Voyager®
Universal Catalog User's Guide

March 2014

Ex Libris Confidential

The bridge to knowledge

CONFIDENTIAL INFORMATION

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

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

DISCLAIMER

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

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

TRADEMARKS

"Ex Libris," the Ex Libris Bridge 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.

Document released: March 2014

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

Contents

About This Document

1 Getting Started

2 Before You Load Records

Contents

Contents

Contents

About This Document

Purpose

This document provides instructions for implementing a Voyager® Universal Catalog (UC) database and how to search the UC database. It contains the following information.

- Prerequisites
- Pre-Initial database load considerations and software settings
- Steps for loading records into your Voyager Universal Catalog database
- Maintenance considerations for a UC database
- Steps for searching the UC database

Intended Audience

This document is intended for Voyager customers who plan to implement one or more UC databases.

Reason for Reissue

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

- Removed Note from Pucatimp on page 3-34
- Removed -P reference from Table 3-28 on page 3-35
- Updated the guide format for consistency with the other user’s guides

How to Use This Document

This document consists of the following:

Chapter 1 — Getting Started
Chapter 1 provides an overview of the Voyager Universal Catalog and implementation considerations.

Chapter 2 — Before You Load Records
Chapter 2 details the Voyager Universal Catalog configuration settings. This includes all of the changes that need to be made before you load records for the first time.

Chapter 3 — Loading Records into and Maintaining the Universal Catalog
Chapter 3 details loading records into the Voyager Universal Catalog database and the process for maintaining records and adding new ones.

Chapter 4 — Searching the Universal Catalog
Chapter 4 details how to search the Voyager Universal Catalog database using the Voyager WebVoyáge client.

Appendix A — Call Numbers in MFHDs
Appendix A details considerations for including call numbers in MFHDs and the process for including these in the Universal Catalog.

Appendix B — Multiple Owning Libraries
Appendix B details the rules and considerations relating to multiple owning libraries in the Universal Catalog

Index — The Index is an alphabetical, detailed cross-reference of topics.

Conventions Used in This Document

The following conventions are used throughout this document:

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

Getting Started

1

Introduction

The Voyager Universal Catalog (UC) is a collection of records from a group of individual Voyager libraries, stored in one combined database. It serves as a bibliographic utility to which you can contribute records and search the collections of many participating libraries at once. In this way, connecting to a UC database allows you to search more efficiently and quickly than searching remote library collections one database at a time or even simultaneously. This is particularly beneficial for consortia or other resource sharing initiatives that have reciprocal agreements for sharing intellectual property.

The Voyager Universal Catalog is also beneficial for distance education, as it points to the resources of many contributing libraries. This is important in that a variety of collections may be searched without being limited to resources at a local library. Likewise, the UC benefits smaller institutions by providing access to a variety of resources that would otherwise be unavailable.

The Voyager Universal Catalog system features the following capabilities.

- Real-time retrieval and display of detailed holdings and item information from local databases
- Holdings "jump bar" feature for easier and quicker navigation of holdings information
- Integration with Universal Borrowing

It is also important to note that many standard Voyager functions and maintenance are unnecessary with the Universal Catalog database. These activities such as the ones done with the Circulation, Acquisitions, and Cataloging modules are performed on the local Voyager databases of contributing libraries. Through the UC “ongoing update” process, local changes are incorporated into the UC database.

Purpose of this Chapter

The purpose of this chapter is to provide introductory information about the Voyager Universal Catalog capability and prerequisite information for implementation of a Universal Catalog database.

The Voyager Universal Catalog capability is integrated with other components of Voyager. As a result, you may want to refer to the *Voyager Cataloging User’s Guide* and the *Voyager System Administration User’s Guide* for additional information.

Prerequisite Skills and Knowledge

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

- Basic Microsoft® Interface navigation
- Basic UNIX® commands and navigation
- Basic Voyager System Administration Module
- Basic WebVoyáge configuration
- Intermediate knowledge of MARC record formats

Installing the UC Database and Server

Since the Universal Catalog database is essentially its own separate Voyager database, you install it the same way you would a regular Voyager database. You also need to install a separate server for storing and running the Voyager UC database.

NOTE:
The only Voyager clients that need to be installed are the WebVoyáge and Cataloging clients for view-only access to records. Record entry and maintenance for Circulation, Acquisitions, and most standard Cataloging activities are to be managed by the local libraries rather than in the Universal Catalog

database. By design, the intent is for the Voyager Universal Catalog database to maintain the currency of its records through the ongoing update process that incorporates changes from the local libraries.

The technical detail regarding how to set up a Voyager database is provided through the standard set of Voyager documentation. As a result, this user's guide assumes that you have a server for Voyager, Voyager software, and related programs properly installed and configured. The focus of this user's guide is to provide customizing and tailoring information specific to the setup and running of a Voyager Universal Catalog database system.

Types of Records in the UC

The UC database stores the following types of records.

- Bibliographic records
- Holding records/MARC Format for Holdings (referred to throughout this user's guide as MFHDs)
- Authority records

Details regarding how records are loaded and maintained in the Voyager Universal Catalog database are described in Initially Loading Records into the UC on page 3-6 and Ongoing Updating of UC Records on page 3-30.

NOTE:
By design, item records are not stored in the Universal Catalog. The intent is for the Universal Catalog database to be a repository that is used for searching purposes. This is accomplished through bibliographic, authority, and MFHD records stored in the UC database with the MFHDs pointing to local databases in order to dynamically pull back detailed holdings and item information (including status) for display. Record entry and maintenance for Circulation, Acquisitions, and most standard Cataloging activities are, therefore, managed by the local library rather than in the Universal Catalog database itself.

Bibliographic Records

Bibliographic records are at the core of the Voyager Universal Catalog database because they contain pertinent information about a particular title. Each bibliographic record has an attached MFHD record indicating to which local library database the title belongs.

If the same bibliographic record is contributed by several library databases, many different MFHDs are attached to the one bibliographic record in the UC database. For instance, if the same bibliographic record is added by three local library databases, one bibliographic record is stored in the UC database as determined by the bibliographic duplicate detection profile. Additionally, the three corresponding MFHD records are attached.

