display the "Save Search as Alert" link
# NOTE: "Save Search" functionality must be enabled for "Save Search as Alert" \
# to work.
#=====#
option.SDI=N
```

Figure 16-4. `option.SDI=`

3. Set the `option.SDI` to `Y`.
4. Customize the SDI display labels for `newhits` and `alertfrequency`.

Search for `newhits` (see [Figure 16-5](#)).

```
#=====#
#
# My Searches
#
#=====#
page.searchMySearches.page.title=WebVoyáge My Searches
page.searchMySearches.page.heading=My Searches
page.searchMySearches.heading.combine=Combine
page.searchMySearches.heading.search=Search
page.searchMySearches.heading.newhits=Results
page.searchMySearches.heading.searchType=Search Type
page.searchMySearches.heading.date=Date
page.searchMySearches.heading.alertfrequency=Alert Frequency
page.searchMySearches.heading.action=Actions
page.searchMySearches.heading.summary=This table displays the search history.
page.searchMySearches.data.action.rerun.linkText=Re-Run
page.searchMySearches.data.action.edit.linkText=Edit
page.searchMySearches.data.action.delete.linkText=Delete
page.searchMySearches.data.newhits.yes=Y
page.searchMySearches.data.newhits.no=N
page.searchMySearches.data.alertfrequency.default=1
page.searchMySearches.data.alertfrequency.1=None
page.searchMySearches.data.alertfrequency.2=Daily
page.searchMySearches.data.alertfrequency.3=Weekly
page.searchMySearches.data.alertfrequency.4=Bi-Monthly
page.searchMySearches.data.alertfrequency.5=Monthly
page.searchMySearches.save.button=Save Changes
page.searchMySearches.combineSearches.button=Search
```

Figure 16-5. newhits and alertfrequency Labels

5. Customize the `newhits` and `alertfrequency` display labels to match your preferences.
6. Save and test your changes.

OPTIONAL:

7. *Back out your changes, if necessary, by deploying your backup files.*

Customizing the Search Results URL Page Sent to Patrons by SDI (sdiemail.ini File)

You must customize the layout of the page(s) containing the URLs sent to patrons by SDI. This is done in the `sdiemail.ini` file. This file is in the `/m1/voyager/xxxxdb/ini` directory.

Components of the sdiemail.ini File

The `sdiemail.ini` file is located in the `/m1/voyager/xxxxdb/ini` directory (see [Figure 16-6](#)).

```
##NOTE 80 character max for length of lines.

[Email]
Subject=Search Updates Results
Body1=The following URL contains the results of your automatic search
Body2=Updates. Clicking the link will take you directly to the
Body3=Webvoyage Titles Index.
SearchString=Search for:
SearchType=Search Type:

[Webvoyage_Server]
Address=localhost:22014
Version7Address=localhost:22008
```

Figure 16-6. sdiemail.ini File

Customize the sections of this file according to your preferences.

Make sure that the `Version7Address=` entry in the `[WebVoyage_Server]` stanza contains the correct address of the server on which webopac is running such as `Version7Address=10.111.111.96:7008` or `Version7Address=www.seekandfind.edu` for example.

Any fields in the `[Email]` stanza that are left blank yield the default text.

NOTE:

The 80-character maximum length per line restriction includes the variable name and equal sign (=).

[Figure 16-7](#) shows an e-mail generated by SDI.

Figure 16-7. Sample SDI URL in E-mail

The footer for this e-mail uses the `emailf.cfg` file `/m1/xxxdb/etc/webvoyage/local`.

Run SDI - Interactive Method

Use the following steps to run SDI Searches interactively.



Procedure 16-2. Running SDI with the Interactive Method

1. Switch to the `/sbin` directory (`/m1/voyager/xxxdb`).
2. Type `Popacjob` and press `Enter`.

A menu containing the following four options displays (see [Figure 16-8](#)):

1 -- Opac Search Log Export

2 -- SDI Searches
3 -- Opac Bib Usage Log Export
99 -- Quit

Figure 16-8. OPAC Job Options Menu

3. At the Process Job # prompt, type 2 and press `Enter`.

SDI Searches now determine which saved search queries to run and e-mail the results to patrons.

NOTE:

Make sure to set the cron to run at least as frequently as the shortest available interval in the New Hits Every drop-down box in the SDI options in WebVoyage. For example, if you give patrons the option of running SDI every day, make sure that the cron runs every day.

Run SID - Cron Method

You can run SDI Searches as a cron, using the following options:

- j Specifies which Popacjob program to run.
 - 1 runs OPAC Search Log Export
 - 2 runs SDI Searches
 - 3 runs OPAC Bib Usage Log Export

Set this to 2 to run SDI Searches.

After running the program, the log file (`opacjob.log`) can be found in the `/rpt` directory (`/ml/voyager/xxxdb/rpt`) on your server.

SDI Searches Program (Classic WebVoyáage)

Included with Voyager is the `Popacjob` program that is comprised of three parts. The second part, SDI Searches, runs search queries stored by patrons at the appropriate intervals. This section describes the setup for the Classic WebVoyáage OPAC.

Saved Searches must be enabled to use SDI. SDI allows patrons to choose intervals for each saved search query to be run automatically (using the IP address and port pulled from the `database_address` table in the LOCAL database) and the results e-mailed to them in the form of a URL.

SDI Searches may be run interactively or set to run automatically as a cron.

Your library must configure SDI in WebVoyáage (including the `sdiemail.ini` file) before running the SDI Searches program.

Enabling SDI

You enable/disable SDI in the `SDIOption` variable of the `[SDI_Page]` stanza in the `opac.ini` file, by setting the variable to `Y`. The following is the default `SDIOption` variable:

`SDIOption=N`

The `SDIOption=` variable must be set to **`Y`** or **`N`**:

Setting this variable to **`Y`** enables SDI.

Setting this variable to **`N`** disables SDI.

Configuring SDI Options

[Figure 16-9](#) contains an example of the [SDI_Page] stanza. [Table 16-3](#) details the functions and descriptions of each variable in the stanza

```
[SDI_Page]
SDIOption=N
RunSearchEvery=Run Search Every:
GetNewHits=Y
NewHits=New Hits
None=None
Daily=Daily
Weekly=Weekly
Bi-Weekly=Bi-Weekly
Monthly=Monthly
```

Figure 16-9. The Default [SDI_Page] Stanza of the opac.ini File.

Table 16-3. Components of the [SDI_Page] Stanza

Variable Name	Description	Possible Values	Default Value
SDIOption=	Switches SDI on and off	Y or N	N
RunSearchEvery=	Heading of column listing options for SDI/saved search intervals	Free text	Run Search Every

Table 16-3. Components of the [SDI_Page] Stanza

Variable Name	Description	Possible Values	Default Value
GetNewHits=	Switches the E-mail search results only if new search result information exists on and off.	Y or N Y enables option to send results of query to patron only if new search result information exists. N Disables option to send results of query to patron only if new search result information exists (results are sent by SDI whenever saved searches are performed).	Y
NewHits=	Heading of column containing the Y/N (yes/no) drop-down to select/deselect E-mail search results only if new search result information exists option. Only available if GetNewHits= above is set to Y.	Free text	New Hits?
None=	Text displayed on Saved Search Query screen when patron has no search queries saved	Free text	None
Daily=	Text in drop-down list box of the Run Search Every column corresponding to the E-mail results every day option.	Free text	Daily
Weekly=	Text in drop-down list box of the Run Search Every column corresponding to the E-mail results weekly option.	Free text	Weekly

Table 16-3. Components of the [SDI_Page] Stanza

Variable Name	Description	Possible Values	Default Value
Bi-Weekly=	Text in drop-down list box of the Run Search Every column corresponding to the E-mail results bi-weekly option.	Free text	Bi-Weekly
Monthly=	Text in drop-down list box of the Run Search Every column corresponding to the E-mail results Monthly option.	Free text	Monthly

Customizing the Search Results URL Page Sent to Patrons by SDI

You must customize the layout of the page(s) containing the URLs sent to patrons by SDI. This is done in the `sdiemail.ini` file. This file is in the `/ml/voyager/xxxdb/ini` directory.

Components of the `sdiemail.ini` File

This file is in the `/ml/voyager/xxxdb/ini` directory (see [Figure 16-10](#) for the default). Customize the sections of this file according to your preferences. [Figure 16-11](#) shows an e-mail generated by SDI.

Make sure that the `Address=` entry in the `[WebVoyage_Server]` stanza contains the correct address of the server webopac is running on. (For example: **`Address=10.111.111.96:7008`** or **`Address=www.seekandfind.edu`**).

Any fields in the `[Email]` stanza left blank will yield the default text.

NOTE:

The 80 character maximum length per line restriction includes the variable name and = sign.

The footer for this e-mail uses the `emailf.cfg` file (`/ml/xxxdb/etc/webvoyage/local`).

```
[Email]
Subject=Search Updates Results
Body1=The following URL contains the results of your automatic
      Search
Body2=Updates. Clicking the link will take you directly to the
Body3=Webvoyage titles index.
SearchString=Search for:
SearchType=Search Type:

[Webvoyage_Server]
Address=XXXX
```

Figure 16-10. Default sdiemail.ini File

Figure 16-11. Sample SDI URL in email

Interactive Method

Use the following steps to run SDI Searches interactively:

Switch to the `/sbin` directory (`/ml/voyager/xxxdb`)

Type **Popacjob**.

Press **Enter**.

A menu containing the following four options displays (see [Figure 16-12](#)):

- 1 -- Opac Search Log Export
- 2 -- SDI Searches
- 3 -- Opac Bib Usage Log Export
- 99 -- Quit

At the Process Job # prompt, type 2.

Press **Enter**.

Figure 16-12. OPAC Job Options Menu

SDI Searches will now determine which saved search queries to run, and email the results to patrons.

NOTE:

Make sure to set the cron to run at least as frequently as the shortest available interval in the New Hits Every drop-down box in the SDI options in WebVoyage. For example, if you give patrons the option of running SDI every day, make sure that the cron runs every day.

For more information on configuring SDI intervals for patron selection, see the Daily, Weekly, Bi-Weekly, and Monthly variables in the *Voyager WebVoyage User's Guide*.

Cron Method

You can run SDI Searches as a cron, using the following options:

- j Specifies which Popacjob program to run.
 - 1 will run OPAC Search Log Export
 - 2 will run SDI Searches
 - 3 will run OPAC Bib Usage Log Export

Set this to 2 to run SDI Searches.

After running the program, the log file (`opacjob.log`) can be found in the `/rpt` directory on your server (`/ml/voyager/xxxdb/rpt`).

OPAC Bib Usage Log Export Program

Included with Voyager is the `Popacjob` program, which is comprised of three parts. The third, OPAC Bib Usage Log Export (formerly performed using the `-b` switch when running `Popaclogexp`), pulls the specified information from the `bib_usage_table` in the database into a comma-delimited text file. The information in the comma-delimited text file is presented in [Table 16-2](#) on [page 16-3](#).

You can run the OPAC Search Log Export and Opac Bib Usage Log Export programs as separate jobs to access both search logging information and bibliographic usage information.

OPAC Bib Usage Log Export may be run interactively, or set to run automatically as a cron.

Interactive Method

Use the following steps to run OPAC Bib Usage Log Export interactively:

Switch to the `/sbin` directory (`/ml/voyager/xxxdb`)

Type `Popacjob`.

Press **Enter**.

A menu containing the following four options displays (see [Figure 16-13](#)):

- 1 -- Opac Search Log Export
- 2 -- SDI Searches

3 -- Opac Bib Usage Log Export

99 -- Quit

At the Process Job # prompt, type 3.

Press **Enter**.

Figure 16-13. OPAC Job Options Menu

The program will prompt you for a date range (see [Figure 16-14](#)). Enter the desired date range and press **Enter**.

The range formats, *yyyy-mm-dd:yyyy-mm-dd* or *today-n* allow you to do one of the following:

- *yyyy-mm-dd:yyyy-mm-dd* format: Specify dates from which Voyager will extract entries from the applicable table in the database (for example, 1997-01-01:1998-03-15 will extract every entry from 1/1/97 through 3/15/98).
- *today-n* format: Specify the number of days before the current date for which Voyager will extract entries (for example, today-7 will extract entries for the current date and the 7 days previous). Excluding the *-n* switch will set the date to the current day.

Figure 16-14. OPAC Bib Usage Log Export Job Date Range Prompt

Cron Method

You can run OPAC Log Export as a cron, using the following options:

- j Specifies which Popacjob program to run.
 - 1 will run OPAC Search Log Export
 - 2 will run SDI Searches
 - 3 will run OPAC Bib Usage Log Export

Set this to 3 to run OPAC Bib Usage Log Export.

-o<filename>Creates an output file with the specified filename. The default filename is bibuselgexport.dat

The range formats, -r<yyyy-mm-dd:yyyy-mm-dd> OR -r<today-n>allow you to do one of the following:

- -r<yyyy-mm-dd:yyyy-mm-dd> format: Specify dates from which Voyager will extract entries from the applicable table in the database (for example, -r1997-01-01:1998-03-15 will extract every entry from 1/1/97 through 3/15/98).
- -r<today-n> format: Specify the number of days before the current date for which Voyager will extract entries (for example, -rtoday-7 will extract entries for the current date and the 7 days previous). Excluding the -n switch will set the date to the current day.

-p Purges the applicable table in the database of the specified entries after saving them to the export file. (Recommended)

-q Quiet mode - will not prompt for any input.

- v Provides version information.
- h Provides help for the program.

After running the program:

Data extracted from the `bib_use_log` table is placed in the `/rpt` directory in the file `bibuselgexport.dat` or in the file specified after the `-o` parameter. The log file for this operation is named `bibuselgexport.log` or uses the filename specified after the `-o` parameter with the file extension of `.log`. A file in the `/log` directory called `log.voyager` gets updated with a record of the extraction operation.

Acquisitions Batch Job - Fix Exchange Rates

17

Introduction

Acquisitions batch job 5, Fix Exchange Rates (formerly Pfixexchangerates script), updates the commitments (encumbrances) on selected Purchase Orders based on the currency conversion rates in the system at the time this job is run. Commitments are monies that are subtracted from the available fund total when an order is approved, but before it has been paid for. This job updates the commitments if the currency conversion rates have changed since the PO was approved.

The fund's commitments, total and pending, are updated as well.

Sites using foreign currencies, where conversion rates vary, may want to use this feature.



IMPORTANT:

This job is not reversible. Once run, the commitments are adjusted permanently.

Fix Exchange Rates

The Fix Exchange Rates job updates commitments on Purchase Orders that use foreign currencies, providing a more accurate picture of commitments as conversion (exchange) rates fluctuate.

Purchase Orders Updated with Acqjob 5

When the batch job is run, the system considers the following criteria when determining which POs are adjusted by this job:

- The Purchase Orders must be in a currency other than the base currency.
- The Purchase Orders must have a status of Received Partial or Approved/Sent.

For each PO that meets the above criteria, the system adjusts all outstanding commitments, omitting line-item copies that are covered by prepaids or have already been invoiced.

A currency rate fluctuation may cause funds to be over committed. If this occurs, the details are reported in the log file. Also, this information is available in the various fund workspaces of the Acquisitions module.

NOTE:

Purchase Orders with a status of Pending, Received Complete, Complete, and Canceled are not considered for adjustment by this job. If a PO has a status of Pending, the system automatically adjusts currency rates when the PO is opened in the Acquisitions client.

Additional Considerations for Acqjob 5

The user can limit the POs considered for adjustment by specifying any of the following criteria on the command line: Currency Codes, Ledger Names, Order Types, and Vendor Codes.

[Table 17-1](#) lists the optional parameters that the user can specify individually or in tandem with the `Pacqjob -j5` batch job.

Table 17-1. Acqjob 5 Batch Job Parameters

Parameter	Description
<code>-c<currency_codes></code>	This optional parameter limits the adjustments of POs to specific currency codes. Specify a single currency code or a comma-separated list of currency codes in the following format: <pre>-c "EUR, CND"</pre>
<code>-m<fund_codes></code>	This optional parameter limits the adjustments of POs to specific ledgers. Specify a single fund code or a comma-separated list of fund codes in the following format: <pre>-m "FY05, Fiscal2004"</pre>
<code>-p<vendor_codes></code>	This optional parameter limits the adjustments of POs to specific vendors. Specify a single vendor code or a comma-separated list of vendor codes in the following format: <pre>-p "EBSCO, B&TA"</pre>
<code>-t<order_types></code>	This optional parameter limits the adjustments of POs to specific Order Types. Specify a single order type or a comma-separated list of order types in the following format: <pre>-t "Approval, FirmOrder"</pre>

Running Acqjob 5

This job can be run in the following ways:

- Entering full command line on the server,
- Using the command line interactively on the server.

- Using WebAdmin, see [Acquisitions Utilities](#) on [page 31-9](#)
- Adding the job to a cron on the server.

Running Acqjob 5 from the Command Line

To run this job from the command line, enter the command along with required and optional parameters from the `/ml/voyager/xxxdb/sbin>` directory. For example, the following command runs the adjustment for all Approved/Received Partial POs that use a foreign currency:

```
Pacqjob -j5
```

If the job is accepted, the screen displays a message, `FixExchangeRates` is running. When the job is finished, the `COMPLETED` message displays, followed by the command-line prompt.

Running Acqjob 5 Interactively

To run this job interactively at the `/ml/voyager/xxxdb/sbin>` prompt, enter the following batch job command without parameters:

```
Pacqjob
```

After the system prompts you to select the batch job (see [Figure 17-1](#)), enter:

```
5
```

```

/m1/voyager/xxxdb/sbin $ Pacqjob
Initializing...
1 -- Order Claim/Cancel Processing
2 -- Open Order Report
3 -- Fund Snapshot Report -- Global Distribution
4 -- Rollover Status Report -- Global Distribution

5 -- Fix Exchange Rate
99 -- Quit

Process Job #?          5
FixExchangeRate is running...
Processing begins...
(*) Finished processing 8 purchase orders.
(*) 1 successfully reconciled; 0 failed.
...COMPLETED
/m1/voyager/xxxdb/sbin $

```

Figure 17-1. Screen Display for Acqjob 5 (Fix Exchange Rates)

NOTE:

The optional parameters that are listed in [Table 17-1](#) on [page 17-3](#) cannot be used in interactive mode.

The system executes the job which has been configured with your database name, username, and password. The screen displays the number of POs considered for update, the number of PO updated, and the number of POs that failed. When completed, the system displays the `COMPLETED` message.

NOTE:

If you want to run a server activity in the background and you do not want to perform it interactively, enter

nohup <activity with parms> &

to make certain that the activity continues, even if the telnet session is lost, where `<activity with parms>` is the command followed by any required and optional parameters. This command will not guarantee that the activity will be completed if the server itself is not available.

Log File

Each time the `Pacqjob -j5` batch job is run, it creates a log file named, `log.foreigncommitments.yyyyymmdd.hhmm`, where *yyyy* is year, *mm* is month, *dd* is day, *hh* is hour, and *mm* is minutes. It is stored in the `/m1/voyager/xxxxdb/rpt` directory as a text file.

This log ([Figure 17-2](#)) can be examined to see changes in individual orders to monitor the fluctuation of currency rates. The fund records in the Acquisitions module can be used to see the current total commitments.

The log includes the following information.

- PO number of PO considered for update
- PO Header information for a PO that is updated
 - PO Number
 - Currency code
 - PO type
 - Status
 - Vendor name
- Line item information
 - Fund
 - Price in foreign currency
 - Conversion rate before adjustment
 - Commitment amount in the base currency before adjustment
 - Conversion Rate after adjustment
 - Commitment amount in the base currency after the adjustment.

NOTE:

If the new commitment causes the fund's total commitments to exceed its limit, this amount is annotated "OVERCOMMIT".


```

PO_ID# 994

000994 0000 000000 0 -----
000994 0000 000000 1 PO number: 994 PO total: GBP 10.00
000994 0000 000000 2 Type: Firm Order Status: Approved/Sent
000994 0000 000000 3 Vendor: Baker & Taylor|

000994 0098 000605 4 * Fund: 04 Test Ledger/04 Test Fund
000994 0098 000605 5 Before @ 0.89000 $ 11.24

000994 0098 000605 4 * Fund: 04 Test Ledger/04 Test Fund
000994 0098 000605 6 After @ 0.75000 $ 13.33 *OVERCOMMIT*

PO_ID# 1000

PO_ID# 1001

PO_ID# 1002

PO_ID# 1004

```

Figure 17-2. Foreign Commitments Log File

The `acqjob.log` file lists the date and time the job began and ended (see [Figure 17-3](#)).

```

Thu Sep 29 15:18:33 2005 FixExchangeRate is running...
Thu Sep 29 15:18:33 2005 ...COMPLETED

```

Figure 17-3. Acqjob.log file

Introduction

This chapter describes the set up information for features that require configuration changes on the server.

UB Barcode Lookup

The UB Barcode Lookup feature enables the Circulation client to automatically determine the Circulation cluster of patrons and items in Universal Borrowing (UB) or multi-cluster environments during charges, discharges, and patron searches - reducing the effort required by circulation clerks to determine the Circulation cluster of the patron or item.

Configuration files are used to enable the feature and to define the barcode patterns that the system uses to determine which patron/item barcodes are associated with a particular cluster or database within a UB or multi-cluster environment.

Configuring the UB Barcode Lookup Feature

The following files are used to configure the UB Barcode Lookup feature:

- `voyager.env` file
- UB Barcode Lookup configuration file

- UB Barcode Lookup schema file

voyager.env

The `UB_BARCODE_CONFIG` environment variable enables the auto-selection of local clusters for patron and item barcodes during charges, discharges, and patron searches. By default, the `UB_BARCODE_CONFIG` environment variable appears as follows in the `/m1/voyager/xxxdb/ini/voyager.env` file:

```
#export UB_BARCODE_CONFIG="$VOYAGER/$DATABASE/ini/
ub_barcode_config.xml"
```

To enable auto-selection of barcodes, uncomment the line and modify the path and file name if different from the default. If the configuration file has not been created, the feature will not become active until the configuration file is created and the Circulation module is restarted.

Creating the UB Barcode Lookup Configuration File

The UB Barcode Lookup Configuration file contains the cluster, database, and barcode patterns that the system uses to determine the local cluster of a patron or item barcode during charges, discharges, and patron searches from the Circulation client.

It is recommended that the configuration file be created and maintained by the consortium administrator. A copy of this file must be installed and enabled via the `voyager.env` file for each database in the consortium.

Before creating the configuration file, verify that all UB and cluster data is configured properly in the database tables (**Voyager_Databases** and **Remote_Circ_Cluster_Cache**) since they will be used to create a skeleton configuration file (see [Figure 18-1](#) for an example).



Procedure 18-1. Creating the UB Barcode Lookup Configuration File

To create the UB Barcode Lookup Configuration file, perform the following steps:

1. Enter the following command from the `/m1/voyager/xxxdb/sbin` directory:

```
./PgenUBBarcodeConfig
```

Result: If the `ub_barcode_config.xml` file already exists in the `/ml/voyager/xxxdb/ini` directory, you will be asked to create a new file. Otherwise, a skeleton configuration file is created (see [Figure 18-1](#)).

2. Open the file you created in [Step 1](#).
3. Check to see if any database or cluster names contain ampersands. If any contain ampersands, change each ampersand to an “and” to prevent syntax errors.
4. For each `<Patron></Patron>` block and `<Item></Item>` block, enter a list of barcode patterns (regular expressions) that the system will use to distinguish the Circulation cluster of patron or item barcode. See [Table 18-1](#) on [page 18-5](#) for more information on the `<BarcodePattern>` field.

NOTE:

If the same pattern is used for more than one cluster, the system will return multiple matches while using this feature in the Circulation module.

[Figure 18-1](#) shows a sample configuration file that has been created using the `PgenBarcodeConfig` command. Note that the `<Patron></Patron>` and `<Item></Item>` blocks are empty and must be filled in manually. See [Table 18-1](#) on [page 18-5](#) for more information.

```
<?xml version="1.0" encoding="UTF-8"?>
<Consortium
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="schema/ub_barcode_config.xsd">
  <Flag>
    <TryLocalClusterFirst>true</TryLocalClusterFirst>
    <UseFirstMatchingPatronCluster>>false</
UseFirstMatchingPatronCluster>
    <UseFirstMatchingItemCluster>>false</
UseFirstMatchingItemCluster>
  </Flag>
  <Cluster>
    <DatabaseName>QAMK UB/Cluster Database</DatabaseName>
    <DatabaseKey>QA20050DB20020613131313</DatabaseKey>
    <ClusterName>QAMK Cluster 1</ClusterName>
    <ClusterId>1</ClusterId>
    <Patron>
    </Patron>
    <Item>
    </Item>
  </Cluster>
```

```
<Cluster>
  <DatabaseName>QAMK UB/Cluster Database</DatabaseName>
  <DatabaseKey>QA20050DB20020613131313</DatabaseKey>
  <ClusterName>QAMK Cluster 2</ClusterName>
  <ClusterId>2</ClusterId>
  <Patron>
  </Patron>
  <Item>
  </Item>
</Cluster>
</Consortium>
```

Figure 18-1. Skeleton UB Configuration File

[Table 18-1](#) describes the fields used in the UB Barcode Lookup Configuration file.

Table 18-1. UB Barcode Lookup Configuration File Commands

Tag Name	Description
<Flag>	This field is used to define initial criteria that is considered by the system to determine the local cluster of a patron or item. It contains the following subfields: • <TryLocalClusterFirst> • <UseFirstMatchingPatronCluster> • <UseFirstMatchingItemCluster> XPath: Consortium/Flag
<TryLocalClusterFirst>	If set to <code>true</code>, this flag indicates that the system will search the local cluster for a match first. If no match is found, the system compares the patron or item barcode to the barcode patterns specified in the configuration file to determine the cluster. If a matching pattern is found, the system uses the cluster that contains the matching pattern; otherwise, the system prompts the user to select a cluster from a drop-down menu. If set to <code>false</code>, the system uses the <UseFirstMatchingPatronCluster> or <UseFirstMatchingItemCluster> flags to find the local cluster of the patron/item. NOTE: This flag has precedence over the other flags. Default: <code>true</code> XPath: Consortium/Flag/TryLocalClusterFirst
<UseFirstMatchingPatronCluster>	If set to <code>true</code>, this flag indicates that the system compares the patron barcode with the patron barcode patterns in the configuration file. If a match is found, the system tries the first matching cluster. If set to <code>false</code>, the system compares the patron barcode with the patron barcode patterns in the configuration file. If multiple matches are found, the system prompts the user to select a cluster from a drop-down menu. NOTE: The <TryLocalClusterFirst> flag has precedence over this flag. Default: <code>false</code> XPath: Consortium/Flag/UseFirstMatchingPatronCluster

Table 18-1. UB Barcode Lookup Configuration File Commands

Tag Name	Description
<UseFirstMatchingItemCluster>	If set to <code>true</code>, this flag indicates that the system compares the item barcode with the item barcode patterns in the configuration file. If a match is found, the system tries the first matching cluster. If set to <code>false</code>, the system compares the item barcode with the item barcode patterns in the configuration file. If multiple matches are found, the system prompts the user to select a cluster from a drop-down menu. NOTE: The <TryLocalClusterFirst> flag has precedence over this flag. Default: <code>false</code> XPath: Consortium/Flag/UseFirstMatchingItemCluster
<Cluster>	This field is used to define the barcode patterns for each cluster in the UB or multi-cluster environment. It contains the following sub-fields: • <DatabaseName> • <DatabaseKey> • <ClusterName> • <ClusterID> • <Patron> • <Item> NOTE: The system automatically populates the cluster field for each cluster defined in the UB or multi-cluster environment. The consortium administrator will need to add the barcode patterns manually. XPath: Consortium/Cluster
<DatabaseName>	For each cluster, the system automatically populates the database name during creation of the configuration file. Data Source: VOYAGER_DATABASES.db_name XPath: Consortium/Cluster/DatabaseName

Table 18-1. UB Barcode Lookup Configuration File Commands

Tag Name	Description
<DatabaseKey>	For each cluster, the system automatically populates the database key during creation of the configuration file. Data Source: VOYAGER_DATABASES.db_key XPath: Consortium/Cluster/DatabaseKey
<ClusterName>	For each cluster, the system automatically populates the cluster name during creation of the configuration file. Data Source: REMOTE_CIRC_CLUSTER_CACHE.cluster_name XPath: Consortium/Cluster/ClusterName
<ClusterID>	For each cluster, the system automatically populates the cluster ID during creation of the configuration file. Data Source: REMOTE_CIRC_CLUSTER_CACHE.remote_circ_cluster_id XPath: Consortium/Cluster/ClusterID
<Patron>	For each cluster, the system automatically populates an empty <Patron> </Patron> block, which is used to manually define the barcode patterns that the system uses to determine the local cluster of the patron barcode. XPath: Consortium/Cluster/Patron
<BarcodePattern> (patron)	A list of regular expressions that represent the barcode patterns for each database/cluster. <pre> <Patron> <BarcodePattern>66.*</BarcodePattern> <BarcodePattern>10301.*</BarcodePattern> </Patron> </pre> XPath: Consortium/Cluster/Patron/BarcodePattern
<Item>	For each cluster, the system automatically populates an empty <Item> </Item> block, which is used to manually define the barcode patterns that the system uses to determine the local cluster of the item barcode. XPath: Consortium/Cluster/Item
<BarcodePattern> (item)	A list of regular expressions that represent the barcode patterns for each database/cluster. <pre> <Item> <BarcodePattern>CL.*</BarcodePattern> </Item> </pre> XPath: Consortium/Cluster/Item/BarcodePattern

UB Barcode Lookup Configuration Schema File

If an XML editor is used to create or modify the UB Barcode Lookup Configuration file, the text (as shown in [Figure 18-2](#)) can be copied and pasted into a file (schema/ub_barcode_config.xsd) on the server to validate the fields in the configuration file as it is being edited.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
>
  <xs:element name="Consortium">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Flag">
          <xs:complexType>
            <xs:sequence>
              <xs:element
                name="TryLocalClusterFirst"
                type="xs:boolean"
                default="true"
              />
              <xs:element name="UseFirstMatchingPatronCluster"
                type="xs:boolean"
                default="false"
              />
              <xs:element
                name="UseFirstMatchingItemCluster"
                type="xs:boolean"
                default="false"
              />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="Cluster" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
```

Figure 18-2. UB Configuration Schema File (Continued on next page)

```

<xs:element
  name="DatabaseName"
  type="xs:string"
  minOccurs="0"
>
  <xs:annotation>
    <xs:documentation>
      Populated from
      VOYAGER_DATABASES.db_name
    </xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element
  name="DatabaseKey"
  type="xs:string"
>
  <xs:annotation>
    <xs:documentation>
      Populated from
      VOYAGER_DATABASES.db_key
    </xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element
  name="ClusterName"
  type="xs:string" minOccurs="0"
>
  <xs:annotation>
    <xs:documentation>
      Populated from
      REMOTE_CIRC_CLUSTER_CACHE.remote_circ_
        cluster_name
      or CIRC_CLUSTER.circ_cluster_name
    </xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element
  name="ClusterId"
  type="xs:string"
>
  <xs:annotation>
    <xs:documentation>
      Populated from

```

Figure 18-2. UB Configuration Schema File (Continued on next page)

```
        REMOTE_CIRC_CLUSTER_CACHE.remote_
            circ_cluster_id
        or CIRC_CLUSTER.circ_cluster_id
    </xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="Patron" minOccurs="0">
    <xs:complexType>
        <xs:sequence minOccurs="0" maxOccurs=
            "unbounded">
            <xs:element ref="BarcodePattern"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="Item" minOccurs="0">
    <xs:complexType>
        <xs:sequence minOccurs="0" maxOccurs=
            "unbounded">
            <xs:element ref="BarcodePattern"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="BarcodePattern">
    <xs:annotation>
        <xs:documentation>
            POSIX compatible regular expression
        </xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:simpleContent>
            <xs:extension base="xs:normalizedString"/>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>
</xs:schema>
```

Figure 18-2. UB Configuration Schema File

Configuration File Validation

The format and consistency of the configuration file (`ub_barcode_config.xml`) is validated each time the Circulation module starts up and attempts to read the barcode patterns from the UB Barcode Lookup configuration file.

If the configuration file contains format errors, an entry is placed in the `/ml/voyager/xxxdb/log/log.circsvr` file. For example, [Figure 18-3](#) shows a log entry which indicates that an invalid value, `flase`, has been entered for a field on Line 6 of the configuration file.

```
circsvr[4694] - ERROR - Thu Oct 27 14:16:28 2005
- LoadBarcodeTables - BarcodePatternTable.cpp[454]
- ReqCirculationDefaults
- vSession::route_request
- vSession::Run
SAXParseException:  Expecting true or false; got flase
System ID:  /ml/voyager/xxxdb/ini/
ub_barcode_config.xml
    Public ID:
        Line:  6
        Column: 39
```

Figure 18-3. UB Configuration Log File

Dynamic Noise Word Reduction

Dynamic noise word reduction improves the performance of keyword searches, particularly in large databases. It eliminates common words from a user's query for the purposes of searching but retains them for the purposes of relevance ranking.

In the server-side `voyager.ini` file, set `NOISEWORDFILTER=Y` to turn on this feature. The default is `NOISEWORDFILTER=N`.

The following is a list of stop words used with this feature.

- AND
- OR
- NOT
- OF
- IN
- THE
- WITH
- TO
- FOR

If the Dynamic Noise Word Reduction feature is enabled, the behavior of keyword searching depends on the boolean operator used in the search. [Table 18-2](#) shows the behavior of boolean searches when this feature is enabled.

Table 18-2. NOISEWORDFILTER and Keyword Searches

String	Search Code	Enabled?	Translation
mice and men	GKEY	N	(GKEY mice) OR (GKEY and) OR (GKEY men)
mice and men	GKEY	Y	(GKEY mice) OR (GKEY men) The “and” is stripped out of the search.
mice and men	GKEY^	N	(GKEY mice) AND (GKEY and) AND (GKEY men)
mice and men	GKEY^	Y	NOISEWORDFILTER is ignored.

Configuring the Z39.50 Server for UTF-8 Encoded Records

By default, the Voyager Z39.50 Server sends MARC-8 encoded records to remote clients that request records via the z39.50 protocol.

With the Unicode release, UTF-8 encoded records can be sent also by configuring the Z39.50 server to handle UTF-8 and MARC-8 encoded records on separate ports. Traditionally, port `xxx90` has been used for MARC-8 encoded records, where `xx` represents the first two digits of your Voyager port. Ex Libris suggests that you use port `xxx91` for UTF-8 encoded records.



Procedure 18-2. Configuring the Z39.50 Server to Send Both MARC-8 and UTF-8 Encoded Records

Use the following procedure to configure the Voyager Z39.50 server to send both MARC-8 and UTF-8 encoded records.

1. Open the `/etc/services` file, locate the Z39.50 port assignment, and insert the following assignment to support UTF-8 encoded messages:

```
xxxdb/Pz3950svr    xx91    # Z39.50 Server UTF8
```

NOTE:

If port `xx91` is already being used, use another port, such as `xxx95`.

2. Exit and save the changes to the `/etc/services` file.
3. Copy the `/m1/voyager/xxxdb/ini/z3950svr.ini` file and save it as `z3950svrUTF8.ini` under the same directory.
4. Open the `/m1/voyager/xxxdb/ini/z3950svrUTF8.ini` file and perform the following steps:
 - a. Change the port assignment of the the `ZPort` parameter to the port used in Step 1 as follows:

```
ZPort=xxx91
```

- b. Modify the `Encoding` parameter of the `[Default]` stanza to use the UTF-8 format as follows:

```
Encoding=UTF8
```

- c. Exit and save the changes to the `z3950svrUTF8.ini` file.

Result: The modifications to the `z3950svrUTF8.ini` file appear as shown in [Figure 18-4](#).

```
# What TCP port number for the server?
ZPort=14091
DBname=VOYAGER
.
.
.

# Encoding
# This is either UTF8 or it defaults to Marc
Encoding=A
```

Figure 18-4. Sample `z3950svrUTF8.ini` File Modifications

5. Open the `/m1/voyager/xxxdb/sbin/Pz3950svr` script file and perform the following steps:
 - a. Insert lines of code into the file (as indicated by the bold text in [Figure 18-5](#)) to run a separate process for the UTF-8 encoded messages.

NOTE:

Use the .ini file created in Step 3 and specify a different log file for UTF-8 encoded transactions.

- b. Exit and save the changes to the `Pz3950svr` script file.

```
if [ $OS = 'Windows_NT' ]; then
    exec $VOYAGER_BIN/z3950svr -i $VOYAGER/$DATABASE/ini/z3950svr.ini \
    -c $VOYAGER/$DATABASE/ini/voyager.ini \
    > $VOYAGER/$DATABASE/log/z3950svr.nohup.out &
else
    /bin/nohup $VOYAGER_BIN/z3950svr -i $VOYAGER/$DATABASE/ini/z3950svr.ini \
    -c $VOYAGER/$DATABASE/ini/voyager.ini \
    > $VOYAGER/$DATABASE/log/z3950svr.nohup.out &
    /bin/nohup $VOYAGER_BIN/z3950svr -i $VOYAGER/$DATABASE/ini/z3950svrUTF8.ini \
    -c $VOYAGER/$DATABASE/ini/voyager.ini \
    > $VOYAGER/$DATABASE/log/z3950svrUTF8.nohup.out &
fi
```

Figure 18-5. Sample Pz3950svr Script File Modifications

6. Restart the Z39.50 server.
-

Introduction

The Patron Record Standard Interface File (SIF) is the format of a file that is used in three ways in Voyager

- Input file for the Patron load done at migration
- Input file for the Patron Update program
- Output of the Patron Extract program.

Purpose of This Chapter

This section discusses the layout of the Patron Record SIF.

- File specification
- File format fixed segment
- File format address segment
- File format variable segment
- File format end-of-record segment

File Specification

The data required for the Patron database records should be supplied in the following manner.

Each patron data record should be one continuous string, terminated by a line-feed (\n in some formats, hex '0A' in others).

The fields must be the indicated length and must all be present and a specific order and format. The notes segment is an exception to the length requirement.

The data fields which are designated numeric must be right-justified and zero-filled on the left.

The data fields which are designated strings (not designated numeric) should be left-justified and blank-filled on the right but not hex-null-terminated.

If the export data is in a continuous stream, you must insert a hex null terminator between records. However, if the export data is broken into one line per record, the null terminator does not need to be inserted.

Patron Record SIF Format

The following conventions are used in the listed tables.

Item # Represents the relative position of the item in the record (sequence).

NOTE:

In all the following tables, the first column, Item #, consists of the field sequence numbers which are not to be included in the record.

Item name Definition of the item.

Offset Actual starting position of the item in the record relative to 1.

Format Format of the variable.

s String, left-justified/blank filled.

n Numeric, right-justified/zero-filled.

d Date in the format ccyy.mm.dd, where c is the century, y is the year, m is the month, and d is the day. The maximum high value that can be used is 2382.12.31. This field may be blank under certain conditions.

b Blanks.

Required Y for yes, the field must have valid data in it (not just blanks).

NOTE:

The Social Security Number (SSAN) or the Institution ID is required for all records.

The maximum record should consist of the following:

1 base segment length: $1 \times 456 = 456$

9 address segments length: $9 \times 429 = 3861$

1 notes segment length: $1 \times 1000 = 1000$

1 end-of-record mark length: $1 \times 1 \text{ or } 2 = 1 \text{ or } 2$

The total maximum record length is 5318 or 5319.

The minimum record should consist of the following:

1 base segment length: $1 \times 456 = 456$

1 address segment length: $1 \times 429 = 429$

1 notes segment length: $1 \times 1 = 1$

1 end-of-record mark length: $1 \times 1 \text{ or } 2 = 1 \text{ or } 2$

The total minimum record length is 887 or 888.

Base Segment

[Table 19-1](#) on [page 19-4](#) defines the base (fixed) segment of all records.

Also, a patron record must have one set, but can have up to three sets of the following fields: patron barcode, patron group, barcode status, and modification date. However, if one of the above fields exists in a set that has data, the other three fields must also have data. The exception to this is the patron barcode, which can be blank.

If more than one patron barcode is present, the associated patron groups must be unique.

NOTE:

The maximum lengths of some of the address and name fields in the database are longer than allowed in the Patron Record SIF. Consequently, these fields will be truncated in the SIF. If you are concerned that some data may exceed the lengths allowed in the SIF, use the XML jobs instead. For more information, see [Patron Batch Jobs in XML](#) on [page 5-1](#).

Table 19-1. Base (Fixed) Segment of Record

Item #	Item Name	Offset	Format	Required	Length	Description
01	patron id	1	n		10	Internal record number for Voyager. This field should be zero-filled. The patron ID will be determined by the SSAN or Institution ID (whichever is unique) and the patron record will be updated if either is found, or added if not.
02	patron bar-code id 1	11	n*		10	ID of first library card barcode.
03	patron bar-code 1	21	s		25	First library card barcode. Footnote 1
04	patron group 1	46	s		10	Valid patron group as defined by library in system administration. Footnote 2
05	barcode status 1	56	n*		1	Valid: 1 is active. 2 is lost. 3 is stolen. 4 is expired. 5 is other.
06	barcode modified date 1	57	d		10	If blank, load run-date will be used.
07	patron bar-code id 2	67	n*		10	ID of second library card barcode.
08	patron bar-code 2	77	s		25	Second library card barcode. Footnote 1

Table 19-1. Base (Fixed) Segment of Record

Item #	Item Name	Offset	Format	Required	Length	Description
09	patron group 2	102	s		10	Valid patron group as defined by library in the System Administration module.
10	barcode status 2	112	n		1	Valid: 1 is active. 2 is lost. 3 is stolen. 4 is expired. 5 is other.
11	barcode modified date 2	113	d		10	If blank, load run-date will be used.
12	patron bar-code id 3	123	n*		10	ID of third library card bar-code.
13	patron bar-code 3	133	s		25	Third library card barcode. Footnote 1
14	patron group 3	158	s		10	Valid patron group as defined by library in the System Administration module.
15	barcode status 3	168	n*		1	Valid: 1 is active. 2 is lost. 3 is stolen. 4 is expired. 5 is other.
16	barcode modified date 3	169	d		10	If blank, load run-date will be used.
17	registration date	179	d		10	The date the record was added to an external system such as administration or human resources.

Table 19-1. Base (Fixed) Segment of Record

Item #	Item Name	Offset	Format	Required	Length	Description
18	patron expiration date	189	d	y	10	Date patron record is to be considered no longer active by the Voyager system. Standard date format.
19	patron purge date	199	d	y	10	Date patron record is to be deleted by the Voyager system. Standard date format.
20	voyager date	209	b		10	The run-date of the load will be used if this is a new patron.
21	voyager updated	219	b		10	The run-date of the load will be used.
22	circulation happening location code	229	s		10	This is the location code for any circulation happening location in the Circulation Policy Group to which the patron belongs. When entering patrons manually, it is the location code of the desk where the patron was added. It must be a valid location code.
23	institution ID	239	s	Foot- note3	30	Any value. If the ID came from the registrar or from human/resources and if it is important to know the source, the indicator should be part of this field (a prefix of reg or h/r).
24	ssn	269	s	Footnote 3	11	Patron's social security account number: with or without dashes.

Table 19-1. Base (Fixed) Segment of Record

Item #	Item Name	Offset	Format	Required	Length	Description
25	statistical category 1	280	s		3	Valid statistical category code as defined by library in the System Administration module. If this is an update to an existing patron record and there is an entry in this field, all existing statistical category records will be removed from the database and replaced with those found in this record.
26	statistical category 2	283	s		3	See statistical category 1.
27	statistical category 3	286	s		3	See statistical category 1.
28	statistical category 4	289	s		3	See statistical category 1.
29	statistical category 5	292	s		3	See statistical category 1.
30	statistical category 6	295	s		3	See statistical category 1.
31	statistical category 7	298	s		3	See statistical category 1.
32	statistical category 8	301	s		3	See statistical category 1.
33	statistical category 9	304	s		3	See statistical category 1.
34	statistical category 10	307	s		3	See statistical category 1.
35	name type	310	n	y	1	Valid data: 1 is personal, the name is an individual. 2 is institutional, the name may be a department or organization.