NOTE:
Multiple bibliographic records may occur in the UC database for a particular title if "multiple owning libraries are defined as part of the UC database system. For more information, see UC Owning Library on page 2-7 and Multiple Owning Libraries on page B-1.

Duplication of Records

Duplicate bibliographic records in the Voyager Universal Catalog are eliminated to facilitate a well-organized and clean database. This process of evaluating duplicate records occurs as records are loaded into the UC database. Duplication is determined by the bibliographic duplicate detection profile that is established in the Voyager System Administration module for the UC that is also a part of the ongoing update process. See *Bibliographic Duplicate Detection Profile* on page 2-21 for more information about the bibliographic duplicate detection setup for the Voyager Universal Catalog database.

MFHDs

MFHDs are generated and attached to bibliographic records during the load of bibliographic records into the UC database. For UC purposes, a MFHD contains the following two fields and subfields:

- 014‡a (Linkage Number)
- 852‡b (Location)

The 014‡a field contains the Voyager bibliographic ID and links the MFHD to its associated bibliographic record in the local database. The 852‡b field indicates the location or to which local library database the record belongs. The 852‡b location information comes from the value set in the pre-bulk configuration file. See Prebulk Configuration File (ucatexp.pl) on page 3-9 for more information.

IMPORTANT:
Since MFHDs in the Universal Catalog usually only contain the 014‡a and the 852‡b fields, they function as pointer. Unlike MFHDs in a normal Voyager database, the UC database MFHDs do not contain detailed holdings information. As part of the search results function of the Universal Catalog software, real-time retrieval and display of detailed holdings and item information are obtained from the MFHD/Item records stored in the local libraries' databases. This is accomplished through a dynamic

connection to the local libraries' local database servers to pull the detailed information.

NOTE:
The MFHD records may also contain 852‡h and 852‡i for call number information. See Call Numbers in MFHDs on page A-1 for more information.

MFHD Options

In an effort to provide Universal Catalog implementers with as much flexibility as possible, call numbers can optionally be stored in the MFHD records during the initial load and/or during ongoing maintenance of the UC database. However, there are special considerations when this is done.

For additional information regarding these considerations and the steps required to implement call numbers in the UC MFHD records, see Call Numbers in MFHDsstarting on A-1.

Authority Records

In addition to bibliographic records and MFHDs, you can include authority records in the UC database. See Loading Authority Records on page 3-29 for more information regarding how to load Authority records into the UC database.

OPTIONAL:
If you include authority records in the UC, you may want to consider loading the entire file of the Library of Congress Name and Subject Headings records for consistency of authority records.

Universal Catalog Setup Checklists

The purpose of this section is to provide you with a checklist of setup considerations specific to the Universal Catalog environment. The primary areas requiring attention include the following.

- Voyager System Administration
- `Opac.ini` file
- `Connect.ini` file

For non-UC specific Voyager setup instructions and guidelines, see the *Voyager System Administration User's Guide* and the *Voyager WebVoyáge User's Guide*.

Use the checklists in Table 1-1, Table 1-2, and Table 1-3 to assist with planning your implementation of a UC database. Mark off each option in the Completed column as it is considered or customized.

NOTE:
"No" in the Required column of the checklist indicates that you may choose to activate the functions or features listed to create the Universal Catalog system as you prefer it. However, default data elements are in place and do not need to be changed.

Table 1-1. Voyager Configuration Checklist - Voyager System Administration (UC)

Topic	Required?	Completed
System > Locations The locations settings in the Universal Catalog environment are used to identify each contributing library database.	**Yes**	
System > Owning Libraries One must specify an owning library for the Universal Catalog environment to which bibliographic records and MFHD records belong. All UC participating library location names are associated with the owning library name.	**Yes**	
Call Slips > Queues A default Call Slip definition must be configured if Universal Borrowing is installed. This allows for routing of Call Slip information throughout the Universal Catalog system. See the *Voyager System Administration User's Guide* and the *Voyager Universal Borrowing User's Guide* for additional configuration information.	**No**	
Search > Database Definitions Database definitions must be set in order for the Universal Catalog system to dynamically access the local databases to retrieve detailed holdings and item information. See `Connect.ini` on page 2-17 for more information.	**Yes**	

Table 1-1. Voyager Configuration Checklist - Voyager System Administration (UC)

Topic	Required?	Completed
System > Default Address A default address must be set. This setting displays in WebVoyáge as the **Library Name**.	**Yes**	
OPAC Configuration > Request Forms OPAC Request Forms may need to be configured if Universal Borrowing is installed. See the *Voyager Universal Borrowing User's Guide* for setup information.	**No**	
OPAC Configuration > Miscellaneous OPAC Miscellaneous must be configured if Universal Borrowing is installed. See the *Voyager Universal Borrowing User's Guide* for setup information.	**No**	
Cataloging > Bibliographic Duplicate Detection Profiles Bibliographic Duplicate Detection Profiles need to be set for the Bulk Import process.	**Yes**	
Cataloging > Authority Duplicate Detection Profiles An Authority Duplicate Detection Profile is needed for loading authority records into the UC database.	**Yes**	
Cataloging > Bulk Import Rules The Bulk Import Rule is needed for importing records using the Bulk Import function.	**Yes**	
Cataloging > Policy Definitions A Cataloging Policy Definition should be established to view/compare two or more detailed bibliographic records.	**Yes**	
Security > Operator Profiles Each operator or group of operators using Voyager staff modules requires an Operator Profile.	**Yes**	

Table 1-1. Voyager Configuration Checklist - Voyager System Administration (UC)

Topic	Required?	Completed
Security > Master Profiles A Master Security Profile is needed to control access to the Voyager System Administration module.	**Yes**	

Table 1-2. WebVoyáge Configuration Checklist - Opac.ini File

Stanza	Required?	Completed
NoHits_Redirect Use the NoHits Redirect function/button to allow patrons to easily search another database when a search returns zero hits in the Universal Catalog database	**No**	
View_Record_Page Enable the Jump Bar function (DBHoldingsJumpBar = Y) from within the [View_Record_Page] stanza. This function provides easy navigation through a potentially long list of items and displays a list of “Held at:” libraries.	**No**	
Login Types Login Types must be configured if Universal Borrowing is installed. See the *Voyager Universal Borrowing User’s Guide* for setup information. It defines if the libraries use more than one type such as social security numbers, barcodes, or institutional IDs.	**No**	
Logon_Page Logon_Page may need to be configured if Universal Borrowing is installed. See the *Voyager Universal Borrowing User’s Guide* for setup information. This includes the Home Library variable which controls the text that appears in front of the database drop-down menu on the Logon page.	**No**	