Table 19-1. Base (Fixed) Segment of Record

Item #	Item Name	Offset	Format	Required	Length	Description
36	surname	311	s	y	30 (50 in DB)	Patron's last name or institution name. Include any suffix (Jr. or PH.D.).
37	first name	341	s		20 (50 in DB)	Patron's first name. Required for name type 1.
38	middle name	361	s		20 (50 in DB)	Patron's middle name or initial.
39	title	381	s		10	Patron's title, if any.
	transaction counters	--	--	--	--	The next 10 fields are transaction counters.
40	historical charges	391	n		10	Total number of charges for patron's entire history. Maximum value is 2147483648.
41	claims returned count	401	n		5	Total number of claims returned for patron's entire history. Maximum value is 32767.
42	self-shelved count	406	n		5	Total number of items self-shelved for patron's entire history. Maximum value is 32767.
43	lost items count	411	n		5	Total number of lost items for patron's entire history. Maximum value is 32767.
44	late media returns	416	n		5	Total number of late media returns for patron's entire history. Maximum value is 32767.
45	historical bookings	421	n		5	Total number of historical bookings for patron's entire history. Maximum value is 32767.

Table 19-1. Base (Fixed) Segment of Record

Item #	Item Name	Offset	Format	Required	Length	Description
46	canceled bookings	426	n		5	Total number of canceled bookings for patron's entire history. Maximum value is 32767.
47	unclaimed bookings	431	n		5	Total number of unclaimed bookings for patron's entire history. Maximum value is 32767.
48	historical callslips	436	n		5	Number of historical call slips for this patron.
49	historical distributions	441	n		5	Total number of historical distributions for this patron's entire history.
50	historical short loans	446	n		5	Total number of historical short loans for this patron's entire history.
51	unclaimed short loans	451	n		5	Total number of unclaimed short loans for this patron's entire history.
52	address count	456	n	Y	1	Valid data: 1 through 9. There must be at least 1 address segment (type 1: permanent) following. There may be as many as 8 additional addresses (2 through 9) which may be either type 2: temporary, or 3: e-mail.

* Numeric if present, blank (not zero filled) otherwise.

1 If any one of the following are defined: patron barcode, patron group, barcode status, and modification date, the remaining three fields must also be defined (except the patron barcode, which can be blank to allow for a blank barcode patron). If a barcode already exists, it will be updated with this information. If a barcode exists for this patron with the same patron group, the status on the existing barcode will be changed to 5 (other) and the new one will be added with the status 1 (active), except where the input barcode is blank and the patron group and barcode status match an existing record. If this is the case, no alterations are made.

2 If there is more than one patron barcode present, the associated patron groups must be unique. At least one patron group must be present.

3 For each patron, either the social security number or the institution ID is required and that number must also be unique to the current record. There can be no duplicates. The social security number will be used in update runs to identify the record. If it is absent, the institution ID will be used in its place. Only one of these two fields may be altered on an update as the other will be required to identify the record in the database.

Address Segment

[Table 19-2](#) defines the address portion of the record. This segment is repeatable up to nine times consecutively. All address segments must follow the base segment. Note that for the address segment, these sequence numbers might vary, according to the number of the address records.

Table 19-2. Address Segment of Record.

Item #	Item Name	Offset	Format	Required	Length	Description
53	address id	457	n		10	Internal record number for Voyager.
54	address type	467	n	y	1	Valid data: 1 = permanent -- only one is permitted 2 = temporary 3 = e-mail Permanent address (type = 1) update: If existing permanent address has the protect_address flag = y, an error will be reported. If not, the address will be updated with the new data. Temporary/e-mail address update: Any existing temporary/e-mail addresses without address protection will be deleted from the database and replaced by the new type 2 and 3 addresses supplied. Type 2 and 3 addresses with address protection will not be deleted. All addresses added/updated will default to not protected.

Table 19-2. Address Segment of Record.

Item #	Item Name	Offset	Format	Required	Length	Description
55	address status code	468	s	y	1	Valid data: n = normal h = hold mail The next two items represent the beginning and ending effective dates for the address. For e-mail and temporary addresses this date will be rejected if it overlaps dates of another address of the same type.
56	address begin date	469	d	y	10	Beginning date.
57	address end date	479	d	y	10	Ending date. Beginning and Ending Dates can <i>not</i> be the same date.
58	address line 1	489	s	y	50 (100 in DB)	Street address, or e-mail address for address type 3 (cannot be blank).
59	address line 2	539	s		40 (100 in DB)	Second line of street address.
60	address line 3	579	s		40 (100 in DB)	Third line of street address.
61	address line 4	619	s		40 (100 in DB)	Fourth line of street address.
62	address line 5	659	s		40 (100 in DB)	Fifth line of street address.
63	city	699	s		40	City
64	state (province) code	739	s		7	Any 2-character state or 7-character province code.

Table 19-2. Address Segment of Record.

Item #	Item Name	Offset	Format	Required	Length	Description
65	zipcode/postal code	746	s		10	Zipcode in format 12345 or 12345-6789 or other postal code.
66	country	756	s		20	Country: any format is valid.
67	phone (primary)	776	s		25	Primary phone: any format is valid.
68	phone (mobile)	801	s		25	Mobile phone: any format is valid.
69	phone (fax)	826	s		25	Fax phone: any format is valid.
70	phone (other)	851	s		25	Other phone: any format is valid.
71	date added/updated	876	b		10	This will be set to the run-date of the load.

Variable Segment

[Table 19-3](#) defines the variable segment of the record. This segment must follow the last address segment. The segment is not required and may be between 0 (zero) and 1000 characters in length. A segment of zero characters (a non-existent segment) indicates that there are no notes for the record.

NOTE:

If the user is performing a Patron Update and extended notes processing is selected, then each note in the Notes segment, must be preceded by a tab character. This includes the first note.

Table 19-3. Variable (Notes) segment of record

Item #	Item Name	Offset	Format	Required	Length	Description
72	Notes	After Last Address	s		1000 Max.	Variable Portion of Record Patron notes. This field may contain any data. A field of zero characters in length indicates that there are no notes.

End-of-Record Segment

[Table 19-4](#) defines the end-of-record marker. This segment must follow the notes segment.

Table 19-4. End-of-record marker

Item #	Item Name	Offset	Format	Required	Length	Description
73	End-of-Record Marker	Last	s	y	1 or 2	Valid data: Line-feed (\n or hex '0A'). A carriage-return/line-feed is also acceptable.

Introduction

The Charge Transaction Record Standard Interface File (SIF) is the format of the file that contains your site's charge transactions that are loaded during migration.

Purpose of This Chapter

This section discusses the layout of the Charge Transaction Record SIF.

- Input File specification
- File format fixed segment

Input File Specification

The data required for the Charge Transaction Records should be supplied in the following manner.

Each charge transaction record should be one continuous string, terminated by a line-feed (\N).

The fields must be the indicated length and must all be present and in a specific order and format.

The data fields which are designated numeric must be right-justified and zero-filled on the left.

The data fields which are designated strings (not designated numeric) should be left-justified and blank-filled on the right but not hex-null-terminated.

The date format should be entered as CCYY.MM.DD and the Time format should be entered as HH:MM, where C is century, Y is year, M is month, D is day, H is hour, and S is second.

The data fields which are supplied (not blanks or zeroes) will be updated if they already exist or added if they do not.

Charge Transaction Record SIF Format

The following conventions are used in the listed tables.

Item # Represents the relative position of the item in the record (sequence).

NOTE:

In all the following tables, the first column, Item #, consists of the field sequence numbers which are not to be included in the record.

Item name Definition of the item.

Type Definition of the type of input.

Offset Actual starting position of the item in the record relative to 1.

Format Format of the variable.

s String, left-justified/blank filled.

n Numeric, right-justified/zero-filled.

d Date in the format ccyy.mm.dd, where c is the century, y is the year, m is the month, and d is the day. The maximum high value is 2382.12.31 which can be used no expiration. This field may be blank under certain conditions.

t Time in the format HH:MM.

b Blanks.

Required Y for yes, the field must have valid data in it (not just blanks).

[Table 20-1](#) defines the format required for all incoming records.

Table 20-1. Charge Transaction Standard Interface File Format

Item #	Item Name	Type	Offset	Format	Required	Length	Description
01	item barcode	varchar	1	s	Y	25	Barcode link to item record.
02	patron barcode	varchar	26	s	Y	25	Barcode link to the patron record.
03	date charged	date	51	d	Y	10	Date the item was checked out to a patron.
04	time charged	time	61	t	Y	5	Time the item was checked out to a patron.
05	date due	date	66	d	Y	10	The most current date the item is expected back.
06	time due	time	76	t	Y	5	The most current time the item is expected back.
07	date renewed	date	81	d	N	10	The most current date the item was renewed.
08	time renewed	time	91	t	N	5	The most current time the item was renewed.
09	renew count	numeric	96	n	N	5	The number of times the item has been renewed.
10	overdue notice count	numeric	101	n	N	3	The number of overdue notices sent for the item record.
11	overdue notice date	date	104	d	N	10	The last date an overdue notice was sent for the item record, if the overdue notice count is greater than zero.

Introduction

The Item Delete Standard Interface File (SIF) is the format of the file to be produced by any staff client when an item is successfully deleted. Items deleted in any of the modules will append to this file.

Purpose of This Chapter

This section discusses the layout of the Item Delete SIF.

- File Format

File Format

All fields are comma delimited.

If a field is not present, it is represented by a comma for the position it would occupy.

The filename is located in the `/m1/voyager/xxxdb/rpt` directory and is named `delete.item`.

NOTE:

This SIF remains in the Latin-1 character set. If elements in the SIF were stored in the database using the Unicode character set, they are converted to Latin-1 before being exported to the SIF.

[Table 21-1](#) describes the Item Delete SIF.

Table 21-1. Item Delete SIF Format

Position	Description	Notes
1	item barcode	Only the active barcode is placed in the SIF.
2	item id	System generated item record id.
3	mfhd id	System generated MFHD record id.
4	bib id	System generated bibliographic record id.
5	title	Title of the item.
6	create operator id	ID of the operator who created the record.
7	delete operator id	ID of the operator who deleted the record.
8	deletion date/time	Date and time item was deleted.
9	bib 010 \$a	LCCN.
10	item type id	Item type id.
11	item type code	Item type code.
12	media type id	Media type id.
13	media type code	Media type code.
14	location id	Location id.
15	location code	Location code.
16	enum	Enumeration of the item.
17	chron	Chronology of the item.
18	year	Year from the item record.
19	caption	Caption in the item record.
20	free text	Note from the item record.
21	spine label	Spine label.
22	copy number	Copy number for this item.
23	pieces	Pieces on the item record.
24	price	Price formatted according to the base currency.

Introduction

The Vendor Record Standard Interface File (SIF) is the format of the file of your site's vendor records loaded during migration.

Purpose of This Chapter

This section discusses the layout for the Vendor Record SIF.

- Input file specification
- File format fixed segment
- File format address segment
- File format variable segment
- File format end-of-record segment

Input File Specification

The data required for the Vendor database records should be supplied in the following manner.

Each vendor data record should be one continuous string, terminated by a line-feed (\N).

The fields must be the indicated length and must all be present and in a specific order and format. The notes segment is an exception to the length requirement.

The data fields which are designated numeric must be right-justified and zero-filled on the left.

The data fields which are designated strings (not designated numeric) should be left-justified and blank-filled on the right but not hex-null-terminated. The notes segment is an exception to the no-null termination requirement.

Vendor SIF Format

The following conventions are used in the listed tables.

Item # Represents the relative position of the item in the record (sequence).

NOTE:

In all the following tables, the first column, Item #, consists of the field sequence numbers which are not to be included in the record.

Item name Definition of the item.

Offset Actual starting position of the item in the record relative to 1.

Format Format of the variable.

s String, left-justified/blank filled.

n Numeric, right-justified/zero-filled.

Required Y for yes, the field must have valid data in it (not just blanks).

The maximum record should consist of the following:

1 base segment length: $1 \times 147 = 147$

6 address segments length: $6 \times 471 = 2826$

1 notes segment length: $1 \times 2000 = 1900$

1 end-of record mark length: $1 \times 1 = 1$

The total maximum record length is 4874.

The minimum record should consist of the following:

1 base segment length: $1 \times 147 = 147$

1 address segment length: $1 \times 471 = 471$

1 notes segment length: $1 \times 1 = 1$

1 end-of record mark length: $1 \times 1 = 1$

The total minimum record length is 620.

Base Segment

[Table 22-1](#) defines the base segment of all incoming records.

Table 22-1. Base Segment of Record

Item #	Item	Offset	Format	Required	Length	Description
01	vendor type	1	s		2	Variable data based on user's set up of the vendor type table.
02	vendor code	3	s	Y	10	Variable data: user's discretion.
03	vendor name	13	s	Y	60	Vendor name.
04	vendor tax id	73	s		11	Vendor tax identification number.*
05	institution id	84	s		25	ID of ordering institution.
06	default currency	109	s		3	Variable data: user's discretion.
07	claim interval	112	n		5	Maximum value is 32767. This field is numeric (right-justified/zero-filled).
08	claim count	117	n		5	Maximum value is 32767. This field is numeric (right-justified/zero-filled).
09	cancel interval	122	n		5	Maximum value is 32767. This field is numeric (right-justified/zero-filled).
10	ship via	127	s		20	Variable data: user's discretion.

Table 22-1. Base Segment of Record

Item #	Item	Offset	Format	Required	Length	Description
11	address count	147	n	Y	1	Valid data: 1 through 6. There must be at least 1 address segment following. There can be as many as 6. This field is numeric (right-justified/zero-filled).
* Although the Federal Tax ID Number is officially defined as ten digits in length, this field has been defined to accept up to 11 digits by design, to account for the possibility of an alphabetic character at the end as per the IRS EIN documentation at http://ftp.fedworld.gov/pub/irs-pdf/p1635.pdf . If the source info. to be loaded is only ten digits in length, this field may be padded with a space at the front of the number.						

Address Segment

[Table 22-2](#) defines the address portion of the incoming record. This segment is repeatable up to six times consecutively. All address segments must follow the base segment. Note that for the address segment, these sequence numbers might vary, according to the number of the address records.

Table 22-2. Address segment of record

Item #	Item	Offset	Format	Required	Length	Description
12	std. address number	148	s		8	Standard address number for the vendor.
13	type: order	156	s		1	Valid data: Y = Yes, N = No. Footnote 1
14	type: payment	157	s		1	Valid data: Y = Yes, N = No. Footnote 1
15	type: return	158	s		1	Valid data: Y = Yes, N = No. Footnote 1
16	type: claim	159	s		1	Valid data: Y = Yes, N = No. Footnote 1
17	type: email	160	s		1	Valid data: Y = Yes, N = No. Footnote 1

Table 22-2. Address segment of record

Item #	Item	Offset	Format	Required	Length	Description
18	type: other	161	s		1	Valid data: Y = Yes, N = No. Footnote 1
19	contact name	162	s		40	Name of the contact person at the vendor.
20	contact title	202	s		40	Title of the contact person at the vendor.
21	address line 1	242	s	desired	50	Address Line 1.
22	address line 2	292	s		40	Second line of street address.
23	address line 3	332	s		40	Third line of the street address.
24	address line 4	372	s		40	Fourth line of street address.
25	address line 5	412	s		40	Fifth line of street address.
26	city	452	s		30	Any value up to the maximum length.
27	state (province) code	482	s		7	Any 7-character state (province) code.
28	zipcode/postal code	489	s		10	Zipcode in format 12345 or 12345-6789
29	country	499	s		20	Any value up to the maximum length.
30 ^D	phone (primary)	519	s	desired	25	Any value up to the maximum length. At least one phone number (primary) should be present.
31	phone (mobile)	544	s		25	Any value up to the maximum length.
32	phone (fax)	569	s		25	Any value up to the maximum length.
33	phone (other)	594			25	Any value up to the maximum length.
1 at least one of these address types must be marked "y" (yes).						

Variable Segment

[Table 22-3](#) defines the variable segment of the incoming record. This segment must follow the last address segment and is an exception to the length and no-null-termination requirements. The notes segment is the only portion which is variable in length. It must be present and it must be null-terminated but it may consist of only a null-terminator (\0 or hex '00').

Table 22-3. Variable (Notes) segment of record

Item #	Item Name	Offset	Format	Required	Length	Description
34	notes	619	s		1900 max	Vendor notes. This field may contain any data the user wants. This segment must be present but may be simply a null-terminator to indicate that there are no notes.

End-of-Record Segment

[Table 22-4](#) defines the end-of-record marker. This segment must follow the notes segment.

Table 22-4. End-of-record marker

Item #	Item Name	Offset	Format	Required	Length	Description
35	End-of-Record Marker		s	Y	1	Valid Data: Line-feed (\N).

Introduction

The Acquisitions Notices Standard Interface File (SIF) is the file produced by activity at acquisitions locations and/or after running acquisitions batch jobs on the server.

Acquisitions activity and/or running acquisitions batch jobs produces a file containing the notice information in a specific format. This file is then used as an input file (`.inp`) for the Reporter module to use to generate notices. See *Creating Input Files for Acquisitions* in the *Voyager Reporter User's Guide* for more information.

Purpose of This Chapter

This section discusses the layout for the Acquisitions Notices SIF.

- File specification
- File format base segment
- File format suffix segment

File Specification

The name of the file produced is `acqnotes.[print location code].inp`, where `xxxx` represents the appropriate printing location code as defined in "Print Locations" in the System Administration module. See *Print Locations* in the *Voyager System Administration User's Guide* for more information. If the print location code is `main`, the file will be called `acqnotes.main.inp`. When the file is generated, it is placed in the `/m1/voyager/xxxdb/rpt` directory.

Each record in the SIF is made up of two pieces: a base segment and a suffix.

The base segment is used in every record in the acquisitions notices SIF and is always at the beginning of the record. After the base segment, one of the suffixes will be attached. Depending on the type of notice for which the record contains data, a different suffix will be attached. A record will only contain one base segment and one suffix.

For example, a sample acquisitions notice record might appear as follows:

```
00|97.2||2|EBSCO|John Adams|10400 Higgins Road||||Des
Plaines|IL|60018|||10/12/1999|Ex Libris Library of
InformationAcquisitions|123 9th Street|||Des
Plaines|IL|60000||||56|08/04/1999||British journal of
anaesthesia.|ISSN 0007-0912|2|2
```

The first 29 fields of the record are the base segment. Because this is a cancellation notice record, the suffix is made up of the last seven fields. The type of suffix attached determines the record type.

The record type can be determined by examining the first field of the record. The two-digit number indicates the type of notice. The notice numbers are listed in the heading for each suffix type.

All fields must be present and pipe-delimited. The pipe for a given field must be present even if the field is empty.

Each record will be a line-feed-terminated string.

Maximum field lengths as imposed below must be adhered to.

NOTE:

This SIF remains in the Latin-1 character set. If elements in the SIF were stored in the database using the Unicode character set, they are converted to Latin-1 before being exported to the SIF.

Acquisitions Notices SIF Format

The following conventions are used in the listed tables.

Item # Represents the relative position of the item in the record (sequence).

NOTE:

In all the following tables, the first column, Item #, consists of the field sequence numbers which are not to be included in the record.

Item name Definition of the item.

Required If Y is entered, the field must have valid data in it and cannot be blank. If nothing is entered, the field is not required.

Description Description of the item.

The total number of fields varies with record type as follows.

- Type 00: Cancellation Notice = 36 fields.
- Type 01: Return Notice = 36 fields.
- Type 02: Order Claim Notice = 37 fields.
- Type 03: Serial Claim Notice = 38 fields.
- Type 04: Voucher/check request = 44 fields.
- Type 05: Cancel Serial Claim Notice = 38 fields.

Base Segment

[Table 23-1](#) defines the base segment for all acquisitions notice type records.

Table 23-1. Base Segment for Acquisitions Notices

Item #	Required	Item Name	Length	Description
1	Y	notice id	2	This field must be 2 characters in length and contain 1 of the following codes indicating the type of notice: 00 is a cancellation notice 01 is a return notice 02 is an order claim notice 03 is a serial claim notice 04 is a voucher/check request 05 is a cancel serial claims
2	Y	version number	30	Program version number to be used by client for version checking.
3		e-mail address	50	Vendor's e-mail address. If this field is not blank, the notice will be sent by e-mail. If it is blank, the notice will be printed for mailing.
4	Y	vendor id	10	Vendor's ID number.
5	Y	vendor name	60	Vendor's name.
6		attention	40	Attention to.
--->>		note		Addresses to be used for various notices are as follows: cancellation notice -- order address. return notice -- return address. order/serial claim notices -- claim address. voucher/check requests -- payment address.
7	Y	address line 1	50	Vendor's address line 1.
8		address line 2	40	Vendor's address line 2.
9		address line 3	40	Vendor's address line 3.
10		address line 4	40	Vendor's address line 4.

Table 23-1. Base Segment for Acquisitions Notices

Item #	Required	Item Name	Length	Description
11		address line 5	40	Vendor's address line 5.
12		city	30	Vendor's city.
13		state/province	7	Vendor's state/province.
14		postal code	10	Vendor's postal code.
15		country	20	Vendor's country.
16		phone	25	Vendor's phone.
17		date	10	Current date.
18	Y	institution	50	Institution name.
19	Y	order site	25	Cancellation notices: order site location name. Return notices: order site location name. Order claim notices: order site location name. Serial claim notices: ship to site. Voucher/check request: bill to site.
20		address line 1	50	Order site address line 1.
21		address line 2	40	Order site address line 2.
22		address line 3	40	Order site address line 3.
23		city	30	Order site city.
24		state/province	7	Order site state/province.
25		postal code	10	Order site postal code.
26		country	20	Order site country.
27		phone	25	Order site phone.
28		fax	25	Order site fax.
29		e-mail address	50	Order site e-mail address.

Suffixes

[Table 23-2](#) through [Table 23-7](#) cover the various suffix segments of the acquisitions notices.

Cancellation Notice Suffix (00)

[Table 23-2](#) describes the cancellation notice suffix.

Table 23-2. Cancellation Notice Suffix (00)

Item #	Required	Item Name	Length	Description
30	Y	P.O. #	25	A single P.O. number.
31	Y	P.O. date	10	Purchase order date.
32		invoice number	25	Voyager invoice number for this line-item.
33	Y	title	255	Title/edition.
34		standard number	100	Standard number (use vendor title/number if available).
35	Y	line item number	10	Single line-item number.
36	Y	number of copies	10	Number of copies to cancel. <u>Last item in the record for</u> cancellation notices.

Return Notice Suffix (01)

[Table 23-3](#) describes the return notice suffix.

Table 23-3. Return Notice Suffix (01)

Item #	Required	Item Name	Length	Description
30	Y	P.O. #	25	A single P.O. number.
31	Y	P.O. date	10	Purchase order date.
32		invoice number	25	Voyager invoice number for this line-item.
33	Y	title	255	Title/edition.
34		standard number	100	Standard number (use vendor title/number if available).
35	Y	line item number	10	Single line-item number.
36	Y	number of copies	10	Number of copies returned. <u>Last item in the record for</u> return notices.

Order Claim Notice Suffix (02)

[Table 23-4](#) describes the order claim notice suffix.

Table 23-4. Order Claim Notice Suffix (02)

Item #	Required	Item Name	Length	Description
30	Y	P.O. #	25	A single P.O. number.
31	Y	P.O. date	10	Purchase order date.
32	Y	title	255	Title/edition.
33		standard number	100	Standard number.
34	Y	line item number	10	Single line-item number.
35		claim type	70	Claim type.
36	Y	number of copies	10	Number of copies claimed.
37		note to vendor	255	Line-item note to vendor. <u>Last item in the record for</u> order claim notices.

Serial Claim Notice Suffix (03)

[Table 23-5](#) describes the serial claim notice suffix.

Table 23-5. Serial Claim Notice Suffix (03)

Item #	Required	Item Name	Length	Description
30	Y	P.O. #	25	A single P.O. number.
31	Y	P.O. date	10	Purchase order date.
32	Y	title	255	Title.
33		issn	100	ISSN.
34	Y	line item number	10	Single line-item number.
35		claim type	70	Claim type.
36	Y	number of copies	10	Number of copies claimed.
37		note to vendor	255	Line-item note to vendor.
38		description	255	Issue information. <u>Last item in the record for</u> serial claim notices.

Voucher/Check Request Suffix (04)

[Table 23-6](#) describes the voucher/check request suffix.

Table 23-6. Voucher/Check Request Suffix (04)

Item #	Required	Item Name	Length	Description
30	Y	voucher id	25	Voyager voucher ID.
31	Y	currency	35	Transaction currency description.
32	Y	invoice amount	10	Voyager invoice amount.
33	Y	invoice number	25	Voyager invoice number.
34	Y	invoice date	10	Voyager invoice date.
35		vendor inst. id	25	Vendor institution ID.
36	Y	P.O. #	25	A single P.O. number.
37		P.O. date	10	Purchase order date.
38	Y	line item number	10	Single line-item number.
39	Y	line item title	255	Title for the item.
40	Y	line item amount	10	Line item amount (for this fund).
41	Y	fund total	10	Total amount for this fund which applies to this invoice.
42	Y	fund description	125	Fund description.
43	Y	ledger id	25	Ledger id used for sort key.
44	Y	fund id	25	Fund id used for sort key. <u>Last item in the record</u> for voucher/check requests.

Cancel Serial Claim Notice Suffix (05)

[Table 23-7](#) describes the cancel serial claim notice suffix.

Table 23-7. Cancel Serial Claim Notice Suffix (05)

Item #	Required	Item Name	Length	Description
30	Y	P.O. #	25	A single P.O. number.
31	Y	P.O. date	10	Purchase order date.
32	Y	title	255	Title.
33		issn	100	ISSN.

Table 23-7. Cancel Serial Claim Notice Suffix (05)

Item #	Required	Item Name	Length	Description
34	Y	line item number	10	Single line-item number.
35		claim type	70	Claim type.
36	Y	number of copies	10	Number of copies claimed.
37		note to vendor	255	Line-item note to vendor.
38		description	255	Issue information. <u>Last item in the record for</u> cancel serial claim notices.

Introduction

The Acquisitions Reports Standard Interface File (SIF) is a file produced by running acquisitions batch jobs on the server.

Running acquisitions batch jobs produces a file containing the report information in a specific format. This file is then used as an input file (`.inp`) for the Reporter module to use to generate reports. See *Creating Input Files for Acquisitions* in the *Voyager Reporter User's Guide* for more information.

Purpose of This Chapter

This section discusses the layout for the Acquisitions Report SIF.

- File specification
- File format base segment
- File format suffix segment

File Specification

The name of the file produced is `acqrprts.[print location code].inp`, where `xxxx` represents the appropriate printing location code as defined in "Print Locations" in the System Administration module. See *Print Locations* in the

Voyager System Administration User's Guide for more information. If the print location code is main, the file will be called `acqrprts.main.inp`. When the file is generated, it is placed in the `/ml/voyager/xxxdb/rpt` directory.

Each record in the SIF is made up of two pieces: a base segment and a suffix.

The base segment is used in every record in the acquisitions reports SIF and is always at the beginning of the record. After the base segment, one of the suffixes will be attached. Depending on the type of report for which the record contains data, a different suffix will be attached. A record will only contain one base segment and one suffix.

For example, a sample acquisitions report record might appear as follows:

```
06|97.2|08/16/1999|1999 (01/01/1999 - 12/31/1999)|Main
Library FY99|1999 (01/01/1999 - 12/31/1999)|Main
Acquisitions Group|American
Literature|1|General|Literature|0|General|||N|
10000.00|10000.00|9874.19|10000.00|0.00|0.00|125.81|0.0
0|110%|102%
```

The first two fields of the record are the base segment. Because this is a Fund Snapshot Before Rollover report record, the suffix is made up of the last 20 fields. The type of suffix attached determines the record type.

The record type can be determined by examining the first field of the record. The two-digit number indicates the type of report. The report numbers are listed in the heading for each suffix type.

All fields must be present and pipe-delimited. The pipe for a given field must be present even if the field is empty.

Each record will be a line-feed-terminated string.

Maximum field lengths as imposed below must be adhered to.

NOTE:

This SIF remains in the Latin-1 character set. If elements in the SIF were stored in the database using the Unicode character set, they are converted to Latin-1 before being exported to the SIF.

Acquisitions Reports SIF Format

The following conventions are used in the listed tables.

Item # Represents the relative position of the item in the record (sequence).

NOTE:

In all the following tables, the first column, Item #, is only the field sequence numbers and are not to be included in the record.

Item name Definition of the item.

Required If Y is entered, the field must have valid data in it and cannot be blank. If nothing is entered, the field is not required.

Description Description of the item.

The total number of fields varies with record type as follows.

- Type 00: Purchase Order Report = 52 fields.
- Type 01: Open Orders Report = 22 fields.
- Type 02: Global Open Orders Report = 22 fields.
- Type 03: Fund Snapshot Report = 26 fields.
- Type 04: Open Orders Before Rollover = 22 fields.
- Type 05: Open Orders After Rollover = 22 fields.
- Type 06: Fund Snapshot Before Rollover = 22 fields.
- Type 07: Fund Snapshot After Rollover = 22 fields.
- Type 08: Fund Rollover Status = 9 fields.
- Type 09: Copy Rollover Status = 11 fields.

Base Segment

[Table 24-1](#) defines the base segment for all acquisitions report type records.

Table 24-1. Base segment for acquisitions reports

Item #	Required	Item Name	Length	Description
1	Y	report id	2	This field must be 2 characters in length and contain 1 of the following codes indicating the type of notice: 00 is a purchase order. 01 is an open order report. 02 is a global open orders report. 03 is a fund shapshot report. 04 is an open order rpt before roll-over. 05 is an open order rpt after rollover. 06 is a fund snapshot rpt before roll-over. 07 is a fund snapshot rpt after roll-over. 08 is a fund rollover status rpt. 09 is a copy rollover status rpt.
2	Y	version number	30	Program version number to be used by client for version checking.

Suffixes

[Table 24-2](#) through [Table 24-7](#) cover the various suffix segments of the acquisitions reports.

Purchase Orders Report (00)

[Table 24-2](#) describes the purchase order report suffix.

Table 24-2. Purchase Orders Report Suffix (00)

Item #	Required	Item Name	Length	Description
3	Y	order_site	50	Institution name.
4		rush_order	25	The word 'rush' if entire order is.
5	Y	p_o_number	25	Purchase order number.
6	Y	p_o_date	10	Date, preferred format: mm/dd/ccyy
7	Y	ordered_by	10	Name of ordering person.
8	Y	vendor_name	60	Vendor's name.
9		address line 1	50	Vendor's address line 1.
10		address line 2	50	Vendor's address line 2.
11		address line 3	50	Vendor's address line 3.
12		address line 4	50	Vendor's address line 4.
13		city	30	Vendor's city.
14		state/province	7	Vendor's state/province.
15		postal code	10	Vendor's postal code.
16		country	20	Vendor's country.
17		ship_via	20	Shipping method.
18		note to vendor	60	Order-level note to vendor.
19	Y	print_id	12	ID number used to keep individual copies of a purchase order separate.
20	Y	line_item_number	4	Line item number.
21		item_rush_word	18	The word rush in some language if this item (but not the entire order) is to be rushed.

Table 24-2. Purchase Orders Report Suffix (00)

Item #	Required	Item Name	Length	Description
22	Y	description	255	A maximum of 255 characters will be printed due to limitations of MSAccess. This item is comprised of the following pieces: Title and piece identifier. The title will always be present. MARC 245[a]s. It may be as long as 150 characters. The piece identifier may be present. The sum of these two items must not exceed 255 characters, it is no longer necessary to calculate how many lines are needed for a line item as line spacing will be handled by the client program.
23		line item note	255	Line-item level note to the vendor.
24		publisher	255	Publisher identification.
25		vendor_title_number	100	Vendor title/number, if it exists. Otherwise the print standard number for the line item preceded by its standard number type description is used. Example: ISSN 123456789 or PUB NO. 987654321 See Vendor Title/Number Field on page 24-8 for additional information.
26	Y	quantity/unit	4	Number of units ordered.
27		unit_cost	12	Cost of 1 unit in format: 9.99
28		adjustments	12	Total of all adjustments to line item.
29		prepay_amount	12	Prepay amount of 1 unit in format: 9.99

Table 24-2. Purchase Orders Report Suffix (00)

Item #	Required	Item Name	Length	Description
30		net_total_cost	12	Extended cost in format: 9.99. The extended cost is calculated by the following formula: ((UNIT COST * (NUMBER OF UNITS) +/- ADJUSTMENTS - PREPAY AMOUNT)).
31		sub_total	12	Sub_total of ordered items on this page: 9.99
32		other_charges	12	Other charges applicable to this page: 9.99
33		order_total	12	Total of all items on this entire purchase order (all pages): 9.99
34	Y	ship_to_name	50	Ship to name.
35	Y	address line 1	50	Ship to address line 1.
36		address line 2	50	Ship to address line 2.
37		address line 3	50	Ship to address line 3.
38		address line 4	50	Ship to address line 4.
39		city	30	Ship to city.
40		state/province	7	Ship to state/province.
41		postal code	10	Ship to postal code.
42		country	20	Ship to country.
43	Y	bill_to_name	50	Bill to name.
44		address line 1	50	Bill to address line 1.
45		address line 2	50	Bill to address line 2.
46		address line 3	50	Bill to address line 3.
47		address line 4	50	Bill to address line 4.
49		city	30	Bill to city.
50		state/province	7	Bill to state/province.
51		postal code	10	Bill to postal code.
52		country	20	Bill to country.
--		record length	1983	Record length (not included as part of record).

Vendor Title/Number Field

The Vendor Title/Number will be provided if present. If not, print_std_num is specified in the line-item. If the latter is the case, use MARC fields and supply the label and the number as follows. If print_std_num is:

IS use: marc 022|a and supply: ISSN + number

IB use: marc 020|a and supply: ISBN + number

CO use: marc 030|a and supply: CODEN + number

GP use: marc 037|a and supply: STOCK NO. + number

PN use: marc 028|a and supply: PUB NO. + number

ST use: marc 027|a and supply: STRN + number

Open Orders Report (01)

[Table 24-3](#) describes the open orders report suffix.

Table 24-3. Open Orders Report Suffix (01)

Item #	Required	Item Name	Length	Description
3	Y	report_date	10	Report run date.
4	Y	for_site	50	For institution name (or all).
5	Y	for_ledger	125	For ledger name (or all), fiscal-period (dates).
6	Y	order_site	50	Institution name.
7	Y	ledger_name	125	Ledger name, fiscal-period (dates).
8	Y	p_o_number	25	Purchase order number.
9	Y	p_o_id	50	Purchase order system ID number.
10	Y	vendor_name	60	Vendor's name.
11	Y	p_o_type	25	Purchase order type.
12	Y	line_item-number	4	Line item number.
13		description	255	This item is comprised of the following pieces: title and piece identifier.
14	Y	copy_number	4	Copy sequence number.
15	Y	location	25	Target location.
16		item_type	25	Line item type.
17	Y	copy_status	25	Copy status.

Table 24-3. Open Orders Report Suffix (01)

Item #	Required	Item Name	Length	Description
18		invoice_status	25	Invoice status.
19		item_amount	20	Line item amount [(\$)987,654,321.1234567].
20	Y	fund_count	3	Number of fund segments to follow.
21	Y	ledger_id	15	Ledger_id: used for sorting fund segments.
22		fund_name	255	Fund name. If copy is allocated to more than 1 fund, etc., additional fund name(s), ledger(s), fiscal period(s), and percentages will be included here as additional (ledger_id/fund_name) segments.
note:		--->>	-->>	Ledger_id and fund_name fields (paired) constitute a repeatable segment. The fund_count field will have the number of these segments.
--		record length	1148	Record length (not included as part of record).

Global Open Orders Report (02)

[Table 24-4](#) describes the global open orders report suffix.

Table 24-4. Global Open Orders Report Suffix (02)

Item #	Required	Item Name	Length	Description
3	Y	report_date	10	Report run date.
4	Y	for_site	50	For institution name (or all).
5	Y	for_ledger	125	For ledger name (or all) , fiscal-period (dates).
6	Y	order_site	50	Institution name.
7	Y	ledger_name	125	Ledger name, fiscal-period (dates).
8	Y	p_o_number	25	Purchase order number.
9	Y	p_o_id	50	Purchase order system ID number.
10	Y	vendor_name	60	Vendor's name.

Table 24-4. Global Open Orders Report Suffix (02)

Item #	Required	Item Name	Length	Description
11	Y	p_o_type	25	Purchase order type.
12	Y	line_item-number	4	Line item number.
13		description	255	This item is comprised of the following pieces: title and piece identifier.
14	Y	copy_number	4	Copy sequence number.
15	Y	location	25	Target location.
16		item_type	25	Line item type.
17	Y	copy_status	25	Copy status.
18		invoice_status	25	Invoice status.
19		item_amount	20	Line item amount [(\$)987,654,321.1234567].
20	Y	fund_count	3	Number of fund segments to follow.
21	Y	ledger_id	15	Ledger_id: used for sorting fund segments.
22		fund_name	255	Fund name. If copy is allocated to more than 1 fund, etc., additional fund name(s), ledger(s), fiscal period(s), and percentages will be included here as additional (ledger_id/fund_name) segments.
note:		--->>	-->>	Ledger_id and fund_name fields (paired) constitute a repeatable segment. The fund_count field will have the number of these segments.
--		record length	1148	Record length (not included as part of record).

Fund Snapshot Report Suffix (03)

[Table 24-5](#) describes the fund snapshot report suffix.

Table 24-5. Fund Snapshot Report Suffix (03)

Item #	Required	Item Name	Length	Description
3	Y	report_date	10	Report run date.
4	Y	for_f_period	100	For fiscal period name (or all).
5	Y	ledger_name	40	Ledger name.
6	Y	fiscal_period	100	Fiscal period name.
7	Y	policy_group	40	Policy group name.
8	Y	fund_name	25	Fund name.
9	Y	fund_cat	2	Fund category: 00=summary, 01=allocated, 02=reporting.
10	Y	fund_type	25	Fund type name (descriptive name).
11		parent_fund_name	25	Fund name (parent).
12		parent_fund_cat	2	Fund category: 00=summary, 01=allocated, 02=reporting.
13		parent_fund_type	25	Fund type name (descriptive name).
14		begin_date	10	Fund begin date.
15		end_date	10	Fund end date.
16	Y	expend_only	1	Yes or no.
17		original_alloc	20	Original allocation [(\$)987,654,321.1234567].
18		net_alloc	20	Net allocation [(\$)987,654,321.1234567].
19		bal_avail	20	Balance available [(\$)987,654,321.1234567].
20		bal_cash	20	Cash balance [(\$)987,654,321.1234567].
21		pend_commit	20	Pending commitments [(\$)987,654,321.1234567].
22		pend_expnd	20	Pending expenditures [(\$)987,654,321.1234567].

Table 24-5. Fund Snapshot Report Suffix (03)

Item #	Required	Item Name	Length	Description
23		commits	20	Commitments [(\$)987,654,321.1234567].
24		expends	20	Expenditures [(\$)987,654,321.1234567].
25		over_commit	4	Over commitment percent (999%).
26		over_expend	4	Over expenditure percent (999%).
--		record length	525	Record length (not included as part of record).

NOTE:

There must be a separate Fund Snapshot Report Record for each fund to be reported. The record has been designed so that all funds may be reported in the same format. However, while all funds may be reported in the same record format, not all records will include data for all of the data items in the record. Be aware that even if a field is not appropriate for the type of fund you are reporting, it (the field) must still be present and represented by the pipe (|) delimiter.

Open Orders Before Rollover Report (04)

There must be one Open Orders Before Rollover record for each line-item which is to be displayed on the report. This report is currently the same layout as the standard Open Orders Report (report type 01).

Open Orders After Rollover Report (05)

There must be one Open Orders After Rollover record for each line-item which is to be displayed on the report. This report uses the same layout as the standard Open Orders Report (report type 01).

Fund Snapshot Before Rollover Report (06)

There must be a separate Fund Snapshot Report Record for each fund to be reported. The record has been designed so that all funds may be reported in the same format. This report uses the same layout as the standard Fund Snapshot Report (report type 03).

Fund Snapshot After Rollover Report (07)

There must be a separate Fund Snapshot Report Record for each fund to be reported. The record has been designed so that all funds may be reported in the same format. This report uses the same layout as the standard Fund Snapshot Report (report type 03).

Fund Rollover Status Report Suffix (08)

[Table 24-6](#) describes the fund rollover status report suffix.

Table 24-6. Fund Rollover Status Report Suffix (08)

Item #	Required	Item Name	Length	Description
3	Y	report_date	10	Report run date.
4	Y	rollover_Run_ID	50	Unique rollover run identifier.
5	Y	for_fiscal_period	100	For fiscal-period (dates).
6		ledger_name	125	Ledger name.
7		fund_name	25	Fund name.
8		rollover_status	255	Rollover status string.
9	Y	time_stamp	25	Rollover time.
--		record length	537	Record length (not included as part of record).

Copy Rollover Status Report Suffix (09)

[Table 24-7](#) describes the copy rollover status report suffix.

Table 24-7. Fund Snapshot Report Suffix (03)

Item #	Required	Item Name	Length	Description
3	Y	report_date	10	Report run date.
4	Y	rollover_run_id	50	Unique rollover run identifier.
5	Y	p_o_number	25	Purchase order number.
6		item_number	4	Line item number.
7		description	255	Item title.
8		vendor title number	100	Vendor's item identification.

Table 24-7. Fund Snapshot Report Suffix (03)

Item #	Required	Item Name	Length	Description
9		copy_location	25	Copy location. To be replaced with copy sequence number when that becomes available.
10		rollover_status	255	Rollover status string.
11	Y	time_stamp	25	Rollover time.
--		record length	781	Record length (not included as part of record).

Introduction

The Cataloging Reports Standard Interface File (SIF) is the file produced by running cataloging batch jobs on the server.

Running the cataloging batch jobs produces a file containing the report information in a specific format. This file is then used as an input file (`.inp`) for the Reporter module to use to generate the reports. See *Creating Input Files for Cataloging* in the *Voyager Reporter User's Guide* for more information.

Purpose of This Chapter

This section discusses the layout for the Cataloging Reports SIF.

- File specification
- File format base segment
- File format suffix segment

File Specification

The name of the file produced is `catrprts.[print location code].inp`, where `xxxx` represents the appropriate printing location code as defined in "Print Locations" in the System Administration module. See *Print Locations* in the

Voyager System Administration User's Guide for more information. If the print location code is main, the file will be called `catrprts.main.inp`. When the file is generated, it is placed in the `/ml/voyager/xxxdb/rpt` directory.

Each record in the SIF is made up of two pieces: a base segment and a suffix.

The base segment is used in every record in the cataloging reports SIF and is always at the beginning of the record. After the base segment, one of the suffixes will be attached. Depending on the type of report for which the record contains data, a different suffix will be attached. A record will only contain one base segment and one suffix.

For example, a sample cataloging report record might appear as follows:

```
09|97.2||198.80.36.91|http://198.80.36.91/ndu/inss/
books/ootw/ ootwhome.html|URL|996|Socket Write
Error|Bibliographic| 61218|09/16/1999
```

The first three fields of the record are the base segment. Because this is an 856 Link Failure Report report record, the suffix is made up of the last eight fields. The type of suffix attached determines the record type.

The record type can be determined by examining the first field of the record. The two-digit number indicates the type of report. The report numbers are listed in the heading for each suffix type.

All fields must be present and pipe-delimited. The pipe for a given field must be present even if the field is empty.

Each record will be a line-feed-terminated string.

Maximum field lengths as imposed below must be adhered to.

NOTE:

This SIF remains in the Latin-1 character set. If elements in the SIF were stored in the database using the Unicode character set, they are converted to Latin-1 before being exported to the SIF.

Cataloging Reports SIF Format

The following conventions are used in the listed tables.

Item #	Represents the relative position of the item in the record (sequence).

NOTE:

In all the following tables, the first column, Item #, consists of the field sequence numbers which are not to be included in the record.

Item name	Definition of the item.
Required	If Y is entered, the field must have valid data in it and cannot be blank. If nothing is entered, the field is not required.
Description	Description of the item.

The total number of fields varies with record type as follows.

- Type 00: Unauthorized Subject Headings = 8 fields.
- Type 01: Unauthorized Name Headings = 8 fields.
- Type 02: Unauthorized Title Headings = 8 fields.
- Type 03: Unauthorized Name/Title Headings = 8 fields.
- Type 04: Unauthorized Subdivision Headings = 8 fields.
- Type 05: Duplicate Authority Records = 9 fields
- Type 06: 'See' Ref's with Linked Bib Records = 10 fields.
- Type 07: 'See' Ref's Authorized in Another Record = 11 fields.
- Type 08: 'See' Ref's without an Authority Record = 10 fields.
- Type 09: 856 Link Failure Report = 11 fields.

Base Segment

[Table 25-1](#) defines the base segment for all cataloging report type records.

Table 25-1. Base segment for cataloging reports

Item #	Required	Item Name	Length	Description
1	Y	report id	2	This field must be 2 characters in length and contain 1 of the following codes indicating the type of notice: 00 is an unauthorized subject heading. 01 is an unauthorized name heading. 02 is an unauthorized title heading. 03 is an unauthorized name/title heading. 04 is an unauthorized subdivision heading. 05 is a duplicate authority record. 06 is a 'see' references with linked bib records. 07 is a 'see' references authorized in another authority record. 08 is a 'see-also' reference without a corresponding authority record. 09 is an 856 link failure report.
2	Y	version number	30	Program version number to be used by client for version checking.
3		date[time]/selection (range)	50	Date (mm/dd/ccyy) with optional time (hh:mm) or a range of dates with optional times or a selected range of headings.

Suffixes

[Table 25-2](#) through [Table 25-11](#) cover the various suffix segments of the cataloging reports.

Unauthorized Subject Headings (00)

[Table 25-2](#) describes the unauthorized subject headings report suffix.

Table 25-2. Unauthorized Subject Headings Report Suffix (00)

Item #	Required	Item Name	Length	Description
4	Y	heading 1	255	First 255 characters of heading.
5		heading 2	45	Last 45 characters of heading.
6	Y	thesaurus	50	Thesaurus.
7	Y	date heading added	10	Date heading added.
8	Y	opac bib count	15	Opac bib count.

Unauthorized Name Headings (01)

[Table 25-3](#) describes the unauthorized name headings report suffix.

Table 25-3. Unauthorized Name Report Suffix (01)

Item #	Required	Item Name	Length	Description
4	Y	heading 1	255	First 255 characters of heading.
5		heading 2	45	Last 45 characters of heading.
6	Y	name type	50	Name type.
7	Y	date heading added	10	Date heading added.
8	Y	opac bib count	15	Opac bib count.

Unauthorized Title Headings (02)

[Table 25-4](#) describes the unauthorized title headings report suffix.

Table 25-4. Unauthorized Title Headings Report Suffix (02)

Item #	Required	Item Name	Length	Description
4	Y	heading 1	255	First 255 characters of heading.
5		heading 2	45	Last 45 characters of heading.
6	Y	title type	50	Title type.
7	Y	date heading added	10	Date heading added.
8	Y	opac bib count	15	Opac bib count.

Unauthorized Name/Title Headings (03)

[Table 25-5](#) describes the unauthorized name/title headings report suffix.

Table 25-5. Unauthorized Name/Title Headings Report Suffix (03)

Item #	Required	Item Name	Length	Description
4	Y	heading 1	255	First 255 characters of heading.
5		heading 2	45	Last 45 characters of heading.
6	Y	name/title type	50	Name/title type.
7	Y	date heading added	10	Date heading added.
8	Y	opac bib count	15	Opac bib count.

Unauthorized Subdivision Headings (04)

[Table 25-6](#) describes the unauthorized subdivision headings report suffix.

Table 25-6. Unauthorized Subdivision Headings Report Suffix (04)

Item #	Required	Item Name	Length	Description
4	Y	heading 1	255	First 255 characters of heading.
5		heading 2	45	Last 45 characters of heading.
6	Y	thesaurus	50	Thesaurus.
7	Y	date heading added	10	Date heading added.
8	Y	opac bib count	15	Opac bib count.

Duplicate Authority Records (05)

[Table 25-7](#) describes the duplicate authority records report suffix.

Table 25-7. Duplicate Authority Records Report Suffix (05)

Item #	Required	Item Name	Length	Description
4	Y	heading type	50	Heading type (name, title).
5	Y	heading 1	255	First 255 characters of heading.
6		heading 2	45	Last 45 characters of heading.
7	Y	normalized heading 1	255	First 255 characters of normalized heading.

Table 25-7. Duplicate Authority Records Report Suffix (05)

Item #	Required	Item Name	Length	Description
8		normalized heading 2	255	Last 45 characters of normalized heading.
9	Y	authority record id	100	Authority record ID.

‘See’ References with Linked Bib Records (06)

[Table 25-8](#) describes the see references with linked bib records report suffix.

Table 25-8. ‘See’ References with Linked Bib Records Report Suffix (06)

Item #	Required	Item Name	Length	Description
4	Y	heading type	50	Heading type (name, title).
5	Y	heading 1	255	First 255 characters of heading.
6		heading 2	45	Last 45 characters of heading.
7	Y	normalized heading 1	255	First 255 characters of normalized heading.
8		normalized heading 2	255	Last 45 characters of normalized heading.
9	Y	4xx authority record id	100	4xx authority record ID.
10	Y	opac bib count	15	Opac bib count.

‘See’ References Authorized in Another Authority Record (07)

[Table 25-9](#) describes the see references authorized in another authority record report suffix.

Table 25-9. ‘See’ References Authorized in Another Authority Record Suffix (07)

Item #	Required	Item Name	Length	Description
4	Y	heading type	50	Heading type (name, title).
5	Y	heading 1	255	First 255 characters of heading.
6		heading 2	45	Last 45 characters of heading.
7	Y	normalized heading 1	255	First 255 characters of normalized heading.
8		normalized heading 2	255	Last 45 characters of normalized heading.

Table 25-9. 'See' References Authorized in Another Authority Record Suffix (07)

Item #	Required	Item Name	Length	Description
9	Y	4xx authority record id	100	4xx authority record ID.
10	Y	1xx authority record id	100	1xx authority record ID.
11	Y	opac bib count	15	Opac bib count.

'See' References without Corresponding Authority Record (08)

[Table 25-10](#) describes the see references without corresponding authority record report suffix.

Table 25-10. 'See' References without Corresponding Authority Record Report Suffix (08)

Item #	Required	Item Name	Length	Description
4	Y	heading type	50	Heading type (name, title).
5	Y	heading 1	255	First 255 characters of heading.
6		heading 2	45	Last 45 characters of heading.
7	Y	normalized heading 1	255	First 255 characters of normalized heading.
8		normalized heading 2	255	Last 45 characters of normalized heading.
9	Y	5xx authority record id	100	5xx authority record ID.
10	Y	opac bib count	15	Opac bib count.

856 Link Failure Report Suffix (09)

[Table 25-11](#) describes the 856 link failure report suffix.

Table 25-11. 856 Link Failure Report Suffix (09)

Item #	Required	Item Name	Length	Description
4	Y	link_host	40	Link host.
5	Y	link	255	Link (URL).
6	Y	link type	3	Link type.
7	Y	error code	10	Error (status) code.
8	Y	error description	255	Error description.

Table 25-11. 856 Link Failure Report Suffix (09)

Item #	Required	Item Name	Length	Description
9	Y	record type	25	Link record type (bibliographic, holdings, or e-item).
10	Y	record id	25	Record ID.
11	Y	update date	10	Date last updated.

Introduction

The Circulations Notices Standard Interface File (SIF) is the file produced by activity at circulation locations and then running circulation batch jobs on the server.

Running the circulation batch jobs produces a file containing the notice information in a specific format. This file is then used as an input file (.inp) for the Reporter module to use to generate the notices. See *Creating Input Files for Circulation* in the *Voyager Reporter User's Guide* for more information.

Purpose of This Chapter

This section discusses the layout for the Circulation Notices SIF.

- File specification
- File format base segment
- File format suffix segment

File Specification

The name of the file produced is `crcnotes.[print location code].inp`, where `xxxx` represents the appropriate printing location code as defined in "Print Locations" in the System Administration module. See *Print Locations* in the *Voyager System Administration User's Guide* for more information. If the print location code is `main`, the file will be called `crcnotes.main.inp`. When the file is generated, it is placed in the `/m1/voyager/xxxdb/rpt` directory.

Each record in the SIF is made up of two pieces: a base segment and a suffix.

The base segment is used in every record in the circulation notices SIF and is always at the beginning of the record. After the base segment, one of the suffixes will be attached. Depending on the type of notice for which the record contains data, a different suffix will be attached. A record will only contain one base segment and one suffix.

For example, a sample circulation notice record might appear as follows:

```
02|97.2||28|Smith|John ||8974 W.117th Street||||Kansas
City|MO|76763||(290) 932-0371|10/15/1999|Ex Libris
Library of Information|Main|2200 E. Devon|Suite
382||Des Plaines|IL|60018|USA||Motor carriers' road
atlas, 1998 : United States, Canada, Mexico / Rand
McNally.|Rand McNally and Company.|12481632||10/13/
1999|1|||
```

The first 33 fields of the record are the base segment. Because this is an overdue notice record, the suffix is made up of the last five fields. The type of suffix attached determines the record type.

The record type can be determined by examining the first field of the record. The two-digit number indicates the type of notice. The notice numbers are listed in the heading for each suffix type.

All fields must be present and pipe-delimited. The pipe for a given field must be present even if the field is empty.

Each record will be a line-feed-terminated string.

Maximum field lengths as imposed below must be adhered to.

NOTE:

This SIF remains in the Latin-1 character set. If elements in the SIF were stored in the database using the Unicode character set, they are converted to Latin-1 before being exported to the SIF.

Circulation Notices SIF Format

The following conventions are used in the listed tables.

Item #	Represents the relative position of the item in the record (sequence).

NOTE:

In all the following tables, the first column, Item #, consists of the field sequence numbers which are not to be included in the record.

Item name	Definition of the item.

Required	If Y is entered, the field must have valid data in it and cannot be blank. If nothing is entered, the field is not required.

Description	Description of the item.

The total number of fields varies with record type as follows.

- Type 00: Cancellation Notice = 33 fields.
- Type 01: Item Available Notice = 34 fields.
- Type 02: Overdue Notice = 38 fields.
- Type 03: Recall Notice = 37 fields.
- Type 04: Recall-Overdue Notice = 38 fields.
- Type 05: Fine/Fee Notice = 39 fields.
- Type 06: Statement of Fines/Fees = 39 fields.
- Type 07: Courtesy (due) Notice = 37 fields.

Base Segment

[Table 26-1](#) defines the base segment for all circulation notice type records.

Table 26-1. Base segment for circulation notices

Item #	Required	Item Name	Length	Description
1	Y	notice id	2	This field must be 2 characters in length and contain 1 of the following codes indicating the type of notice: 00 is a cancellation notice 01 is an item available notice 02 is an overdue notice 03 is a recall notice 04 is a recall overdue notice 05 is a fine/fee notice 06 is a statement of fines/fees 07 is a courtesy (due) notice
2	Y	version number	30	Program version number to be used by client for version checking.
3		e-mail address	50	Patron's e-mail address. If this field is not blank, the notice will be sent by e-mail. If it is blank, the notice will be printed for mailing.
4	Y	patron id	10	Patron's ID number.
5	Y	last name	30	Patron's last name.
6		first name	20	Patron's first name.
7		title	20	Patron's title.
8		address line 1	50	Patron's address line 1.
9		address line 2	40	Patron's address line 2.
10		address line 3	40	Patron's address line 3.
11		address line 4	40	Patron's address line 4.
12		address line 5	40	Patron's address line 5.
13		city	30	Patron's city.
14		state/province	7	Patron's state/province.
15		postal code	10	Patron's postal code.

Table 26-1. Base segment for circulation notices

Item #	Required	Item Name	Length	Description
16		country	20	Patron's country.
17		phone	25	Patron's phone.
18	Y	date	10	Current date.
19	Y	institution	50	Institution name.
20		library	25	Library location name.
21		address line 1	50	Library's address line 1.
22		address line 2	40	Library's address line 2.
23		address line 3	40	Library's address line 3.
24		city	30	Library's city.
25		state/province	7	Library's state/province.
26		postal code	10	Library's postal code.
27		country	20	Library's country.
28		phone	25	Library's phone.
29		item title	255	Item title.
30		item author	255	Item author.
31		item id	25	Item barcode.
32		item call #	255	Item call number.
33		enum/chron	255	Enum/chron. This is the last common item and the <u>last item in the record</u> for circulation notices.

Suffixes

[Table 26-2](#) through [Table 26-9](#) cover the various suffix segments of the circulation notices.

Cancellation Notice Suffix (00)

[Table 26-2](#) describes the cancellation notice suffix.

Table 26-2. Cancellation Notice Suffix (00)

Item #	Required	Item Name	Length	Description
--		note	--	All items are included in the common portion of record. There is nothing unique for cancellation notices.

Item Available Notice Suffix (01)

[Table 26-3](#) describes the item available notice suffix.

Table 26-3. Item Available Notice Suffix (01)

Item #	Required	Item Name	Length	Description
34		expiration date	10	Expiration date. This is the <u>last item in the record</u> for item available notices.

Overdue Notice Suffix (02)

[Table 26-4](#) describes the overdue notice suffix.

Table 26-4. Overdue Notice Suffix (02)

Item #	Required	Item Name	Length	Description
34	Y	due date	10	Due date.
35		sequence	2	Notice sequence number.
36		proxy patron last name	30	Proxy patron patron's last name.
37		proxy patron first name	20	Proxy patron patron's first name.
38		proxy patron title	20	Proxy patron's title. This is the <u>last item in the record</u> for overdue notices.
39		copy number	5	Contains the copy number of the item that is overdue.

Recall Notice Suffix (03)

[Table 26-5](#) describes the recall notice suffix.

Table 26-5. Recall Notice Suffix (03)

Item #	Required	Item Name	Length	Description
34	Y	due date	10	Due date.
35		proxy patron last name	30	Proxy patron patron's last name.
36		proxy patron first name	20	Proxy patron patron's first name.
37		proxy patron title	20	Proxy patron's title.

Recall-Overdue Notice Suffix (04)

[Table 26-6](#) describes the recall-overdue notice suffix.

Table 26-6. Recall-Overdue Notice Suffix (04)

Item #	Required	Item Name	Length	Description
34	Y	due date	10	Due date.
35		sequence	2	Notice sequence number.
36		proxy patron last name	30	Proxy patron patron's last name.
37		proxy patron first name	20	Proxy patron patron's first name.
38		proxy patron title	20	Proxy patron's title. This is the <u>last item in the record</u> for recall-overdue notices.

Fine/Fee Notice Suffix (05)

[Table 26-7](#) describes the fine/fee notice suffix.

Table 26-7. Fine/Fee Notice Suffix (05)

Item #	Required	Item Name	Length	Description
34	Y	fine/fee date	10	Date of this fine/fee.
35	Y	fine/fee description	25	This is the description of a fine/fee due.

Table 26-7. Fine/Fee Notice Suffix (05)

Item #	Required	Item Name	Length	Description
36		fine/fee amount	10	Total amount of fine/fee described above.
37		fine/fee balance	10	Net amount of fine/fee described above.
38		previously billed	10	Previously billed amounts total.
39		total fines/fees	10	Total of all fines/fees. This is the <u>last item in the record</u> for notice of fines and fees.
40		due date	10	Contains the due date for the item associated with the fine/fee.
41		due date when fined	10	Contains the due date for the item when the fine was incurred.

Statement of Fines and Fees Suffix (06)

[Table 26-8](#) describes the statement of fines and fees notice suffix.

Table 26-8. Statement of Fines and Fees Suffix (06)

Item #	Required	Item Name	Length	Description
34	Y	fine/fee date	10	Date of this fine/fee.
35	Y	fine/fee description	25	This is the description of a fine/fee due.
36		fine/fee amount	10	Total amount of fine/fee described above.
37		fine/fee balance	10	Net amount of fine/fee described above.
38		previously billed	10	Previously billed amounts total.
39		total fines/fees	10	Total of all fines/fees. This is the <u>last item in the record</u> for statement of fines and fees.
40		due date	10	Contains the due date for the item associated with the fine/fee.
41		due date when fined	10	Contains the due date for the item when the fine was incurred.

Courtesy (Due) Notice Suffix (07)

[Table 26-9](#) describes the courtesy (due) notice suffix.

Table 26-9. Courtesy (Due) Notice Suffix (07)

Item #	Required	Item Name	Length	Description
34	Y	due date	10	Due date.
35		proxy patron last name	30	Proxy patron's last name.
36		proxy patron first name	20	Proxy patron's first name.
37		proxy patron title	20	Proxy patron's title. This is the <u>last item in the record</u> for courtesy (due) notices.

Introduction

The Circulation Reports Standard Interface File (SIF) is the file produced by running circulation batch jobs on the server.

Running the circulation batch jobs produces a file containing the report information in a specific format. This file is then used as an input file (`.inp`) for the Reporter module to use to generate the reports. See *Creating Input Files for Circulation* in the *Voyager Reporter User's Guide* for more information.

Purpose of This Chapter

This section discusses the layout for the Circulation Report SIF.

- File specification
- File format base segment
- File format suffix segment

File Specification

The name of the file produced is `crcrprts.[print location code].inp`, where `xxxx` represents the appropriate printing location code as defined in "Print Locations" in the System Administration module. See *Print Locations* in the

Voyager System Administration User's Guide for more information. If the print location code is main, the file will be called `crcrprts.main.inp`. When the file is generated, it is placed in the `/ml/voyager/xxxdb/rpt` directory.

Each record in the SIF is made up of two pieces: a base segment and a suffix.

The base segment is used in every record in the circulation reports SIF and is always at the beginning of the record. After the base segment, one of the suffixes will be attached. Depending on the type of report for which the record contains data, a different suffix will be attached. A record will only contain one base segment and one suffix.

For example, a sample circulation report record might appear as follows:

```
03|97.2|09/16/1999||TL230 .A523|America's light
trucks.|39550000173749|main|07/30/1999
```

The first three fields of the record are the base segment. Because this is a Missing in Transit report record, the suffix is made up of the last six fields. The type of suffix attached determines the record type.

The record type can be determined by examining the first field of the record. The two-digit number indicates the type of report. The report numbers are listed in the heading for each suffix type.

All fields must be present and pipe-delimited. The pipe for a given field must be present even if the field is empty.

Each record will be a line-feed-terminated string.

Maximum field lengths as imposed below must be adhered to.

NOTE:

This SIF remains in the Latin-1 character set. If elements in the SIF were stored in the database using the Unicode character set, they are converted to Latin-1 before being exported to the SIF.

Circulation Reports SIF Format

The following conventions are used in the listed tables.

Item #	Represents the relative position of the item in the record (sequence).

NOTE:

In all the following tables, the first column, Item #, consists of the field sequence numbers which are not to be included in the record.

Item name Definition of the item.

Required If Y is entered, the field must have valid data in it and cannot be blank. If nothing is entered, the field is not required.

Description Description of the item.

The total number of fields varies with record type as follows.

- Type 00: Reserved Items Active Report = 9 fields.
- Type 01: Reserved Items Expired Report = 9 fields.
- Type 02: Hold Shelf Expired Report = 11 fields.
- Type 03: Missing in Transit Report = 8 fields.
- Type 04: Circ Transactions Statistics Rpt = 9 fields.
- Type 05: Circ Item-related Exceptions Rpt = 9 fields
- Type 06: Circ Patron-related Exceptions Rpt = 9 fields.
- Type 07: Circ Transactn-related Exceptions Rpt = 11 fields.
- Type 08: Global Circ Transactions Statistics = 9 fields.
- Type 09: Distribution Item Order Rpt = 22 fields.

Base Segment

[Table 27-1](#) defines the base segment for all circulation report type records.

Table 27-1. Base segment for circulation reports

Item #	Required	Item Name	Length	Description
1	Y	report id	2	This field must be 2 characters in length and contain 1 of the following codes indicating the type of notice: 00 is a reserved items active report. 01 is a reserved items expired report. 02 is a hold shelf expired report. 03 is a missing in transit report. 04 is a circ transaction statistics report. 05 is a circ item-related override/exception. 06 is a circ patron-related override/exception. 07 is a circ transaction-related override/exception. 08 is a global circ transaction statistics report. 09 is a distribution item order list
2	Y	version number	30	Program version number to be used by client for version checking.
3		date[time]/selection (range)	50	Date (mm/dd/ccyy) with optional time (hh:mm) or a range of dates with optional times. Only relevant to reserved items (reports 00 and 01) and circ transaction stats (reports 04 and 08).

Suffixes

[Table 27-2](#) through [Table 27-11](#) cover the various suffix segments of the circulation reports.

Reserved Items Active Report (00)

[Table 27-2](#) describes the reserved items active report suffix.

Table 27-2. Reserved Items Active Report Suffix (00)

Item #	Required	Item Name	Length	Description
4	Y	location	10	Location code.
5		call number	144	Call number.
6	Y	title	100	Title.
7		item barcode	25	Item barcode.
8	Y	reserve list name	40	Reserve list name.
9	Y	effective date	10	Effective date.

Reserved Items Expired Report (01)

[Table 27-3](#) describes the reserved items expired report suffix.

Table 27-3. Reserved Items Expired Report Suffix (01)

Item #	Required	Item Name	Length	Description
4	Y	location	10	Location code.
5		call number	255	Call number.
6	Y	title	100	Title.
7		item barcode	25	Item barcode.
8	Y	reserve list name	40	Reserve list name.
9	Y	expiration date	10	Expiration date.

Hold Shelf Expired Report (02)

[Table 27-4](#) describes the hold shelf expired report suffix.

Table 27-4. Hold Shelf Expired Report Suffix (02)

Item #	Required	Item Name	Length	Description
4	Y	location	10	Location code.
5		call number	255	Call number.
6	Y	title	100	Title.

Table 27-4. Hold Shelf Expired Report Suffix (02)

Item #	Required	Item Name	Length	Description
7		item barcode	25	Item barcode.
8	Y	expiration date	10	Date reserve expires.
9	Y	patron's first name	20	Patron's first name (Patron.first_name).
10	Y	patron's middle name	20	Patron's middle name (Patron.middle_name).
11	Y	patron's last name	20	Patron's last name (Patron.last_name).

Missing in Transit Report (03)

[Table 27-5](#) describes the missing in transit report suffix.

Table 27-5. Missing in Transit Report Suffix (03)

Item #	Required	Item Name	Length	Description
4		source location	10	Location code where item is coming from.
5		call number	255	Call number.
6	Y	title	255	Title.
7		item barcode	25	Item barcode.
8		target location	10	Location code where item is going to.
9	Y	transit date	10	Transit date.

Circulation Statistics Report (04)

[Table 27-6](#) describes the circulation statistics report suffix.

Table 27-6. Circulation Statistics Report Suffix (04)

Item #	Required	Item Name	Length	Description
4	Y	location	10	Circulation location code.
5	Y	operator id	10	Operator ID.
6		charges	10	Number of charges.
7		discharges	10	Number of discharges.
8		renewals	10	Number of renewals.

Table 27-6. Circulation Statistics Report Suffix (04)

9		fines	12	Amount of fines collected.

Circulation Item-Related Exceptions Report (05)

[Table 27-7](#) describes the circulation item-related exceptions report suffix.

Table 27-7. Circulation Item--related Exceptions Report Suffix (05)

Item #	Required	Item Name	Length	Description
4	Y	location	10	Home location code.
5		description	50	Exception description.
6	Y	title	255	Title.
7		item barcode	25	Item barcode.
8	Y	date	10	Exception date.
9	Y	operator id	10	Operator ID.

Circulation Patron-Related Exceptions Report (06)

[Table 27-8](#) describes the circulation patron-related exceptions report suffix.

Table 27-8. Circulation Patron-related Exceptions Report Suffix (06)

Item #	Required	Item Name	Length	Description
4	Y	location	10	Home location code.
5	Y	description	50	Exception description.
6	Y	patron name	50	Patron name.
7		patron id	25	Patron barcode.
8	Y	date	10	Exception date.
9	Y	operator id	10	Operator ID.

Circulation Transaction-Related Exceptions Report (07)

[Table 27-9](#) describes the circulation transaction-related exceptions report suffix.

Table 27-9. Circulation Transaction-related Exceptions Report Suffix (07)

Item #	Required	Item Name	Length	Description

Table 27-9. Circulation Transaction-related Exceptions Report Suffix (07)

4	Y	location	10	Home location code.
5	Y	description	50	Exception description.
6	Y	patron name	50	Patron name.
Item #	Required	Item Name	Length	Description
7		patron id	25	Patron barcode.
8	Y	title	255	Title.
9		item barcode	25	Item barcode.
10	Y	date	10	Exception date.
11	Y	operator id	10	Operator ID.

Global Circulation Statistics Report (08)

[Table 27-10](#) describes the global circulation statistics report suffix.

Table 27-10. Global Circulation Statistics Report Suffix (08)

Item #	Required	Item Name	Length	Description
4	Y	location	10	Circulation location code.
5	Y	operator id	10	Operator ID.
6		charges	10	Number of charges.
7		discharges	10	Number of discharges.
8		renewals	10	Number of renewals.
9		finer	12	Amount of fines collected.

NOTE:

Report 08, Global Circ Transactions Statistics, is a duplicate of report 04 except that it includes all locations while all other reports only include the happening locations assigned to the particular Print Location which is issuing the report.

Distribution Item Order List Report (09)

[Table 27-11](#) describes the distribution order list report suffix.

Table 27-11. Distribution Item Order List Report Suffix (09)

Item #	Required	Item Name	Length	Description
4	Y	vendor code	25	Vendor code.

Table 27-11. Distribution Item Order List Report Suffix (09)

5	Y	vendor_name	60	Vendor's name.
6		address line 1	50	Vendor's address line 1.
7		address line 2	50	Vendor's address line 2.
Item #	Required	Item Name	Length	Description
8		address line 3	50	Vendor's address line 3.
9		address line 4	50	Vendor's address line 4.
10		city	30	Vendor's city.
11		state/province	7	Vendor's state/province.
12		postal code	10	Vendor's postal code.
13		country	20	Vendor's country.
14	Y	item title	255	Item title.
15		item author	255	Item author.
16	Y	item barcode	25	Item barcode.
17		item call #	255	Item call number.
18		enum/chron	255	Enum/chron.
19	Y	order date	10	Order date.
20	Y	quantity	10	Number of items to order.
21	Y	expected date	10	Expected date.
22	Y	operator	10	Operator ID. This is the last common item and the <u>last item in the record</u> for distribution item order list report.

Introduction

The Media Scheduling Notices Standard Interface File (SIF) is the file produced by activity at media scheduling locations and then running media scheduling batch jobs on the server.

Running the media scheduling batch job produces a file containing the notice information in a specific format. This file is then used as an input file (`.inp`) for the Reporter module to use to generate the notice. See *Creating Input Files for Media Scheduling* in the *Voyager Reporter User's Guide* for more information.

Purpose of This Chapter

This section discusses the layout for the Media Scheduling Notices SIF.

- File specification
- File format base segment
- File format suffix segment

File Specification

The name of the file produced is `mednotes.[print location code].inp`, where `xxxx` represents the appropriate printing location code as defined in *Print Locations* in the System Administration module. See *Print Locations* in the *Voyager System Administration User's Guide* for more information. If the print location code is `main`, the file will be called `mednotes.main.inp`. When the file is generated, it is placed in the `/m1/voyager/xxxdb/rpt` directory.

Each record in the SIF is made up of two pieces: a base segment and a suffix.

The base segment is used in every record in the media scheduling notices SIF and is always at the beginning of the record. After the base segment, one of the suffixes will be attached. Depending on the type of notice for which the record contains data, a different suffix will be attached. A record will only contain one base segment and one suffix.

For example, a sample media scheduling notice record might appear as follows:

```
00|97.2|mayer@endinfosys.com|8548742|Mayer|Henry||123
8th Drive #843||||Chicago|IL|60000||(847) 296-2200|10/
19/99| Institution of Laughing Person|Main
Library|||||||(847)296-2200 x 4621|Women
jogging.|John K. Skrupnid|87943243||||||Random House
Video/Media|||5t5698|09/02/99 19:54|09/02/99 22:00|9
```

The first 28 fields of the record are the base segment. Because this is an overdue notice record, the suffix is made up of the last 16 fields. The type of suffix attached determines the record type.

The record type can be determined by examining the first field of the record. The two-digit number indicates the type of notice. The notice numbers are listed in the heading for each suffix type.

All fields must be present and pipe-delimited. The pipe for a given field must be present even if the field is empty.

Each record will be a line-feed-terminated string.

Maximum field lengths as imposed below must be adhered to.

NOTE:

This SIF remains in the Latin-1 character set. If elements in the SIF were stored in the database using the Unicode character set, they are converted to Latin-1 before being exported to the SIF.

Media Scheduling Notices SIF Format

The following conventions are used in the listed tables.

Item # Represents the relative position of the item in the record (sequence).

NOTE:

In all the following tables, the first column, Item #, consists of the field sequence numbers which are not to be included in the record.

Item name Definition of the item.

Required If Y is entered, the field must have valid data in it and cannot be blank. If nothing is entered, the field is not required.

Description Description of the item.

The total number of fields in the notice record is as follows.

- Type 00: Overdue Notice = 44 fields.

Base Segment

[Table 28-1](#) defines the base segment for the media scheduling notice type records.

Table 28-1. Base Segment for Media Scheduling Notices

Item #	Required	Item Name	Length	Description
1	Y	notice id	2	This field must be 2 characters in length and contain the following code indicating the type of notice: 00 is an overdue notice.
2	Y	version number	30	Program version number to be used by client for version checking.
3		e-mail address	50	Patron's e-mail address. If this field is not blank, the notice will be sent by e-mail. If it is blank, the notice will be printed for mailing.
4	Y	patron id	10	Patron's id number.
5	Y	last name	30	Patron's last name.

Table 28-1. Base Segment for Media Scheduling Notices

Item #	Required	Item Name	Length	Description
6		first name	20	Patron's first name.
7		title	20	Patron's title.
8		address line 1	50	Patron's address line 1.
9		address line 2	40	Patron's address line 2.
10		address line 3	40	Patron's address line 3.
11		address line 4	40	Patron's address line 4.
12		address line 5	40	Patron's address line 5.
13		city	30	Patron's city.
14		state/province	7	Patron's state/province.
15		postal code	10	Patron's postal code.
16		country	20	Patron's country.
17		phone	25	Patron's phone.
18	Y	date	10	Current date.
19	Y	institution	50	Institution name.
20		library	25	Library location name.
21		address line 1	50	Library address line 1.
22		address line 2	40	Library address line 2.
23		address line 3	40	Library address line 3.
24		city	30	Library city.
25		state/province	7	Library state/province.
26		postal code	10	Library postal code.
27		country	20	Library country.
28		phone	25	Library phone.

Suffixes

[Table 28-2](#) covers the suffix segment of the media scheduling notice.

Overdue Notice Suffix (00)

[Table 28-2](#) describes the overdue notice suffix.

Table 28-2. Overdue Notice Suffix (00)

Item #	Required	Item Name	Length	Description
29		item title	255	Item title.
30		item author	255	Item author.
31		item id	25	Item barcode.
32		item call #	255	Item call number.
33		enum/chron	255	Enum/chron.
34		equipment type	255	Equipment type.
35		equipment no	15	Unit/group ID.
36		equipment id	25	Equipment ID.
37		barcode	25	Unit/group barcode.
38		manufacturer	100	Equipment manufacturer.
39		model	100	Equipment model designation.
40		serial number	100	Equipment serial number.
41	Y	confirmation no.	10	Booking confirmation number.
42		booking start	25	Booking end date/time.
43		booking end	25	Booking end date/time.
44		sequence	2	Notice sequence number. This is the <u>last item in the record</u> for overdue notices.
45		copy number	5	Contains the copy number of the item that is overdue.

Introduction

The Media Scheduling Reports Standard Interface File (SIF) is the file produced by running media scheduling batch jobs on the server.

Running media scheduling batch jobs produce a file containing the report information in a specific format. This file is then used as an input file (`.inp`) for the Reporter module to use to generate the reports. See *Creating Input Files for Media Scheduling* in the *Voyager Reporter User's Guide* for more information.

Purpose of This Chapter

This section discusses the layout for the Media Scheduling Report SIF.

- File specification
- File format base segment
- File format suffix segment

File Specification

The name of the file produced is `medrprts.[print location code].inp`, where `xxxx` represents the appropriate printing location code as defined in "Print Locations" in the System Administration module. See *Print Locations* in the

Voyager System Administration User's Guide for more information. If the print location code is main, the file will be called `medrpts.main.inp`. When the file is generated, it is placed in the `/ml/voyager/xxxdb/rpt` directory

Each record in the SIF is made up of two pieces: a base segment and a suffix.

The base segment is used in every record in the media scheduling reports SIF and is always at the beginning of the record. After the base segment, one of the suffixes will be attached. Depending on the type of report for which the record contains data, a different suffix will be attached. A record will only contain one base segment and one suffix.

For example, a sample media scheduling report record might appear as follows:

```
00|97.2||TEST|Television|media|101|Classrooms|MED101|7|
|2|TV002|122||0|Charged||||
```

The first three fields of the record are the base segment. Because this is a media equipment inventory report record, the suffix is made up of the last 18 fields. The type of suffix attached determines the record type.

The record type can be determined by examining the first field of the record. The two-digit number indicates the type of report. The report numbers are listed in the heading for each suffix type.

All fields must be present and pipe-delimited. The pipe for a given field must be present even if the field is empty.

Each record will be a line-feed-terminated string.

Maximum field lengths as imposed below must be adhered to.

Media Scheduling Reports SIF Format

The following conventions are used in the listed tables.

Item #	Represents the relative position of the item in the record (sequence).

NOTE:

In all the following tables, the first column, Item #, consists of the field sequence numbers which are not to be included in the record.

Item name	Definition of the item.

Required If Y is entered, the field must have valid data in it and cannot be blank. If nothing is entered, the field is not required.

Description Description of the item.

The total number of fields varies with record type as follows.

- Type 00: Media Equipment Inventory Report = 21 fields.
- Type 01: Media Scheduling Statistics Report = 8 fields.
- Type 02: Media Scheduling Exceptions Report = 9 fields.
- Type 03: Media Scheduling Charge Statistics Report = 14 fields.

Base Segment

[Table 29-1](#) defines the base segment for all media scheduling report type records.

Table 29-1. Base Segment for Media Scheduling Reports

Item #	Required	Item Name	Length	Description
1	Y	report id	2	This field must be 2 characters in length and contain 1 of the following codes indicating the type of notice: 00 is a media equipment inventory report. 01 is a booking statistics report. 02 is a booking exceptions report. 03 is a booking charge statistics report.
2	Y	version number	30	Program version number to be used by client for version checking.
3		date[time]/selection (range)	50	Date (mm/dd/ccyy) with optional time (hh:mm) or a range of dates with optional times.

Suffixes

[Table 29-2](#) through [Table 29-5](#) cover the various suffix segments of the media scheduling reports.

Media Equipment Inventory Report (00)

[Table 29-2](#) describes the media equipment inventory report suffix.

Table 29-2. Media Equipment Inventory Report Suffix (00)

Item #	Required	Item Name	Length	Description
4	Y	policy group	50	Policy group.
5	Y	equipment type	50	Equipment type.
6	Y	location name	25	Equipment location name.
7	Y	room number	15	Equipment room number.
8	Y	room type	50	Equipment room type code.
9		room name	100	Equipment room name.
10		group id	25	Equipment group ID.
11		group number	15	Equipment group number.
12	Y	equipment id	25	Equipment ID.
13	Y	equipment number	15	Equipment number.
14		equipment barcode	25	Equipment barcode.
15		acquisition date	10	Acquisition date.
16	Y	equipment value	12	Equipment value.
17	Y	equipment status	100	Equipment status.
18		manufacturer	100	Equipment manufacturer.
19		model	100	Model number/description.
20		equipment serial no.	100	Equipment serial number.
21		last maintenance date	10	Date of last maintenance.

Booking Statistics Report (01)

[Table 29-3](#) describes the booking statistics report suffix.

Table 29-3. Booking Statistics Report Suffix (01)

Item #	Required	Item Name	Length	Description
4	Y	location	10	Media Scheduling location code.

Table 29-3. Booking Statistics Report Suffix (01)

5	Y	operator id	10	Operator ID.
6	Y	bookings made	10	Number of bookings made.
7	Y	bookings charged	10	Number of charges.
8	Y	bookings discharged	10	Number of discharges.
9	Y	bookings canceled	10	Number of cancellations.

Booking Exceptions Report (02)

[Table 29-4](#) describes the booking exceptions report suffix.

Table 29-4. Booking Exceptions Report Suffix (02)

Item #	Required	Item Name	Length	Description
4	Y	location	10	Home location code.
5	Y	description	25	Exception description.
6	Y	equipment type	255	Equipment type.
7		equipment bar-code	25	Equipment barcode.
8	Y	date	10	Exception date.
9	Y	operator id	10	Operator ID.

Booking Charge Statistics Report (03)

[Table 29-5](#) describes the booking charge statistics report suffix.

Table 29-5. Booking Charge Statistics Report Suffix (03)

Item #	Required	Item Name	Length	Description
4	Y	location	10	Media scheduling location code.
5	Y	operator id	10	Operator ID.
6	Y	staff delivery	10	Number of staff deliveries.
7	Y	delivery/return items	10	Number of items delivered.

Table 29-5. Booking Charge Statistics Report Suffix (03)

8		delivery/return equipment	10	Number of equipment items delivered.
9	Y	patron picked-up	10	Number of patron pick-ups.
10	Y	pickup items	10	Number of items picked-up.
11	Y	pickup equip	10	Number of equipment items picked-up.
12	Y	schedule room	10	Number of rooms scheduled.
13	Y	library items	10	Number of library items.
14	Y	library equip	10	Number of library equipment items.

Introduction

The Voyager database views provide you with a simplified way of retrieving data from the database. A view is like a table that, instead of holding new information, collects specific related data from a number of sources in the database.

A view, however, does not actually copy the information to a second location when it is created. Views are a part of SQL - each view is an SQL script. Each view contains an SQL query that draws the appropriate information from the database.

You can view (query) different sections of different tables at the same time. When you enter an SQL query on the view name, it finds the description of the view and returns with the table results.

Using SQL Plus, the following is an example of what you might enter to display the results of a Authheading_vw.

Enter: *select * from Authheading_vw*

These commands will be different depending on what SQL editor you use.

Purpose of This Chapter

This chapter includes Views tables.

Views

The following information is described about each table:

Field Name The name of the field as defined by the view.

Description and Use General information.

Report Whether this would be useful in a report.

Query Select Whether this would be useful as a criterion for querying based on this field.

Normalized Sort Whether the data in this field can be used to sort in some meaningful way.

Link to Field Whether this field would be helpful in linking this view to other tables in the database. An asterisk (*) indicates that this field may be linked to a table in another database.

Authblob_vw

This view provides access to the entire MARC authority record (see [Table 30-1](#)).

Table 30-1. Authblob_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Auth_id	Authority ID number	Y	Y		Y
Marc_record	Whole MARC authority record in its native format, contained in a single row	Y	N	N	N/A

NOTE:

Normally you will run MARC record parsing functions to access a particular field in the record.

Authhistory_vw

This view, provides information on the create and last update history of authority records (see [Table 30-2](#)).

Table 30-2. Authhistory_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Auth_id	Authority ID number	Y	Y		Y
Create_operator_id	ID of operator who created the record	Y	Y	Y	Y
Create_date	Date and time of record creation	Y	Y	Y	
Create_location_id	ID of happening location where record was created		Y		Y
Update_operator_id	ID of operator that updated the record	Y	Y	Y	Y
Update_date	Date and time of record update	Y	Y	Y	
Update_location_id	ID of happening location where record was updated		Y		Y

Authheading_vw

This view is used by the heading_vw. It may be of limited use to the end-user (see [Table 30-3](#)).

Table 30-3. Authheading_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Heading_id_pointer	The heading pointed to by the reference				Y
Heading_id_pointee	The heading pointed from by the reference				Y
Auth_id	Authority record that established this reference				Y

Table 30-3. Authheading_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Reference_type	Cross reference type	Y	Y	Y	

Authority1xx4xx_vw

This view can be used to make a report for authority maintenance purposes (see [Table 30-4](#)).

Table 30-4. Authority1xx4xx_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Index_type	The type of the heading index. (name, subject)	Y	Y	Y	
Auth_id_1XX	The authorized heading authority record ID	Y	Y	Y	Y
Auth_id_4XX	The see from heading authority record ID	Y	Y	Y	Y
Staffbibs	Count of all bibliographic records using this heading	Y	Y	Y	
Opacbibs	Count of non-suppressed bibliographic records using this heading	Y	Y	Y	Y
Display_heading	The heading, including subfield markers	Y	Y	Y	

Authority5xx1xx_vw

This view can be used to make a report for authority maintenance purposes (see [Table 30-5](#)).

Table 30-5. Authority5xx1xx_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Index_type	The type of the heading index. (name, subject)	Y	Y	Y	
Auth_id_5XX	The see-also from heading authority record ID	Y	Y	Y	Y

Table 30-5. Authority5xx1xx_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Display_heading	The heading, including subfield markers	Y	Y	Y	

Authoritydupe_vw

This view can be used to make a report for authority maintenance purposes. These records may be complete duplicate records or different authority records which may be authorizing the same heading for use in the same heading-index type (see [Table 30-6](#)).

Table 30-6. Authoritydupe_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Auth_id	The authority record ID	Y	Y	Y	Y
Display_heading	The heading	Y	Y	Y	

Bib_vw

This is a very powerful view that links the bibliographic record to the holdings record if one exists. This view will have a row for every MFHD in the database and every bib record that does not have a MFHD. It can be linked to the bib_text or bib_id for additional bibliographic information to display in a report (see [Table 30-7](#)).

Table 30-7. Bib_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Title	Title Statement (245 a and b)	Y			
Sort_title	Normalized title for sorting a report by title non-filing words removed, punctuation removed		Y	Y	
Bib_id	The bibliographic record ID	Y	Y	Y	Y

Table 30-7. Bib_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Create_date	The bibliographic record create date	Y	Y	Y	
Create_opid	The bibliographic record create operator	Y	Y	Y	Y
Mfhd_id	Mfhd_id attached to bib record. This field will be blank if no MFHDs are attached to the bib record. If more than one MFHD is attached to a bib record, there will be one row for each MFHD in the table	Y	Y	Y	Y
Call_no	The call number for the MFHD	Y			
Call_no_type	The MFHD indicator of call number type- Dewey, LC	Y	Y	Y	Y
Normalized_call_no	The normalized call number		Y	Y	
Mfhd_location_id	The internal location id of the MFHD				Y
Mfhd_location_code	The MFHD location code	Y	Y	Y	
Mfhd_location_name	The MFHD location name	Y		Y	
Mfhd_create_date	The MFHD record create date	Y	Y	Y	
Mfhd_create_opid	The MFHD record create operator	Y	Y	Y	Y

Bibblob_vw

This view provides access to the entire MARC bibliographic record (see [Table 30-8](#)).

Table 30-8. Bibblob_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Bib_id	Bibliographic ID number	Y	Y		Y
Marc_record	Whole MARC bibliographic record in its native format, contained in a single row	Y	N	N	N/A

NOTE:

Normally you will run MARC record parsing functions to access a particular field in the record.