Table 1-2. WebVoyáge Configuration Checklist - Opac.ini File

Stanza	Required?	Completed
Patron_Info_Page Patron_Info_Page may need to be configured if Universal Borrowing is installed. See the *Voyager Universal Borrowing User's Guide* for setup information. This allows you to define the various sections of the Patron Information page as well as control any labels and colors on the page.	**No**	
Patron_Request_Page Patron_Request_Page may need to be configured if Universal Borrowing is installed. See the *Voyager Universal Borrowing User's Guide* for setup information. This allows you to define color settings for the page that displays after the successful submission of a UB request.	**No**	
Request Dialogs Request Dialogs must be configured if Universal Borrowing is installed. See the *Voyager Universal Borrowing User's Guide* for setup information. This includes three variables that must be defined for each UB library. • UBLibrary • UBPickupLibrary • UBPickupDesk	**No**	
Request_Page Request_Page may need to be configured if Universal Borrowing is installed. See the *Voyager Universal Borrowing User's Guide* for setup information. This allows one to define color settings for the page that displays if there is a validation error after a patron clicks the "Submit" button on the UB request form.	**No**	

Table 1-2. WebVoyáge Configuration Checklist - Opac.ini File

Stanza	Required?	Completed
Request_Select_Page Request_Select_Page may need to be configured if Universal Borrowing is installed. See the *Voyager Universal Borrowing User’s Guide* for setup information. This allows one to define color settings for the page on which a patron selects a request form.	**No**	
Title_Bar Title_Bar may need to be configured if Universal Borrowing is installed. See the *Voyager Universal Borrowing User’s Guide* for setup information. The Title_Bar contains variables that control the text in the title bar of various WebVoyáge pages including the UB Request page.	**No**	
UB_Page UB_Page may need to be configured if Universal Borrowing is installed. See the *Voyager Universal Borrowing User’s Guide* for setup information. The UB_Page contains the TurnOffLibraryDropDown variable that can be turned on or off to control the library drop-down list on the UB request form.	**No**	

Table 1-3. WebVoyáge Configuration Table - Connect.ini File

Stanza	Required?	Completed
Add Other Database Information Like other Voyager systems, the connect.ini file must be configured on the UC system for simultaneous searching and for library database names to display on the drop-down list on the databases page. This enables searching outside the Universal Catalog database like Z39.50 sites, or other Voyager sites outside the set of libraries participating in the UC.	**No**	

Before You Load Records

2

Universal Catalog System Setup: Pre-Initial Load

The following configurations must be set for the Universal Catalog in the System Administration module before records can be loaded and maintained.

- Locations for contributing libraries
- UC owning library
- UC default address
- Database definitions
- Bibliographic duplicate detection profile
- Authority duplicate detection profile
- Bulk Import rule
- Cataloging policy definition
- Security setup

NOTE:
You will see some additional settings in the Universal Catalog System Administration module that you have not set up on your own such as Circulation Policy Groups. These settings are automatically created on the UC server during the load process. The only System Administration settings you need to define are the ones discussed in the remainder of this chapter.

Locations for Contributing Libraries

You must define a location for each contributing library database on the Universal Catalog server. This involves the use of the 040‡d and 852‡b fields that indicate modifying agency (NUC code) and location, respectively.

The procedure for adding and defining a location for contributing libraries is shown in Procedure 2-1, Defining a Location for Contributing Libraries, on page 2-2.

Procedure 2-1. Defining a Location for Contributing Libraries

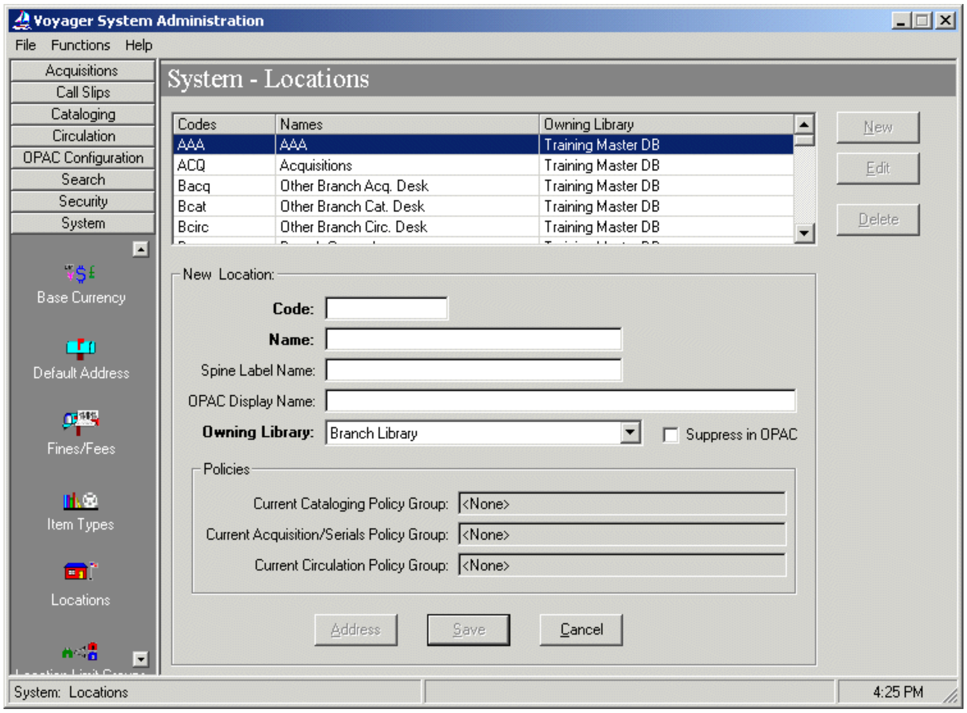
Use the following to add and define a location in Voyager System Administration.