Bibhistory_vw

This view provides information on the create and last update history of bibliographic records (see [Table 30-9](#)).

Table 30-9. Bibhistory_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Bib_id	Bibliographic ID number	Y	Y		Y
Create_operator_id	ID of operator who created the record	Y	Y	Y	Y
Create_date	Date and time of record creation	Y	Y	Y	
Create_location_id	ID of happening location where record was created		Y		Y
Update_operator_id	ID of operator that updated the record	Y	Y	Y	Y
Update_date	Date and time of record update	Y	Y	Y	

Table 30-9. Bibhistory_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Update_location_id	ID of happening location where record was updated		Y		Y

Bibloc_vw

This view extracts the coded location from the bib 008 field (see [Table 30-10](#)).

Table 30-10. Bibloc_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Bib_id	Bib ID number	Y	Y		Y
Marcloccode	Three-character MARC location code	Y	Y	Y	*

Circcharges_vw

This report brings together circulation statistics from current charges and archive charges (see [Table 30-11](#))

Table 30-11. Circcharges_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Patron_group_id	The internal number of the patron group types				Y
Patron_group_code	The patron group code	Y	Y	Y	
Patron_group_name	The patron group name	Y		Y	
Item_id	The internal ID number of the item	Y	Y	Y	Y

Table 30-11. Circcharges_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Mfhd_id	Mfhd_id attached to bib record. This field will be blank if no MFHDs are attached to the bib record. If more than one MFHD is attached to a bib record, there will be one row for each MFHD in the table	Y	Y	Y	Y
Bib_id	Bib ID number	Y	Y		Y
Perm_location_code	The permanent location code of the item being charged	Y	Y	Y	
Perm_location	The permanent location of the item	Y		Y	
Gov_location_code	The temporary location code of the item if it exists or else the permanent location	Y	Y	Y	
Gov_location	The temporary location code of the item if it exists, or else the permanent location code	Y		Y	Y
Perm_item_type_code	The permanent item type code		Y		
Perm_item_type	The permanent item type	Y	Y	Y	
Gov_item_type_code	The temporary item type if it exists or else the permanent item type code		Y		
Gov_item_type	The temporary item type if it exists or else the permanent item type	Y	Y	Y	

Table 30-11. Circcharges_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Charge_date	Date item was charged	Y	Y	Y	
Charge_date_time	Date and time item was charged	Y	Y	Y	
Charge_date_only	Date item was charged	Y	Y	Y	
Charge_oper_id	Operator ID of the person who made the charge	Y	Y	Y	
Charge_location	ID of circulation desk where item was charged out				Y
Charge_location_code	Location code of circulation desk		Y		
Charge_location_name	Location name of circulation desk	Y		Y	
Renewal_count	Item renewal count	Y	Y	Y	
Notice_count	Number of notices sent including overdue recall notices	Y	Y	Y	

Circrenewal_vw

This view brings together circulation statistics from charged (archived) items that have been renewed (see [Table 30-12](#)).

Table 30-12. Circrenewal_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Patron_group_id	The internal number of the patron group types				Y
Patron_group_code	The patron group code	Y	Y	Y	
Patron_group_name	The patron group name	Y		Y	
Item_id	The internal ID number of the item	Y	Y	Y	Y

Table 30-12. Circrenewal_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Mfhd_id	Mfhd_id attached to bib record. This field will be blank if no MFHDs are attached to the bib record. If more than one MFHD is attached to a bib record, there will be one row for each MFHD in the table				
Bib_id	Bib ID number	Y	Y		Y
Perm_location_code	The permanent location code of the item being charged	Y	Y	Y	
Perm_location	The permanent location of the item	Y		Y	
Gov_location_code	The temporary location code of the item if it exists or else the permanent location code	Y	Y	Y	Y
Gov_location	The temporary location of the item if it exists or else the permanent location	Y		Y	Y
Perm_item_type_code	The permanent item type code		Y		
Perm_item_type	The permanent item type	Y	Y	Y	
Gov_item_type_code	The temporary item type code if it exists or else the permanent item type code		Y		
Gov_item_type	The temporary item type if it exists or else the permanent item type	Y	Y	Y	
Charge_date_time	Date and time item was charged	Y	Y	Y	

Table 30-12. Circrenewal_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Charge_date_only	Date item was charged	Y	Y	Y	
Charge_oper_id	Operator ID of the person who made the charge	Y	Y	Y	
Charge_location	ID of circulation desk where item was charged out				Y
Charge_location_code	Location code of circulation desk		Y		
Charge_location_name	Location name of circulation desk	Y		Y	
Renewal_count	Item renewal count	Y	Y	Y	
Renew_date_only	Date item was renewed	Y	Y	Y	
Renew_oper_id	Operator ID of the person who renewed the item	Y	Y	Y	
Renew_location_code	ID of circulation desk where item was renewed				Y
Location_name	Location name of circulation desk where item was renewed	Y		Y	

Fundledger_vw

This view summarizes funds (see [Table 30-13](#)).

Table 30-13. Fundledger_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Fundline	The name of the fund and all funds that connect it back to the main fund			Y	

Table 30-13. Fundledger_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Fiscal_period_id	The ID of the fiscal period that the ledger and fund is in		Y		Y
Fiscal_period_name	The name of the fiscal period, the ledger, and the fund	Y	Y	Y	
Fiscal_period_start	The start date of the fiscal period	Y	Y	Y	
Fiscal_period_end	The end date of the fiscal period	Y	Y	Y	
Ledger_id	The id of the ledger that the fund is in		Y		Y
Ledger_name	Ledger name	Y	Y	Y	Y
Normal_ledger_name	Normalized ledger name		Y	Y	
Policy_name	The name of the acquisitions policy associated with this ledger	Y	Y	Y	
Fund_type	Fund type as defined in System Administration	Y	Y	Y	
Fund_category	Fund category-effects the meaning of allocation, commitment and expenditure amounts	Y	Y	Y	
Fund_id	The ID of the fund		Y		Y
Fund_name	Fund name	Y	Y	Y	
Normal_fund_name	Normalized fund name		Y	Y	
Parent_fund_id	Will be 0 (zero) if this is in the first level of the ledger; otherwise, the ID number of the parent fund (the fund above)	Y	Y		Y
Parent_fund	Name of the parent fund-this will be blank if the parent fund is the ledger	Y	Y	Y	Y

Table 30-13. Fundledger_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Institution_id	Institution assigned ID-optional	Y	Y	Y	*
Begin_date	Fund beginning date	Y	Y	Y	
End_date	Fund ending date	Y	Y	Y	
Original_allocation	Original fund allocation in base currency	Y	Y	Y	
Current_allocation	Original allocation +/- fund transfers and adjustments in base currency	Y	Y	Y	
Cash Balance	Cash balance of fund in base currency	Y	Y	Y	
Free_balance	Free balance of fund in base currency	Y	Y	Y	
Expenditures	Total expenditures in base currency	Y	Y	Y	
Commitments	Total commitments in base currency	Y	Y	Y	
Commit_pending	Pending commitments in base currency	Y	Y	Y	
Expend_pending	Pending expenditures in base currency	Y	Y	Y	

Heading_vw

Pulls together the headings from the heading index and the associated authority records with bibliographic records that count (see [Table 30-14](#)).

Table 30-14. Heading_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Auth_id	The authority record for this heading if it exists	Y	Y		Y
Heading_id	The heading's ID number		Y		Y

Table 30-14. Heading_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Reference_type	The reference type of this heading if an authority record exists	Y	Y	Y	
Normal_heading	The normalized version of the heading		Y	Y	
Display_heading	The display version of the heading	Y			
Opacbibs	The count of records NOT suppressed from the OPAC using this heading	Y	Y	Y	
Create_date	The date that this heading was first added to the database, either by a bibliographic record or authority record	Y	Y	Y	
Index_type	The index of the heading-name, subject	Y	Y	Y	
Heading_type	The thesaurus or sub-index type. For example, corporate, personal, LCSH, MESH	Y	Y	Y	

Issues_vw

The issues_vw view combines predicted and unpredictable issues into a single table. It is used by serials_vw to give a complete view of serials check-in (see [Table 30-15](#)).

Table 30-15. Issues_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Issue_id	Issue id number-combined with the component_id to form a key				Y

Table 30-15. Issues_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Component_id	Component ID number-combined with issue_id to form a key				Y
Enumchron	The enumeration/chronology field of the issue	Y		Y	
Expected_date	The expected receipt date of the issue	Y	Y	Y	
Receipt_date	The date the issue was received or NULL if it has not been received	Y	Y	Y	
Received	The number of copies received	Y			

Item_vw

Pulls information from holding and item records to give a complete description of the item. The MFHD ID can be used to link this to retrieve bibliographic information (see [Table 30-16](#)).

Table 30-16. Item_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Mfhd_id	Item's parent MFHD	Y	Y		Y
Call_no	Call number for MFHD	Y			
Call_no_type	MFHD indicator of call number type-Dewey, LC	Y	Y	Y	Y
Normalized_call_no	Normalized call number		Y	Y	
Item_id	The internal ID number of the item	Y	Y		Y
Barcode	The active barcode of the item, if one is set	Y	Y	Y	*
Perm_location_code	The permanent location code of the item	Y	Y	Y	

Table 30-16. Item_yvw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Perm_location	The permanent location of the item	Y		Y	
Gov_location_code	The temporary location code of the item if it exists, or else the permanent location code	Y	Y	Y	
Gov_location	The temporary location code of the item if it exists, or else the permanent location code	Y		Y	Y
Perm_item_type_code	The permanent item type code		Y		
Perm_item_type	The permanent item type	Y	Y	Y	
Gov_item_type_code	The temporary item type code if it exists or else the permanent type code		Y		
Gov_item_type	The temporary item type if it exists, or else the permanent item type	Y	Y	Y	
Media_type_code	The media type code if the item has been assigned a media type	Y	Y	Y	
Media_type	The media type if this item has been assigned a media type	Y	Y	Y	
Enumeration		Y		Y	
Chronology		Y			
Year		Y		Y	
Caption		Y			
Historical_browses	The number of times this item has been browsed	Y	Y	Y	
Historical_charges	The number of times this item has been charged out	Y	Y	Y	

Table 30-16. Item_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Historical_bookings	the number of times this item has been booked in Media Scheduling	Y	Y	Y	
Hold_placed	The number of holds placed against this item	Y	Y	Y	
Recalls_placed	The number of recalls placed against this item	Y	Y	Y	
Create_date	The date this item record was created	Y	Y	Y	
Create_opid	The operator who created this record	Y	Y	Y	Y

LCclass_vw

Takes Library of Congress classification numbers and breaks them down into individual classes used to do reports based on call numbers, call number ranges (see [Table 30-17](#)).

Table 30-17. LCclass_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Mfhd_id	The MFHD ID	Y	Y		Y
Firstletter	The first letter of the class number	Y	Y	Y	*
Class	The first 1 to 3 characters in the normalized call number field	Y	Y	Y	*
Longclass	A 7-character string that includes the class letters and class number up to the decimal point. The number is right justified	Y	Y	Y	*
Classnumber	Numeric field with the class number, including decimal places	Y	Y + class		

Marccomputer_vw

This view gives information from the bib 008 for computer files (see [Table 30-18](#)).

Table 30-18. Marccomputer_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Bib_id	Bib record ID	Y	Y		Y
Bibtype	The bibliographic record type from the leader/06	Y	Y	Y	*
Biblevel	The bibliographic level from the leader/07	Y	Y	Y	*
Audience	Target audience 008/22	Y	Y	Y	*
Filetype	Type of computer file 008/26	Y	Y	Y	*
Governmentpub	Government publication 008/28	Y	Y	Y	*

Marcbook_vw

This view extracts key information from the bib 008 for books (see [Table 30-19](#)).

Table 30-19. Marcbook_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Bib_id	Bib record ID	Y	Y		Y
Bibtype	The bibliographic record type from the leader/06	Y	Y	Y	*
Biblevel	The bibliographic level from the leader/07	Y	Y	Y	*
Audience	Target audience 008/22	Y	Y	Y	*
Itemform	Form of item 008/22	Y	Y	Y	*
Entirenature	Nature of entire form 008/24	Y	Y	Y	*
Governmentpub	Government publication 008/28	Y	Y	Y	*

Table 30-19. Marcbook_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Conferencepub	Conference publication 008/29	Y	Y	Y	*
Literaryform	Literary form 008/33	Y	Y	Y	*
Biography	Biography 008/34	Y	Y	Y	*

Marcmap_vw

This view extracts key information from the bib 008 for maps (see [Table 30-20](#)).

Table 30-20. Marcmap_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Bib_id	Bib record ID	Y	Y		Y
Bibtype	The bibliographic record type from the leader/06	Y	Y	Y	*
Biblevel	The bibliographic level from the leader/07	Y	Y	Y	*
Projection	Projection 008/23	Y	Y	Y	*
Cartographictyp	Type of cartographic material 008/25	Y	Y	Y	*
Governmentpub	Government publication 008/28	Y	Y	Y	*
Indexed	Index 008/31	Y	Y	Y	*

Marcmusic_vw

This view extracts key information from the bib 008 for music (see [Table 30-21](#)).

Table 30-21. Marcmusic_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Bib_id	Bib record ID	Y	Y		Y
Bibtype	The bibliographic record type from the leader/06	Y	Y	Y	*

Table 30-21. Marcmusic_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Biblevel	The bibliographic level from the leader/07	Y	Y	Y	*
Compositionform	Form of composition 008/18-19	Y	Y	Y	*
Musicformat	Format of music 008/20	Y	Y	Y	*
Audience	Target audience 008/22	Y	Y	Y	*
Itemform	Form of item 008/23	Y	Y	Y	*

Marcserial_vw

This view extracts key information from the bib 008 for serials (see [Table 30-22](#)).

Table 30-22. Marcserial_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Bib_id	Bib record ID	Y	Y		Y
Bibtype	The bibliographic record type from the leader/06	Y	Y	Y	*
Biblevel	The bibliographic level from the leader/07	Y	Y	Y	*
Frequency	MARC frequency code 008/18	Y	Y	Y	*
Regularity	MARC regularity code 008/19	Y	Y	Y	*
Type	MARC serial type 008/21	Y	Y	Y	*
Originalform	Form of original item 008/22	Y	Y	Y	*
Itemform	Form of item 008/23	Y	Y	Y	*
Entirenature	Nature of entire form 008/24	Y	Y	Y	*
Governmentpub	Government publication 008/28	Y	Y	Y	*
Conferencepub	Conference publication 008/29	Y	Y	Y	*

Marcvisual_vw

This view extracts key information from the bib 008 for visual materials (see [Table 30-23](#)).

Table 30-23. Marcvisual_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Bib_id	Bib record ID	Y	Y		Y
Bibtype	The bibliographic record type from the leader/06	Y	Y	Y	*
Biblevel	The bibliographic level from the leader/07	Y	Y	Y	*
Runningtime	Running time 008/18-20	Y	Y		
Audience	Target audience 008/22	Y	Y	Y	*
Governmentpub	Government publication 008/28	Y	Y	Y	*
Visualtype	Type of visual material 008/33	Y	Y	Y	*
Technique	Technique 008/34	Y	Y	Y	*

MFHDblob_vw

This view provides access to entire MARC MFHDs (holding records) (see [Table 30-24](#)).

Table 30-24. MFHDblob_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Mfhd_id	MFHD ID number	Y	Y		Y
Marc_record	Whole MARC MFHD record in its native format, contained in a single row	Y	N	N	N/A

NOTE:

Normally you will run MARC record parsing functions to access a particular field in the record.

MFHDhistory_vw

This view provides information on the create and last update history of MFHDs (see [Table 30-25](#)).

Table 30-25. MFHDhistory_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Mfhd_id	MFHD ID number	Y	Y		Y
Create_operator_id	ID of operator who created the record	Y	Y	Y	Y
Create_date	Date and time of record creation	Y	Y	Y	
Create_location_id	ID of happening location where record was created		Y		Y
Update_operator_id	ID of operator that updated the record	Y	Y	Y	Y
Update_date	Date and time of record update	Y	Y	Y	
Update_location_id	ID of happening location where record was updated		Y		Y

NLMclass_vw

This view takes National Library of Medicine classification numbers and breaks them down into individual classes used to do reports based on call numbers, call number ranges (see [Table 30-26](#)).

Table 30-26. NLMclass_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Mfhd_id	The MFHD ID	Y	Y		Y
Firstletter	The first letter of the class number	Y	Y	Y	*
Class	The first 1 to 3 characters in the normalized call number field	Y	Y	Y	*

Table 30-26. NLMclass_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Longclass	A 7-character string that includes the class letters and class number up to the decimal point. The number is right justified	Y	Y	Y	*
Classnumber	Numeric field with the class number, including decimal places	Y	Y + class		

Recordcount_vw

This view counts various records in the database (see [Table 30-27](#)).

Table 30-27. Recordcount_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Recordtype	Record type name	Y	Y	Y	
Count	Count of records	Y			

Serials_vw

This view lists all issues for serial or multi-part check-in components (see [Table 30-28](#).)

Table 30-28. Serials_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Bib_id	The bib ID that is linked to this serial	Y	Y		Y
Mfhd_id	The MFHD ID that is linked to this serial	Y	Y		Y
Component_id	The serial check-in component ID				Y
Component_name	The check-in component name	Y			

Table 30-28. Serials_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Component_name_norm	The normalized version of the component name		Y	Y	
Predict	Whether the serial is predicted or not; will be Y or N	Y	Y		
Next_issue_id	The ID number of the next expected issue	Y	Y	Y	Y
Note	Note field	Y			
Issue_id	ID of issue		Y		Y
Enumchron	The enumeration/ chronology field of the issue	Y		Y	
Expected_date	The expected receipt date of the issue	Y	Y	Y	
Receipt_date	The date the issue was received or NULL if the issue has not yet been received	Y	Y	Y	
Received	The number of copies received	Y			

Sudocclass_vw

This view takes SuDoc (Superintendent of Documents) classification numbers and breaks them down into individual classes used to do reports based on agency, call number ranges (see [Table 30-29](#)).

Table 30-29. Sudocclass_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Mfhd_id	The MFHD ID	Y	Y		Y
Class	The first character letters in the normalized SuDoc call number field	Y		Y	*

Table 30-29. Sudocclass_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Longclass	The character string that includes letters and numbers up to the decimal point. The number is right justified	Y	Y	Y	*

Vendorinvoice_vw

This view includes actual fund amounts invoiced by vendors (see [Table 30-30](#)).

Table 30-30. Vendorinvoice_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Vendor_code	The vendor code	Y	Y	Y	
Vendor_name	The vendor name	Y	Y	Y	
Vendor_type	The vendor type assigned to the vendor	Y	Y	Y	
Institution_id	The institution ID of the vendor if one has been assigned to the vendor	Y	Y	Y	*
Bill_to_location_code	The billing location code on the invoice	Y	Y	Y	
Bill_to_location	The billing location on the invoice	Y		Y	
Invoice_date	The date of the invoice	Y	Y	Y	
Invoice_number	The invoice number	Y	Y	Y	*
Currency_code	The currency code the invoice ID is in. The amounts expressed in this view are already in the base currency.	Y	Y	Y	Y
Currency_name	Name of currency	Y		Y	

Table 30-30. Vendorinvoice_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Invoice_id	The invoice ID				Y
Invoice_status	The status of the invoice in Voyager	Y	Y	Y	
Invoice_status_date	The date the above status was set	Y	Y	Y	
Voucher_number	The voucher number printed on the voucher/check request	Y	Y	Y	*
Expenditures	Expenditures of this invoice in base currency	Y	Y	Y	
Expend_pending	Pending expenditures of this invoice—pending invoices only	Y	Y	Y	
Policy_name	The name of the acquisitions policy associated with this ledger	Y	Y	Y	
Fiscal_period_name	The name of the fiscal period used to expend funds	Y	Y	Y	
Fiscal_period_start	The beginning date of the fiscal period	Y	Y	Y	
Fiscal_period_end	The ending date of the fiscal period	Y	Y	Y	
Ledger_name	The ledger name	Y	Y	Y	Y
Fund_name	The fund name	Y	Y	Y	
Institution_fund_id	The institution assigned ID—optional	Y	Y	Y	*

Vendororder_vw

This view summarizes orders by purchase order line item and vendor (see [Table 30-31](#)).

Table 30-31. Vendororder_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Vendor_code	The vendor code	Y	Y	Y	
Vendor_name	The vendor name	Y	Y	Y	
Vendor_type	The vendor type assigned to the vendor	Y	Y	Y	
Institution_id	The institution ID of the vendor if one has been assigned to the vendor	Y	Y	Y	*
Po_number	The purchase order number	Y	Y	Y	
Po_type	The purchase order type	Y	Y	Y	
Order_location_code	The order location code	Y	Y	Y	
Order_location	The order location	Y	Y	Y	
Po_status	The purchase order status	Y	Y	Y	
Po_status_date	The date that the purchase order was sent	Y	Y	Y	
Currency_name	The name of the currency that was used on the purchase order	Y	Y	Y	
Total	The total amount of the purchase order in base currency	Y	Y		
Unit_price	The single unit price of the line item in base currency	Y	Y		
Quantity	The number of units ordered	Y	Y		
Line_price	The line item total in base currency (unit_price * quantity) +/- adjustments	Y	Y		
Mfhd_id	The MFHD id of the title ordered	Y	Y		Y

Table 30-31. Vendororder_vw

Field Name	Description and Use	Report	Query Select	Normalized Sort	Link to Field
Po_line_status	The line item status	Y	Y	Y	
Invoice_status	The invoice status of a line	Y	Y	Y	
Line_status_date	The date of the last change to the line item status or line item invoice status	Y	Y	Y	

Introduction

Voyager's WebAdmin (Server Utilities) allows operators to run various server batch jobs by selecting from the options that appear on the opening page. Instead of starting a telnet session and entering the batch job commands at a command-line prompt, you can simply point your web browser at WebAdmin and click the job that you want to run. WebAdmin uses the existing batch jobs in order to do the processing, but provides a more user-friendly interface for operators. Using WebAdmin, you can run the following batch jobs:

- Acquisitions (Pacqjob)
- Cataloging (Pcatjob)
- Circulation (Pcircjob)
- Bursar Transfer (Pbursar)
- Bulk Import (Pbulkimport)
- MARC Export (Pmarcexport)
- OPAC Reports and Processes (Popacjob)
- Media Scheduling Utilities (Pmediajob)

You can also access the log and report files that are created by the batch jobs through WebAdmin.

Information on the Circulation, Cataloging, and Acquisitions batch jobs is found in this user's guide and the following sections of the *Voyager Reporter User's Guide*:

- *Creating Input Files for Circulation*
- *Creating Input Files for Cataloging*
- *Creating Input Files for Acquisitions*

See also [Bursar Transfer System](#) for additional information on the Bursar Transfer, [Bulk Import, Replace, and Merge of MARC Records](#) for additional information on the Bulk Import batch job, [Bulk Export of MARC Records](#) for additional information the MARC Export batch job, and [Popacjob](#) for additional information OPAC Search Logging is found in this user's guide.

The Purpose of this Chapter

This chapter discusses

- Setting up WebAdmin
- Starting the Voyager Server Utilities
- Changing your WebAdmin password
- Using WebAdmin

Setting Up WebAdmin

This section discusses how to set up WebAdmin.

System Requirements

In order to run WebAdmin, you must have Perl installed on the server. Perl is commonly installed in the `/m1/shared/perl/version` directory.

You must also have the Perl DBI (Database Interface) installed as well as perl DBD:Oracle.

Perl and the DBI and DBD:Oracle components are included with the standard Voyager installation.

Each copy of WebAdmin works on only one database. If you want to run WebAdmin on multiple databases, you must install a separate copy of WebAdmin for each database you want to access. Each copy must have its own directory structure and set of files.

Starting the Server Dæmon

The WebAdmin dæmon must be running in order for the jobs to be run.

The dæmon should start when you start Voyager.

The script for the WebAdmin dæmon is called `Pwebadmind` and is located in the `/m1/voyager/xxxdb/sbin` directory. You can use the `-f` parameter to specify the interval in minutes at which you want the dæmon to run the batch jobs. For example, to run the dæmon at five minute intervals, you would enter the following at the command line:

```
Pwebadmind -f5
```

You may want to add this line to the startup script for your server so that it automatically starts up when you start your server.

NOTE:

The dæmon should not be set to run more than every two to five minutes. Setting the dæmon to run more often than that will diminish the performance of not just the dæmon but also any batch jobs being run manually.

When starting up the dæmon, you must be logged in as Voyager in order for the dæmon to run.

Creating WebAdmin Users and Passwords

The file that defines and encrypts users and passwords is named `htpasswd` and it is located in the `/m1/shared/apache2/bin` directory.

The login and password information will be stored in a text file named `xxxdb.users` which is located in the `/m1/shared/apache2/conf` directory. Here, `xxxdb` is the name of the database.



Procedure 31-1. Creating a New Login/Password for the First Time

Use the following to create a new login/password for the first time:

1. Login as `root` and enter: `cd /m1/shared/apache2/bin`
2. At the prompt enter: `./htpasswd -c /m1/shared/apache2/conf/AuthorizedUsers/xxxdb.users webadmin` where `webadmin` is the user name.

3. Then, the system prompts you to enter that user's password two times.

Result: A new login and password is created.



Procedure 31-2. Adding an Additional Login/Password to the Existing File

Use the following to add an additional login/password to the existing file.

1. Login as **root** and enter: **cd /m1/shared/apache2/bin**
2. At the prompt enter: **./htpasswd /m1/shared/apache2/conf/AuthorizedUsers/xxxdb.users newname** where newname is the new user.
3. Then, the system prompts you to enter the password for newname two times.

Result: An additional login and password is created.



Procedure 31-3. Deleting a Login/Password for an Existing User

Use the following to delete a login/password for an existing user.

1. Login as **root** and enter: **cd /m1/shared/apache2/conf**
2. Open the **xxxdb.users** file using **vi** or another server text editor.
3. Delete the line for the specific user you want to delete then save the **xxxdb.users** file.

Result: The login and password are deleted.



Procedure 31-4. Modifying a Login/Password for an Existing User

Use the following to modify a login/password for an existing user.

1. Login as **root** and move to the **/bin** directory, enter:
cd /m1/shared/apache2/bin

2. At the bin prompt, enter:
`./htpasswd /m1/shared/apache2/conf/AuthorizedUsers/
xxxdb.users webadmin` where webadmin is the user name.
3. Then, the system prompts you to enter the new password two times.

Result: The login and password are modified.

Adding the WebAdmin Authorization Section to the httpd.conf File

Add the following section to the `httpd.conf` file to set up the authorization for the WebAdmin login page.

The `httpd.conf` file is located in the `/m1/shared/apache2/conf/AuthorizedUsers` directory.



Procedure 31-5. Adding the WebAdmin Authorization Section to the httpd.conf File

Use the following to add WebAdmin authorization to the `httpd.conf` file.

1. Login as root and navigate to the `/m1/shared/apache2/conf/AuthorizedUsers` directory and backup the `httpd.conf` file, at the prompt enter
`cp httpd.conf httpd.$$backup`
2. Then open the `httpd.conf` file using `vi` or another server text editor.
3. After the line in the file that reads
`DocumentRoot /m1/voyager/xxxdb/webvoyage/html` enter:

```
#####Begin WebAdmin Configuration Change#####
```

```
ScriptAlias /webadmin/cgi-bin/ /m1/voyager/xxxdb/webadmin/  
cgi-bin/
```

```
Alias /webadmin /m1/voyager/xxxdb/webadmin
```

```
DirectoryIndex webvoy.htm index.htm index.html
```

```
<Directory /m1/voyager/xxxdb/webadmin >
```

```
AuthName "xxxdb security"
```

```

AuthType Basic

AuthUserFile /m1/shared/apache2/conf/AuthorizedUsers/
xxxdb.users

require valid-user

</Directory>

#####End WebAdmin Configuration Change#####

```

Result: You are now ready to access WebAdmin.

Starting the Voyager Server Utilities

To start WebAdmin, in your browser address box, enter the URL of the server where your database resides followed by /webadmin.

The login dialog opens (see [Figure 31-1](#)).

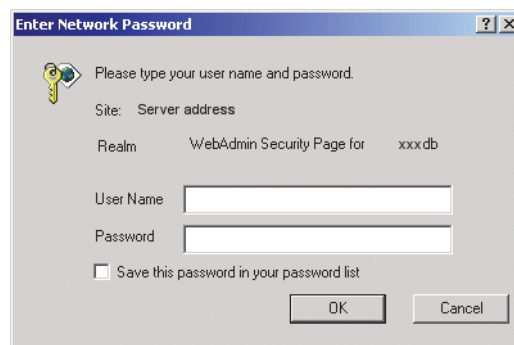


Figure 31-1. WebAdmin Log in Dialog Box



Procedure 31-6. Logging into WebAdmin

Use the following to log into WebAdmin.

1. Enter your user name into the **User Name** field.

2. Enter your password into the **Password** field.

You can have your user name and password appear the next time you start WebAdmin by selecting the **Save this password in your password list** check box. If your user name and password currently appear automatically in the dialog, you can prevent them from appearing by unselecting this check box.

3. Click the **OK** button.

Result: The WebAdmin main page, Voyager Server Utilities, opens (see [Figure 31-2](#) on [page 31-8](#)).

Using WebAdmin

WebAdmin is entirely web-based. From this main page users select the batch jobs to run or files to examine. Each batch job or set of files has its own web page, accessed by clicking the link to that page.

Voyager Server Utilities Page

The Voyager Server Utilities (main) page ([Figure 31-2](#)) contains all of the links to the various batch jobs and report and log files. This page comes from the file `index.html` on the server. You can edit this file like you would a standard html page.

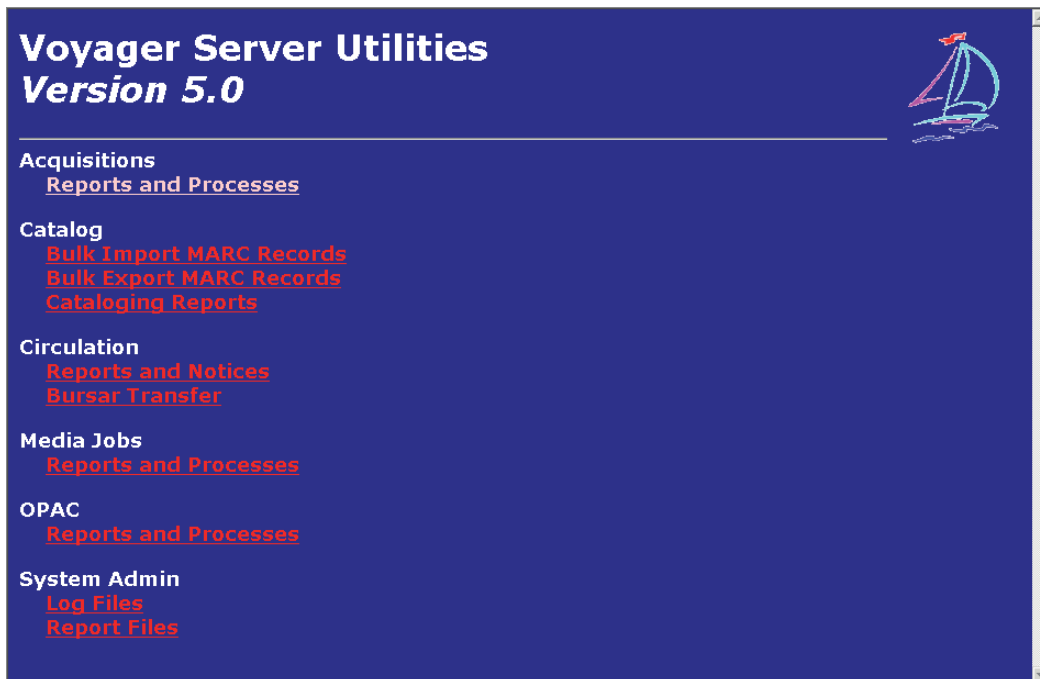


Figure 31-2. Webadmin Main Page

The activities available on the main page are broken down by module in the following manner:

- Acquisitions (see [Acquisitions Utilities](#) on [page 31-9](#))
 - Reports and Processes
Acquisitions Batch Job page (Pacqjob)
- Catalog (see [Cataloging Utilities](#) on [page 31-12](#))
 - Bulk Import MARC Records
Bulk Import Batch Job page (Pbulkimport)
 - Bulk Export MARC Records
MARC Export Batch Job page (Pmarcexport)
 - Cataloging Reports
Cataloging Batch Job page (Pcatjob)

- Circulation (see [Circulation Utilities](#) on [page 31-24](#))
 - Reports and Notices
Circulation Batch Job page (Pcircjob)
 - Bursar Transfer
Bursar Transfer Batch Job (Pbursar)
- Mediajobs (see [Media Scheduling Utilities](#) on [page 31-34](#))
 - Reports and Processes
- OPAC (see [OPAC Reports Utilities](#) on [page 31-36](#))
 - Reports and Processes
Search Logging Batch Jobs (Popacjob)
- System Admin (see [System Admin Utilities](#) on [page 31-38](#))
 - Log Files
Contents of the /log subdirectory
 - Report Files
Contents of the /rpt subdirectory
 - EDI File Maintenance

Acquisitions Utilities

On the main page, the Acquisitions utility available is for acquisitions reports and notices. The Reports and Processes utility allows you to run all acqjobs.

See *The Acquisitions Batch Jobs* in the *Voyager Reporter User's Guide* for an explanation of Acqjob 1-4 jobs and for instructions on running them in WebAdmin.

See [Acquisitions Batch Job - Fix Exchange Rates](#) on [page 17-1](#) for information on Acqjob 5.

The procedure for running Acqjob 5 using Webadmin is shown in [Procedure 31-7, Running Acqjob 5 Using WebAdmin](#).



Procedure 31-7. Running Acqjob 5 Using WebAdmin

Use the following to run Acqjob 5 (Fix Exchange Rates) with optional parameters.

1. From the WebAdmin main page ([Figure 31-2](#)), Acquisitions section, click the **Reports and Processes** link.

Result: The **Acquisitions Reports and Notices** page opens (see [Figure 31-3](#)).

Figure 31-3. WebAdmin Acquisitions Reports and Notices Page

2. From the drop-down menu, select **Fix Exchange Rates**.
3. Enter your e-mail address in the **E-Mail Address** field.

NOTE:

You must enter the entire email address (that is, user@hostname.com) in order for the e-mail to be sent.

4. Click the **Submit Acqjob** button.

Result: The Fix Exchange Rate Options page opens (see [Figure 31-4](#)).

Choose from the options below to setup your job.

Currency Code:

Vendor Code:

Ledger:

PO Type:

Figure 31-4. Fix Exchange Rate Options Page

5. As needed, select items from the following fields to limit the adjustment:
 - **Currency Code**
 - **Vendor Code**
 - **Ledger**
 - **PO Type**
6. Select **OK** to run the job.

Result: The job runs, the **Job Scheduler** page opens, and an e-mail is sent to the user (see [Figure 31-5](#)).

Job Scheduler

[Return to Main Page](#)

Email notification to: m.phillips@endinfosys.com

job: acqjob -j 5

Figure 31-5. Job Scheduler Page after Submitting Acqjob 5

Cataloging Utilities

On the main page, the Cataloging utility is available for cataloging reports.

Users run Bulk Import MARC Records, Bulk Export MARC Records, and other Cataloging Reports.

See *The Cataloging batch jobs* in the *Voyager Reporter User's Guide* for explanations of the various cataloging jobs. See the *Voyager Reporter User's Guide* for instructions on how to run cataloging jobs one through ten.

Bulk Import MARC Records

The Bulk Import MARC Records utility allows you to automatically import a large number of MARC records from a file located on a local PC or a network drive. This WebAdmin utility provides the same options that are available in the Bulk Import batch job.

See [Bulk Import, Replace, and Merge of MARC Records](#) on [page 13-1](#) for more information on bulk import. Also refer to chapter 4 in the *Voyager System Administration User's Guide* for information regarding duplicate detection profiles and bulk import rules.



Procedure 31-8. Running Bulk Import Using WebAdmin

Use the following to run the Bulk Import Utility.

1. After logging in to WebAdmin, from the main page, Cataloging section, click the **Bulk Import MARC Records** link.

Result: The **MARC record Bulk Import** page opens (see [Figure 31-6](#)).

MARC Record Bulk Import

Use the Browse button to select the MARC file on your PC to upload to the Voyager database. Uploading large files (more than 1000 records) may take more time and may require several minutes. The file will not be deleted until you press the Submit button at the bottom of the screen. When the file is sent you will see notification of a successful upload.

Local Filename:

Voyager Batch Upload Information

All of the fields in this part of the form come directly from the Voyager Conversion and Technical Manual, from the section Bulk Import and Playback of MARC records. Please consult the document for information about these fields.

Import Code: required

Operator Name: optional

Dedup Location Code: optional

Holdings Location Code: optional

Begin Record: (1st or 'first')

End Record: (1st or 'last')

Copy Number: optional

Generate Keyword Index? Default is no keyword index generated ☐

Import file has interleaved bib and holdings records?: ☐

Delete bib records matched in import file?: ☐

Delete holdings records matched in import file?: ☐

Mark imported records as OK for export?: ☐

Don't create 035 during import bib?: ☐ optional

Use the Browse button to select the USEMARC.CON file on your PC.

USEMARC.CON Initialization Filename: optional

Show/Approve MARC display before database load?: ☒

Allow multiple instances of bulkimport?: ☐

Email Address: required

Figure 31-6. MARC Record Bulk Import Page

2. Complete the MARC Bulk Import Record page for your batch import. See [Table 31-1](#) for a description of your options.

Result: If you did not turn on the **Show/Approve** check box, the **Job Scheduler** page tells you when the import will begin.

If you did turn on the **Show/Approve** check box, you will be presented with a display of the records you want to import. To accept these records and begin the import, click the **Accept these records** button. Then the **Job Scheduler** page tells you when the import will begin. To reject these records, click the **Back** button in your browser until you reach the main page.

Table 31-1. MARC Bulk Import Record Options

Option	Description
Local Filenam	Enter the filename that you want to import into your Voyager database in the Local Filename field. Or click the Browse button to select the file you want to import. This opens the File Upload dialog box. Select a file to load and click the Open button.
Import Code	In the Import Code field, select the import code of the Import/Replace profile that you want to apply to this import session. The Import Code comes from the Rule Code field on the Bulk Import Rules dialog in the System Administration module.
Operator Name	In the Operator Name field, select the operator name that you want saved with the record in the database as having last modified the record.
Dedupe Location Code	In the Dedupe Location Code field, select the code for the location at which the records were last modified.
Holdings Location Code	In the Holdings Location Code field, select the code for the location that you want saved with the record in the database as the location that the records come from.
Begin Record	In the Begin Record field, enter the sequential number of the record which you want to begin importing. (For example, to start importing at the fifth record, enter 5.) Enter first to begin with the first record.
End Record	In the End Record field, enter the sequential number of the record which you want to end importing. (For example, to end after record ten, enter 10). To end with the last record, enter last.
Copy Number	Optionally, enter a copy number to add upon import.

Table 31-1. MARC Bulk Import Record Options

Option	Description
Generate Keyword Index? Default it no keyword index generated	Select this option to specify that you want a keyword index generated (<code>-K ADDKEY</code>) when running Bulk Import. See Parameters on page 13-4 for more information. Running a keyword index regen during Bulk Import creates updated keyword index files in addition to updating the bibliographic records in the database. NOTE: When you import a large number of records, performance may be negatively affected. See Generating Keyword Index Files on page 13-3 for more information. By default, no keyword index is generated when running Bulk Import. As a result, bibliographic records are added to the database but the keyword index files are not updated.
Import file has interleaved bib and holdings records?	If the import file contains both bibliographic and holdings records that are interleaved, select the Import file has interleaved bib and holdings records? check box.
Delete bib records matched in import file?	If you want to delete any bibliographic records in your database that match the bibliographic records in the file, select the Delete bib records matched in import file? check box. Note that this will not import any records - it will only delete bibliographic records from the database.
Delete holdings records matched in import file?	If you want to delete any holdings records in your database that match the holdings records in the file, select the Delete holdings records matched in import file? check box. Note that this will not import any records - it will only delete holdings records from the database.
Mark imported records as OK for export?	If you want the OK to export check box (on the System tab of bibliographic, authority, and holdings records on view in the Cataloging module) to be turned on for all of the newly-imported records in the database, select the Mark imported records as OK for export? check box.
Don't create 035 during import of bib?	Select this option to indicate that you do not want the 035 field created when importing bibliographic records during Bulk Import. The bulk import default behavior is to create the 035 from the 001 of the imported bibliographic record.

Table 31-1. MARC Bulk Import Record Options

Option	Description
USEMARCON Initialization filename	For USEMARCON Initialization filename , enter the full path. This points to the translation files needed to translate the records into MARC21 format before running Bulk Import as usual.
Show/Approve MARC display before database load?	Select the Show/Approve MARC display before database load? check box to view all of the bibliographic record that you are importing onscreen before performing the import. If the records are not satisfactory, you may cancel the process without importing the records.
Allow multiple instances of bulk import?	Select this option to specify your intention to run multiple concurrent bulkimport sessions. This can be done safely if there is no data overlap between runs.  IMPORTANT: <i>Using this option assumes that you have confirmed that there is no data overlap that might be an issue for the multiple concurrent bulkimport sessions you are running with this option selected.</i> When this option is not selected, the bulkimport script checks for a running bulkimport process for the current database. If it finds a running bulkimport process, it terminates and displays a message indicating that there is a bulkimport process that is already running.
E-mail Address	In the E-mail Address field, enter your email address. The utility will send an e-mail to that address when the job is done processing. You must enter the entire email address (user@hostname.com) in order for the e-mail to be sent.
Submit Batch Import	When you have verified that the settings on the page are all correct, click the Submit Batch Import button to begin the importing process.

3. Click **Submit Batch Import**.

4. Click the **Back** button.

Result: Returns to the main menu.

5. Click **Report Files** from the WebAdmin main menu to view resulting reports.

Refer to [Procedure 31-18, Accessing Report Files on the Server Using Webadmin](#), on page [31-39](#). Also refer to [Log File](#) on [page 13-10](#).

Bulk Export MARC Records

The Bulk Export MARC Records utility allows you to automatically export a large number of MARC records (such as bibliographic, holdings/MFHD, or authority) from the Voyager database to a file on the server.

NOTE:

The MARC records remain in your Voyager database. A copy is extracted to a flat file on the server.

All of the same options in the MARC Export batch job are available when you import records this way. See [Bulk Export of MARC Records](#) on [page 11-1](#).



Procedure 31-9. Running MARC Record Bulk Export Using WebAdmin

Use the following to run the MARC Record Bulk Export Utility.

1. After logging in to WebAdmin, from the main page, Cataloging section, click the **Bulk Export MARC Records** link.

Result: The **MARC record Bulk Export** page opens (see [Figure 31-7](#)).

MARC Record Bulk Export

This utility allows you to export several different types of MARC records (bib, holdings, authority, authority - main, authority - subject, authority - series, bib-holdings groups). Please complete the required information below and click on the submit button.

When the job is completed, you will receive notification via email that the job has ended as well as any additional information which was provided by the job.

Finally, check the "Voyager's Report Files" link for both log exp [date] and marc exp [date] output files with the current date and time which will be generated from this run.