1. Click **System** on the listbar and click **Locations**.

   Result: A list of any previously defined locations displays.

2. Click the **New** button.

   Result: The **New Location** group box opens. See Figure 2-1.



Codes	Names	Owning Library
AAA	AAA	Training Master DB
ACQ	Acquisitions	Training Master DB
Bacq	Other Branch Acq. Desk	Training Master DB
Bcat	Other Branch Cat. Desk	Training Master DB
Bcirc	Other Branch Circ. Desk	Training Master DB

New Location:

Code:

Name:

Spine Label Name:

OPAC Display Name:

Owning Library: Branch Library

□ Suppress in OPAC

Policies

Current Cataloging Policy Group: <None>

Current Acquisition/Serials Policy Group: <None>

Current Circulation Policy Group: <None>

Figure 2-1. Voyager System Administration - New Location dialog box

3. Enter location information to match your requirements. See Table 2-4 for more information about each field in the **New Location** group box.

   Result: New Location information is customized for your institution.

Table 2-4. Voyager System Administration New Location settings

Name	Description	Required	Type and Range
[Location] Code	The system stores Location Codes to display location information in MARC holdings records (used for an item’s permanent home location) and in item records (used for an item’s permanent or temporary “home” location). The code must be unique. For Universal Catalog purposes, enter the code for the contributing library.	Yes	Alphanumeric, punctuation, and spaces. Uppercase and lowercase allowed. 10 characters.
[Location] Name	Voyager clients display Location Names in A-Z drop-down lists in staff modules. Using a common prefix helps group related locations. This is especially useful if your Voyager system contains data from multiple, physically distinct collections. Unless you define a Location OPAC (WebVoyáge) Display Name, the Location Name displays in WebVoyáge. For Universal Catalog purposes, enter the name for the contributing library.	Yes	Alphanumeric, punctuation, and spaces. Uppercase and lowercase allowed. 25 characters.
Spine Label Name	Enter a spine label name if you print locations on spine labels. If left blank, only call numbers print on spine labels. Multiple locations may share the same Spine Label Name.	No	Alphanumeric, punctuation, and spaces. Uppercase and lowercase allowed. 25 characters.

Table 2-4. Voyager System Administration New Location settings

Name	Description	Required	Type and Range
OPAC Display Name	If left blank, the system automatically uses the Location Name. OPAC display names (names that display in WebVoyáge) are often helpful if you want to give more specific location information to your patrons, such as items at this location are non-circulating. You can use the same Location OPAC Display Name for different locations if you need to distinguish locations for staff but not patrons. That is, different location codes in staff records display the same Location OPAC Display Name to patrons. Notice that for some locations, the examples use floor information. Since the MARC holdings or item record stores only the location code, the system merely points to a display name and uses whatever it finds. This means you can change the Location OPAC Display Name without changing Location Codes.	No	Alphanumeric, punctuation, and spaces. Uppercase and lowercase allowed. 60 characters.
Owning Library	Select the owning library defined for the Universal Catalog. You define this owning library for the UC database through the Voyager System Administration client. See the section UC Owning Library on page 2-7 for additional information. For more information, see the *Voyager System Administration User's Guide*.	Yes	Select from the drop-down list.

Table 2-4. Voyager System Administration New Location settings

Name	Description	Required	Type and Range
Suppress in OPAC	If you do not want items at this location to display in WebVoyáge, select it. Any OPAC Display Name that you define does not apply. This is not retroactive. Suppressing a location does not retroactively suppress holdings records assigned to that location. This option only impacts new holding records (MFHDs). You must manually suppress existing MFHDs through the Cataloging module. (See the *Voyager Cataloging User’s Guide.*) If you are cataloging restricted or classified materials and you want them automatically suppressed from WebVoyáge display, you should only assign them to those locations where this value is selected. You can also suppress individual records from WebVoyáge display on a record-by-record basis. If an item is temporarily housed in a staff department, you still probably want it displayed in WebVoyáge with an accurate location.	No	Check box The default is unchecked.
Policies: Current Cataloging Policy Group	The Policies section displays the Cataloging Policy Group associated with the selected location.		
Policies: Current Acquisitions/ Serials Policy Group	The Policies section displays the Acquisitions/Serials Policy Group associated with the selected location.		

Table 2-4. Voyager System Administration New Location settings

Name	Description	Required	Type and Range
Policies: Current Circulation Policy Group	The Policies section displays the Circulation Policy Group associated with the selected location.		
Address button	When checked the Address information for this location displays in the System Address dialog box. For more information, see the *Voyager System Administration User's Guide*. Once a location code and name are provided the Address button becomes active.		Button

4. Click **Save** or **Cancel**.

   Result: This saves or cancels the New Location settings.

Repeat the steps in Procedure 2-1, Defining a Location for Contributing Libraries, on page 2-2 until a separate location for each participating library has been established.

UC Owning Library

You must specify an owning library for the UC to which bibliographic records and MFHD records belong. Location names entered for each UC database participant are associated with the UC database owning library name.

NOTE:
Owning libraries' records in local catalogs are reduced to a single bibliographic record through bibliographic duplicate detection processing.

See the *Voyager System Administration User's Guide* for more information about owning libraries.

RECOMMENDED:
We recommend that only one owning library be created for the UC database. However, the Universal Catalog program has the flexibility to accommodate more than one owning library. See Multiple Owning Libraries on page B-1 for a description of considerations and steps associated with implementing multiple owning libraries in the UC database.

Owning Libraries Setup

The procedure for setting up an owning library for the Universal Catalog database is shown in Procedure 2-2, Setting Up an Owning Library for the UC Database, on page 2-8.