Record Type: Bibliographic

Export from Specific Library (Bib Export Only):

Export OCLC Only (Bib Export Only): ☐

System Control (003) Identifier:

System Control (040\$d) Identifier:

Export Mode:	Export Target:
☒ MARC ID Input File	Browse...
☐ ISBN Input File (Bib Export Only)	Browse...
☐ Range of MARC IDs	☒ All ☐ ID Range (00000-00000) -
☐ Date Range -- Create Dates	☒ Today - days ☐ Date Range (YYYY-MM-DD) -
☐ Date Range -- Update Dates	☒ Today - days ☐ Date Range (YYYY-MM-DD) -
☐ Date Range -- Create or Update Dates	☒ Today - days ☐ Date Range (YYYY-MM-DD) -
☐ Date Range -- OK to Export	☒ Today - days ☐ Date Range (YYYY-MM-DD) -
☐ Date Range -- Suppressed	☒ Today - days ☐ Date Range (YYYY-MM-DD) -
☐ Date Range -- Excluded	☒ Today - days ☐ Date Range (YYYY-MM-DD) -
☐ Date Range -- Excluded and Suppressed	☒ Today - days ☐ Date Range (YYYY-MM-DD) -

Select an Exclude Type	Select Exclude Location File
☒ Exclude from create location and last update location	Select the Location file on your PC: Browse...
☐ Exclude from most recent location (create or update)	

Create Control Number (001) from LCCN (010a) ? ☐

Update \$52a on Holdings Export with the following text:

Copy OCLC Number from Bib to Holdings ? ☐

Ignore Suppressed Records ? ☐

Suppress Serials Checkin Data? ☐

Convert Exported Record to:

Use the **Browse** button to select the USEMARCON file on your PC.
 USEMARCON Initialization Filename: Browse... optional

Email Address: required

Submit Bulk Export

Figure 31-7. Marc Record Bulk Export Page

2. On the MARC Record Bulk Export page, select the options to match your requirements from the following:
 - a. In the **Record Type** drop-down menu, select the type of record that you want to export.
 - b. If you want to export only bibliographic records from a specific library, select the library from the **Export From Specific Library** drop-down menu.
 - c. If you want to export only OCLC-created bibliographic records, select the **Export OCLC Only** check box.
 - d. If you choose to create a control number (001) from the LCCN (101a), you are required to enter a System Control Identifier. Enter this identifier into the **System Control (003) Identifier** field. This code will be placed into the 003 of each exported bibliographic or authority MARC record. See Create Control Number (001) from LCCN (010a) (step #7).
 - e. If you want to place a code for the Modifying Agency in the 040\$d of each exported record, enter the identifying code into the **System Control (040\$d) Identifier** field.
 - f. From the table, select an **Export Mode**. Click one of the radio buttons in the Export Mode column to select one of the following options:
 1. **MARC ID Input.** (Bib Export Only.) Allows you to specify a file with a list of MARC ID numbers of the records you want to export. Enter the filename and path in the field or click the **Browse** button in the Export Target column to select the name of the list file.
 2. **ISBN Input File.** (Bib Export Only.) Allows you to specify a file with a list of ISBN numbers of the records you want to export. Enter the filename and path in the field or click the **Browse** button in the Export Target column to select the name of the list file.
 3. **Range of MARC IDs.** Select the All radio button in the Export Target column to designate that all records are to be exported; or, select the **ID Range** radio button in the Export Target column to specify a range of MARC IDs to be exported. Enter the beginning of the range into the first field and the ending of the range into the last field. Both the first and last records specified will be exported.
 4. **Date Range -- Create Dates.** Select the **Today** radio button in the Export Target column to specify that you want to export all records created between the current day and a number of days previous; then enter the number of days previous in the days field. If you want to export all records created between two dates, select the **Date Range** radio button in the Export Target column. Then enter the starting date into the first field and the ending date in the second field. Dates must be entered in the format YYYY-MM-DD.

5. **Date Range -- Update Dates.** Select the **Today** radio button in the Export Target column to specify that you want to export all records updated between the current day and a number of days previous; then enter the number of days previous in the days field. If you want to export all records updated between two dates, select the **Date Range** radio button in the Export Target column. Then enter the starting date into the first field and the ending date in the second field. Dates must be entered in the format YYYY-MM-DD.
6. **Date Range -- Create or Update Dates.** Select the **Today** radio button in the Export Target column to specify that you want to export all records created or updated between the current day and a number of days previous; then enter the number of days previous in the days field. If you want to export all records created or updated between two dates, select the **Date Range** radio button in the Export Target column. Then enter the starting date into the first field and the ending date in the second field. Dates must be entered in the format YYYY-MM-DD.
7. **Date Range -- OK to Export.** Select the **Today** radio button in the Export Target column to specify that you want to export all records that have had the OK to Export flag set or updated between the current day and a number of days previous; then enter the number of days previous in the days field. If you want to export all records that have had the OK to Export flag set or updated between two dates, select the **Date Range** radio button in the Export Target column. Then enter the starting date into the first field and the ending date in the second field. Dates must be entered in the format YYYY-MM-DD.
8. **Date Range -- Suppressed.** Select the **Today** radio button in the Export Target column to specify that you want to export all records suppressed between the current day and a number of days previous; then enter the number of days previous in the days field. If you want to export all records suppressed between two dates, select the **Date Range** radio button in the Export Target column. Then enter the starting date into the first field and the ending date in the second field. Dates must be entered in the format YYYY-MM-DD.
9. **Date Range -- Excluded.** Select the **Today** radio button in the Export Target column to specify that you want to export all records excluded between the current day and a number of days previous; then enter the number of days previous in the days field. If you want to export all records excluded between two dates, select the **Date Range** radio button in the Export Target column. Then enter the starting date into the first field and the ending date in the second field. Dates must be entered in the format YYYY-MM-DD.

10. **Date Range -- Excluded and Suppressed.** Select the **Today** radio button in the Export Target column to specify that you want to export all records excluded and suppressed between the current day and a number of days previous; then enter the number of days previous in the days field. If you want to export all records excluded and suppressed between two dates, select the **Date Range** radio button in the Export Target column. Then enter the starting date into the first field and the ending date in the second field. Dates must be entered in the format YYYY-MM-DD.
- g. From the table, in the **Select an Exclude Type** section, click one of the radio buttons in the **Exclude Type** column:
 1. Exclude from create location or last update location.
 2. Exclude from most recent activity location
 - h. From the table, in the **Select Exclude Location File** section, enter or browse for the Location file on your PC.
 - i. If you want to change the LCCN number of the records being exported to a control number, select the check box for **Create Control Number (001) from LCCN (010a)**. If you select this option, you may also specify a code to be placed into the 003 field in the System Control (003) Identifier.
 - j. If you want to enter any text into the 852a field of the exported record, enter the text into the **Update 852a on Holdings Export with the following text** field.
 - k. If you want to copy the OCLC number from the bibliographic record's 035 \$a to the holdings record upon export, select the **Copy OCLC Number from Bib to Holdings?** check box. See the explanation for -b in [Parameters](#) on [page 11-3](#) for more information.
 - l. If you want to ignore suppressed records, that is not export them, select the **Ignore Suppressed Records?** check box.
 - m. If you want to suppress serials checkin data when exporting holdings records, select the **Suppress Serials Checkin Data?** check box. The default is to not suppress serials checkin data.
 - n. If you want the records being exported to be converted to a specific cataloging character set (RLIN Legacy Encoding, OCLC, USMARC (MARC21 MARC8), select the character set from the **Convert Exported Record to** drop-down menu. The Unicode character set is the default.
 - o. If you are running USEMARCON, enter the name of the **USEMARCON initialization file**.
 - p. In the **E-mail Address** field, enter your e-mail address. The Utility will send an email to that address when the job is done processing. You must enter the entire email address (user@hostname.com) in order for the e-mail to be sent.

- q. When you have verified that the settings on the page are all correct, click the **Submit Bulk Export** button to begin the export process. The **Job Scheduler** page will appear with information about when the export will begin.

Result: The Bulk Export runs.

3. Click the **Back** button.

Result: Returns to the main menu.

4. Click **Report Files** from the main menu of WebAdmin to view the export log.

See [Procedure 31-18, Accessing Report Files on the Server Using Webadmin](#), on page [31-39](#).

For output file and log file information, see [Output File Specification](#) on [page 11-11](#).

Cataloging Reports

Users can run cataloging batch jobs using WebAdmin. Cataloging reports and the Global Headings Change (GHC) jobs.

See [Global Heading Change Jobs](#) on [page 14-1](#) for GHC information and the *Voyager Reporter User's Guide* for more information on the cataloging reports.



IMPORTANT:

The cataloging GHC jobs should be run in order and in concert with specific cataloging module activities.



Procedure 31-10. Running Global Headings Change Jobs Using WebAdmin

Use the following to run the GHC jobs.

1. After logging in to WebAdmin, from the main page, Cataloging section, click the **Cataloging reports** link.

The **Cataloging reports and processes** page opens.

2. Select from the drop-down menu the GHC job you wish to run, catjob 11, catjob 12, or catjob 13 (see [Figure 31-8](#)).

-
- | |
|---|
| 1. Unauthorized subject headings report |
| 2. Unauthorized name headings report |
| 3. Unauthorized title headings report |
| 4. Unauthorized name/title headings report |
| 5. Duplicate authority records report |
| 6. 'See' References with linked bib records report |
| 7. 'See' References authorized in another authority record report |
| 8. 'See-also' Reference without corresponding authority record report |
| 9. HTTP Verify - Date Range |
| 10. HTTP Verify - All |
| 11. Process global headings change queue -- step 1 |
| 12. Generate global headings change prevue information -- step 2 |
| 13. Apply global headings changes for marked fields -- step 3 |
-

Figure 31-8. Drop-Down Menu Options on the Cataloging Reports and Processes Page

- a. If you select either catjob 11 or catjob 12
 1. In the **E-mail Address** field, enter your e-mail address. The Utility will send an email to that address when the job is done processing. You must enter the entire email address (user@hostname.com) in order for the e-mail to be sent.
 2. When you have verified that the settings on the page are all correct, click the **Submit** button to begin the job. The **Job Scheduler** page will appear.
 - b. If you select catjob 13, the system prompts you to provide the optional variables of location and operator (see [Figure 31-9](#)).
-

Make selections below to complete the setup of the job.

Location Code: optional

Operator Name: optional

Figure 31-9. Select Location and Operator to Run Catjob 13

1. select the location code from the drop-down menu
2. select the operator name from the drop-down menu

3. In the **E-mail Address** field, enter your e-mail address. The Utility will send an email to that address when the job is done processing. You must enter the entire email address (user@hostname.com) in order for the e-mail to be sent.
4. When you have verified that the settings on the page are all correct, click the **Submit** button to begin the job. The **Job Scheduler** page opens.

Result: The catjob runs.

Circulation Utilities

From the main Voyager WebAdmin page, the Circulation batch jobs are accessed from the following Circulation options:

- Reports and Notices
- Bursar Transfer

Additional information regarding Circulation batch jobs can be found in [Circulation Batch Jobs](#) on [page 9-1](#).

Voyager WebAdmin Circulation options provide batch processing that is used in a variety of ways. In particular, a subset of the circjobs that can be run from WebAdmin generate data that is used as input to Microsoft Access to generate standardized reports and notices. See the *Voyager Reporter User's Guide* for more information regarding these alternatives.

Additional information regarding patron export, import, and purge can be found in [Patron Batch Jobs in XML](#) on [page 5-1](#).

Reports and Notices - Forgiving Fines and Fees

Forgiving fines and fees circjobs are described in the procedures in this section.



Procedure 31-11. Forgiving fines by patron ID (circjob 40)

Use the following to forgive fines by Patron ID in the WebAdmin client.

**IMPORTANT:**

Before starting this job, you should have a file listing patron IDs delimited by carriage-returns.

1. From Voyager Server Utilities (the main page of the WebAdmin client), click **Reports and Notices** under the **Circulation** heading.
2. Scroll down the page until you see the **Forgive Fines by Patron ID** option in the left column of the table ([Figure 31-10](#)). Select the radio button beside it.

☐ Forgive Fines by Patron ID	Operator Id: Required Input File: Required Run In Test Mode? ☐

Figure 31-10. Circjob 40, WebAdmin

3. In the corresponding right column, enter your Operator ID. (This will tag the job to your ID in the system.)
4. In the **Input File** field, enter the name of the input file. If the file resides somewhere besides `/m1/voyager/xxxdb/local`, include the path with the file name.
5. To run the file in test mode (and view the logs before running the actual batch job), select the check box beside **Run In Test Mode?**
6. Scroll down to the bottom of the page and enter your e-mail address in the required address field, then click the **Submit CircJob** button to run the job.

Result: Voyager runs the batch job, forgiving the fines and fees for those patrons whose IDs are listed in the file. If you ran the job in test mode, the database is not updated but Voyager produces an error log and an audit report. For information about these, see page [9-35](#).



Procedure 31-12. Forgiving fines by date created (circjob 41)

Use the following to forgive fines by date created in the WebAdmin client.

1. From Voyager Server Utilities (the main page of the WebAdmin client), click **Reports and Notices** under the **Circulation** heading.
2. Scroll down the page until you see the **Forgive Fines by Fine Create Date** option in the left column of the table ([Figure 31-11](#)). Select the circle beside it.

☐ Forgive Fines by Fine Create Date	Operator Id:	
	Required	
	Forgive Stub Record Fines? ☐	
	Forgive Fines from ONLY these Circulation Locations:	
	CIRC ELECTRONIC Offsite Res main media sc	Fine Create Date Range (YYYY-MM-DD): - Required
	Select All	
	Run In Test Mode? ☐	

Figure 31-11. Circjob 41, WebAdmin

3. In the corresponding right column, enter your Operator ID. (This will tag the job to your ID in the system.)

4. Click the **Forgive Stub Record Fines?** check box to have fines for Universal Borrowing stub records forgiven.
5. To limit the Circulation locations from where you want the fines removed, select any of the locations in the **Circulation Locations** list box.

NOTE:

The list box is populated with locations where the fines and fees were first created.

6. Enter the **Fine Create Date Range (YYYY-MM-DD)**: to specify the incur dates for which you want to forgive.
7. To run the job in test mode and view logs, select the **Run In Test Mode?** check box.
8. Scroll down to the bottom of the page and enter your e-mail address in the required address field, then click the **Submit CircJob** button to run the job.

Result: Voyager runs the batch job, forgiving the fines and fees for the create date range with any limits you have set. If you ran the job in test mode, the database is not updated but Voyager produces an error log and an audit report. For information about these, see page [9-35](#).



Procedure 31-13. Forgiving fines by patron group and expiration date (circjob 42)

Use the following to forgive fees and fines using patron group and expiration date.

1. From Voyager Server Utilities (the main page of the WebAdmin client), click **Reports and Notices** under the **Circulation** heading.
2. Scroll down the page until you see the **Forgive Fines by Patron Group and Expiration Date** option in the left column of the table ([Figure 31-12](#)). Select the radio button beside it.

☐ Forgive Fines by Patron Group and Expiration Date	Operator Id: Required
	Forgive Stub Record Fines? ☐
	Forgive Fines from ONLY these Circulation Locations:
	CIRC ELECTRONIC Offsite Res main Expiration Date Range (YYYY- MM-DD); - Required
	Select All
Forgive Fines for ONLY these Patron Groups:	
BINDERY CARREL GRAD RESTRICTED	
Select All	
Required	
Run In Test Mode? ☐	

Figure 31-12. Circjob 42, WebAdmin

3. In the corresponding right column, enter your Operator ID. (This will tag the job to your ID in the system.)
4. Click the **Forgive Stub Record Fines?** check box to have fines for stub records forgiven.
5. To limit the Circulation Locations from where you want the fines removed, select any of the locations in the **Circulation Locations** list box.
6. Enter the **Expiration Date Range (YYYY-MM-DD)**: to specify the expiration date range for patron records whose fines you want to forgive.

7. To limit the patron groups whose fines are being forgiven, select one or more group names from the **Patron Groups** list box.
8. To run the job in test mode and view logs, select the **Run In Test Mode?** check box.
9. Scroll down to the bottom of the page and enter your e-mail address in the required address field, then click the **Submit CircJob** button to run the job.

Result: Voyager runs the batch job, forgiving the fines and fees for the patron group(s) and expiration date range specified. If you ran the job in test mode, the database is not updated but Voyager produces an error log and an audit report. For information about these, see page [9-35](#).

Reports and Notices - XML Patron Purge

The XML Patron Purge option enables you to delete patrons from your database by using either date criteria or a file of patron IDs (one ID per row in a flat file located on your local PC). See [Figure 31-13](#).

☒ XML Patron Purge (Purge patrons, log data in XML format)	Please select patron purge option: ☐ By Date ☐ By File of Patron IDs Delete patrons from database? : ☐ Output File: <i>Default directory is \$VOYAGER/\$DATABASE/rpt. Default filename is xml.patpurge.YYYYMMDD.HHMMSS.</i> Audit File: <i>Default directory is \$VOYAGER/\$DATABASE/rpt. Default filename is log.patpurge.YYYYMMDD.HHMMSS.</i> Exception File Path: <i>Default directory is \$VOYAGER/\$DATABASE/rpt.</i>

Figure 31-13. XML Patron Purge Option

When you select the **By File** radio button, the display dynamically refreshes to present you with the filename entry/browse options. If you enter the filename manually, you need to enter the fully qualified name such as
C:\Delete_Patrons\IDs_Summer_2014.txt.

Refer to [PpatronPrg \(Patron Purge\)](#) on [page 5-11](#) for more information.

Bursar Transfer

The Bursar Transfer utility allows you to transfer information from your Voyager database to another system, such as the bursar's accounting system. All of the options available to the Bursar Transfer batch job are available on the Bursar Transfer page. See [Bursar Transfer System](#) on [page 8-1](#).



Procedure 31-14. Running Bursar Transfer using WebAdmin

Use the following to run the Bursar Transfer Utility.

1. After logging in to WebAdmin, from the main page, Circulation section, click on the **Bursar Transfer** link.

Result: The **Bursar Transfer** page opens (see [Figure 31-14](#)).

Transfer fines/fees from ONLY these circulation locations: CIRC ELECTRONIC ILLTEST LAWCIRC LAWcirc OPACCIRC Offsite Res Select All	Transfer fines/fees for ONLY these patron groups: BINDERY CARREL CC FAC GLOBAL GRAD ILL Law Select All	Transfer fines/fees for ONLY these fine/fee types: Information Fee Lost Equipment Processing Lost Item Processing Lost Item Replacement Lost Library Card Media Booking Late Charge Media Booking Usage Fee Overdue Select All
☒ Create transfer record for each fine/fee transferred ☐ Create one summary transfer record per patron		
Transfer fines/fees greater than or equal to: \$ 0 .00		
Transfer fines/fees 0 days after fine created		
Operator ID: BURSAR		
Email Address: required		
Submit Bursar Transfer		

Figure 31-14. Bursar Transfer Page

2. On the Bursar Transfer page, select the options to match your requirements from the following:
 - a. From the **Transfer fines/fees from ONLY these circulation locations** list, select the locations for which you want to transfer information. Click the **Select All** button to select all of the locations in the list.

- b. From the **Transfer fines/fees for ONLY these patron groups** list, select the patron groups for which you want to transfer information. Click the **Select All** button to select all of the patron groups in the list.
- c. From the **Transfer fines/fees for ONLY these fine/fee types** list, select the fine/fee types that you want to transfer. Click the **Select All** button to select all of the fine/fee types in the list. Refer to [Configuration File on page 8-5](#) for additional information regarding fine/fee type transfers to the bursar system using the Pbursar script and a configuration file.
- d. Click the **Create transfer record for each fine/fee transferred** radio button to transfer individual fine/fee records and patron information, as well as specific information about the fines and fees.

OR

Click the **Create one summary transfer record per patron** radio button to transfer only patron information and the patron's current balance.

- e. In the **Transfer fines/fees greater than or equal to:** field, enter the minimum amount that the fine or fee must be in order to be transferred (if **Create transfer record...** is selected)

OR

enter the minimum that a fine/fee must be in order to be added to the total (if **Create one...** is selected).

- f. In the **Transfer fines/fees [] days after fine created** field, enter the number of days that must pass after the fine create date before the fine can be transferred (if **Create transfer record...** is selected)

OR

enter the number of days that must pass after the fine create date before the fine can be added to the total (if **Create one...** is selected).

- g. Select an operator ID from the **Operator ID** drop down list that is to be used with the postings. Defaults to Bursar.
- h. In the **E-mail Address** field, enter your e-mail address. The Utility will send an email to that address when the job is done processing. You must enter the entire email address (user@hostname.com) in order for the e-mail to be sent.
- i. When you have verified that the settings on the page are all correct, click the **Submit Bursar Transfer** button

Result: The transfer process begins. The **Job Scheduler** page displays information about the export.

3. Click the **Back** button.

Result: Returns to the main menu.

Media Scheduling Utilities

On the main page, the Media Jobs Reports and Notices allows users to run all of the Media Scheduling jobs using WebAdmin.

For explanations of the various jobs see *The Media Scheduling batch jobs* in the *Voyager Reporter User's Guide*, and [Retain Patron IDs \(Mediajob 5\)](#) on [page 10-1](#) in this user's guide.



Procedure 31-15. Running Retain Patron IDs Media Scheduling, Mediajob 5

Use the following steps to run Mediajob 5 from Webadmin.

1. After logging in to WebAdmin, click the **Reports and Processes** link from the **Media Scheduling** section of the main screen.

Result: The Media Scheduling Reports and Processes page opens.(see [Figure 31-15](#)).

Mediajob Reports and Processes

This utility allows you to run any of the Mediajob reports and processes described in the manual. Select the mediajob you want to run, select submenu for this job(if applicable), select date range (if applicable) and press the submit button.

Select Job to Run:	
☐ OverdueNotices	
☐ Inventory Reports	SUBMENU: ☐ Date Range For All Media Policies ☐ All Media Policies ☐ Specific Media Policy The Specific Media Policy option requires Media Policy Group name Please enter the name of Media Policy A Date Range (YYYY-MM-DD) is required only for the task Date Range For All Media Policies option -
☐ Booking Statistics	SUBMENU: ☐ All Booking Desks ☐ Global Booking Desk ☐ Specific Booking Desk The Specific Booking Desk option requires a Booking Desk location Please enter the name of the Booking Desk A Date Range (YYYY-MM-DD) is required for all tasks in this submenu -
☐ Charge Statistics	SUBMENU: ☐ All Booking Desks ☐ Global Booking Desk ☐ Specific Booking Desk The Specific Booking Desk option requires a Booking Desk location Please enter the name of the Booking Desk A Date Range (YYYY-MM-DD) is required for all tasks in this submenu -
☒ Retain Patron IDs	Number of patron IDs to retain :

Email Address: **required**

Figure 31-15. Mediajob Reports and Processes Page

2. To run Mediajob 5, click the **Retain Patron IDs** radio button.
3. Enter the number of patron IDs in the **Number of patron IDs to retain** field.
4. Click the **Submit Media Job** button.

Result: The job runs. The **Job Scheduler** page displays information about the job (see [Figure 31-16](#)).



Figure 31-16. Job Scheduler Page after Submitting the Job

5. Click the **Back** button on your browser to return to the **Mediajob Reports and Processes** page. Otherwise, click **Return to Main Page** to return to the Webadmin Home Page.

OPAC Reports Utilities

The OPAC Reports utility allows you to run the OPAC Search logging reports.

Popacjob

The `Popacjob` program is comprised of three parts.

- The first, OPAC Log Export (formerly *Popaclogexp*), pulls the specified information from the `OPAC_search_log` table in the database into a comma-delimited text file.
- The second, SDI Searches, runs search queries stored by patrons at the appropriate intervals. SDI allows patrons to choose intervals for each saved search query to be run automatically (on the local database) and the results e-mailed to them in the form of a URL.
- The third, OPAC Bib Usage Log Export (formerly performed using the `-b` switch when running *Popaclogexp*), pulls the specified information from the `bib_usage_table` in the database into a comma-delimited text file.

See [Popacjob](#) on [page 16-1](#) for more information.



Procedure 31-16. Running OPAC Reports and Processes Using WebAdmin

Use the following to run the OPAC Utility.

1. After logging in to WebAdmin, from the main page, OPAC section, click on the **Reports and Processes** link.

Result: The **OPAC Reports and Processes** page opens (see [Figure 31-17](#)).

OPAC Reports and Processes

This utility allows you to run any of the OPAC reports and processes described in the Accessories manual. Select the report you want to run and press the submit button.

Select Job to Run:	
☐ OPAC Search Log Export ☐ Bib Usage Log Export	☒ Today - days ☐ Date Range (YYYY-MM-DD) - Purge Log after export ? ☐
☐ SDI Searches	
☐ Run Jobs 1 - 3	☒ Today - days

Figure 31-17. OPAC Reports and Processes Page

2. Complete the fields on this page, see [Popacjob](#) on [page 16-1](#) for more information.
3. In the **E-mail Address** field, enter your e-mail address. The Utility will send an email to that address when the job is done processing. You must enter the entire email address (user@hostname.com) in order for the e-mail to be sent.

4. When you have verified that the settings on the page are all correct, click the **Submit OPAC Job** button

Result: The transfer process begins. The **Job Scheduler** page displays information about the job.

5. Click the **Back** button.

Result: Returns to the main menu.

System Admin Utilities

On the main page, the System Administrative utilities available are directory listings of the log files and of the report files.

Log Files Utility

The Voyager Log Files page provides you with a list of your batch job log files. This page lists all of the files that exist in the log directory on your server. This directory is commonly the `/m1/voyager/xxxxdb/log` directory. You can view any of the files in the directory.



Procedure 31-17. Accessing Log Files on the Server Using WebAdmin

Use the following to access the log files:

1. After logging in to WebAdmin, from the main page, System Admin section, click the **Log Files** link.

Result: The **Voyager Log Files** page opens.

2. Double-click the file that you want to view.

Result: The log file will display on the screen.

3. When you are done viewing the file, click the **Back** button twice.

Result: Returns to the log file list, then to the main menu.

Report Files Utility

The Voyager Report Files page provides you with a list of your batch job report files. This page lists all of the files that exist in the report directory on your server. This directory is commonly the `/m1/voyager/xxxdb/rpt` directory. You can view any of the files in the directory.



Procedure 31-18. Accessing Report Files on the Server Using Webadmin

Use the following to access the log files.

1. After logging in to WebAdmin, from the main page, System Admin section, click the **Report Files** link.

Result: The **Voyager Report Files** page opens.

2. Double-click the file that you want to view.

Result: The file opens.

3. When you are done viewing the file, click the **Back** button twice.

Result: Returns to the report file list, then to the main menu.

FPC Admin Utilities

The FPC (Fiscal Period Close) Admin utilities are optional features of WebAdmin. See [Figure 31-18](#).

To enable the FPC Admin utilities in WebAdmin, see [Enabling FPC Admin Utilities](#) on [page 31-42](#).

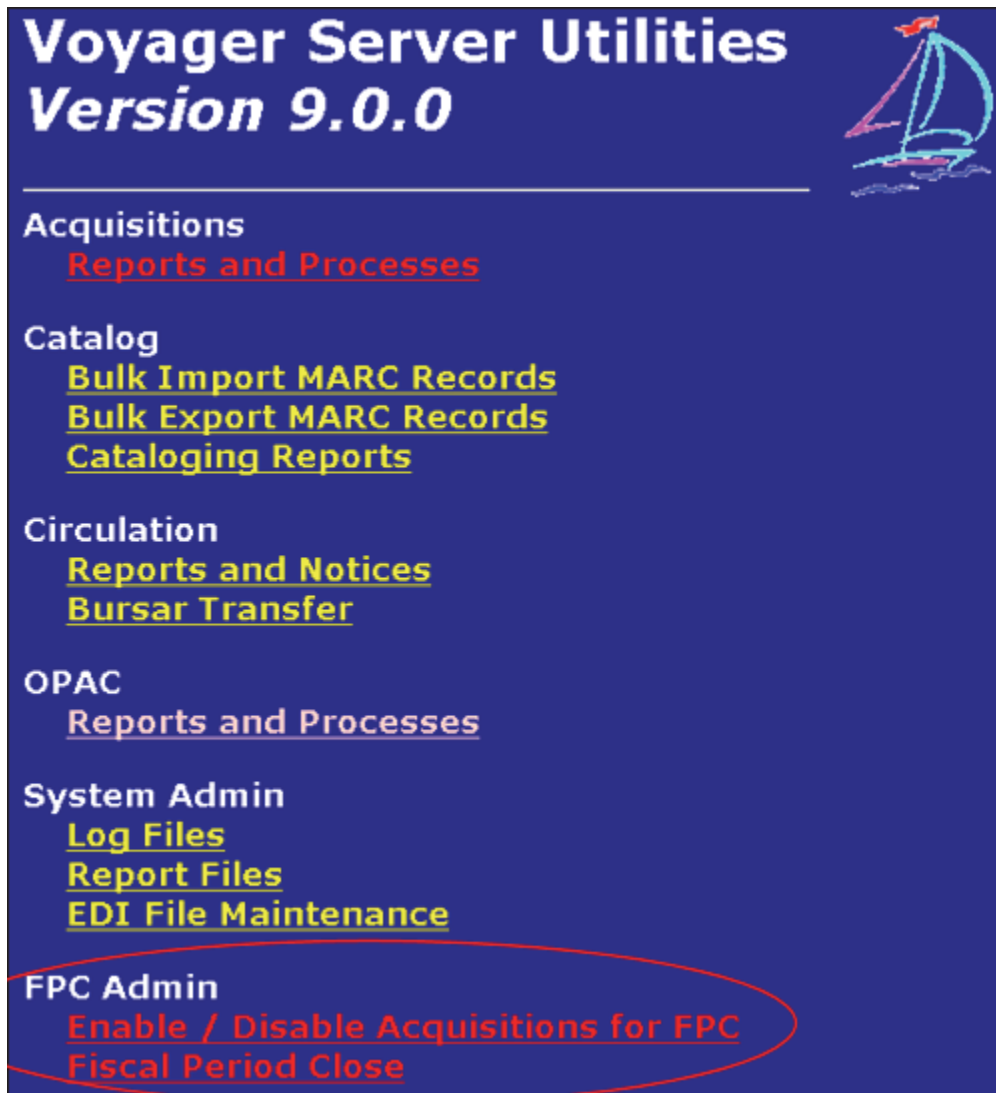


Figure 31-18. WebAdmin Main Menu with Optional FPC Admin Utilities

With these utilities, you can:

- Enable and disable Acquisitions for the purposes of your fiscal period close
See [Figure 31-19](#).

Use these options to turn off Acquisitions while you run FPC and then subsequently, turn Acquisitions on when you are finished.

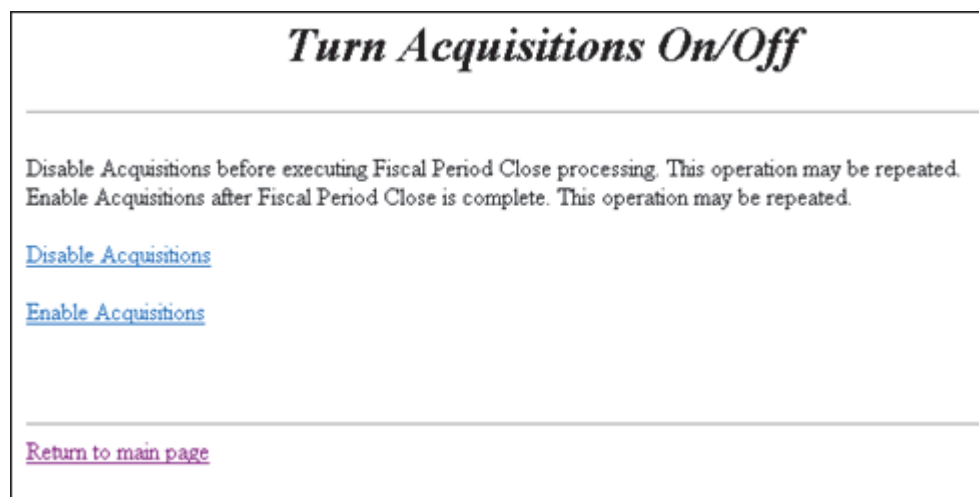


Figure 31-19. Disable and Enable Acquisitions Options

- Run your fiscal period close
See [Figure 31-20](#).
Use these options to execute fiscal period save, fiscal period close, and, optionally, generate reports.

Fiscal Period Close

This utility allows you to run Fiscal Period Close processing.
In normal operations, you should execute *Fiscal Period Save* and wait for it to finish. Then execute *Fiscal Period Run*.

Select the job you want to run, and press the submit button.

CAUTION: DO NOT RUN THIS MULTIPLE TIMES

When a job is completed, you will receive notification via email that the job has ended as well as any additional information which was provided by the job.

Finally, check the "Voyager's Report Files" link for output files with the current date and time which will be generated from this run.

Select Job to Run:

☒ FPC Save

☐ FPC Run

Reports (Optional)

☐ Open Orders Reports

☐ Fund Snapshot Reports

☐ Rollover Status Reports

Email Address: **required**

Figure 31-20. Fiscal Period Close Options

Refer to the *Voyager Acquisitions User's Guide* for a detailed explanation of Voyager's fiscal period close processing.

Enabling FPC Admin Utilities

To enable the FPC Admin utilities, you need to remove the commenting from the FPC Admin section of the `index.html` file that is located in the `/ml/voyager/xxxdb/webadmin` directory. See [Figure 31-21](#) and [Figure 31-22](#). If you are a Voyager TotalCare installation, contact Support to enable the FPC Admin utilities.

[illegible]

Figure 31-21. FPC Disabled in index.html with Commenting

[illegible]

Figure 31-22. FPC Enabled in index.html without Commenting

Introduction

Oracle stored functions can assist SQL queries by providing a quick way to return common search information, such as MARC fields and invoice details. This section lists the following groups of stored functions that may be called in an SQL query:

- MARC Functions
- Advanced MARC Functions
- Miscellaneous Functions

These groups break the functions up by likely usage in SQL.

NOTE:

The functions are not accessible via Access. These functions are for use with Oracle SQL, either with SQLPLUS on the server or with an SQL client that can pass through SQL exactly as created.

MARC Functions

[Table 32-1](#) on [page 32-2](#) lists the MARC functions that provide basic access to the MARC record and allow staff to perform many basic MARC queries.

For example, the following query returns the 900 tag from all bibliographic records in the database:

```
select Bib_Id, GetBibTag( Bib_Id, '900') from
bib_master;
```

Table 32-1. MARC Functions

Function	Description
<code>GetAllAuthTag</code> <code>(Auth_Id, tag, format)</code>	This function returns all tags for an <i>Auth_Id</i>. Inputs: • <i>Auth_Id</i> specifies Auth ID for the MARC record. • <i>tag</i> specifies the MARC tag, such as '150' or '4XX' (where X indicates a wild card). This can also be a list of tags delimited by spaces such as: 650 651 653 • <i>format</i> specifies one of the following output formats: 1 — plain style or 2 — diagnostic style (includes MARC coding). If you omit the parm, you get format 1. If you include any other value than 1 as the parm, you get format 2. Outputs: Returns all matching tags.

Table 32-1. MARC Functions

Function	Description
GetAllBibTag (Bib_Id, tag, format)	This function returns all tags for a <i>Bib_Id</i>. Inputs: • <i>Bib_Id</i> specifies Bib ID for the MARC record. • <i>tag</i> specifies the MARC tag, such as '245' or '1XX' (where X indicates a wild card). This can also be a list of tags delimited by spaces such as: 650 651 653 • <i>format</i> specifies one of the following output formats: 1 — plain style or 2 — diagnostic style (includes MARC coding). If you omit the parm, you get format 1. If you include any other value than 1 as the parm, you get format 2. Outputs: Returns all matching tags.
GetAllMFHDTAG (MFHD_Id, tag, format)	This function returns all tags for an <i>MFHD_Id</i>. Inputs: • <i>MFHD_Id</i> specifies MFHD ID for the MARC record. • <i>tag</i> specifies the MARC tag, such as '852' or '85X' (where X indicates a wild card). This can also be a list of tags delimited by spaces such as: 650 651 653 • <i>format</i> specifies one of the following output formats: 1 — plain style or 2 — diagnostic style (includes MARC coding). If you omit the parm, you get format 1. If you include any other value than 1 as the parm, you get format 2. Outputs: Returns all matching tags.

Table 32-1. MARC Functions

Function	Description
GetAuthTag (<i>Auth_Id</i> , <i>tag</i>)	This function returns the MARC field for an <i>Auth_ID</i> and <i>tag</i>. Inputs: • <i>Auth_Id</i> specifies Auth ID for the MARC record. • <i>tag</i> specifies the MARC tag, such as '100' or '5XX' (where X indicates a wild card). This can also be a list of tags delimited by spaces such as: 650 651 653 Outputs: Returns the MARC field.
GetBibTag (<i>Bib_Id</i> , <i>tag</i>)	This function returns the MARC field for a <i>Bib_ID</i> and <i>tag</i>. Inputs: • <i>Bib_Id</i> specifies Bib ID for the MARC record. • <i>tag</i> specifies the MARC tag, such as '245' or '1XX' (where X indicates a wild card). This can also be a list of tags delimited by spaces such as: 650 651 653 Outputs: Returns the MARC field.

Table 32-1. MARC Functions

Function	Description
GetAuthSubfield (<i>Auth_Id</i> , <i>tag</i> , <i>subfield</i>)	This function returns the authority subfield information for a specified <i>Auth_ID</i>, <i>tag</i> and <i>subfield</i>. Inputs: • <i>Auth_Id</i> specifies Auth ID for the MARC record. • <i>tag</i> specifies the MARC tag, such as '150' or '1XX' (where <i>X</i> indicates a wild card). • <i>subfield</i> specifies the MARC subfields, such as 'a' or 'ab'. Outputs: Returns the first MARC subfield found.
GetBibSubfield (<i>Bib_Id</i> , <i>tag</i> , <i>subfield</i>)	This function returns the bibliographic subfield information for a specified <i>Bib_ID</i>, <i>tag</i> and <i>subfield</i>. Inputs: • <i>Bib_Id</i> specifies Bib ID for the MARC record. • <i>tag</i> specifies the MARC tag, such as '245' or '6XX' (where <i>X</i> indicates a wild card). • <i>subfield</i> specifies the MARC subfields, such as 'a' or 'ab'. Outputs: Returns the first MARC subfield found.

Table 32-1. MARC Functions

Function	Description
<code>GetMFHDSubfield</code> (<i>MFHD_Id</i> , <i>tag</i> , <i>subfield</i>)	This function returns the MARC subfield information for a specified <i>MFHD_ID</i>, <i>tag</i> and <i>subfield</i>. Inputs: • <i>MFHD_Id</i> specifies MFHD ID for the MARC record. • <i>tag</i> specifies the MARC tag, such as '852' or '86X' (where <i>X</i> indicates a wild card). • <i>subfield</i> specifies the MARC subfields, such as 'h' or 'hi'. Outputs: Returns the first MARC subfield found.
<code>GetMFHDTag</code> (<i>MFHD_Id</i> , <i>tag</i>)	This function returns the MARC field for an <i>MFHD_ID</i> and <i>tag</i>. Inputs: • <i>MFHD_Id</i> specifies MFHD ID for the MARC record. • <i>tag</i> specifies the MARC tag, such as '852' or '9XX' (where <i>X</i> indicates a wild card). This can also be a list of tags delimited by spaces such as: 650 651 653 Outputs: Returns the first MARC tag found.

NOTE:

Since Oracle limits the size of the output data to 4000 characters, these functions may truncate some MARC records.

Users may enter IDs (such as `Bib_ID`) either numerically or as a reference to the appropriate MARC ID column.

Advanced MARC Functions

[Table 32-2](#) on [page 32-7](#) lists the functions that provide more advanced access to the MARC records than the MARC functions listed in [Table 32-1](#) on [page 32-2](#), which may call these functions to retrieve fields from the MARC record.

Table 32-2. Advanced MARC Functions

Function	Description
<pre>GetAllTags (binary_integer id, char rectype, char tag, binary_integer format)</pre>	This function returns all matching tags. Inputs: <i>id</i> specifies one of the following ID types: Bib ID, Auth ID, or MFHD ID. <i>rectype</i> indicates one of the following record types: Bib ('B'), Auth ('A'), or MFHD ('M'). <i>tag</i> specifies the MARC tag, such as '100' or '1XX' (where X indicates a wild card). <i>format</i> specifies one of the following output formats: 1 — plain style, 2 — diagnostic style, or 3 — normalized style. The default value is 1. Outputs: Returns all matching tags.
<pre>getAuthBlob (Auth_Id)</pre>	This function extracts raw authority blob information for a specific MARC record. Inputs: <i>Auth_Id</i> specifies Auth ID for the MARC record. Outputs: Returns the raw authority blob.
<pre>getBibBlob (Bib_Id)</pre>	This function extracts raw bibliographic blob information for a specific MARC record. Inputs: <i>Bib_Id</i> specifies Bib ID for the MARC record. Outputs: Returns the raw bibliographic blob.

Table 32-2. Advanced MARC Functions

Function	Description
GetMarcField (Bib_Id, Auth_Id, MFHD_Id, Taglist, Indicators, Subfields, TagOccurrence, TagOccurrenceCompare, FieldLength, FieldLengthCompare, SubfieldOrder, SubfieldOrderCompare, SubfieldLength, SubfieldLengthCompare, SubfieldPattern, Text, MatchType, CompareType, Format)	This function populates the inputs to GetMarcField() and returns the MARC field. Inputs: • <i>Bib_Id</i> specifies the ID of a bibliographic record. • <i>Auth_Id</i> specifies the ID of an authority record. • <i>MFHD_Id</i> specifies the ID of a holdings record. • <i>Taglist</i> specifies a list of MARC tags, separated by either a comma or a space. • <i>Indicators</i> specifies the MARC indicators to be filtered (such as '1_', where _, #, and x are treated as wildcards). • <i>Subfields</i> specifies a list of subfields for which to search. • <i>TagOccurrence</i> specifies the tag occurrence (such as 2, which returns the second occurrence of each tag specified in <i>Taglist</i>). The default value is 0, which returns the first occurrence. • <i>TagOccurrenceCompare</i> indicates which of the following operations to use for the tag occurrence filter: 0 — first matching occurrence, 1 — first occurrence that is not equal to specified occurrence, 2 — next matching occurrence, 3 — previous matching occurrence. The default value is 0. • <i>FieldLength</i> specifies the length of the field. • <i>FieldLengthCompare</i> indicates which of the following operations to use for the field length filter: 0 — equal, 1 — not equal, 2 — greater than, 3 — less than. (Continued on next page.)

Table 32-2. Advanced MARC Functions

Function	Description
GetMarcField() (Continued from previous page.)	• <i>SubfieldOrder</i> specifies the subfield order (such as 2, which returns each subfield that matches subfields in the second position). The default value is 0, which returns the first matching subfield in any order. • <i>SubfieldOrderCompare</i> indicates which of the following operations to use for the subfield order filter: 0 — first matching subfield in any order, 1 — first matching subfield that is not in <i>SubfieldOrder</i>, 2 — first matching subfield that comes after <i>SubfieldOrder</i>, 3 — first matching subfield that comes before <i>SubfieldOrder</i>. The default value is 0. • <i>SubfieldLength</i> specifies the length of the subfield. • <i>SubfieldLengthCompare</i> indicates which of the following operations to use for the subfield length filter: 0 — equal, 1 — not equal, 2 — greater than, 3 — less than. • <i>SubfieldPattern</i> specifies a pattern in which the subfields are ordered within a field. • <i>Text</i> contains the search pattern. • <i>MatchType</i> indicates the following search positions: 0 — text may appear successively in any position within the field or subfield, 1 — text must start at the beginning of field or subfield, 2 — text must appear at the end of a field or subfield, 3 — text is treated similarly to the <code>Like</code> command in SQL, allowing % and _ as wildcards. (Continued on next page.)

Table 32-2. Advanced MARC Functions

Function	Description
<code>GetMarcField()</code> (Continued from previous page.)	• <i>CompareType</i> indicates the following types of search treatments: 0 — full normalization, 1 — case free, or 2 — raw. The default value is 1. • <i>Format</i> specifies one of the following output formats: 1 — plain style, 2 — diagnostic style, or 3 — normalized style. The default value is 2. Outputs: Returns the requested MARC field.
<code>getMFHDBlob</code> (<i>MFHD_Id</i>)	This function extracts raw holdings blob information for a specific MARC record. Inputs: <i>MFHD_Id</i> specifies Bib ID for the MARC record. Outputs: Returns the raw holdings blob.

Table 32-2. Advanced MARC Functions

Function	Description
GetTag (<i>id</i> , <i>rectype</i> , <i>tag</i> , <i>occurrence</i> , <i>indicators</i> , <i>subfields</i> , <i>format</i>)	This function populates the inputs to GetMarcField() and returns the MARC field. Inputs: • <i>id</i> specifies one of the following ID types: Bib ID, Auth ID, or MFHD ID. • <i>rectype</i> indicates one of the following record types: Bib ('B'), Auth ('A'), or MFHD ('M'). • <i>tag</i> specifies the MARC tag, such as '100' or '1XX' (where X indicates a wild card). This can also be a list of tags delimited by spaces such as: 650 651 653 • <i>occurrence</i> specifies the tag occurrence. • <i>indicators</i> specifies the MARC indicators. • <i>subfields</i> specifies the MARC subfields. • <i>format</i> specifies one of the following output formats: 1 — plain style, 2 — diagnostic style, or 3 — normalized style. The default value is 1. Outputs: Returns the MARC field.

Miscellaneous Functions

[Table 32-3](#) on [page 32-12](#) lists the functions that may assist staff in acquisition and circulation queries by performing the business logic of the queries for them.

For example, the following circulation query returns the active email address from all patron records in the database:

```
select first_name, last_name, GetPatronEmailAddress(
Patron_Id) as email from patron;
```

For example, the following acquisitions query returns the base currency amount from all invoices in the database:

```
select invoice_number, ToBaseCurrency( Total,
Currency_Code, Conversion_Rate) from invoice;
```

Table 32-3. Miscellaneous Functions

Function	Description
<code>getBaseCurrencyCode</code>	This function returns the base currency code. Inputs: None. Outputs: <i>Base_Currency_Code</i> contains the base currency code.
<code>getCallNoClass</code> (<i>Norm_Call_Num</i> , <i>Call_Num_Type</i>)	This function returns the class of the call number. Inputs: • <i>Norm_Call_Num</i> specifies the normalized call number. • <i>Call_Num_Type</i> specifies the Call Number Type. Outputs: Returns the call number class.

Table 32-3. Miscellaneous Functions

Function	Description
<code>getConversionRate</code> <code>(Currency_Code, Date)</code>	This function returns a currency conversion rate. Inputs: • <i>Currency_Code</i> specifies the currency code. • <i>Date</i> specifies the date of the currency rate. Defaults to current date if not supplied. Outputs: Returns a conversion rate to a format that the database can use to perform calculations.
<code>getFirstPatronBarcodeID</code> <code>(Patron_ID)</code>	This function returns the first active patron group for a patron on the basis of status and date. Inputs: <i>Patron_ID</i> is a patron ID. Outputs: Returns the most recent active barcode ID for a patron.
<code>getItemBarcode</code> <code>(Item_ID)</code>	This function returns the active item barcode. Inputs: <i>Item_ID</i> specifies the item's ID. Outputs: Returns the active barcode for the item or an empty string if not active.
<code>getLatestItemStatus</code> <code>(Item_ID)</code>	This function returns the ID of the current status for an item. Inputs: <i>Item_ID</i> specifies the item's ID. Outputs: Returns the ID of the most recently applied status.

Table 32-3. Miscellaneous Functions

Function	Description
<code>getPatronActiveAddress</code> (<i>Patron_ID</i>)	This function returns the active address ID for a patron. Inputs: <i>Patron_ID</i> specifies the patron's ID. Outputs: Returns the address ID for the patron.
<code>getPatronEmailAddress</code> (<i>Patron_ID</i>)	This function returns the active email address for a patron. Inputs: <i>Patron_ID</i> is a patron ID. Outputs: Returns the active email address for the patron or an empty string if not active.
<code>setCurrencyDecimals</code> (<i>Amount, Currency_Code</i>)	This function uses the <i>Currency_Code</i> to convert the database representation of an amount to a more familiar currency amount. Inputs: • <i>Amount</i> specifies the amount to convert. • <i>Currency_Code</i> specifies the currency code. If not specified, the base conversion code is used for the query. Outputs: Returns the display version of the currency amount.
<code>setConvDecimals</code> (<i>Rate</i>)	This function converts an integer <i>Rate</i> to a floating-point number. Inputs: <i>Rate</i> specifies the integer rate to convert. Outputs: Returns a display version of a conversion rate.

Table 32-3. Miscellaneous Functions

Function	Description
<code>toBaseCurrency</code> (<i>Amount</i> , <i>Currency_Code</i> , <i>Rate</i>)	This function converts the database representation of an amount to a base currency amount. Inputs: • <i>Amount</i> specifies the amount to convert. • <i>Currency_Code</i> specifies the currency code. • <i>Rate</i> specifies the conversion rate. Outputs: Returns a base currency amount.
<code>truncField</code> (<i>Field</i>)	This function limits strings to 4000 characters (the maximum number of characters that can be output by SQLPLUS). Inputs: <i>Field</i> specifies the string to truncate. Outputs: Returns the truncated string.

Introduction

SMS (Short Message Service) capability has been incorporated into Voyager to provide institutions with more flexibility for staff to communicate with patrons. Specifically, patrons may choose to receive circulation notices (per [Circulation Notices](#) on [page 33-5](#)) via their mobile devices.

Since SMS is a configurable feature, each institution has the option to choose if it is implemented. The SMS feature is intended to augment the existing features for print and/or e-mail notices. These traditional capabilities remain available.

Implementation of SMS includes the following components:

- Voyager ILS SMS feature
- Ex Libris SMS Proxy
- Third-party SMS gateway/service provider's software
- Patron's mobile device

See [Figure 33-1](#) for an illustration of the flow of a notice from the Voyager ILS to the patron's mobile device and the components involved.

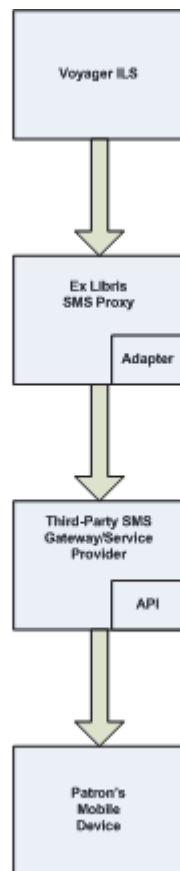


Figure 33-1. Components/Path for Voyager SMS Notices Sent

NOTE:

The Voyager SMS feature is designed for one-way transmission, messages to be sent outbound to patrons. There is no mechanism for receiving messages.

Implementation

In order to implement SMS, you need to:

- Purchase credits from an SMS gateway/service provider (see [SMS Gateway/Service Providers](#) on [page 33-3](#))
- Install the SMS gateway/service provider's software (see [SMS Gateway/Service Providers](#) on [page 33-3](#))

- Install Ex Libris SMS Proxy software (see [Ex Libris SMS Proxy](#) on [page 33-4](#))
- Set up/configure Voyager (see [Voyager Setup/Configuration](#) on [page 33-4](#))
 - Circulation notices (see [Circulation Notices](#) on [page 33-5](#))
 - `webvoyage.properties` (see [webvoyage.properties](#) on [page 33-5](#))
 - `voyager.env` (see [voyager.env](#) on [page 33-6](#))
 - `SmsProxyConf.xml` (see [SmsProxyConf.xml](#) on [page 33-7](#))
 - `Pcircjob/Pmediajob` (see [Pcircjob/Pmediajob](#) on [page 33-9](#))
- Set up the patron (see [Patron SMS Number](#) on [page 33-9](#))
- Set up logging for SMS (see [SMS Send Status Log](#) on [page 33-11](#))

SMS Gateway/Service Providers

The SMS gateway/service provider's software/service bridges the connection between the Ex Libris SMS Proxy software (see [Ex Libris SMS Proxy](#) on [page 33-4](#)) and the patron's mobile device. The SMS gateway handles SMS operations such as routing, that is, forwarding and storing messages on their way to the desired endpoints.

Since each provider has a different method (API) of receiving SMS requests, the Ex Libris SMS Proxy software is needed to convert SMS requests from the main Ex Libris products (such as Voyager, Primo, and so forth) to the appropriate format as defined by the SMS gateway/service providers. This enables the transfer of the following message/routing components:

- User ID
- Password
- Sender phone number
- Destination phone number
- Message text

Prior to implementing Voyager SMS, you need to purchase credits from an SMS gateway/service provider. When you purchase this service, you get a user ID and password from the provider that needs to be configured into `SmsProxyConf.xml` (see [SmsProxyConf.xml](#) on [page 33-7](#)).

Ex Libris SMS Proxy

The Ex Libris SMS Proxy software (provided with your installation) is used to pass SMS messages to gateway/service providers following the providers' defined methods for handling the transaction. The component inside the SMS proxy software that is used to process the transaction is known as an adapter.

With the SMS proxy software, Ex Libris provides adapters for the following SMS gateway/service providers:

- Clickatell (www.clickatell.com)
- SimpleWire (www.openmarket.com)

The architecture of the Ex Libris SMS Proxy software enables adding adapters that you may have written to allow support of providers that are already in use by your institution. Refer to the Developer Zone on the EL Commons collaborative Web site (<http://www.exlibrisgroup.org>).

For more information regarding the Ex Libris SMS Proxy software, refer to the *SMS Proxy User's Guide* available in the Cross Product section of the Ex Libris Documentation Center (Cross-Product > Technical Documentation > SMS Proxy).

Voyager Setup/Configuration

For the Voyager setup, you need to:

- Confirm that the existing notices meet your requirements or modify them
See [Circulation Notices](#) on [page 33-5](#)).

- Determine if you want patrons to enter their own mobile device numbers via the WebVoyage Patron Information page

If you choose to have patrons enter this information, you need to configure `webvoyage.properties` to display this option.

See [webvoyage.properties](#) on [page 33-5](#).

- Modify the `voyager.env` file to trigger the Pcirjob and Pmediajob to evaluate if patron SMS numbers should be used for distributing circulation notices.

See [voyager.env](#) on [page 33-6](#).

- Configure the `SmsProxyConf.xml` file with SMS gateway/service provider information

See [SmsProxyConf.xml](#) on [page 33-7](#).

Circulation Notices

The format/content of circulation notices for SMS messages is set up in the `smsnotices.properties` file that is located in `/ml/voyager/xxxxdb/ini/`. See [Figure 33-2](#).

```
<entry key="Media-00">You have a media item that was due on \d: \T</entry>
<entry key="Circ-01">You have an item available for pickup until \d: \T</entry>
<entry key="Circ-02">You have an item that was due on \d: \T</entry>
<entry key="Circ-03">You have an item that has been recalled and is now due on \d: \T</
entry>
<entry key="Circ-04">You have a recalled item that is overdue: \d: \T</entry>
<entry key="Circ-07">You have an item that will be due on \d: \T</entry>
```

Figure 33-2. smsnotices.properties File

The following notice types can be sent with SMS messaging:

- Circulation and media overdue notices (circjob2 and mediajob1)
- Circulation hold request item available notices (circjob5)
- Circulation recall notices (circjob3)
- Circulation recall overdue notices (circjob 2)
- Circulation courtesy notices (circjob 7)

Notices can be a maximum of 160 characters in length with the following formatting characteristics:

- `\d` is used as the variable placeholder for date
You can also specify the date format in the `smsnotices.properties` file. The default format is month-day-year (MM-dd-yyyy).
- `\T` is used as the variable placeholder for the title of the item
- Title references in the messages may be truncated, when sent, if the complete message is going to exceed 160 characters

webvoyage.properties

The settings in the `webvoyage.properties` file enables the SMS option to display on the Patron Information page. See [Figure 33-3](#).

The `webvoyage.properties` file is located in `/ml/voyager/xxxxdb/tomcat/vwebv/context/vwebv/ui/en_US/` where `xxxxdb` represents your database name and `en_US` represents the default skin path.

```
#=====
# SMS number
# Change the Option.SMS to Y to allow patrons to enter an SMS (cell phone) number through
# WebVoyage. Patrons who provide an SMS number are allowing the library to send selected
# circ and media notices via SMS message (text message), as long as the library has
# configured
# such messages to be sent
#=====
option.SMS=N
```

Figure 33-3. webvoyage.properties File

To activate this feature, you need to change the `option.SMS` setting to `Y`.

The help text associated with the SMS phone number field on the Patron Information page, "If you would like to receive circulation notices from the library on your mobile device ...," (see [Figure 33-8](#)) is set in `pageProperties.xml` (see [Figure 33-4](#)) and may be modified to match your site's requirements.

The `pageProperties.xml` file is located in `/m1/voyager/xxxdb/tomcat/vwebv/context/vwebv/ui/en_US/xsl/userTextConfigs/` where `xxxdb` represents your database name and `en_US` represents the default skin path.

```
<page name="page.myAccount.viewPersonalInfo.smsInstruction" position="aboveContent">
  <div class="instructionText">
    If you would like to receive circulation notices from the library on your mobile
    device, please
    enter the mobile device number in the box below. Please be sure to include
    the country code (1 for the United States) before the area code. Example: 18885551234.
  </div>
</page>
```

Figure 33-4. pageProperties.xml File

voyager.env

When `Pcircjob2`, `Pcircjob3`, `Pcircjob5`, or `Pmediajob1` are run, they check the `voyager.env` file to determine if the SMS feature is to be used for sending circulation notices. To activate this feature, the `export` row needs to be uncommented and set to the correct server IP and port (see [Figure 33-5](#)) which is generally done at installation. The `voyager.env` file is located in `/m1/voyager/xxxdb/ini/`.

```
#####
## SMS messages configuration
## Uncomment the following line in order to send SMS (text/cellphone) messages
## for selected
## circulation notices. By uncommenting the line, you are informing the
## circulation and media
## jobs to attempt to send SMS notices using this URL. You may temporarily
## suspend sending
## SMS messages by commenting the line. See the circulation documentation for
## further
## configuration required to allow SMS messages to be sent.
#export SMSPROXY=http://<server IP>:7038/core-sms-proxy/sms
```

Figure 33-5. voyager.env File

SmsProxyConf.xml

The `SmsProxyConf.xml` file is used to identify/configure the SMS gateway/service provider information. See [Figure 33-6](#). When you purchase credits from an SMS gateway/service provider, you receive credentials (user ID and password) that need to be entered into the `SmsProxyConf.xml` file.

The `SmsProxyConf.xml` file is configured with the SimpleWire and Clickatell provider information by default since the Ex Libris SMS proxy software provides the adapters for these providers. However, you may use other providers with the adapters that you have created for them. See [Ex Libris SMS Proxy](#) on [page 33-4](#), [Ex Libris EL Commons](#), and the *SMS Proxy User's Guide* available in the Cross Product section of the Ex Libris Documentation Center
(Cross_Product_Information_Distributors >
Cross_Product_Information_Customers > SMS Proxy).

```
<Senders>
  <Sender>xx.xxx.x.xx</Sender>
</Senders>
<Institutions>
  <Institution name="Simplewire Demo Library" code="xxxdb">
    <ProviderCode>simplewire</ProviderCode>
    <ProviderUser>SMS_PROXY_OPENMARKET_USER</ProviderUser>
    <ProviderPassword>SMS_PROXY_OPENMARKET_PASSWORD</ProviderPassword>
  </Institution>
  <Institution name="Clickatell Demo Library" code="xxxdb">
    <ProviderCode>clickatell</ProviderCode>
    <ProviderUser>SMS_PROXY_CLICKATELL_USER</ProviderUser>
    <ProviderPassword>SMS_PROXY_CLICKATELL_PASSWORD</ProviderPassword>
    <ProviderMo>1</ProviderMo>
    <ProviderFrom>1111</ProviderFrom>
  </Institution>
</Institutions>

<Providers>
  <Provider name="SimpleWire" code="simplewire">
    <component>com.exlibris.core.sms.proxy.SimpleWireProvider</component>
  </Provider>
  <Provider name="Clickatell" code="clickatell">
    <component>com.exlibris.core.sms.proxy.ClickatellProvider</component>
  </Provider>
</Providers>
</SmsProxyRoot>
```

Figure 33-6. SmsProxyConf.xml File

For each institution, you need to define:

- Institution name and code (Voyager System Administration database code/ xxxdb)
- Provider code
- User ID
- Password
- ProviderMo (remains at 1)
- ProviderFrom (the two-way number assigned to your account)

One institution needs to be defined for each database (xxxdb).

Also defined in this configuration file is the sender and provider information. The sender information is the IP address of the server that sends the outgoing message. The provider component information is the fully qualified class name of the Java class that implements the adapter for the vendor selected. The provider component is supplied by default for SimpleWire and Clickatell. If you create your own adapter, you need to provide this information.

The `SmsProxyConf.xml` file is located in `/ml/voyager/xxxdb/tomcat/smsproxy/context/core-sms-proxy/WEB-INF/classes/`.

Pcircjob/Pmediajob

The following existing jobs, when run, trigger the SMS notices to be sent:

- `circjob2`
- `circjob3`
- `circjob5`
- `mediajob1`

The triggers that cause these jobs to send an SMS message to a patron are:

- SIF message type of circulation overdue, recall, item available, and/or media overdue notice
- Existence of the SMS Proxy URL in the `voyager.env` file
- Existence of the patron SMS number (mobile device number) in the patron table in the Voyager database (see [Patron SMS Number](#) on [page 33-9](#)).

All of these triggers need to exist for the SMS message to be sent.

Use WebAdmin to verify that the circulation and/or media jobs are running as scheduled. View the information provided in Circulation > Reports and Notices.

Patron SMS Number

The patron's mobile device number may be entered in two ways:

- By adding or editing a patron record in the Circulation client (see [Figure 33-7](#))
- Entered by patrons on the Patron Information page in WebVoyage (see [Figure 33-8](#))

NOTE:

Only one mobile number is permitted per patron regardless of where it is entered.

Add New Patron Record

Name	Barcode	Address	Phone
Last: First: Middle: Title: Name Type: Personal Institution Id: SSN: ___-__-__ SMS Number: Expires: 9/21/2011 ... Purge Date: 9/30/2011 ...			

Patron Record

<New Patron>

Figure 33-7. Add New Patron

Personal Information

[My Account](#)[Edit Preferences](#)

Name: Sandy Smith

Email: sandy.smith@abcinstitute.edu

Permanent Address: 532 N. 3rd St.

Watseka IL 60970

If you would like to receive circulation notices from the library on your mobile device, enter your mobile device number in the field provided starting with the country code. Examples: 18885551234, 1-888-555-1234, or 1(888) 555-1234, +61 9 1234 1234 or 61912341234

SMS Number:

[Save](#)

Figure 33-8. Patron Information Page - Mobile Device Number

SMS Send Status Log

To track the status of SMS messages that are sent, you need to configure the `sms-log4j.properties` file. See [Figure 33-10](#). The `sms-log4j.properties` file identifies the specific log file for logging the status.

The status logged includes:

- SMS proxy send status
- Provider receive status
- Patron ID

- Message type
- Patron SMS mobile device number

See [Figure 33-9](#) for an example.

```
2010-04-09 16:48:22,427 0      [main] INFO  SMSLOG - Send SMS result for patron : 1000008
SMS number : 18005551234
Message Type : Circulation overdue notice
SMS Proxy Status: request accepted
SMS Provider Status: ID: 12912c01604d3c7aaa712372cd349dbf
Send date : 2010-03-04 13:04:50
```

Figure 33-9. SMS Status Log Example

Confirm that the correct path (using the your xxxdb) is identified for the
`log4j.appender.LOGFILE.File=` option.

```
log4j.rootLogger=WARN, CONSOLE
log4j.appender.CONSOLE=org.apache.log4j.ConsoleAppender
log4j.appender.CONSOLE.encoding=UTF-8
log4j.appender.CONSOLE.layout=org.apache.log4j.PatternLayout
log4j.appender.CONSOLE.layout.ConversionPattern=%-4r [%t] %-5p %c %x - %m%n
log4j.logger.com.endinfosys=DEBUG, LOGFILE
#log4j.logger.com.endinfosys.web.util=DEBUG
#log4j.logger.com.endinfosys=FATAL
log4j.appender.LOGFILE=org.apache.log4j.DailyRollingFileAppender
log4j.appender.LOGFILE.File=/m1/voyager/xxxdb/tomcat/smsproxy/logs/sms.log
log4j.appender.LOGFILE.datePattern='.'yyyy-MM-dd
log4j.appender.LOGFILE.append=true
log4j.appender.LOGFILE.encoding=UTF-8
log4j.appender.LOGFILE.layout=org.apache.log4j.PatternLayout
log4j.appender.LOGFILE.layout.ConversionPattern=%d{ISO8601} %-4r [%t] %-5p %c %x - %m%n
```

Figure 33-10. sms-log4j.properties File

The `sms-log4j.properties` file is located in `/m1/voyager/xxxdb/tomcat/smsproxy/context/core-sms-proxy/WEB-INF/classes/`.

Introduction

The system monitoring job provides a way for a Voyager system administrator to obtain information about the server. It can be set to alert a system administrator via email when a threshold is met for:

- Free space
- Oracle table space
- Free space on the file system

Purpose of This Chapter

This chapter provides the following:

- An overview of the monitor program (see [Overview](#) on [page 34-2](#))
- Parameters to run the monitor program (see [Parameters](#) on [page 34-2](#))
- Examples of how to run the monitor program (see [Examples](#) on [page 34-3](#))
- How to setup the monitor program (see [Setup](#) on [page 34-3](#))

Overview

The system monitor job (Pmonjob) can be setup and run on each database. It can be run from either the command line interactively, a cronjob, or a scheduled task on Windows to send automated email messages to any specified email address.

Parameters

The System Monitor Job utilizes the parameters described in [Table 34-1](#).

Table 34-1. System Monitor Job Parameters

Parameter	Description
No option	Provides a list of all the checks that should be addressed as soon as possible
-r	Displays a full report of information reviewed
-x	Indicates that emails should not be sent to Ex Libris Customer Support
-h	Provides a detailed usage of the program
-w	Provides all checks that are flagged as warnings or urgent

Examples

See below for examples of the System Monitoring Job reporting/alerts.

```
cd /m1/voyager/<db>/sbin/
./Pmonjob -h
Usage: /m1/voyager/bin/2008.2.1/monitor [ -r Output reporting info] [ -x : Do not send
      error reports to Ex Libris ] [ -w <Warning Threshold> ]

./Pmonjob
Warnings for Voyager database <db> on voyqa
* FILESYSTEM /m1 - 91.9 % used.  33023 Mb remain
* ZFS Pool no:

* ORACLE LOG - ORA-01110: data file 201: 'D:\ORACLE\ORADATA\VGER2\TEMP01.DBF'
* ORACLE LOG - ORA-01578: ORACLE data block corrupted (file # 201, block # 4194057)
* ORACLE LOG - ORA-01578: ORACLE data block corrupted (file # 201, block # 4194058)
* ORACLE LOG - ORA-01578: ORACLE data block corrupted (file # 201, block # 4194059)

./Pmonjob -r
Oracle Tablespace info for database <db>:
Tablespace      Maximum      Used      Allocated  % Used
<DB>            20000 mb    778 mb    1300 mb    3.9 %

Filesystem fs for database <db>:
Mountpoint      Maximum      Used      % Used
/m1             81920 mb    79717 mb  97.3 %

The dynamic keyword file is 5 % of the static file size.

Checking Alert Log for errors:
Checking System Logs for errors:
Warnings for Voyager database <db> on <hostname>
* FILESYSTEM /m1 - 97.3 % used.  2202 Mb remain

./Pmonjob -w
Warnings for Voyager database <db> on <hostname>
* TABLESPACE <DB> - 3.9 % used.  19221 Mb remain
* FILESYSTEM /m1 - 97.3 % used.  2202 Mb remain
```

Setup

Ensure that the scripts are set up for the database.

If /m1/voyager/bin/<version>/monitor doesn't exist, then do the following:

Unix

```
su - voyager
cd /m1/voyager/bin/<version>
cp -p /m1/utility/monitor monitor
chmod 750 monitor
```

Windows

```
cd D:/ml/voyager/bin/<version>
cp -p D:/ml/utility/monitor monitor
```

If /ml/voyager/<db>/sbin/Pmonjob doesn't exist, then do the following:

Unix

```
su - voyager
cd /ml/voyager/<db>/sbin
cp -p /ml/utility/Pmonjob Pmonjob
chmod 750 Pmonjob
```

Windows

```
cd D:/ml/voyager/bin/<version>
cp -p D:/ml/utility/monitor monitor
```

Automate running the System Monitor Job for <db> by doing the following:

Unix

```
su - voyager
crontab -e
```

To set this database to run every day at 3:30 AM, specify:

```
30 3 * * * /ml/voyager/<db>/sbin/Pmonjob
```

Windows

1. Log in as Administrator, Voyager, or any ID that is a member of the Local Administrators group.
2. Open **Windows Explorer> My Computer> Control panel> Scheduled tasks**.
3. Double click the **Add Scheduled Task** icon.
The Scheduled Task Wizard windows opens.
4. Click **Next**.
5. In the Scheduled Task Wizard window, from the **Click the program you want Windows to run** list, select the KornShell application and click **Next**.

6. For the **Type a name for this task** field:
 - a. Enter **<db> Monitor Job** (not case sensitive).
 - b. Select the **Daily** radio button and click **Next**.
7. For the **Select the time and day you want this task to start** field:
 - a. Enter the start time for which you want the job to begin running in the **Start time** field.
 - b. Leave the default of **Every Day**.
 - c. Leave the default **Start Date**.
8. Enter the user name, password (password confirmation), and click **Next**.
The default user name is the name of the operator currently logged in.
9. In the Schedule Task Wizard window, select the **Open advanced properties for this task when I click Finish** check box and click **Finish**.
10. In the <db> Monitor Job window, in the **Run** field, at the end of the command line add the following to the end of the command line:


```
D:/m1/voyager/<db>/sbin/?monjob > D:/m1/voyager/tmp/<db>.monitor.log
```

This will run the job and send a copy of its output to the <db>.monitor.log.

The full command line must be:

```
c:\WINNT\system32\cmd.exe /c %ROOTDIR%\bin\ncenv.cmd 6
sh &&%ROOTDIR%\mksnt\sh.exe -L D:/m1/voyager/<db>/sbin/
?monjob > D:/m1/voyager/tmp/<db>.monitor.log
```
11. Select the **Settings** tab.
12. For the **Stop the task if it runs for** field, change the default of **72 hours** to **30 minutes** and click **OK**.
The Set Account Information window displays.
13. Type the password (password confirmation) and click **OK**.
In the Schedule Task window, the daily backup is listed. The backup is automated.

NOTE:

If email is configured to be sent from this server, an email with any errors or warnings when running this job will be sent to the voyager user on the server.

UseMARCON Configuration for Use with Voyager



What is UseMARCON?

The UseMARCON API converts records from one MARC format to another. It interfaces with Voyager by converting records to or from the MARC21 format (used by Voyager) so that incoming records in non-MARC21 format can be used by Voyager and Voyager records can be used by non-MARC21 format processing applications. UseMARCON is used in conjunction with Bulk Import, MARC Export, and Z39.50 servers.

In This Chapter

This chapter contains information regarding the UseMARCON API's interaction with Voyager and the Voyager configuration required to make UseMARCON interact properly with Voyager. Throughout this document, the distinction between the processes performed by the UseMARCON API and those specific to Voyager and Z39.50 servers is maintained to facilitate troubleshooting problems.

Functions, files, and messages specific to UseMARCON's actual conversion of MARC records from one MARC format to another are referred to as UseMARCON functions, files, or messages. Functions, files, and messages specific to Bulk Import, MARC Export, and Voyager Z39.50 database interaction are referred to as Voyager functions, files, and messages.

For background information regarding UseMARCON, refer to the Koninklijke Bibliotheek (National Library of the Netherlands) and access the following link:

<http://www.kb.nl/hrd/bibinfra/usemarcon/usema-en.html>

For information regarding the UseMARCON converter and creating and editing system/rules files refer to The British Library website and access the following links:

<http://www.bl.uk/bibliographic/pdfs/guidetousemarcon.pdf>

or

http://www.bl.uk/bibliographic/pdfs/usemarcon_manual.pdf

Supported MARC Formats

The MARC formats supported by Voyager's interface with UseMARCON are:

- Unimarc
- Intermarc
- Ukmarc
- Normarc
- Librismarc
- Danmarc
- Finmarc
- Canmarc
- Picamarc
- AMARC21
- Ibermarc

UseMARCON File Structure

The following section details the file structure of UseMARCON as it pertains to Voyager functions.

UseMARCON Conversion Directories

Each conversion consists of *one* direction of MARC format translation *only*. For example, MARC21 to FinMARC and FinMARC to MARC21 are two separate conversions. Therefore, they each require their own conversion directories and

files. These directories contain the files necessary for UseMARCON to make the conversions in that direction, as well as the initialization file to direct UseMARCON to the required files.

Create a `usemarcon` subdirectory in the `ini` directory (`/m1/voyager/xxxdb/ini`) and place the conversion directories in it. A sample would look like Figure A-1.

UNI2US	US2UNI	FIN2US	US2FIN

Figure A-1. Sample Contents of `/m1/voyager/xxxdb/ini/usemarcon`

For more information on the UseMARCON conversion directories, see the *UseMARCON Technical Manual*.

Figure A-2 shows sample contents of the UNI2US conversion profile in the sample `usemarcon` subdirectory in Figure A-1 (for the Unimarc to MARC21 format).

<code>uni2.ii</code>	<code>standard.trs</code>	<code>uni2us.rul</code>	<code>uni.mrc</code>	<code>unius.mrc</code>	<code>uni.chk</code>
<code>us.chk</code>					

Figure A-2. Sample Contents of UNI2US Conversion Profile

Note that the `uni2.ini` (UseMARCON initialization) file and the six required files are in the sample `usemarcon` subdirectory as detailed in [Required Files](#) on [page A-4](#).

For more information on the UseMARCON file structures and the files that it contains, see “System Files” and “Rules Files” sections of the *UseMARCON Technical Manual*.

UseMARCON Initialization (`ini`) File

UseMARCON uses an initialization file for each conversion which points UseMARCON to the files it needs to perform conversions (for that direction). For more information on the UseMARCON initialization file, see the “Initialization File (`.ini`)” section of the “System Files” chapter of the *UseMARCON Technical Manual*.

Each UseMARCON initialization file resides in the conversion directory for its conversion. For more information on the conversion directories, see [UseMARCON Conversion Directories](#) on [page A-2](#).

Required Files

UseMARCON requires certain files to be in each conversion directory. These files must be specified in the UseMARCON initialization file for each conversion.

A sample UseMARCON .ini file can be found in [Sample UseMARCON Initialization File](#) on [page A-5](#).

ErrorLogFile



IMPORTANT:

All of these required files must be readable, and the MARC output file must be writable, using the Voyager ID.

Table A-1. Required Fields

Required File	Description/Function
Error Log File	The path to the error log, usemarcon.errmsgs, is set with the ErrorLogFile= parameter in the initialization file. See Figure A-3 .
Rule (.rul) file	Translation file used by UseMARCON to convert one MARC format to another.
Translation (.trs) file	Character mapping table for any special characters that need to be converted when translating from one MARC format to another.
Input format checking (.chk) table	These files act as validity checking devices to make certain that the records being converted contain information in the requisite fields and subfields, especially where there are required fields necessary to convert records.  IMPORTANT: <i>Make certain that the field requirements of these format checking files are not overly restrictive, to avoid a large number of error messages with the conversions.</i>
Output format checking (.chk) table	

Table A-1. Required Fields

Required File	Description/Function
Marc input (.mrc) file, and	UseMARCON requires two empty files, though they are not used in conversion.
Marc output (.mrc) file	

Sample UseMARCON Initialization File

Figure A-3 is a sample UseMARCON initialization (.ini) file containing examples of the required files.

Line

```
[DEFAULT_FILES]
ErrorLogFile=/m1/voyager/xxxdb/ini/usemarcon.errmsgs
1 RuleFile=/m1/voyager/xxxdb/ini/usemarcon/uni2us/
   uni2us.rul
2 TranscodingCharacterTable=/m1/voyager/xxxdb/ini/
   usemarcon/uni2us/iso2us.trs
3 InputFormatCheckingTable=/m1/voyager/xxxdb/ini/
   usemarcon/uni2us/uni.chk
4 OutputFormatCheckingTable=/m1/voyager/xxxdb/ini/
   usemarcon/uni2us/us.chk
5 MarcInputFile=/m1/voyager/xxxdb/ini/usemarcon/uni2us/
   uni.mrc
6 MarcOutputFile=/m1/voyager/xxxdb/ini/usemarcon/
   uni2us/unius.mrc
```

Figure A-3. Sample of the required files in the UseMARCON initialization file

UseMARCON Log File

The UseMARCON log file is called `usemarcon.log`. All Voyager applications that run UseMARCON write log information to this file.

The log file is stored in the directory from which the application is run. In most cases, that is the `/m1/voyager/xxxdb/sbin` directory since that is where the Pscript is run and where `xxxdb` is your database name.

See [Figure A-4](#) for an example log file.

Line#	
1	-----
2	Started at : Thu Aug 16 11:04:07 2001
3	
4	WARNING(2105)-Unexpected field found in input record : Notice '13819' : field '092'
5	ERROR (1300)-Completed translating MARC record. :
6	WARNING(2102)-Invalid first indicator found in input record : Notice '15654' : field '082' (ind ' ')
7	WARNING(2105)-Unexpected field found in input record : Notice '15654' : field '092'
8	ERROR (3001)-Character not transcoded (unable to find it in transco table) : Notice '15654' : field '300' (Unknown character '¾' (\be)) : table 'us2iso.trs'
9	ERROR (5100)-Rule analysis error : ERROR `Syntax error' in rule 041\$a 101\$a (no) If (I1=0) Then Sto(0); To(3);
10	Redo;
11	Mem(0); From(4) To(6);
12	Redo; Mem(0); From(7) To(9) "
13	ERROR (1300)-Completed translating MARC record. :
14	WARNING(2105)-Unexpected field found in input record : Notice 'UNI0000079' : field '200'
15	WARNING(2105)-Unexpected field found in input record : Notice 'UNI0000079' : field '205'
16	WARNING(2102)-Invalid first indicator found in input record : Notice 'UNI0000079' : field '210' (ind ' ')
17	WARNING(2104)-Invalid or redundant subfield found in input record : Notice 'UNI0000079' : field '210' (subfield '\$c')

Figure A-4. Sample log file

Line#	
18	WARNING(2104)-Invalid or redundant subfield found in input record : Notice 'UNI0000079' : field '210' (subfield '\$c')
19	WARNING(2104)-Invalid or redundant subfield found in input record : Notice 'UNI0000079' : field '210' (subfield '\$d')
20	WARNING(2106)-Mandatory field expected in input record : Notice 'UNI0000079' : field '008'
21	WARNING(2106)-Mandatory field expected in input record : Notice 'UNI0000079' : field '245'
22	WARNING(5004)-Only one indicator has been found : I2 is missing in field 700
23	WARNING(7103)-Invalid second indicator found in output record : Notice 'UNI0000080' : field '700' (ind ' ')
24	WARNING(7106)-Mandatory field expected in output record : Notice 'UNI0000080' : field '100'
25	WARNING(7106)-Mandatory field expected in output record : Notice 'UNI0000080' : field '101'
26	WARNING(7106)-Mandatory field expected in output record : Notice 'UNI0000080' : field '200'
27	ERROR (1300)-Completed translating MARC record. :

Figure A-4. Sample log file (Continued)

Fatal/Non-Fatal UseMARCON Errors

UseMARCON divides errors into fatal and non-fatal categories. When errors occur, it processes the records and writes the errors to a log.

Fatal Errors

Fatal errors occur when something prevents UseMARCON from converting a record. Typically, this occurs when UseMARCON cannot find one of the six required files for translation. For more information on the required files, see [Sample of the required files in the UseMARCON initialization file](#) on [page A-5](#).

In the event of a fatal error, UseMARCON writes an error message to the log file for that session for the function being performed during that session stating that UseMARCON could not complete the translation.

The error warning for fatal errors when converting MARC files with UseMARCON, is `END non OK`. For an example of a UseMARCON fatal error message, see the [Sample Fatal Error \(END non OK\) Message in UseMARCON Log File](#) on [page A-8](#).

Non-Fatal Errors

Non-Fatal errors occur when UseMARCON encounters a problem that does not prevent it from converting the record that causes the error. An example of this might be a redundant MARC field in a record.

After the record is converted, an error message is created in the log file for that session. For more information on the UseMARCON log file, see [UseMARCON Log File](#) on [page A-5](#).

The two error classifications of Non-fatal errors as they display in the UseMARCON logs are `ERROR` and `WARNING`. For the purposes of Voyager's interaction with UseMARCON, the `ERROR` and `WARNING` messages mean the same thing.

Sample Fatal Error (END non OK) Message in UseMARCON Log File

Figure A-5 is an example of a fatal error message in a UseMARCON log file.

```
-----  
Started at : Thu Aug 16 11:03:09 2001  
  
END non OK : No rulefile specified
```

Figure A-5. Sample Fatal Error Message in a UseMARCON Log File

Troubleshooting UseMARCON Error Conditions

UseMARCON generates an error type, an error number, a brief description of the error, and the field in which the error occurred for each error, as a line in the UseMARCON log for each session. This information is what enables troubleshooting of UseMARCON conversion errors.

Use the UseMARCON error messages to determine what changes to make to the conversion files used for that session.

Patterns of recurring errors in the UseMARCON logs (for example, recurring Mandatory field expected in input record messages in the same field) may indicate necessary adjustment of the UseMARCON files (making a field that is considered required by the format checking file a non-required field), or discrepancies/inconsistencies in the records of the database (the field set as required in the format checking file does not exist in many of the records, or is in the wrong field).

UseMARCON Log File Example

For example, the log file in [Figure A-4](#) on [page A-6](#) contains the following error message:

```
WARNING(2105)-Unexpected field found in input record : Notice '13819' : field '092'
```

Figure A-6. Sample UseMARCON Log File Error Message Line

Table A-2 contains a description of the format of each error message line (the numbers in the left column correspond to the numbered sections of the sample error message in Figure A-6):

Table A-2. Descriptions of Error Message Lines in UseMARCON Log File

#	Name	Format	Description
1	Error type	One of three values: • ERROR • WARNING • End non OK	Severity of error: • ERROR and WARNING are both non-fatal. • END non OK indicates a fatal error, preventing the conversion of records.

Table A-2. Descriptions of Error Message Lines in UseMARCON Log File

#	Name	Format	Description
2	Error number	(nnnn)	UseMARCON error code number. For error codes, see UseMARCON Error Numbers on page A-16 .
3	Error description	Text	Brief description of error.
4	UseMARCON record ID number	Notice nnnn	The nnnn variable is the record ID number (which UseMARCON generates for each file) of the record which caused the error.
5	Error field	Field 'nnn'(subfield '\$x')	The nnn indicates the field which caused the error. The x in the subfield error field indicator specifies the subfield which caused the error (if applicable).

NOTE:

Fatal errors do not include all of the record specific error information, because the error prevents it from converting any records (for a sample fatal error message in a UseMARCON log file, see [Sample Fatal Error \(END non OK\) Message in UseMARCON Log File](#) on [page A-8](#)).

For more information about the error logs generated for the individual Voyager processes (MARC Export, Bulk Import, Z39.50), see the individual Error Reporting discussions for each Voyager process.

Bulk Import and UseMARCON

This section discusses how UseMARCON and the bulk import program interact.

Overview

UseMARCON interacts with Bulk Import by translating records specified in the **bulkimport** command into MARC21 format before importing the records into your Voyager database. Using the command line option **-y** allows you to specify the location of a UseMARCON initialization file which points Bulk Import to the translation files it needs to translate the records into MARC21 format before it runs Bulk Import as usual.

To convert a batch of non-MARC21 format records to MARC21 format when using Bulk Import, add the **-y** switch to the **pbulkimport** command, followed by the path and filename of the UseMARCON initialization file you wish to use for the conversion. The initialization file points the Bulk Import command to the directory and files needed by UseMARCON to perform the conversion.

Bulk Import Example

The following example shows a sample Bulk Import command line with a UseMARCON activation switch.

```
pbulkimport -f/m1/xxxdb/sbin/marcfile.txt -imarcive -y/m1/xxxdb/ini/usemarcon/UNI2US/uni2us.ini
```



IMPORTANT:

UseMARCON translations are for Bibliographic records and Authority records only. See [Bulk Import, Replace, and Merge of MARC Records on page 13-1](#) for more information.

Bulk Import Error Reporting

When UseMARCON encounters errors during Bulk Import batch jobs, it generates error messages in the log file for that session.

Two possible error messages can occur in the Bulk Import log for each session.

- For *each* record not translated, a `record not translated` message is sent to the Bulk Import log. Check the UseMARCON log file for the session to determine whether the error causing the `record not translated` message prevented translation completely or just affected some records.
- If the UseMARCON initialization file is missing, in the wrong place, or not configured properly, an `ini path not valid` message is sent to the log file. This message indicates a fatal error (which prevents translation), since UseMARCON cannot convert records without a valid initialization file.

MARC Export and UseMARCON

This section discusses how UseMARCON and the MARC export program interact.

Overview

UseMARCON interacts with MARC Export by using translation rules to convert the exported MARC records from MARC21 to the specified MARC format.

Using the command line option `-y` allows you to specify the location of a UseMARCON initialization file which points to the translation files needed to translate the records into the specified MARC format.

To use UseMARCON to convert a batch of MARC21 format records to another MARC format when using MARC Export, add the `-y` switch to the `pmarcexport` command, followed by the path and filename of the UseMARCON initialization file you wish to use for the conversion. The initialization file directs MARC Export to the directory and files needed by UseMARCON to perform the conversion. The following example shows a sample MARC Export command line with a UseMARCON activation switch.

MARC Export Example

```
pmarcexport -rA -mU -t1997-10-25:1998-10-27 -y/m1/xxxdb/ini/
usemarcon/US2UNI/us2uni.ini
```



IMPORTANT:

UseMARCON translations only work on these record types:

- B (Bibliographic records)
- A (Authority records)

Therefore the required `-r` switch in the MARC Export command line can only be followed by the preceding record types when using UseMARCON for translations with MARC export. For more information on the MARC Export record type switch, see [Bulk Export of MARC Records](#) on [page 11-1](#).

MARC Export Error Reporting

When UseMARCON encounters errors during MARC Export batch jobs, it generates error messages to the log file for that session.

Two possible error messages can occur in the MARC Export log for each session.

- For *each* record not translated, a `record not translated` message is sent to the log. Check the log file for the session to determine whether the error causing the `record not translated` message prevented translation completely or just affected some records.
- If the UseMARCON initialization file is missing, in the wrong place, or not configured properly, an `ini path not valid` message is sent to the log file. This message indicates a fatal error (which prevents translation), since UseMARCON cannot convert records without a valid initialization file.

Z39.50 Server and UseMARCON Interaction

This section discusses how UseMARCON and the Z39.50 Server interact

Overview

UseMARCON interacts with Z39.50 servers enabling remote Z39.50 clients which only request MARC records in formats other than MARC21, to search your Voyager databases.

UseMARCON converts the records to the MARC format requested by the Z39.50 client before sending those records.

When a Z39.50 client sends a request for non-MARC21 records, your Voyager Z39.50 server locates the MARC conversion profile (specified in the `z3950svr.ini` file) corresponding to the requested format (if one exists). This conversion profile points to the UseMARCON initialization file containing the required conversion files. UseMARCON then converts the requested records to the specified format and then sends them to the Z39.50 client in the requested format.

The `z3950svr.ini` File

To use UseMARCON with your Z39.50 server(s), you must first specify the UseMARCON initialization file paths for each conversion in the `z3950svr.ini` file.



Procedure A-1. Setting Up Initialization Paths in z3950svr.ini File

Use the following to specify the initialization paths.