Procedure 2-2. Setting Up an Owning Library for the UC Database

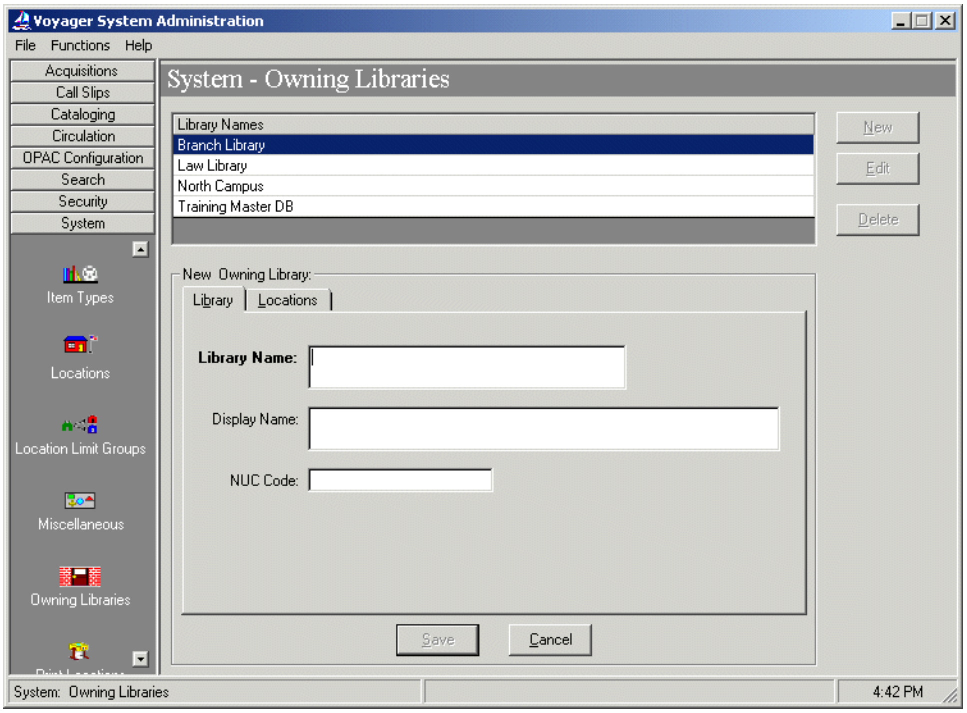
Use the following to set up an owning library for the Universal Catalog database.

1. Click **System** on the listbar and click **Owning Libraries**.

   Result: A list of any previously defined owning libraries displays.

2. Click the **New** button.

   Result: The **New Owning Library** group box opens. See Figure 2-2.



Library Names
Branch Library
Law Library
North Campus
Training Master DB

Library Name:

Display Name:

NUC Code:

System: Owning Libraries

4:42 PM

Figure 2-2. New Owning Library group dialog box

3. Enter the **Library** tab options. See Table 2-5 for more information about each field on the **Library** tab.

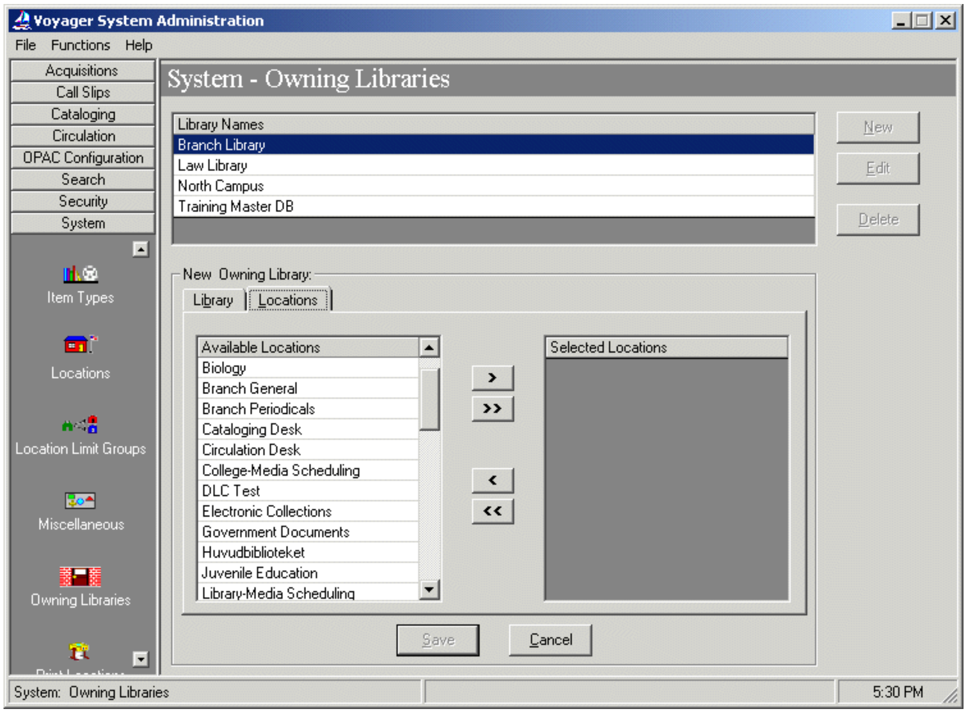
   Result: This completes the definition of the **Library** portion of an Owning Library.

Table 2-5. Library tab options

Options	Description
Library Name	Enter the Library Name up to 50 characters in length. This is the Owning Library Name that displays on the bibliographic record and holding record in the Universal Catalog.
Display Name	Enter the Display Name up to 80 characters in length. This is the name that displays on the Titles Search Results page in WebVoyáge. For this information to display, the Owning Library (title name created in Search - Title List Column Names) needs to be-selected in Voyager System Administration Search - Indexes - Composite Definitions on the **Search Results** tab.
NUC Code	Enter the Owning Library’s NUC code. During bulk import or online import, the National Union Catalog (NUC) code identifies the owning library. The NUC value used for a new bibliographic or authority record creates an 040 tag. The 040 tag should be your code for the owning library.

4. Click the **Locations** tab.

   Result: A list of **Available Locations** and **Selected Locations** display. See Figure 2-3.



Library Names
- Branch Library
- Law Library
- North Campus
- Training Master DB

Available Locations
- Biology
- Branch General
- Branch Periodicals
- Cataloging Desk
- Circulation Desk
- College-Media Scheduling
- DLC Test
- Electronic Collections
- Government Documents
- Huvudbiblioteket
- Juvenile Education
- Library-Media Scheduling

Selected Locations

Figure 2-3. New Owning Library Locations tab

5. Highlight the location for each contributing library that you want associated with the UC Owning Library from the list of **Available Locations**, and click the **right arrow (>)** button.

   Result: This moves the intended location to the **Selected Locations** list.

OPTIONAL:

6. *Optionally click the **double right arrow (>>)** button to associate all contributing library locations with the UC Owning Library.*

   Result: This moves all the **Available Locations** to the **Selected Locations** list.

NOTE:
Usually, all contributing library locations are selected unless multiple owning libraries are being implemented. See Multiple Owning Libraries on page B-1 for more information about implementing multiple owning libraries.

7. Click **Save** or **Cancel**.

Result: This saves or cancels the **New Owning Library** settings.

Database Definitions

In the Universal Catalog environment, database definitions are required to enable the real-time retrieval and display of detailed holdings and item information from the local databases. The Universal Catalog system needs to know the connection information to access the UC participating libraries databases' detailed holdings and item information. The information in this section highlights the installation steps for database definitions in a UC environment. More detailed information about database definitions can be found in the *Voyager System Administration User's Guide*.

Creating Database Definitions

Database definitions need to be created on the UC server for each local library that the UC server connects to for detailed holdings and item information.

The procedure for creating database definitions is shown in Procedure 2-3, Creating Database Definitions, on page 2-11.

Procedure 2-3. Creating Database Definitions

Use the following to create database definitions on the Universal Catalog system.

1. Log into the Voyager System Administration module.

   Result: The Voyager System Administration listbar displays.

2. Click **Search**.

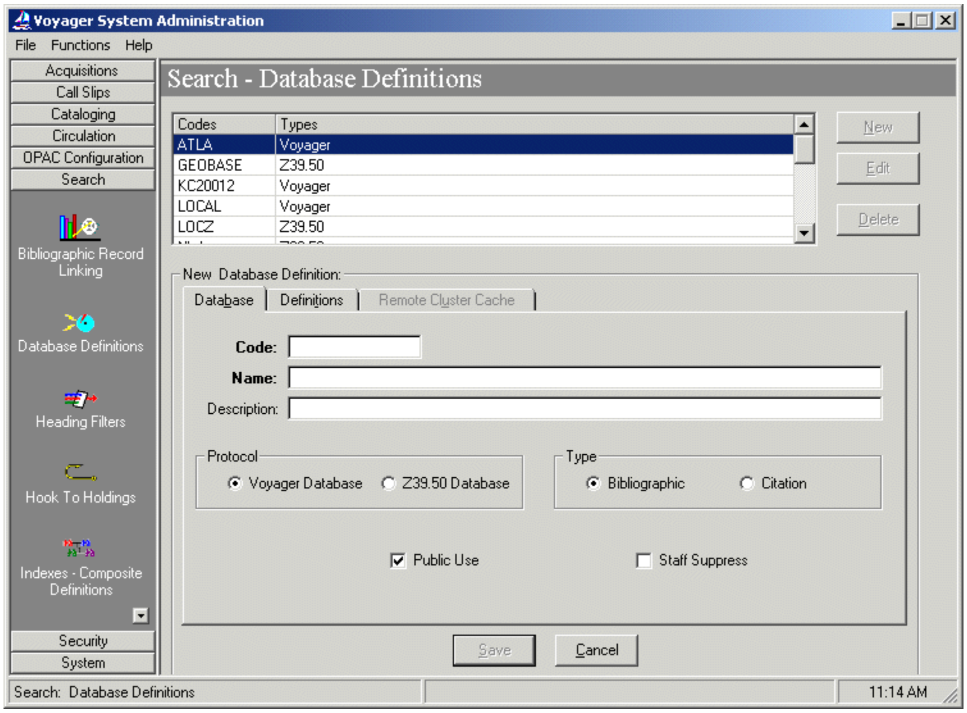
   Result: The **Search** options display in the listbar.

3. Click **Database Definitions**.

   Result: A list of previously defined database definitions displays.

4. Click **New** to create a database definition for one of the local libraries to which the UC system will connect for detailed holdings and item information.

   Result: The **New Database Definition** group box opens. See Figure 2-4.



Codes	Types
ATLA	Voyager
GEOBASE	Z39.50
KC20012	Voyager
LOCAL	Voyager
LOCZ	Z39.50

Code:

Name:

Description:

Protocol: ● Voyager Database ○ Z39.50 Database

Type: ● Bibliographic ○ Citation

☑ Public Use ☐ Staff Suppress

Figure 2-4. New Database Definition group box

5. Enter the **Database** tab options to match your requirements. See Table 2-6 for more information about each field on the **Database** tab.

   Result: This completes the definition of the **Database** tab portion of a Database Definition.

Table 2-6. Database tab options

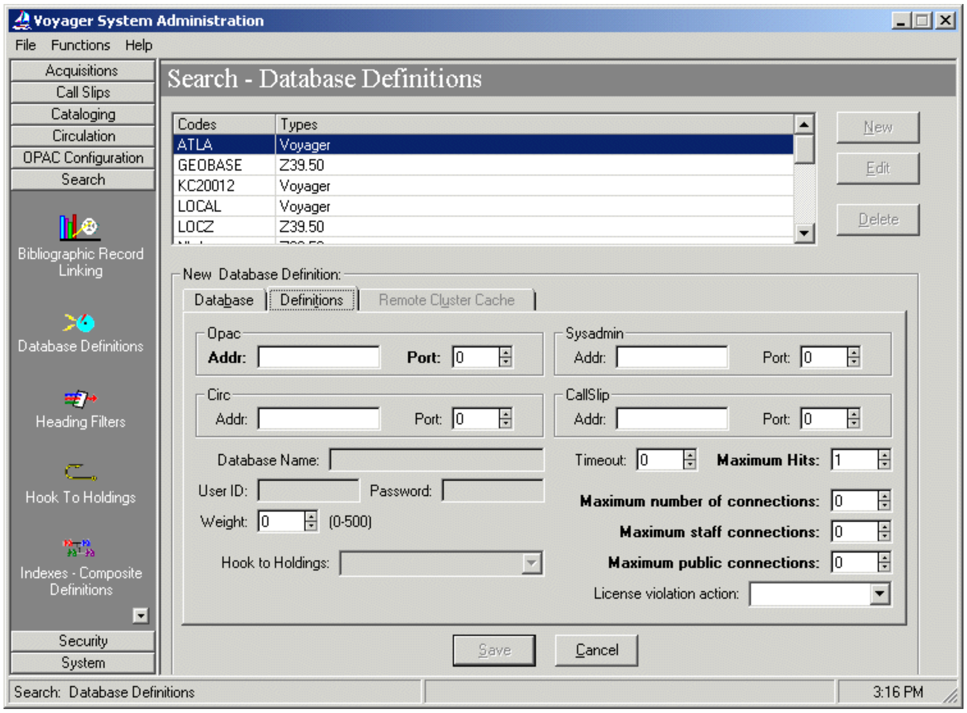
Options	Description
Code	Enter the Code for the database definition up to 8 characters. This code needs to be unique for each local database definition created.
Name	Enter the Name for the database definition up to 100 characters in length.