1. Add a stanza to the `z3950svr.ini` file entitled `[usemarcon translations]` (if one does not already exist).
2. Add a line for each MARC format supported by your library in the `[usemarcon translations]` stanza in the `z3950svr.ini` file, using the following format:

`Marcformat=/path/usemarconinifile.name`

3. Enter the `Marcformat` portions of the lines as they are formatted in Table A-3 (the table includes the Z39.50 format codes of the formats in case you need them).
-

Table A-3. MARC Formats and Corresponding Z39.50 Format Codes

MARC Format	Z39.50 Format Code
Unimarc	1.2.840.10003.5.1
Intermarc	1.2.840.10003.5.2
Ukmarc	1.2.840.10003.5.11
Normarc	1.2.840.10003.5.12
Librismarc	1.2.840.10003.5.13
Danmarc	1.2.840.10003.5.14
Finmarc	1.2.840.10003.5.15
Canmarc	1.2.840.10003.5.17
Picamarc	1.2.840.10003.5.19
AMARC21	1.2.840.10003.5.20
Ibermarc	1.2.840.10003.5.21

4. Set the path to the initialization file that converts MARC21 to the specified non-MARC21 format (for Z39.50 server processes, the records are converted from your Voyager, MARC21 records to the format specified by the Z39.50 client).

Do this for *each* non-MARC21 format supported by your library.

**IMPORTANT:**

Only one non-MARC21 format can be requested by the Z39.50 client per session. If you request another MARC format by way of the Z39.50 client in a session after choosing the first one, the server and client receive an error message.

For a sample [usemarcon translations] stanza in the z3950svr.ini, see [A-15](#).

Sample [usemarcon translations] Stanza in z3950svr.ini File (Lines 1-3)

Figure A-7 shows a sample [usemarcon translations] stanza in the z3950svr.ini file containing UseMARCON initialization file paths for two MARC formats.

(Line #)

1	# Translations
2	Unimarc=/m1/voyager/xxxdb/ini/usemarcon/US2UNI/ us2.ini
3	Finmarc=/m1/voyager/xxxdb/ini/usemarcon/US2FIN/ us2.ini
4	
5	# STOP!!
6	#
7	# That should be all you have to edit in this file. Make sure you have your
8	# sapi.ini file configured properly.

Figure A-7. Sample [usemarcon translations] Stanza

UseMARCON/Z39.50 Error Reporting

When UseMARCON encounters errors during Z39.50 server sessions, it generates error messages specific to UseMARCON translations in the UseMARCON log file (default= usemarcon.z39.date.time) for that session. It also sends error messages, detailing how the UseMARCON translation error affected the Z39.50 process, to the normal Z39.50 log (default= z3950svr.log).

The possible error messages in the Z39.50 log for each session are:

- For *each* record not translated, a *record not translated message* gets sent to the Z39.50 log. Check the UseMARCON log file for the session to determine whether the error causing the *record not translated message* prevented translation completely or just affected some records.
- If the UseMARCON initialization file is missing, in the wrong place, or not configured properly, an *ini path not valid message* is sent to the Z39.50 log file.
- If the client requests a MARC format which has no entry, a *No USEMARCON ini entry for this OID: oid error message* is generated.
- If the client requests more than one non-MARC21 format in the same session, an *Only one type of USEMARCON conversion allowed per session message* is generated.

A fatal error (an error which prevents UseMARCON from converting the record) sends a message stating that there was a translation error, to the UseMARCON log, *and* an error message stating that the record was not translated by UseMARCON to the normal Z39.50 error log (*z3950svr.log*, by default).



IMPORTANT:

In the case of a fatal error, the Z39.50 client displays a general “Failed to retrieve record” message. In the case of non-fatal errors, the client receives no messages detailing the errors in the conversion. Because of this, troubleshooting errors in the initial phases of setting up UseMARCON for use with Z39.50 servers is very important.

UseMARCON Error Numbers

The following sections detail the UseMARCON error messages.

General Error Numbering

Errors are numbered 000-9999. The first digit describes the part of the process in which the error appears. Table A-4 shows the general error categories and corresponding error number ranges.

Table A-4. General UseMARCON Error Message Guidelines

Error #	Error
0xxx	The error is due to improper interface use.

Table A-4. General UseMARCON Error Message Guidelines

Error #	Error
1xxx	The error appears during reading of a MARC file, that is, a format problem.
2xxx	The error occurs during MARC checking in input (in comparison with the input checking table).
3xxx	The error occurs during character translation.
4xxx	The error occurs during coded data translation.
5xxx	The error occurs during rule analysis or conversion.
7xxx	The error occurs during checking of the MARC output.
8xxx	The error occurs during writing of the MARC file.
9xxx	Other internal errors.

Error Message Numbers

A complete list of error messages used by the USEMARCON software is shown in Table A-5.

Table A-5. UseMARCON Error Messages (Page 1 of 7)

Error #	Error
501	Invalid rule pattern to search
502	No more patterns found
503	Maximum errors to be encountered limit has been reached
504	The last record to be converted has been reached
505	Type or browse a MARC file to open before
506	Pattern not found
507	Unable to evaluate an empty rule
508	No defined box to search. Please point to the box to search in
509	No search is available on this selection. Please select another one
510	Comment is too long and will be truncated

Table A-5. UseMARCON Error Messages (Page 2 of 7)

Error #	Error
511	Line is too long and will be truncated
512	Unknown input format comment
513	Unknown output format comment
514	Please select RI boundaries
515	Please select No boundaries
516	Invalid RI boundaries
517	Invalid No boundaries
590	Specified file does not exist (please check path)
1001	Unable to reset a writing mode opened file
1002	Invalid SCW encountered when attempting to read a MARC notice
1003	Unable to go further in MARC file reading
1004	Invalid length encountered when attempting to read a MARC notice
1005	Unable to go further in MARC file writing
1006	Unable to flush the MARC output file
1007	Invalid MARC data location address
1009	Unable to set the content of the field
1101	Invalid MARC tag
1102	Invalid MARC indicators
1202	Unable to set the label
1501	MARC buffer allocation failure
1502	Error encountered when attempting to read the MARC file
2001	Invalid input format checking rule (item expected)
2002	Invalid or absent tag in input format checking rule
2003	Format checking rule redundancy in input format checking file
2004	Invalid or absent first indicators list in input format checking rule
2005	Invalid or absent second indicators list in input format checking rule
2006	Invalid or absent subfield in input format checking rule
2101	Not repeatable but redundant field found in input record
2102	Invalid first indicator found in input record

Table A-5. UseMARCON Error Messages (Page 3 of 7)

Error #	Error
2103	Invalid second indicator found in input record
2104	Invalid or redundant subfield found in input record
2105	Unexpected field found in input record
2106	Mandatory field expected in input record
2107	Mandatory subfield expected in input record
2108	Field without any subfield found in input record
2501	TControlField allocation failure when attempting to load a new input format checking rule
2502	TCtrlSubfield allocation failure when attempting to load a new input format checking rule
2503	First indicators list allocation failure when attempting to load a new input format checking rule
2504	Second indicators list allocation failure when attempting to load a new input format checking rule
3000	Memory allocation error
3001	Character not transcoded (unable to find it in transco table)
4001	Coded data not loaded
5000	Memory allocation error
5001	The selected rule file does not exist
5002	Unable to load the invalid rule
5003	Unable to find the label in any CD
5004	Only one indicator has been found
5005	Content of indicator is too long (>1)
5100	Rule analysis error
5101	A CDOut like TTT(no)... has an invalid subfield occurrence number (no, nso or nto)
5102	A CDOut like ...SS(no) has an invalid field occurrence number (no, nso or nto)
5103	A CDOut like TTT(nton)... has an invalid subfield occurrence number (no, nso or nto)
5104	A CDOut like ...SS(nso) has an invalid field occurrence number (no,nso or nto)
5200	Expected CD tag

Table A-5. UseMARCON Error Messages (Page 4 of 7)

Error #	Error
5201	Invalid CD tag (three characters are required)
5202	Invalid CD tag (only numerics or letters are allowed)
5203	Invalid CD subfield (only two characters required)
5204	Invalid CD subfield (only '1' or '2' is expected behind a 'I' subfield)
5205	Invalid CD subfield (only numerics or letters are allowed behind a '\$' subfield)
5206	Invalid CD subfield (only I1, I2 or \$? are allowed)
5207	Invalid position settings
5208	Invalid CD tag occurrence number
5209	Invalid CD subfield occurrence number
5210	Misplaced rule : please insert this rule before the previous one
5211	Misplaced rule : please insert this rule after the next one
5212	Invalid character found in rule
5301	Invalid output occurrence number 'no'
5302	Invalid output tag occurrence number 'nto'
5303	Invalid output sub occurrence number 'nso'
5304	Invalid input occurrence number 'n'
5305	Invalid input tag occurrence number 'nt'
5306	Invalid input sub occurrence number 'ns'
5307	Unknown main input CD (please type or load it before evaluating)
5308	Unknown old output CD (please type it before evaluating)
5309	Unknown other input CD (please type it before evaluating)
5501	TRule allocation failure when attempting to analyse the rule
5502	Unable to allocate space for setting text of analysed rule
5503	Unable to allocate space for setting comment of analysed rule
5504	TCD allocation failure when attempting to analyse the rule
5505	TCDLib allocation failure when attempting to deal with other input CDs
5506	Buffer allocation failure when attempting to split the rule
5507	TCD allocation failure when attempting to load CD from MARC record
7001	Invalid output format checking rule (item expected)

Table A-5. UseMARCON Error Messages (Page 5 of 7)

Error #	Error
7002	Invalid or absent tag in output format checking rule
7003	Format checking rule redundancy in output format checking file
7004	Invalid or absent first indicators list in output format checking rule
7005	Invalid or absent second indicators list in output format checking rule
7006	Invalid or absent subfield in output format checking rule
7101	Redundant field (not repeatable) found in output record
7102	Invalid first indicator found in output record
7103	Invalid second indicator found in output record
7104	Invalid or redundant subfield found in output record
7105	Unexpected field found in output record
7106	Mandatory field expected in output record
7107	Mandatory subfield expected in output record
7108	Field without any subfield found in output record
7501	TControlField allocation failure when attempting to load a new output format checking rule
7502	TCtrlSubfield allocation failure when attempting to load a new output format checking rule
7503	First indicators list allocation failure when attempting to load a new output format checking rule
504	Second indicators list allocation failure when attempting to load a new output format checking rule
8001	Unable to delete the Error Log File
9001	TRuleFile allocation failure when attempting to load the Rule File
9011	TCheckFile allocation failure when attempting to load the Input Check File
9012	TCheckFile allocation failure when attempting to load the Output Check File
9013	TTransFile allocation failure when attempting to load the Translation Character Table
9021	TMARCFFile allocation failure when attempting to open the Input MARC File
9022	TMARCFFile allocation failure when attempting to open the Output MARC File
9031	TMARCRecord allocation failure when attempting to load the Input MARC File

Table A-5. UseMARCON Error Messages (Page 6 of 7)

Error #	Error
9032	TMARCRecord allocation failure when attempting to load the Output MARC File
9041	TMARCField allocation failure when attempting to load the notice into memory fields
9101	TRuleDoc not created
9102	TMARCDoc not created
9103	TDummyDoc not created
9104	TTransDoc not created
9105	TCheckDoc not created
9201	TCD allocation failure when attempting to search for another CD
9202	Label is mandatory and has not been converted
9203	TMARCField allocation failure when attempting to merge CDs into fields
9301	TCDLib allocation failure when attempting to split a field into CDs
9401	Find SLIST allocation failure when attempting to memorize precedent find/replace request
9402	Replace SLIST allocation failure when attempting to memorize precedent find/replace request
9403	TRule allocation failure when attempting to search/replace items
9404	TCD allocation failure when attempting to search/replace items
9501	Unable to open the ASCII mode file
9502	Unable to open the binary mode file
9503	Unable to delete the file
9504	Unable to get the next line of a binary file
9505	Invalid #include 'file' directive found in file
9506	Unable to read two first lines of an ASCII file
9601	Unable to open the MARC Input Window
9602	Unable to open the MARC Output Window
9603	Unable to open the Rule Edit Window
9604	Unable to open the Rule Eval. Window
9700	Two identical CDs found in a record !
9701	Unable to save the MARC edit configuration file

Table A-5. UseMARCON Error Messages (Page 7 of 7)

Error #	Error
9703	Invalid tag field to add to the list of tags without indicator
9704	Unable to add the selected tag to the list of tags without indicator
9705	Unable to remove the selected tag from the list of tags without indicator
9706	(No) is not filled
9800	Unable to open the help file usemarco.csc/hlp
9999	Unknown error

WebVoyage Patron Authentication Adapter Feature



Overview

The WebVoyage Patron Authentication Adapter feature makes WebVoyage compatible with an external patron authentication program. This feature enables libraries to develop customized security solutions for WebVoyage. To allow WebVoyage to communicate with the external patron authentication programs, libraries must develop a patron authentication adapter.

Once the adapter is created, you can use the external authentication system in lieu of the normal WebVoyage patron login page or give patrons the option of using either login page.

This document details the process that occurs between WebVoyage, Voyager, and the patron authentication adapter. It also includes the relevant WebVoyage configuration and behavior. Use this information to develop your patron authentication adapter suited to your third party authentication program.

The Voyager patron login functionality that accommodates the external authentication is designed to be used with any external authentication program. No particular type of authentication system is assumed, and the system is only discussed in terms of its interaction with Voyager and WebVoyage.

Some benefits of the WebVoyage interaction design are as follows:

- You can create the adapter in any programming language.
- For added security, patron information is not communicated through the network.

- If you have a pre-established patron ID value in your external database and are not using Institution IDs, you can populate the Institution ID fields of your Voyager Patron records with your external database patron ID's.
- WebVoyáge gives control of the browser to the authentication adapter/system so it may collect patron credentials as needed.

NOTE:

Throughout this document, the term “WebVoyáge Patron Authentication Adapter feature” refers to the WebVoyáge functionality that allows WebVoyáge to communicate with an external authentication program, via a customer-developed authentication adapter. The term “patron authentication adapter” is used to refer to the customer-developed adapter which provides the communication bridge between WebVoyáge and the external authentication program.



Procedure B-1. Patron Authentication Workflow Overview

The following steps outline the patron authentication process between WebVoyáge and the patron authentication adapter, at a very general level. For a more detailed description of the interaction between WebVoyáge and the patron adapter, see [Detailed WebVoyáge Patron Adapter Interaction](#) on [page B-3](#).

[Figure B-1](#) illustrates the interaction between WebVoyáge and the patron authentication adapter, at an overview level.

When a patron performs a WebVoyáge function requiring login:

1. WebVoyáge is redirected to the external adapter.
2. The patron is authenticated using the third party authentication system.
3. The patron authentication adapter inserts information into the Voyager database.
4. The patron authentication adapter sends a URL to WebVoyáge indicating the authentication status.

If the adapter returns a positive authentication status, WebVoyáge uses the information placed into the Voyager database by the adapter, to locate the corresponding Voyager patron.

5. (Not shown in diagram) WebVoyáge logs in that patron.

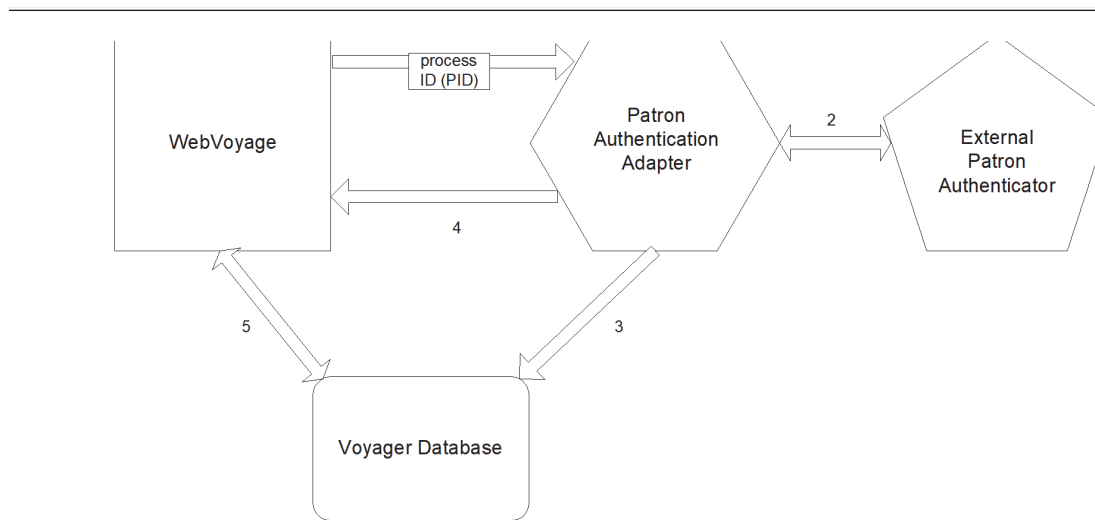


Figure B-1. Patron Authentication Adapter Flow Diagram

NOTE:

Voyager patron records must exist in the Voyager database before they can be retrieved using an external authentication program in conjunction with an adapter.

Patron records in the Voyager database must be uniquely identified by information in the Institution ID field. The information in the Institution ID field in the Voyager patron records in the Voyager database must also exist in the external patron authenticator (or database) and the adapter must be able to match this information in both patron record sources (the voyager patron records and external records).

Detailed WebVoyage Patron Adapter Interaction

This section includes a more detailed description of the WebVoyage/adapter interaction process (once the adapter behavior is enabled in WebVoyage, and the adapter is created).

1. A WebVoyage user performs a WebVoyage function requiring log in.
2. The login page displays.

If	Then
Bypassing the Patron Login page	- WebVoyage sends a form to the URL specified in <code>option.extAuthSystemURL=</code> including the process ID (PID) of the WebVoyage session. The form information is submitted as a GET. - The external authentication page opens.
If	Then
Accessing the external authentication system via a button on the WebVoyage Patron Login page	- The WebVoyage Patron Login page opens, with a redirect button to the external authentication program. - When you click the button, WebVoyage sends a form to the URL specified in the <code>option.extAuthSystemURL=</code> including the process ID (PID) of the WebVoyage session.
3. WebVoyage gives control of the browser to the adapter for purposes of patron authentication.	
4. The patron is authenticated externally.	
If...	Then...
authentication fails	- The adapter includes a failure status in the URL redirect to WebVoyage in the following format: <pre>http://<host>:<port>/vwebv/externalLogin.do?<redirect string>&authenticate=N</pre> For more on this URL format, see Format of the Redirect URL from the Authentication Adapter to WebVoyage on page B-8. - WebVoyage provides the patron with another opportunity to log in, using whichever login page was originally provided (WebVoyage with the redirect button, or the external authentication page itself). - Return to step 2 in this process until successfully authenticated.

If...	Then...
authentication succeeds	continue to step 5
5. The adapter inserts a record into the WOPAC_PID_PATRON_KEYS table on the Voyager server that contains 2 fields:	• PID: Contains the encrypted WebVoyage process ID obtained from the WebVoyage redirect URL (step 2 in this process) . When storing the redirect URL into the PID field of the WOPAC_PID_PATRON_KEYS table, the encryption and escape codes must be retained. • PATRON_KEY: Contains a value unique to the patron, obtained from the external authentication. This value must correspond to the value in the Voyager patron's Institution ID information field, in the Voyager patron record (though the actual string in the Institution ID field need not actually be an institution ID per se).
6. The adapter sends a URL back to the WebVoyage session which does the following:	- Indicates to WebVoyage an authentication success status. - Returns control of the browser to WebVoyage. This URL takes the following form: <pre>http://<host>:<port>/vwebv/externalLogin.do?<redirect string>&authenticate=Y</pre>
7. WebVoyage looks in the PATRON_KEY field of the Voyager database for the unique patron value added by the adapter.	
8. WebVoyage looks for a Voyager patron containing the value of the PATRON_KEY (from step 5) in its Institution ID field in the Voyager patron record.	

If...	Then...
WebVoyáge does not find a patron containing the PATRON_KEY information in his or her voyager patron record	a. The adapter includes a failure status in the URL redirect to WebVoyáge in the following format: <pre>http://<host>:<port>/vwebv/externalLogin.do?<redirect string>&authenticate=N</pre> For more on this URL format, see Format of the Redirect URL from the Authentication Adapter to WebVoyáge on page B-8. b. WebVoyáge provides the patron with another opportunity to log in, using whichever login page was originally provided (WebVoyáge with the redirect button, or the external authentication page itself). c. Return to step 2 in this process until authentication success is attained and a patron match is found.

If...	Then...
WebVoyáge finds a patron containing the PATRON_KEY information in the voyager patron record	continue to step 9
9. WebVoyáge logs the corresponding patron in to the WebVoyáge session with the specified process ID.	

WebVoyáge Configuration

This section details the WebVoyáge configuration required to use the Patron Authentication Adapter feature.

The `webvoyage.properties` file controls the WebVoyáge behavior options for the Patron Authentication Adapter.

- [Enabling and Disabling Webvoyáge Adapter Behavior](#)
- [Enabling and Disabling the WebVoyáge Patron Login Bypass](#)

- [Setting the URL to the External Authentication System](#)
- [Setting the Text of the External Authentication Link Button](#) (used if *not* bypassing WebVoyage Patron Login page)

See [Figure B-2](#) for an example of the `webvoyage.properties` file settings.

```
#=====
# Should WebVoyage users be able to use an external authentication system when logging in?
# If Y, WebVoyage uses the external authentication system as configured below
# If N, WebVoyage displays the logon form as previously configured
#=====
option.extAuthSystemEnabled=Y
#=====
# URL to the external authentication system
#=====
option.extAuthSystemURL=
#=====
# Should WebVoyage bypass the logon form if using an external authentication system?
#=====
option.extAuthBypassLoginScreen=N
...
# use of the external authentication link is optional,
# this line will have no effect if option.extAuthSystemEnabled=N
#=====
page.logIn.extAuth.linkText=Go to External Patron Login System
```

Figure B-2. `webvoyage.properties` example

Enabling and Disabling Webvoyage Adapter Behavior

The `option.extAuthSystemEnabled=` variable enables or disables the WebVoyage patron authentication adapter behavior.

- To enable the WebVoyage patron authentication adapter behavior, set the `option.extAuthSystemEnabled=` variable to `Y`.
- To disable the behavior, set the variable to `N`.

If the variable is set to `N`, WebVoyage only displays the standard WebVoyage Patron Login page for login functions.

Setting the URL to the External Authentication System

Specify the URL to the external authentication program in the `option.extAuthSystemURL= variable`.

Enabling and Disabling the WebVoyage Patron Login Bypass

The `option.extAuthBypassLoginScreen= variable` determines whether WebVoyage displays the normal WebVoyage Patron Login page or bypasses it and goes straight to the external authentication page.

- To bypass the WebVoyage Patron Login and display the external authentication system when performing WebVoyage functions requiring login, set the `option.extAuthBypassLoginScreen= variable` to Y.
- To display the WebVoyage Patron Login page, along with a button connecting to the external authentication system, set the `option.extAuthBypassLoginScreen= variable` to N.

If you set this variable to N, configure the `page.logIn.linkText` variable (see [Setting the Text of the External Authentication Link Button](#)).

Setting the Text of the External Authentication Link Button

If you do not bypass the WebVoyage Patron Login page, a link displays on the login page that links to the external authentication system. The text of the link is set in the `page.logIn.extAuth.linkText` variable.

Format of the Redirect URL from the Authentication Adapter to WebVoyage

The format for the redirect URL that the Authentication Adapter must use to return control to WebVoyage combines fields in a conventional WebVoyage session URL with the authentication result field:

```
http://<host>:<port>/vwebv/externalLogin.do?<redirect
string>&authenticate=<status>
```

[Table B-1](#) details the URL field components.

Table B-1. URL field Components

Component	Description
<host>	IP or domain name of the host where WebVoyage is running.
<port>	TCP port of the WebVoyage process.
<redirect string>	A string, such as <code>searchBasic</code> , consisting of form fields that the adapter must use to redirect back to the instance of WebVoyage that invoked it.
<status>	Adapter must set this field to: • Y if patron authentication succeeded. • N if patron authentication failed or cancelled.

When control is returned to WebVoyage, it will check the authenticate field in the redirect URL to determine the patron status. If authentication was successful, i.e. the authenticate field was set to Y, WebVoyage send its process ID in a patron login request to the OPAC server. This causes the server to retrieve the unique patron key, corresponding to the encrypted WebVoyage process ID, from the WOPAC_PID_PATRON_KEYS table in the Voyager database.

Example

This section includes an example process, using a WebVoyage session which does not bypass the WebVoyage Patron Login page. The external authentication program in the example uses an LDAP database as its patron database, with the attribute of *empid* (employee ID) acting as the LDAP match-point to the Institution ID in the Voyager patron records in the Voyager database. The empid values from the LDAP directory have also been added to the Institution ID field of the corresponding patron records (to serve as match-points between the two databases).

1. A WebVoyage user performs a WebVoyage function requiring log in.
2. The login page displays including the redirect link.
3. Patron clicks the **Go to External Patron Login System** link.
4. The external authentication program displays a login page.

5. The adapter stores the encrypted process ID (PID) from the WebVoyage form submitted, retaining escape codes and encryption.
6. The user logs in to the external authentication program.
7. The external authentication program verifies the patron against the LDAP directory.

If authentication fails, the adapter returns the failure status in the URL `http://<host>:<port>/vwebv/externalLogin.do?<redirect string>&authenticate=N`. WebVoyage then returns to step [2](#).

8. If authentication succeeds, the patron authentication adapter retrieves the empid from the patron's information in the LDAP directory.
9. The adapter inserts the encrypted process ID (PID) stored in step [5](#) into the PID field of the WOPAC_PID_PATRON_KEYS table in the Voyager database.
10. The adapter inserts the empid value into the PATRON_KEY field in the WOPAC_PID_PATRON_KEYS table in the Voyager database.
11. The adapter sends a URL to the web server from which it received the request, in the following format

```
http://<host>:<port>/vwebv/externalLogin.do?<redirect
string>&authenticate=Y
```

12. When WebVoyage receives this URL, it retrieves the PATRON_KEY from the Voyager database for that PID.
13. WebVoyage searches for a voyager patron record that contains the PATRON_KEY value in its Institution ID field.

If WebVoyage does not find a corresponding patron record, it returns to step [2](#).

14. If it WebVoyage finds a corresponding patron record, it logs the Voyager patron into the WebVoyage session with the encrypted process ID in the PID field of the WOPAC_PID_PATRON_KEYS table.
-

Index

Numerics

- [008] stanza, [12-16](#)
- 035 field
 - stripping a non-standard 035 field, [12-10](#)
- 856 link failure report suffix segment
 - cataloging report, [25-8](#)

A

- About this document, [xxiii](#)
- Accept these records button, [31-14](#)
- Access database file
 - Reports.mdb, [4-48](#)
- accrued fines and demerits, [9-6](#)
 - running circulation batch job, [9-7](#)
- Acqbib.cfg file, [4-50](#)
- acqjob 5, [17-1](#)
- acquisitions
 - batch job
 - fix exchange rates, [17-1](#)
 - running batch job 5, [17-3](#)
- acquisitions notice
 - base segment, [23-4](#)
 - cancel serial claim notice suffix segment, [23-8](#)
 - cancellation notice suffix segment, [23-6](#)
 - order claim notice suffix segment, [23-7](#)
 - return notice suffix segment, [23-6](#)
 - serial claim notice suffix segment, [23-7](#)
 - standard interface file (SIF), [23-1](#)
 - suffix segment, [23-5](#)
 - voucher/check request suffix segment, [23-8](#)
- acquisitions report
 - base segment, [24-4](#)
 - fund snapshot after rollover report suffix segment, [24-13](#)
 - fund snapshot before rollover report suffix segment, [24-12](#)
 - fund snapshot report suffix segment, [24-11](#)
 - global open orders report suffix segment, [24-9](#)
 - open orders after rollover report suffix segment, [24-12](#)
 - open orders before rollover report suffix segment, [24-12](#)
 - open orders report suffix segment, [24-8](#)
 - purchase order report suffix segment, [24-5](#)
 - standard interface file (SIF), [24-1](#)
 - suffix segment, [24-4](#)
- Acquisitions Reports and Notices page, [31-10](#)
- acquisitions utilities
 - WebAdmin, [31-9](#)
- Add Bib/Item dialog box, [4-50](#)
- adding
 - clients after installation, [4-9](#)
- adding a new login/password
 - WebAdmin, [31-4](#)
- AIX server, [3-4](#)
 - starting
 - Apache, [3-10](#)
 - Oracle services, [3-6](#)
 - Voyager, [3-8](#)
 - stopping
 - Apache, [3-10](#)
 - Oracle services, [3-6](#)
 - Voyager, [3-8](#)
- archive and expire call slip requests, [9-4](#)
 - running circulation batch job, [9-4](#)
- archive short loans, [9-5](#)
 - running circulation batch job, [9-5](#)
- audience
 - of this document, [xxiii](#)
- audit file
 - bursar transfer program, [8-10](#)
 - marcexport program, [11-11](#)
 - patron extract program, [6-5](#)
 - patron update program, [7-7](#)
- Authblob_vw, [30-2](#)
- Authheading_vw, [30-3](#)
- Authhistory_vw, [30-3](#)
- authority data
 - loading, [2-2](#)
- Authority record has a status of D, S, or X written to reject.imp file message, [13-12](#)
- Authority records, [A-12](#)
- Authority1xx4xx_vw, [30-4](#)
- Authority5xx1xx_vw, [30-4](#)
- Authoritydupe_vw, [30-5](#)
- AutoUpdate
 - AcquisitionsStart.exe, [4-47](#)
 - CatalogStart.exe, [4-47](#)
 - characteristics, [4-45](#)
 - settings, [4-45](#)
 - transition, [4-2](#)
 - [Upgrade] stanza, [4-44](#)

B

backup

- cron, [3-11](#)
- determining device driver, [3-12](#)
- identifying data filesystem/volume, [3-15](#)
- restoring, [3-19](#)

barcode lookup, [18-1](#)

- configuration file, [18-2](#)
- schema file, [18-8](#)

base segment

- acquisitions notice, [23-4](#)
- acquisitions report, [24-4](#)
- cataloging report, [25-4](#)
- circulation notice, [26-4](#)
- circulation report, [27-4](#)
- media scheduling notice, [28-3](#)
- media scheduling report, [29-3](#)

batch forgive jobs

- audit report, [9-29](#), [9-32](#), [9-36](#)
- error logging, [9-28](#), [9-32](#), [9-35](#)
- error report, [9-29](#), [9-32](#), [9-35](#)

batch job

- acquisitions
 - fix exchange rates, [17-1](#)
- cataloging
 - process GHC queue-step 1, [14-4](#)
 - process GHC queue-step 2, [14-5](#)
 - process GHC queue-step 3, [14-6](#)
- circulation
 - accrued fines and demerits, [9-6](#)
 - archive and expire call slip requests, [9-4](#)
 - archive short loans, [9-5](#)
 - export OPAC requests, [9-4](#)
 - forgive demerits, [9-16](#)
 - forgive fines by create date, [9-30](#)
 - forgive fines by patron group and expire date, [9-33](#)
 - forgive fines by patron ID, [9-27](#)
 - patron purge, [9-23](#)
 - patron suspension, [9-7](#)
 - place items on active course reserve list, [9-10](#)
 - place recalls and holds for items on active course reserve list, [9-12](#)
 - purge UB patron stub record, [9-5](#), [9-8](#)
 - retain patron IDs, [9-17](#)
 - takes items on inactive course reserve lists off reserve, [9-14](#)
 - update remote circulation cluster cache, [9-9](#)
 - update shelving status, [9-3](#)
- media scheduling

- retain patron IDs, [10-1](#)

- running acquisitions batch job 5, [17-3](#)

Begin Record field, [31-14](#)

Bib Usage Log Export

- see OPAC Bib Usage Log Export program

Bib_usage_log table, [16-3](#)

Bib_vw, [30-5](#)

Bibblob_vw, [30-7](#)

Bibhistory_vw, [30-7](#)

bibliographic data

- loading, [2-2](#)

bibliographic records, [A-11](#), [A-12](#)

bibliographic usage logging

- configuring, [16-5](#)

Bibloc_vw, [30-8](#)

Bi-Weekly variable, [16-18](#)

Bmarcfix.cfg file, [4-49](#)

booking charge statistics report suffix segment

- media scheduling report, [29-5](#)

booking exceptions report suffix segment

- media scheduling report, [29-5](#)

booking statistics report suffix segment

- media scheduling report, [29-4](#)

bulk export of MARC records (marcexport) program, [11-2](#)

bulk import, [A-11](#)

UseMARCON, [A-10](#)

- batch jobs, [A-11](#)

- error reporting with, [A-11](#)

- sample script with UseMARCON switch, [A-11](#)

bulk import program, [13-1](#)

- call number hierarchy, [13-9](#)

- delete file, [13-10](#)

- discard file, [13-11](#)

- error and log file messages, [13-12](#)

- error file, [13-12](#)

- input file, [13-9](#)

- log file, [13-10](#)

message

- No item option in use - No item records created, [13-13](#)

- No Matches found for Input Cancel Record. Import Record Discarded!, [13-13](#)

- Non-Bibliographic Record detected! Holdings record written to error file, [13-14](#)

messages

- Authority record has a status of D, S, or X written to reject.imp file, [13-12](#)

- Could not delete Mfhd NNNNN. There are attachments., [13-12](#)

- DB Bib Record cancelled/deleted by Import Re-

- cord!, 13-12
- Delete Failed - Items Attached, 13-12
- Delete Failed - MFHDs Attached, 13-12
- Delete Failure, 13-12
- Duplicate detected. Discarding dupe import record!, 13-13
- Duplicate detection failure!, 13-13
- Duplicates above replace threshold. Adding anyway, 13-13
- Duplicates above thresholds. Cannot resolve!, 13-13
- Existing database record replaced by imported, 13-13
- Failed to load converter, 13-13
- Record does not match format for " change your import rule, 13-14
- Record nnn discarded -- no matching records in the database, 13-14
- Record Parse Failure, 13-14
- Record Retrieval Failure, 13-14
- Unparseable record written to error file, 13-14
- parameters
 - a MFHD location code, 13-5
 - b begin record, 13-5
 - d delete MFHDs, 13-5
 - e end record, 13-5
 - f filename, 13-4
 - h help, 13-7
 - i import code, 13-4
 - K ADDKEY, 13-3
 - k OK to export, 13-6
 - l location, 13-5
 - m load MFHDs, 13-5
 - o operator name, 13-4
 - x delete bibliographic records, 13-6
 - X NOKEY, 13-7
- reject file, 13-11
- replace file, 13-11
- running, 13-9
- WebAdmin, 31-12
- bulk importing
 - interleaved file generated by prebulk, 12-4
- bulkimport command, A-11
- bursar transfer program, 8-1
 - audit file, 8-10
 - bursar (SIF), 8-9
 - configuration file, 8-5
 - error file, 8-11
 - output file, 8-9
 - parameters
 - c configuration file, 8-7

- d database name, 8-7
- h help, 8-8
- i interval for transfer, 8-7
- m minimum fine/fee amount, 8-7
- o operator ID, 8-8
- p patron mode, 8-7
- q quiet mode, 8-8
- t test mode, 8-8
- u username and password, 8-7
- v version, 8-8
- refunding, 8-11
- running, 8-8
- stub patron records, 8-2
- transferring itemized patron fines/fees, 8-4
- transferring total patron balances, 8-4
- viewing information, 8-11
- WebAdmin, 31-30
- button
 - Accept these records, 31-14
 - Post, 8-12
 - Preview, 14-5
 - Send Record To, 4-29
 - Submit Batch Import, 31-16
 - Submit Bulk Export, 31-22
 - Submit Bursar Transfer, 31-33
 - Submit OPAC Job, 31-38

C

- call number hierarchy
 - bulk import program, 13-9
- Callslip.ini file, 4-50
- [CALLTYPES] stanza, 12-12
- cancel serial claim notice suffix segment
 - acquisitions notice, 23-8
- cancellation notice suffix segment
 - acquisitions notice, 23-6
 - circulation notice, 26-6
- cataloging batch job
 - process GHC fields-step 3, 14-6
 - process GHC queue-step 1, 14-4
 - process GHC queue-step 2, 14-5
- cataloging report
 - 856 link failure report suffix segment, 25-8
 - base segment, 25-4
 - duplicate authority records suffix segment, 25-6
 - see references with linked bib records suffix segment, 25-7
 - see refs authorized in other auth record suffix segment, 25-7

see refs without corresponding auth record suffix segment, 25-8

standard interface file (SIF), 25-1

suffix segment, 25-4

unauthorized name headings suffix segment, 25-5

unauthorized name/title headings suffix segment, 25-6

unauthorized subdivision headings suffix segment, 25-6

unauthorized subject headings suffix segment, 25-5

unauthorized title headings suffix segment, 25-5

cataloging utilities

- WebAdmin, 31-12

catjob 11

- cataloging batch job
- process GHC queue-step 1, 14-4

catjob 12

- cataloging batch job
- process GHC queue-step 2, 14-5

catjob 13

- cataloging batch job
- process GHC queue-step 3, 14-6

catjob.log file, 14-2

changing location codes of item records or MFHDs, 15-2

checking records with prebulk, 12-3

[Circ Locations] stanza, 8-5

Circ.ini file, 4-49

Circbib.cfg file, 4-49

Circcharges_vw, 30-8

circjob 1, 9-3

circjob 26, 9-4

circjob 27, 9-5

circjob 29, 9-5, 9-8

circjob 30, 9-6

circjob 31, 9-7

circjob 34, 9-10, 9-12

- output file, 9-11
- parameters
 - e end date, 9-10, 9-12
 - s start date, 9-10
 - start date, 9-12

circjob 35

- output file, 9-13
- parameters
 - D due date, 9-13
 - e end date, 9-12
 - i operator id, 9-12
 - L location code, 9-12
 - P print location, 9-13
 - s start date, 9-12

circjob 36, 9-14

- output file, 9-15
- parameters
 - e end date, 9-15
 - s start date, 9-15

circjob 37, 9-16

- parameters
 - q number of demerits to forgive, 9-16

circjob 38, 9-17

- parameter
 - T number of IDs to retain, 9-19
- running
 - command line, 9-20
 - interactively, 9-21

circjob 39, 9-23

circjob 40, 9-27

circjob 41, 9-30

circjob 42, 9-33

circjob 43, 9-3

circjob 44, 9-3, 9-41

circjob 8, 9-4

circjob.log, 9-1, 9-3, 9-4, 9-5, 9-7, 9-8, 9-9, 9-28, 9-32, 9-35

Circrenewal_vw, 30-10

CircSC.ini file, 4-49

circulation

- batch job
 - accrued fines and demerits, 9-6
 - archive and expire call slip requests, 9-4
 - archive short loans, 9-5
 - export OPAC requests, 9-4
 - forgive demerits, 9-16
 - forgive fines by create date, 9-30
 - forgive fines by patron group and expire date, 9-33
 - forgive fines by patron ID, 9-27
 - patron suspension, 9-7
 - place items on active course reserve list, 9-10
 - place recalls and holds for items on active course reserve list, 9-12
 - purge UB patron stub record, 9-5, 9-8
 - retain patron IDs, 9-17
 - takes items on inactive course reserve lists off reserve, 9-14
 - update remote circulation cluster cache, 9-9
 - update shelving status, 9-3
- patron purge, 9-23

circulation data (transactions)

- loading, 2-3

- circulation item exceptions report suffix segment
 - circulation report, 27-7
- circulation notice
 - base segment, 26-4
 - cancellation notice suffix segment, 26-6
 - courtesy (due) notice suffix segment, 26-9
 - fine/fee notice suffix segment, 26-7
 - item available notice suffix segment, 26-6
 - overdue notice suffix segment, 26-6
 - recall notice suffix segment, 26-7
 - recall-overdue notice suffix segment, 26-7
 - standard interface file (SIF), 26-1
 - statement of fines and fees suffix segment, 26-8
 - suffix segment, 26-5
- circulation patron exceptions report suffix segment
 - circulation report, 27-7
- circulation report
 - base segment, 27-4
 - circulation item exceptions report suffix segment, 27-7
 - circulation patron exceptions report suffix segment, 27-7
 - circulation statistics report suffix segment, 27-6
 - circulation transaction exceptions report suffix segment, 27-7
 - distribution item order list report suffix segment, 27-8
 - global circulation statistics report suffix segment, 27-7, 27-8
 - hold shelf expired report suffix segment, 27-5
 - missing in transit report suffix segment, 27-6
 - reserved items active report suffix segment, 27-5
 - reserved items expired report suffix segment, 27-5
 - standard interface file (SIF), 27-1
 - suffix segment, 27-4
- circulation reports and notices
 - WebAdmin, 31-24
- circulation statistics report suffix segment
 - circulation report, 27-6
- circulation transaction exceptions report suffix segment
 - circulation report, 27-7
- client installation options, 4-2
- clients
 - uninstalling, 4-15
- command
 - bulkimport, A-11
 - MARC Export, A-12
 - nohup, 6-2, 7-2, 8-3, 11-2, 13-3
 - pbulkimport, 13-9, A-11
 - pbursar, 8-8
 - pmarcexport, 11-10, A-12
 - pprebulk, 12-20
 - pptrnextr, 6-4
 - pptrnupdt, 7-6
 - pstrgvfy, 15-10
 - pwebadmin, 31-3
- command file
 - pstrgvfy program, 15-8
- comments
 - about this document, xxvii
- configuration file
 - Acqbib.cfg, 4-50
 - Bmarcfix.cfg, 4-49
 - bursar transfer program, 8-5
 - Circbib.cfg, 4-49
 - Country.cfg, 4-49
 - EditBook.cfg, 4-49
 - EditSerial.cfg, 4-49
 - emailf.cfg, 16-13, 16-18
 - exceptdates.cfg, 9-7
 - Lang.cfg, 4-49
 - MARC21 Repertoire.cfg, 4-49
 - NewBook.cfg, 4-49
 - NewSerial.cfg, 4-49
 - prebulk program, 12-5
 - promote.cfg, 9-9
 - Special Characters.cfg, 4-49
 - Spinelabel.cfg, 4-50
- configuring
 - bibliographic usage logging, 16-5
 - OPAC search logging, 16-4
- conventions used
 - in this document, xxvi
- converting
 - data, 2-1
- Could not delete Mfhd NNNNN. There are attachments. message, 13-12
- Country.cfg file, 4-49
- course reserve list
 - circjob 35, 9-12
 - circjob 36, 9-14
 - circjob34, 9-10
- courtesy (due) notice suffix segment
 - circulation notice, 26-9
- Create one summary transfer record per patron radio button, 31-33
- Create transfer record for each fine/fee transferred radio button, 31-33
- CREATEMFHD variable, 12-8
- creating
 - holdings from interleaved file with prebulk, 12-5

interleaved bib-MFHD file with prebulk, 12-4
login/password for WebAdmin, 31-3

cron

- backup, 3-11
- batch jobs set up at installation, 3-19
- running OPAC Bib Usage Log Export program as, 16-23
- running OPAC Search Log Export program as, 16-8
- running Selective Dissemination of Information as, 16-14, 16-21

customizing

- search results page of Selective Dissemination of Information, 16-12, 16-18

D

Daily variable, 16-17

data

- converting, 2-1

data filesystem, 3-15

database view, 30-1

- Authblob_vw, 30-2
- Authheading_vw, 30-3
- Authhistory_vw, 30-3
- Authority1xx4xx_vw, 30-4
- Authority5xx1xx_vw, 30-4
- Authoritydupe_vw, 30-5
- Bib_vw, 30-5
- Bibblob_vw, 30-7
- Bibhistory_vw, 30-7
- Bibloc_vw, 30-8
- Circcharges_vw, 30-8
- Circrenewal_vw, 30-10
- Fundledger_vw, 30-12
- Heading_vw, 30-14
- Issues_vw, 30-15
- Item_vw, 30-16
- LCclass_vw, 30-18
- Marcbook_vw, 30-19
- Marccomputer_vw, 30-19
- Marcmaps_vw, 30-20
- Marcmusic_vw, 30-20
- Marcserial_vw, 30-21
- Marcvisual_vw, 30-22
- MFHDblob_vw, 30-22
- MFHDhistory_vw, 30-23
- Nlmclass_vw, 30-23
- Recordcount_vw, 30-24
- Serials_vw, 30-24