Table 2-6. Database tab options

Options	**Description**
Description	Enter a description for this database definition up to 200 characters in length. The description is for your reference to assist in differentiating database definitions.
Protocol	Select Voyager Database to identify the protocol for the database definition.
Type	Select the Bibliographic option to identify the database as a bibliographic database.
Staff Suppress	Select Staff Suppress to hide the database definition from view for remote searches in Cataloging (does not display in the remote search list of database definitions).

6. Click the **Definitions** tab.

   Result: The **Definitions** tab options display. See Figure 2-5.



Search - Database Definitions

Codes	Types
ATLA	Voyager
GEOBASE	Z39.50
KC20012	Voyager
LOCAL	Voyager
LOCZ	Z39.50

New Database Definition:

Opac Addr: Port: 0

Sysadmin Addr: Port: 0

Circ Addr: Port: 0

CallSlip Addr: Port: 0

Database Name:

Timeout: 0 Maximum Hits: 1

User ID: Password:

Maximum number of connections: 0

Weight: 0 (0-500)

Maximum staff connections: 0

Maximum public connections: 0

Hook to Holdings:

License violation action:

Search: Database Definitions 3:16 PM

Figure 2-5. Definitions tab (Database Definitions)

7. Enter the **Definitions** tab options to match your requirements. See Table 2-7 for more information about each field on the **Definitions** tab.

   Result: This completes the **Definitions** tab portion of a Database Definition.

Table 2-7. Definitions tab options

Options	Description
OPAC	Enter the IP **Address (Addr)** and **Port** information for the local database you are defining. In general, the OPAC **Addr** and **Port** fields are the only fields that need to be completed. However, the **Addr** and **Port** information needs to be entered for all four modules when Universal Borrowing is implemented. Port suffixes usually follow this format. The *XX* represents the port prefix. *XX*00

Table 2-7. Definitions tab options

Options	**Description**
Sysadmin	Enter the IP **Address (Addr)** and **Port** information for the database you are defining. In general, the OPAC **Addr** and **Port** fields are the only fields that need to be completed. However, the **Addr** and **Port** information needs to be entered for all four modules when Universal Borrowing is implemented. Port suffixes usually follow this format. The *XX* represents the port prefix. *XX*50
Circ	Enter the IP **Address (Addr)** and **Port** information for the database you are defining. In general, the OPAC **Addr** and **Port** fields are the only fields that need to be completed. However, the **Addr** and **Port** information needs to be entered for all four modules when Universal Borrowing is implemented. Port suffixes usually follow this format. The *XX* represents the port prefix. *XX*30
CallSlip	Enter the IP **Address (Addr)** and **Port** information for the database you are defining. In general, the OPAC **Addr** and **Port** fields are the only fields that need to be completed. However, the **Addr** and **Port** information needs to be entered for all four modules when Universal Borrowing is implemented. Port suffixes usually follow this format. The *XX* represents the port prefix. *XX*80
Database Name User ID Password	Enter the **Database Name** when defining a Z39.50 database connection. A **User ID** and **Password** may also need to be entered if required by the target server. **NOTE:** These options are inactive because Voyager Database is checked as the database protocol on the Database tab of Database Definitions. See Figure 2-4.

Table 2-7. Definitions tab options

Options	Description
Weight	Leave the database Weight amount at its default setting. Normally, Weight is used to prioritize database search results. For the Universal Catalog system database prioritization is handled through the ongoing bulk import process.
Hook to Holdings	Enter the **Hook to Holdings** information when defining a citation database connection. This displays as an inactive option in Figure 2-5 because the Voyager Database protocol was selected in Figure 2-4.
Timeout	Enter a value greater than zero for the Timeout amount. This determines the number of seconds that the UC system has to attempt a connection to the target local database.
Maximum Hits	Enter a Maximum Hits amount to identify the maximum number of titles to be retrieved during a search.
Maximum number of connections	Enter the **Maximum number of connections**. Your institution may be required to conform to license restrictions regarding connections to specific remote databases. Use these options to identify the maximum number of concurrent connections you are permitted to a remote database. See the *Voyager System Administration User’s Guide* for a more detailed explanation.
Maximum staff connections	Enter the **Maximum staff connections**. Your institution may be required to conform to license restrictions regarding connections to specific remote databases. Use these options to identify the maximum number of concurrent connections you are permitted to a remote database. See the *Voyager System Administration User’s Guide* for a more detailed explanation.

Table 2-7. Definitions tab options

Options	Description
Maximum public connections	Enter the **Maximum public connections**. Your institution may be required to conform to license restrictions regarding connections to specific remote databases. Use these options to identify the maximum number of concurrent connections you are permitted to a remote database. See the *Voyager System Administration User's Guide* for a more detailed explanation.
License violation action	Enter the **License violation action**. Your institution may be required to conform to license restrictions regarding connections to specific remote databases. Use these options to identify the maximum number of concurrent connections you are permitted to a remote database. See the *Voyager System Administration User's Guide* for a more detailed explanation.

8. Click **Save** or **Cancel**.

   Result: This saves or cancels the **New Database Definition** settings.

Connect.ini

In order to connect to a database outside the Universal Catalog database, database codes (see Figure 2-4, "New Database Definition group box," on page 12) must be added to the `connect.ini` file. The `connect.ini` file works in conjunction with Database Definitions in Voyager System Administration to specify the connections (Voyager, Z39.50, Citation) that display on the Select Database page in WebVoyáge.