- Sudocclass_vw, 30-25
- Vendorinvoice_vw, 30-26
- Vendororder_vw, 30-28

date range, 16-7, 16-22

- format, 16-8, 16-22

Date Range radio button, 31-19

days after fine created, 31-33

days after fine created field, 31-33

DB Bib Record cancelled/deleted by Import Record! message, 13-12

DEFAULTCALLIND variable, 12-9

DEFAULTCALLNO variable, 12-8

Delete Failed - Items Attached message, 13-12

Delete Failed - MFHDs Attached message, 13-12

Delete Failure message, 13-12

delete file

- bulk import program, 13-10

delete.auth.imp.ccyymmdd.hhmm file, 13-11

delete.bib.imp.ccyymmdd.hhmm file, 13-11

deleted.mfhd.marc file, 13-11

deleting

- clients after installation, 4-9
- login/password for WebAdmin, 31-4

determining backup device driver, 3-12

dialog box

- Add Bib/Item, 4-50
- File Upload, 31-14
- Run, 4-10, 4-14
- Search, 4-31

discard file

- bulk import program, 13-11

discard.imp.ccyymmdd.hhmm file, 13-11

distribution item order list report suffix segment

- circulation report, 27-8

document summary, xxiv

duplicate authority records suffix segment

- cataloging report, 25-6

Duplicate detected. Discarding dupe import record! message, 13-13

Duplicate detection failure! message, 13-13

Duplicates above replace threshold. Adding anyway message, 13-13

Duplicates above thresholds. Cannot resolve! message, 13-13

E

EDI_msg.ini file, 4-50

EditBook.cfg file, 4-49

EditSerial.cfg file, 4-49

- E-Mail Address field, 31-10
- E-mail Address field, 31-16, 31-21, 31-23, 31-24, 31-33, 31-37
- [E-mail stanza, 4-27
- [Email] stanza, 16-12, 16-18
- emailf.cfg file, 16-13, 16-18
- END non OK message, A-8
- End Record field, 31-14
- err.burs.yymmdd.hhmm file, 8-11
- err.forgive.YYYMMDD.HHMM, 9-29, 9-32, 9-35
- err.imp.yyyymmdd.hhmm file, 13-12
- err.prebulk.YYYMMDD.HHMM file, 12-22
- err.pupd.YYYMMDD.HHMM file, 7-8
- err.pxtr.YYYMMDD.HHMM file, 6-6
- error
 - fatal, A-10, A-16
- error and log file messages
 - bulk import program, 13-12
- error classifications, A-8
- error description, A-10
- error field indicator, A-10
- error file
 - bulk import program, 13-12
 - bursar transfer program, 8-11
 - patron extract program, 6-6
 - patron update program, 7-8
 - prebulk program, 12-22
- error message
 - END non OK, A-8
 - WARNING, A-8
- ERROR messages, A-8
- Error numbers, A-10
- error type, A-9
- exceptdates.cfg file, 9-7
- executable file
 - VoyagerInstall.exe, 4-17
- Existing database record replaced by imported message, 13-13
- expire patron barcode, 9-3, 9-41
- export OPAC requests, 9-4
 - , 9-4
- external patron authentication, B-1

F

- Failed to load converter message, 13-13
- Failed to retrieve record message, A-16
- fatal error, A-10, A-16
- feedback, customer, xxvii
- field

- Begin Record, 31-14
- E-Mail Address, 31-10, 31-16
- E-mail Address, 31-21, 31-23, 31-24, 31-33, 31-37
- End Record, 31-14
- Holdings Location Code, 31-14
- Import Code, 31-14
- Local Filename, 31-14
- Location Code, 2-2, 31-14
- Operator Name, 31-14
- Password, 31-7
- Patron Group Code, 2-3
- Rule Code, 31-14
- System Control (003) Identifier, 31-19
- System Control (040\$d) Identifier, 31-19
- Transfer fines/fees greater than or equal to, 31-33
- User Name, 31-6
- field indicators
 - error, A-10
- file
 - Access database
 - Reports.mdb, 4-48
 - configuration
 - Acqbib.cfg, 4-50
 - Bmarcfix.cfg, 4-49
 - bursar transfer program, 8-5
 - Circbib.cfg, 4-49
 - Country.cfg, 4-49
 - EditBook.cfg, 4-49
 - EditSerial.cfg, 4-49
 - emailf.cfg, 16-13, 16-18
 - exceptdates.cfg, 9-7
 - Lang.cfg, 4-49
 - MARC21 Repertoire.cfg, 4-49
 - NewBook.cfg, 4-49
 - NewSerial.cfg, 4-49
 - prebulk program, 12-5
 - promote.cfg, 9-9
 - Special Characters.cfg, 4-49
 - Spinelabel.cfg, 4-50
 - delete.auth.imp.ccyymmdd.hhmm, 13-11
 - delete.bib.imp.ccyymmdd.hhmm, 13-11
 - deleted.mfhd.marc, 13-11
 - discard.imp.ccyymmdd.hhmm, 13-11
 - err.burs.yymmdd.hhmm, 8-11
 - err.imp.yyyymmdd.hhmm, 13-12
 - err.prebulk.YYYMMDD.HHMM, 12-22
 - err.pupd.YYYMMDD.HHMM, 7-8
 - err.pxtr.YYYMMDD.HHMM, 6-6
- executable

VoyagerInstall.exe, 4-17
 httpd2.conf, 31-5
 index.html, 31-7
 initialization
 Callslip.ini, 4-50
 Circ.ini, 4-49
 CircSC.ini, 4-49
 EDI_msg.ini, 4-50
 Limits.ini, 4-50
 Mediahelp.ini, 4-50
 opac.ini, 16-5
 sdiemail.ini, 16-10, 16-12, 16-15, 16-18
 Template.ini, 4-49
 UseMARCON, A-3
 Voyager.ini, 4-9, 4-17
 voyager.ini, 16-4
 Z39.50 server, A-13, A-15
 log, A-5
 catjob.log, 14-2
 circjob.log, 9-1
 mediajob.log, 10-3
 Z39.50, A-15
 log.bursar.yymmdd.hhmm, 8-10
 log.exp.yyyymmdd.hhmm, 11-11
 log.imp.yyyymmdd.hhmm, 13-10
 log.prebulk.YYYYMMDD.HHMM, 12-21
 log.pupd.YYYYMMDD.HHMM, 7-7
 log.pxtr.YYYYMMDD.HHMM, 6-5
 log.strgvfy.date.time, 15-9
 marc.exp.yymmdd.hhmm, 11-11
 out.prebulk.YYYYMMDD.HHMM, 12-21
 reject.imp.ccyymmdd.hhmm, 13-11
 replace.imp.ccyymmdd.hhmm, 13-11
 sif.burs.yymmdd.hhmm, 8-9
 standard interface
 acquisitions notice, 23-1
 acquisitions report, 24-1
 cataloging report, 25-1
 charge transaction record, 20-1
 circulation notice, 26-1
 circulation report, 27-1
 item delete, 21-1
 media scheduling notice, 28-1
 media scheduling report, 29-1
 patron record, 19-1
 vendor record, 22-1
 File Upload dialog box, 31-14
 files
 log
 UseMARCON Bulk Import log, A-5
 UseMARCON MARC Export log, A-5

UseMARCON Z39.50 log, A-5
 filesystem
 data, 3-15
 [FINE FEE TYPE] stanza, 8-5
 fine/fee notice suffix segment
 circulation notice, 26-7
 Fines/Fees History tab, 8-11
 fix exchange rates, 17-1
 forgive demerits, 9-16
 running circulation batch job, 9-17
 forgive fines
 by date created, 31-26
 by patron group and expire date, 31-27
 by Patron ID, 31-24
 create date, 9-30
 patron group and expire date, 9-33
 patron ID, 9-27
 FTP client installation software, 4-61
 fund snapshot after rollover report suffix segment
 acquisitions report, 24-13
 fund snapshot before rollover report suffix segment
 acquisitions report, 24-12
 fund snapshot report suffix segment
 acquisitions report, 24-11
 Fundledger_vw, 30-12

G

GetNewHits variable, 16-17
 Getting Started, 1-1
 prerequisite skills and knowledge, 1-1
 GHC program
 parameters
 -d database name, 14-2
 -h help, 14-4
 -j job number, 14-3
 -l list job options, 14-3
 -L location code, 14-3, 14-6
 -O operator, 14-6
 -o operator name, 14-3
 -u username and password, 14-3
 -v version, 14-3
 WebAdmin, 31-22
 global circulation statistics report suffix segment
 circulation report, 27-7, 27-8
 Global Headings Change (GHC) program, 14-1
 global open orders report suffix segment
 acquisitions report, 24-9
 [GlobalLog stanza, 4-22, 4-23

H

Heading_vw, 30-14
[HelpMenuLink] stanza, 4-48
History tab, 13-6, 15-1, 15-3, 15-7
hold shelf expired report suffix segment
 circulation report, 27-5
Holdings Location Code field, 31-14
httpd2.conf file, 31-5

I

ID number
 record, A-10
 UseMARCON record, A-10
ID Range radio button, 31-19
identifying data filesystem/volume, 3-15
Import Code field, 31-14
index.html file, 31-7
ini path not valid message, A-11, A-13, A-16
initialization file
 Callslip.ini, 4-50
 Circ.ini, 4-49
 CircSC.ini, 4-49
 EDI_msg.ini, 4-50
 Limits.ini, 4-50
 Mediahelp.ini, 4-50
 opac.ini, 16-5
 sdiemail.ini, 16-10, 16-12, 16-15, 16-18
 Template.ini, 4-49
 UseMARCON, A-3
 Voyager.ini, 4-9, 4-17
 voyager.ini, 16-4
 Z39.50 server, A-13, A-15
input file
 bulk import program, 13-9
 patron update program, 7-3
 prebulk program, 12-5
 pstrgvfy program, 15-4
installation
 modifying, 4-9
installation files
 VoyagerInstall.exe, 4-3, 4-4
 VoyagerInstall.msi, 4-3
installation method
 .bat file, 4-39
 ftp, 4-39
 http, 4-36
 traditional, 4-35

installing clients, 4-2
intended audience
 of this document, xxiii
Issues_vw, 30-15
item available notice suffix segment
 circulation notice, 26-6
Item_vw, 30-16

J

Job Scheduler page, 31-12
Job Scheduler screen, 31-14, 31-22, 31-23, 31-33,
 31-36, 31-38

L

Lang.cfg file, 4-49
LCclass_vw, 30-18
Limits.ini file, 4-50
Linux, 3-4
loading
 authority data, 2-2
 bibliographic data, 2-2
 circulation data (transactions), 2-3
 patron data, 2-3
 vendor data, 2-4
Local Filename field, 31-14
Location Code field, 2-2, 31-14
[LOCATIONS] stanza, 12-12
log (audit) file
 prebulk program, 12-21
Log Export
 see OPAC Log Export program
log file, A-5
 bulk import (UseMARCON), A-5
 bulk import program, 13-10
 cataloging, 14-2
 circulation, 9-1
 MARC export (UseMARCON), A-5
 media scheduling, 10-3
 Pfixexchangerates, 17-6
 pstrgvfy program, 15-9
 Z39.50, A-15
 Z39.50 (UseMARCON), A-5
log file utility
 WebAdmin, 31-38
log.bursar.yymmdd.hhmm file, 8-10
log.exp.yyyyymmdd.hhmm file, 11-11

log.forgive.YYYYMMDD.HHMM, 9-29, 9-32, 9-36
log.imp.yyyymmdd.hhmm file, 13-10
log.prebulk.YYYYMMDD.HHMM file, 12-21
log.pupd.YYYYMMDD.HHMM file, 7-7
log.pxtr.YYYYMMDD.HHMM file, 6-5
log.strgvfy.date.time file, 15-9
LOGBIBUSAGE= variable, 16-5
logging in
 using Single Client login, 4-23
LOGSEARCH= variable, 16-4

M

[MAPPING] stanza, 12-15
MARC Export, A-12-??
 sample script with UseMARCON switch, A-12
 UseMARCON error reporting with, A-13
MARC export command, A-12
MARC Export error reporting, A-13
MARC Export record type switch, A-12
MARC Formats
 supported by UseMARCON, A-2
[MARC POSTing] stanza, 4-27, 4-28
MARC records, A-12
marc.exp.yymmdd.hhmm file, 11-11
MARC21 format, A-1, A-11
MARC21 Repertoire.cfg file, 4-49
Marcbook_vw, 30-19
Marccomputer_vw, 30-19
marcexport program, 11-2
 audit file, 11-11
 output file, 11-11
 parameters
 -a assign character mapping, 11-8
 L Latin-1, 11-8
 M MARC21 MARC8, 11-8
 O OCLC, 11-8
 Voyager Legacy, 11-8
 -c system control identifier, 11-4
 -D Suppress Serials Checkin Data, 11-9
 -h help, 11-9
 -I ignore suppressed bibliographic records, 11-7
 -L library, 11-7
 -m export mode, 11-4
 B both create and update dates, 11-4
 C create dates only, 11-4

I ISBN input file, 11-4

K OK to export file, 11-4

M MARC ID input file, 11-4

R MARC ID number range, 11-4

S suppressed, 11-4

U update dates only, 11-4

-n create control number (001) from LCCN 9010a), 11-4

-O only OCLC records, 11-8

-o USMARC output filename, 11-3

-q quiet, 11-9

-r record type, 11-3

A authority records, 11-3

B bibliographic records, 11-3

E series authority records, 11-3

G Bib-MFHD record group, 11-3

H holdings records, 11-3

M main or added entry authority records, 11-3

S subject authority records, 11-3

-s modifying agency, 11-4

-t export target, 11-4

-v version information, 11-9

-w MFHD 852\$a updating, 11-8

-X exclude, 11-6

WebAdmin, 31-17

Marcmaps_vw, 30-20

Marcmusic_vw, 30-20

Marcserial_vw, 30-21

Marcvisual_vw, 30-22

Matches found for Input Cancel Record. Import Record Discarded! message, 13-13

media equipment inventory report suffix segment
 media scheduling report, 29-4

media scheduling

 batch job

 retain patron IDs, 10-1

Media Scheduling batch jobs, 10-4

media scheduling notice

 base segment, 28-3

 overdue notice suffix segment, 28-5

 standard interface file (SIF), 28-1

 suffix segment, 28-4

media scheduling report

 base segment, 29-3

 booking charge statistics report suffix segment, 29-5

 booking exceptions report suffix segment, 29-5

- booking statistics report suffix segment, [29-4](#)
- media equipment inventory report suffix segment, [29-4](#)
- standard interface file (SIF), [29-1](#)
- suffix segment, [29-3](#)
- media scheduling reports and notices
 - WebAdmin, [31-34](#)
- Mediahelp.ini file, [4-50](#)
- mediajob 5, [10-1](#)
 - parameter
 - n number of IDs to retain, [10-3](#)
 - running
 - command line, [10-4](#)
 - interactively, [10-5](#)
- mediajob.log file, [10-3](#)
- message
 - END non OK, [A-8](#)
 - Failed to retrieve record, [A-16](#)
 - ini path not valid, [A-11](#), [A-13](#), [A-16](#)
 - Matches found for Input Cancel Record. Import Record Discarded, [13-13](#)
 - No item option in use - No item records created, [13-13](#)
 - no USEMARCON ini entry for this OID, [A-16](#)
 - Non-Bib Record detected! Holdings record written to error file, [13-14](#)
 - oid error, [A-16](#)
 - only one type of USEMARCON conversion allowed per session, [A-16](#)
 - record not translated, [A-11](#), [A-13](#), [A-16](#)
 - WARNING, [A-8](#)
- Messages
 - record not translated, [A-11](#)
- MFHDblob_vw, [30-22](#)
- MFHDhistory_vw, [30-23](#)
- [MFHDTAG] stanza, [12-11](#)
- middle dot character, [11-9](#)
- missing in transit report suffix segment
 - circulation report, [27-6](#)
- modifying a login/password
 - WebAdmin, [31-4](#)
- modifying installation, [4-9](#)
- Monthly variable, [16-18](#)

N

- NewBook.cfg file, [4-49](#)
- NewHits variable, [16-17](#)
- NewSerial.cfg file, [4-49](#)
- NImclass_vw, [30-23](#)

- nnnn variable, [A-10](#)
- No item option in use - No item records created message, [13-13](#)
- No USEMARCON ini entry for this OID message, [A-16](#)
- nohup command, [6-2](#), [7-2](#), [8-3](#), [11-2](#), [13-3](#)
- noisewordfilter, [18-12](#)
- Non-Bibliographic Record detected! Holdings record written to error file message, [13-14](#)
- None variable, [16-17](#)
- Non-MARC21 format, [A-1](#), [A-11](#)
- Non-MARC21 records, [A-13](#)
- non-standard 035 field
 - stripping, [12-10](#)

O

- oid error message, [A-16](#)
- Only one type of USEMARCON conversion allowed per session message, [A-16](#)
- OPAC Bib Usage Log Export program, [16-1](#), [16-21](#), [31-36](#)
 - running as cron, [16-23](#)
 - running interactively, [16-21](#)
- OPAC Log Export program, [16-1](#), [31-36](#)
- OPAC reports
 - WebAdmin, [31-36](#)
- OPAC Search Log Export program, [16-6](#)
 - running as cron, [16-8](#)
 - running interactively, [16-7](#)
- OPAC search logging
 - configuring, [16-4](#)
- OPAC_Search_Log table, [16-2](#)
- opac.ini file, [16-5](#)
- open orders after rollover report suffix segment
 - acquisitions report, [24-12](#)
- open orders before rollover report suffix segment
 - acquisitions report, [24-12](#)
- open orders report suffix segment
 - acquisitions report, [24-8](#)
- Operator Name field, [31-14](#)
- Oracle stored functions, [32-1](#)
- order claim notice suffix segment
 - acquisitions notice, [23-7](#)
- out.prebulk.YYYYMMDD.HHMM file, [12-21](#)
- output file
 - bursar transfer program, [8-9](#)
 - circjob 34, [9-11](#)
 - circjob 35, [9-13](#)
 - circjob 36, [9-15](#)

marcexport program, 11-11
patron extract program, 6-5
prebulk program, 12-21
overdue notice suffix segment
 circulation notice, 26-6
 media scheduling notice, 28-5
[Overrides] stanza, 12-8

P

page
 Acquisitions Reports and Notices, 31-10
 Job Scheduler, 31-12
 Voyager Server Utilities (main), 31-7
 webadmin main page, 31-10
parameter
 circjob 38
 -T number of IDs to retain, 9-19
 mediajob 5
 -n number of IDs to retain, 10-3
parameters
 bulk import program
 -a MFHD location code, 13-5
 -b begin record, 13-5
 -d delete MFHDs, 13-5
 -e end record, 13-5
 -f filename, 13-4
 -h help, 13-7
 -i import code, 13-4
 -K ADDKEY, 13-3
 -k OK to export, 13-6
 -l location, 13-5
 -m load MFHDs, 13-5
 -o operator name, 13-4
 -x delete bibliographic records, 13-6
 -X NOKEY, 13-7
 bursar transfer program
 -c configuration file, 8-7
 -d database name, 8-7
 -h help, 8-8
 -i interval for transfer, 8-7
 -m minimum fine/fee amount, 8-7
 o operator ID, 8-8
 -p patron mode, 8-7
 -q quiet mode, 8-8
 -t test mode, 8-8
 -u username and password, 8-7
 -v version, 8-8
 circjob 34
 -e end date, 9-10, 9-12

 -s start date, 9-10, 9-12
 circjob 35
 -D due date, 9-13
 -e end date, 9-12
 -i operator id, 9-12
 -L location code, 9-12
 -P print location, 9-13
 -s start date, 9-12
 circjob 36
 -e end date, 9-15
 -s start date, 9-15
 circjob 37
 -q number of demerits to forgive, 9-16
GHC program
 -d database name, 14-2
 -h help, 14-4
 -j job number, 14-3
 -l list job options, 14-3
 -L location code, 14-3, 14-6
 -O operator, 14-6
 -o operator name, 14-3
 -u username and password, 14-3
 -v version, 14-3
marcexport program
 -a assign character mapping, 11-8
 L Latin-1, 11-8
 M MARC21 MARC8, 11-8
 O OCLC, 11-8
 Voyager Legacy, 11-8
 -c system control identifier, 11-4
 -D Suppress Serials Checkin Data, 11-9
 -h help, 11-9
 -I ignore suppressed bibliographic records, 11-7
 -L library, 11-7
 -m export mode, 11-4
 B both create and update dates, 11-4
 C create dates only, 11-4
 I ISBN input file, 11-4
 K OK to export file, 11-4
 M MARC ID input file, 11-4
 R MARC ID number range, 11-4
 S suppressed, 11-4
 U update dates only, 11-4
 -n create control number (001) from LCCN 9010a), 11-4
 -O only OCLC records, 11-8

- o USMARC output filename, 11-3
- q quiet, 11-9
- r record type, 11-3
 - A authority records, 11-3
 - B bibliographic records, 11-3
 - E series authority records, 11-3
 - G Bib-MFHD record group, 11-3
 - H holdings records, 11-3
 - M main or added entry authority records, 11-3
 - S subject authority records, 11-3
- s modifying agency, 11-4
- t export target, 11-4
- v version information, 11-9
- w MFHD 852\$a updating, 11-8
- X exclude, 11-6
- patron extract program
 - a audit filename, 6-3
 - d database name, 6-2
 - e error filename, 6-3
 - h help, 6-4
 - m monitor records to process, 6-4
 - p patron output filename, 6-3
 - r number of records to process, 6-4
 - u username and password, 6-3
 - y, 6-2
- patron update program
 - a audit filename, 7-4
 - d database name, 7-3
 - e error filename, 7-4
 - h help, 7-6
 - i index type, 7-3
 - D program selected, 7-4
 - I institution ID, 7-4
 - S ssan, 7-3
 - m monitor records to process, 7-6
 - n notes type, 7-5
 - G general, 7-5
 - P popup, 7-5
 - n number of records to process, 7-6
 - o overlay previous notes, 7-5
 - A add new notes, 7-5
 - I ignore notes processing, 7-5
 - R replace existing notes, 7-5
 - p patron input filename, 7-3
 - u username and password, 7-3
 - x extended notes processing, 7-5
- prebulk program
 - a append tag for log file, 12-20
 - c configuration file, 12-19
 - e error file, 12-19
 - f force overwrite of output files switch, 12-20
 - i input file, 12-19
 - l prebulk log file, 12-20
 - o output file, 12-19
 - p parameter file, 12-19
- pstrgvfy program
 - a alter mode flag, 15-6
 - b update MFHD 852b, 15-6
 - c command filename help, 15-7
 - d database name, 15-5
 - g cataloging location, 15-7
 - h help, 15-8
 - i input file, 15-5
 - l log filename, 15-8
 - m location, 15-5
 - o operator ID, 15-7
 - u username and password, 15-5
 - v verify mode flag, 15-6
- Password field, 31-7
- patron
 - authentication, B-1
- Patron Authentication Adaptor, B-1
- patron barcode
 - expire, 9-3, 9-41
- patron data
 - loading, 2-3
- patron extract program, 6-1
 - audit file, 6-5
 - error file, 6-6
 - output file, 6-5
 - parameters
 - a audit filename, 6-3
 - d database name, 6-2
 - e error filename, 6-3
 - h help, 6-4
 - m monitor records to process, 6-4
 - p patron output filename, 6-3
 - r number of records to process, 6-4
 - u username and password, 6-3
 - y, 6-2
 - patron record standard interface file (SIF), 6-5
 - running, 6-4
 - Patron Group Code field, 2-3
 - [Patron Groups] stanza, 8-5
 - patron purge, 9-3, 9-23
 - patron suspension, 9-7
 - running circulation batch job, 9-8, 9-9

patron update program, 7-1
 audit file, 7-7
 error file, 7-8
 input file, 7-3
 parameters
 -a audit filename, 7-4
 -d database name, 7-3
 -e error filename, 7-4
 -h help, 7-6
 -i index type, 7-3
 D program selected, 7-4
 I institution ID, 7-4
 S ssan, 7-3
 -m monitor records to process, 7-6
 -n notes type, 7-5
 G general, 7-5
 P popup, 7-5
 -n number of records to process, 7-6
 -o overlay previous notes, 7-5
 A add new notes, 7-5
 I ignore notes processing, 7-5
 R replace existing notes, 7-5
 -p patron input filename, 7-3
 -u username and password, 7-3
 -x extended notes processing, 7-5
 patron record standard interface file (SIF), 7-3
 running, 7-6
Pbibredirect.cgi script, 4-28
pbulkimport command, 13-9, A-11
pbursar command, 8-8
Pcircjob, 9-3, 9-4, 9-5, 9-6, 9-7, 9-8, 9-9, 9-11, 9-14,
 9-15, 9-17, 9-20, 9-21, 9-25, 9-26, 9-27, 9-28, 9-30,
 9-31, 9-33, 9-34, 9-39, 9-41
Pfixexchangerates
 log file, 17-6
photocopying
 documentation, xxvii
Phttplinkresolver.cgi script, 4-28
place items on active course reserve list, 9-10
 running circulation batch job, 9-11
place recalls and holds for items on active course
 reserve list, 9-12
pmarcexport command, 11-10, A-12
Popacjob, 16-1
Popaclogexp
 see OPAC Log Export program
populating operator or location on MFHD's history tab,
 15-3
Post button, 8-12

POST request, 4-28
pprebulk command, 12-20
pptrnext command, 6-4
pptrnupdt command, 7-6
prebulk program, 12-1
 checking records, 12-3
 configuration file, 12-5
 location, 12-6
 name, 12-6
 stanza types, 12-7
 creating
 holdings from interleaved file, 12-5
 interleaved file, 12-4
 creating interleaved bib-MFHD file, 12-4
 error file, 12-22
 importing interleaved file, 12-4
 input file, 12-5
 log (audit) file, 12-21
 output file, 12-21
 parameters
 -a append tag for log file, 12-20
 -c configuration file, 12-19
 -e error file, 12-19
 -f force overwrite of output files switch, 12-20
 -i input file, 12-19
 -l prebulk log file, 12-20
 -o output file, 12-19
 -p parameter file, 12-19
 running, 12-20
 striping fields, 12-3
Preview button, 14-5
process
 GHC fields-step 3 catjob 13, 14-6
 GHC queue-step 1
 running cataloging batch job, 14-4
 GHC queue-step 1 catjob 11, 14-4
 GHC queue-step 2
 running cataloging batch job, 14-5
 GHC queue-step 2 catjob 12, 14-5
Process box, 14-5
program
 bulk export of MARC records (marcexport), 11-2
 bulk import, 13-1
 bursar transfer, 8-1
 stub patron records, 8-2
 Global Headings Change (GHC), 14-1
 marcexport, 11-2
 patron extract, 6-1
 patron update, 7-1
 prebulk, 12-1
 pstrgvfy, 15-1

- storage barcode verify (pstrgvfy), 15-1
- promote.cfg file, 9-9
- pstrgvfy command, 15-10
- pstrgvfy program, 15-1
 - changing location codes of item records or MFHDs, 15-2
 - command file, 15-8
 - input file, 15-4
 - log file, 15-9
 - parameters
 - a alter mode flag, 15-6
 - b update MFHD 852b, 15-6
 - c command filename, 15-7
 - d database name, 15-5
 - g cataloging location, 15-7
 - h help, 15-8
 - i input file, 15-5
 - l log filename, 15-8
 - m location, 15-5
 - o operator ID, 15-7
 - u username and password, 15-5
 - v verify mode flag, 15-6
 - populating operator or location on MFHD's history tab, 15-3
 - verifying barcodes of item records or MFHDs, 15-4
- purchase order report suffix segment
 - acquisitions report, 24-5
- purge UB patron stub record, 9-5, 9-8, 9-6
- purpose
 - of this document, xxiii
- pwebadmind command, 31-3
- Pz3950svr, 18-14

R

- radio button
 - Create one summary transfer record per patron, 31-33
 - Create transfer record for each fine/fee transferred, 31-33
 - Date Range, 31-19
 - ID Range, 31-19
 - Today, 31-19
- recall notice suffix segment
 - circulation notice, 26-7
- recall-overdue notice suffix segment
 - circulation notice, 26-7
- record

- bibliographic, A-12
- Record does not match format for " change your import rule message, 13-14
- record ID number, A-10
- Record nnn discarded -- no matching records in the database message, 13-14
- Record not translated message, A-11
- record not translated message, A-11, A-13, A-16
- Record Parse Failure message, 13-14
- Record Retrieval Failure message, 13-14
- Recordcount_vw, 30-24
- refunding
 - bursar transfer program, 8-11
- reissue
 - reason for, xxiii
- reject file
 - bulk import program, 13-11
- reject.imp.ccyymmdd.hhmm file, 13-11
- repair, 4-13
- repairing
 - installation, 4-13
- replace file
 - bulk import program, 13-11
- replace.imp.ccyymmdd.hhmm file, 13-11
- report file utility
 - WebAdmin, 31-39
- Reports.mdb file, 4-48
- reproduction, of documentation, xxvii
- reserved items active report suffix segment
 - circulation report, 27-5
- reserved items expired report suffix segment
 - circulation report, 27-5
- restarting Tomcat, 3-18
- restoring
 - from backup, 3-19
- retain patron IDs, 9-17
 - media batch job, 10-1
- return notice suffix segment
 - acquisitions notice, 23-6
- Rule Code field, 31-14
- Run dialog box, 4-10, 4-14
- running
 - bulk import program, 13-9
 - bursar transfer program, 8-8
 - cataloging batch job
 - process GHC field-step 3, 14-6
 - process GHC queue-step 1, 14-4
 - process GHC queue-step 2, 14-5
 - circjob 38
 - command line, 9-20
 - interactively, 9-21

- circulation batch job
 - accrued fines and demerits, 9-7
 - archive and expire call slip requests, 9-4
 - archive short loans, 9-5
 - export OPAC requests, 9-4
 - forgive demerits, 9-17
 - patron suspension, 9-8, 9-9
 - place items on active course reserve list, 9-11
 - purge UB patron stub record, 9-6
 - update shelving status, 9-3, 9-14, 9-15
- mediajob 5
 - command line, 10-4
 - interactively, 10-5
- OPAC Bib Usage Log Export program as cron, 16-23
- OPAC Bib Usage Log Export program interactively, 16-21
- OPAC Search Log Export program as cron, 16-8
- OPAC Search Log Export program interactively, 16-7
- patron extract program, 6-4
- patron update program, 7-6
- prebulk program, 12-20
- Selective Dissemination of Information as cron, 16-14, 16-21
- Selective Dissemination of Information interactively, 16-13, 16-20
- running acqjob 5, 17-3
- running acquisitions batch jobs
 - webadmin, 31-9
- RunSearchEvery variable, 16-16

S

- screen
 - Job Scheduler, 31-14, 31-22, 31-23, 31-33, 31-36, 31-38
- SDI
 - see Selective Dissemination of Information
- [SDI_Page] stanza, 16-15, 16-16
- sdiemail.ini file, 16-10, 16-15
 - components of, 16-12, 16-18
- SDIOption= variable, 16-15, 16-16
- Search dialog box
 - additional button, 4-31
- search dialog box
 - URI link, 4-31
- [SearchURI stanza, 4-31
- see references with linked bib records suffix segment
 - cataloging report, 25-7

- see refs authorized in other auth record suffix segment
 - cataloging report, 25-7
- see refs without corresponding auth record suffix segment
 - cataloging report, 25-8
- Selective Dissemination of Information, 16-9, 16-15
 - configuring, 16-16
 - enabling, 16-10, 16-15
 - running as cron, 16-14, 16-21
 - running interactively, 16-13, 16-20
 - search results page
 - customizing, 16-12, 16-18
- Send Record To button, 4-29
- Send Record To option, 4-29
- serial claim notice suffix segment
 - acquisitions notice, 23-7
- Serials_vw, 30-24
- server
 - AIX, 3-4
 - Solaris, 3-4
- setting up
 - WebAdmin, 31-2
- sif.burs.yymmdd.hhmm file, 8-9
- Single Client Login, 4-23
- SMS, 33-1
- Solaris server, 3-4
 - starting
 - Apache, 3-10
 - Oracle services, 3-6
 - Voyager, 3-8
 - stopping
 - Apache, 3-10
 - Oracle services, 3-6
 - Voyager, 3-8
- Special Characters.cfg file, 4-49
- Spinelabel.cfg file, 4-50
- SQL queries, 32-1
- standard interface file
 - acquisitions notice, 23-1
 - base segment, 23-4
 - cancel serial claim notice suffix segment, 23-8
 - cancellation notice suffix segment, 23-6
 - order claim notice suffix segment, 23-7
 - return notice suffix segment, 23-6
 - serial claim notice suffix segment, 23-7
 - suffix segment, 23-5
 - voucher/check request suffix segment, 23-8
 - acquisitions report, 24-1
 - base segment, 24-4
 - fund snapshot after rollover report suffix segment, 24-13

- fund snapshot before rollover report suffix segment, 24-12
- fund snapshot report suffix segment, 24-11
- global open orders report suffix segment, 24-9
- open orders after rollover report suffix segment, 24-12
- open orders before rollover report suffix segment, 24-12
- open orders report suffix segment, 24-8
- purchase order report suffix segment, 24-5
 - suffix segment, 24-4
- cataloging report, 25-1
 - 856 link failure report suffix segment, 25-8
 - base segment, 25-4
 - duplicate authority records suffix segment, 25-6
 - see references with linked bib records suffix segment, 25-7
 - see refs authorized in other auth record suffix segment, 25-7
 - see refs without corresponding auth record suffix segment, 25-8
 - suffix segment, 25-4
 - unauthorized name headings suffix segment, 25-5
 - unauthorized name/title headings suffix segment, 25-6
 - unauthorized subdivision headings suffix segment, 25-6
 - unauthorized subject headings suffix segment, 25-5
 - unauthorized title headings suffix segment, 25-5
- circulation notice, 26-1
 - base segment, 26-4
 - cancellation notice suffix segment, 26-6
 - courtesy (due) notice suffix segment, 26-9
 - fine/fee notice suffix segment, 26-7
 - item available notice suffix segment, 26-6
 - overdue notice suffix segment, 26-6
 - recall notice suffix segment, 26-7
 - recall-overdue notice suffix segment, 26-7
 - statement of fines and fees suffix segment, 26-8
 - suffix segment, 26-5
- circulation report, 27-1
 - base segment, 27-4
 - circulation item exceptions report suffix segment, 27-7
 - circulation patron exceptions report suffix segment, 27-7
 - circulation statistics report suffix segment, 27-6
 - circulation transaction exceptions report suffix segment, 27-7
 - distribution item order list report suffix segment, 27-8
 - global circulation statistics report suffix segment, 27-7, 27-8
 - hold shelf expired report suffix segment, 27-5
 - missing in transit report suffix segment, 27-6
 - reserved items active report suffix segment, 27-5
 - reserved items expired report suffix segment, 27-5
 - suffix segment, 27-4
- media scheduling notice, 28-1
 - base segment, 28-3
 - overdue notice suffix segment, 28-5
 - suffix segment, 28-4
- media scheduling report, 29-1
 - base segment, 29-3
 - booking charge statistics report suffix segment, 29-5
 - booking exceptions report suffix segment, 29-5
 - booking statistics report suffix segment, 29-4
 - media equipment inventory report suffix segment, 29-4
 - suffix segment, 29-3
- standard interface file (SIF)
 - bursar transfer program, 8-9
 - charge transaction record, 20-1
 - item delete, 21-1
 - patron record, 6-5, 7-3, 19-1
 - vendor record, 22-1
- stanza
 - [008], 12-16
 - [CALLTYPES], 12-12
 - [Circ Locations], 8-5
 - [E-mail], 4-27
 - [Email], 16-12, 16-18
 - [FINE FEE TYPE], 8-5
 - [GlobalLog], 4-22, 4-23
 - [HelpMenuLink], 4-48
 - [LOCATIONS], 12-12
 - [MAPPING], 12-15
 - [MARC POSTing], 4-27, 4-28
 - [MFHDTAG], 12-11
 - [Overrides], 12-8
 - [Patron Groups], 8-5
 - [SDI_Page], 16-15, 16-16

- [SearchURI], [4-31](#)
- [Strip], [12-9](#)
- [Upgrade] (Voyager 7), [4-32](#)
- [Upgrade] (Voyager 8 and later), [4-44](#)
- [Usemarcon translations], [A-14](#)
- [Webvoyage_Server], [16-12](#), [16-18](#)
- starting
 - Apache
 - AIX server, [3-10](#)
 - Solaris server, [3-10](#)
 - Oracle services
 - AIX server, [3-6](#)
 - Solaris server, [3-6](#)
 - Voyager
 - AIX server, [3-8](#)
 - Solaris server, [3-8](#)
 - WebAdmin, [31-6](#)
 - WebAdmin server daemon, [31-3](#)
- statement of fines and fees suffix segment
 - circulation notice, [26-8](#)
- StatString= variable, [16-5](#)
- stopping
 - Apache
 - AIX server, [3-10](#)
 - Solaris server, [3-10](#)
 - Oracle services
 - AIX server, [3-6](#)
 - Solaris server, [3-6](#)
 - Voyager
 - AIX server, [3-8](#)
 - Solaris server, [3-8](#)
- storage barcode verify (pstrgvfy) program, [15-1](#)
- [Strip] stanza, [12-9](#)
- striping fields with prebulk, [12-3](#)
- stripping
 - non standard 035 field, [12-10](#)
- Submit Batch Import button, [31-16](#)
- Submit Bulk Export button, [31-22](#)
- Submit Bursar Transfer button, [31-33](#)
- Submit OPAC Job button, [31-38](#)
- Sudocclass_vw, [30-25](#)
- suffix segment
 - acquisitions notice, [23-5](#)
 - acquisitions report, [24-4](#)
 - cataloging report, [25-4](#)
 - circulation notice, [26-5](#)
 - circulation report, [27-4](#)
 - media scheduling notice, [28-4](#)
 - media scheduling report, [29-3](#)
- System Control (003) Identifier field, [31-19](#)
- System Control (040\$d) Identifier field, [31-19](#)

- system monitoring job, [34-1](#)
- system requirements
 - WebAdmin, [31-2](#)
- System tab, [13-6](#), [31-15](#)

T

- tab
 - Fines/Fees History, [8-11](#)
 - History, [13-6](#), [15-1](#), [15-3](#), [15-7](#)
 - System, [13-6](#), [31-15](#)
- takes items on inactive course reserve lists off reserve, [9-14](#)
- Template.ini file, [4-49](#)
- Today radio button, [31-19](#)
- Tomcat
 - restarting, [3-18](#)
- Transfer fines/fees greater than or equal to field, [31-33](#)
- transferring
 - itemized patron fines/fees, [8-4](#)
 - total patron balances, [8-4](#)

U

- unauthorized name headings suffix segment
 - cataloging report, [25-5](#)
- unauthorized name/title headings suffix segment
 - cataloging report, [25-6](#)
- unauthorized subdivision headings suffix segment
 - cataloging report, [25-6](#)
- unauthorized subject headings suffix segment
 - cataloging report, [25-5](#)
- unauthorized title headings suffix segment
 - cataloging report, [25-5](#)
- Unicode character set, [31-21](#)
- uninstalling clients, [4-15](#)
- Unparseable record written to error file message, [13-14](#)
- update remote circulation cluster cache
 - circjob 33, [9-9](#)
- update shelving status, [9-3](#)
 - running circulation batch job, [9-3](#), [9-14](#), [9-15](#)
- [Upgrade] stanza (Voyager 7), [4-32](#)
- [Upgrade] stanza (Voyager 8 and later), [4-44](#)
- USE001FOR014 variable, [12-9](#)
- USE003FORLOC variable, [12-9](#)
- UseMARCON

- API, [11-2](#), [13-1](#), [A-1](#)
- bulk import, [A-10](#)
- conversion directories, [A-2–A-3](#)
- error reporting, [A-11](#)
 - END non OK message, [A-8](#)
 - MARC Export, [A-13](#)
 - Non-fatal errors, [A-8](#)
 - Z39.50, [A-15](#)
- file structure, [A-2–A-5](#)
- initialization file, [A-3](#)
- log file, [A-5](#)
- record ID number, [A-10](#)
- required files, [A-4](#)
 - Input format checking (.chk) file, [A-4](#)
 - MARC Input (.mrc) file, [A-5](#)
 - MARC Output (.mrc) file, [A-5](#)
 - Output format checking (.chk) file, [A-4](#)
 - Rule (.rul) file, [A-4](#)
 - Translation (.trs) file, [A-4](#)
- translations, [A-11](#)
- troubleshooting error conditions, [A-8–??](#)
- [Usemarcon translations] stanza, [A-14](#)
- usemarcon.bip.date.time file, [A-5](#)
- usemarcon.mxp.date.time file, [A-5](#)
- usemarcon.z39.date.time file, [A-5](#)
- User Name field, [31-6](#)
- using
 - prebulk to check records, [12-3](#)
 - prebulk to create interleaved file, [12-4](#)
 - prebulk to strip fields, [12-3](#)
 - WebAdmin, [31-7](#)

V

- variable
 - Bi-Weekly, [16-18](#)
 - Daily, [16-17](#)
 - GetNewHits, [16-17](#)
 - LOGBIBUSAGE=, [16-5](#)
 - LOGSEARCH=, [16-4](#)
 - Monthly, [16-18](#)
 - NewHits, [16-17](#)
 - nnnn, [A-10](#)
 - None, [16-17](#)
 - RunSearchEvery, [16-16](#)
 - SDIOption=, [16-15](#), [16-16](#)
 - StatString=, [16-5](#)
 - Weekly, [16-17](#)
- vendor data
 - loading, [2-4](#)

- Vendorinvoice_vw, [30-26](#)
- Vendororder_vw, [30-28](#)
- verifying barcodes of item records or MFHDs, [15-4](#)
- viewing
 - bursar transfer program information, [8-11](#)
- voucher/check request suffix segment
 - acquisitions notice, [23-8](#)
- Voyager Server Utilities (main) page, [31-7](#)
- voyager.env, [18-1](#)
- Voyager.ini file, [4-9](#), [4-17](#)
- voyager.ini file, [16-4](#)
 - module stanzas, [4-20](#)
- voyager.ini file (server side)
 - noisewordfilter, [18-12](#)
- VoyagerInstall.exe file, [4-17](#)

W

- WARNING message, [A-8](#)
- WebAdmin, [31-1](#)
 - acquisitions utilities, [31-9](#)
 - adding a login/password, [31-4](#)
 - bulk import program, [31-12](#)
 - bursar transfer program, [31-30](#)
 - cataloging utilities, [31-12](#)
 - circulation reports and notices, [31-24](#)
 - creating a new login/password, [31-3](#)
 - deleting a login/password, [31-4](#)
 - GHC program, [31-22](#)
 - log file utility, [31-38](#)
 - marcexport program, [31-17](#)
 - media scheduling reports and notices, [31-34](#)
 - modifying a login/password, [31-4](#)
 - OPAC reports, [31-36](#)
 - report file utility, [31-39](#)
 - setting up, [31-2](#)
 - starting, [31-6](#)
 - sever daemon, [31-3](#)
 - system requirements, [31-2](#)
 - using, [31-7](#)
- webadmin
 - running acquisitions reports and notices, [31-9](#)
- webadmin main page, [31-10](#)
- WebVoyage, [16-15](#)
- [Webvoyage_Server] stanza, [16-12](#), [16-18](#)
- webvoyage.properties, [16-10](#)
- Weekly variable, [16-17](#)

X

XML entity reference, [11-8](#)

Z

Z39.50, [A-13](#)

Z39.50 clients, [A-13](#)

Z39.50 Server, [A-13](#)

 initialization file, [A-13](#)

 log file, [A-15](#)

 UTF-8 encoded records, [18-13](#)

z3950svr.ini, [18-13](#)

z3950svr.ini file, [A-13](#)

z3950svr.log file, [A-15](#)