The `connect.ini` file is located in the `/m1/voyager/`*`xxx`*`db/etc/webvoyage` directory. In Figure 2-6, KSU and ETD in the `[Voyager Libraries]` stanza are examples of database codes. You must add the database code for each of the databases defined in Voyager System Administration to the `connect.ini` file before you can connect to the databases from WebVoyáge.

Databases do not display on the Select Database page unless defined in the `connect.ini` file. There is no limit to the number of databases that may be listed in the `connect.ini` file. Figure 2-6 is an example of one way in which the `connect.ini` can be structured.

```
[Folders]
1=Voyager Libraries
2=Some Z3950 Sites

[Voyager Libraries]
1=KSU
2=ETD
3=Groups of Libraries

[Some Z3950 Sites]
1=DRAKE
2=LC

[ETD]
Config=ETDConfiguration
Bitmap=ETDLogo.bmp

[Groups of Libraries]
Type=G
Name=Groups of Voyager Libraries

[List Groups of Libraries]
1=UAB
2=LACLL
```

Figure 2-6. Structure of Sample connect.ini file

For more information regarding the `connect.ini` file, see the *Voyager WebVoyáge User’s Guide*.

UC Default Address

You must define a default address for the Universal Catalog database. The **Library Name** portion of this address is actually what displays in WebVoyáge when you are searching the UC database.

The procedure for defining a default address is shown in Procedure 2-4, Defining a Default Address, on page 2-19.

Procedure 2-4. Defining a Default Address

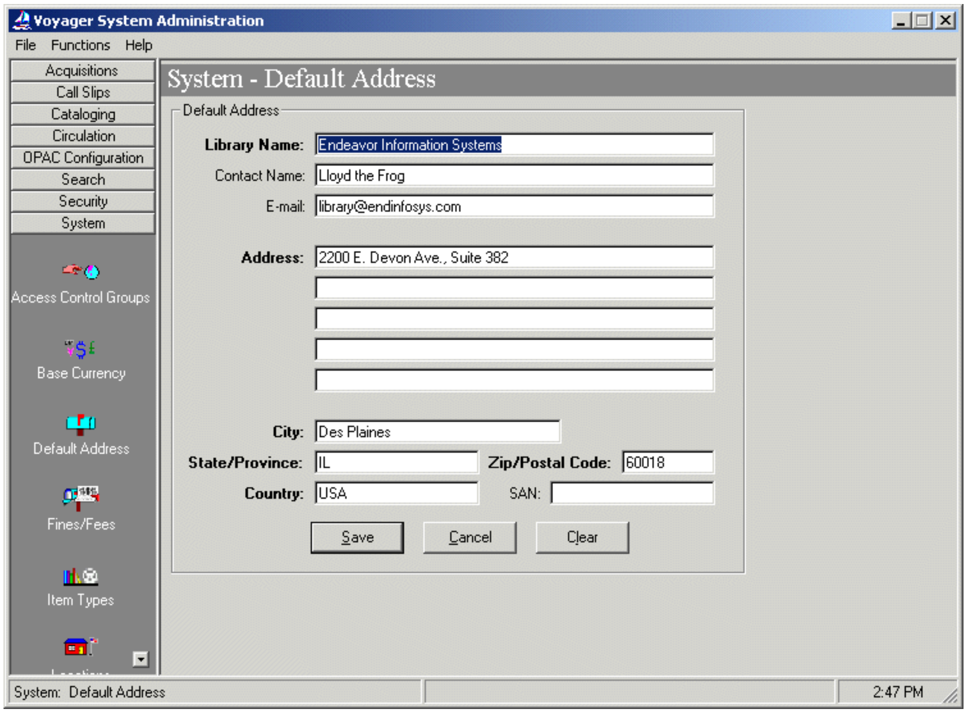
Use the following to define a default address for the UC database.

1. Click **System** from the listbar in Voyager System Administration.

   Result: The **System** configuration options display within the listbar.

2. Click **Default Address**.

   Result: The Default Address dialog box opens. See Figure 2-7.



System - Default Address

Default Address

Library Name: Endeavor Information Systems

Contact Name: Lloyd the Frog

E-mail: library@endinfosys.com

Address: 2200 E. Devon Ave., Suite 382

City: Des Plaines

State/Province: IL

Zip/Postal Code: 60018

Country: USA

SAN:

System: Default Address

Figure 2-7. Default Address dialog box

3. Enter the **Default Address** information to match your requirements. See Table 2-8 for more information about each field in the **Default Address** dialog box.

Result: This completes the **Default Address** definition.

Table 2-8. Fields in the System - Default Address work space

Name	Description	Required	Type and Range
Library Name	This is the name that displays in WebVoyáge when you are searching the Universal Catalog.	Yes	Alphanumeric, punctuation, and spaces. Uppercase and lowercase allowed. 50 characters.
Contact Name	Normally, this contact name is used as a default on acquisitions and circulation notices. However, this does not apply to the UC system. These activities are handled by the local institutions.	No	Alphanumeric, punctuation, and spaces. Uppercase and lowercase allowed. 50 characters.
E-Mail	This is the default e-mail address.	No	50 characters.
Address (5 lines)	The address to be used as a default on acquisitions and circulation notices. However, this does not apply to the UC system. These activities are handled by the local institutions.	Yes You must enter data on at least the first line of the address in order to save the Default Address successfully.	Alphanumeric, punctuation, and spaces. Uppercase and lowercase allowed. 50 characters per line.
City	Name of the city used in the default address.	Yes	Alphanumeric, punctuation, and spaces. Uppercase and lowercase allowed. 30 characters.
State / Province	Name of the state/province used in the default address.	Yes	Alphanumeric, punctuation, and spaces. Uppercase and lowercase allowed. 7 characters.

Table 2-8. Fields in the System - Default Address work space

Name	Description	Required	Type and Range
Zip / Postal Code	Zip code or postal code of the city used in the default address.	Yes	Numeric. 10 characters.
Country	Name of the country used in the default address. Users may use the codes provided in *Codes for the Representation of Names of Countries*, ISO/ANSI/NISO 3166, distributed by the American National Standards Institute. *Reference website: http://www.din.de/gremien/nas/nabd/iso3166ma/index.html.*	Yes	Alphanumeric, punctuation, and spaces. Uppercase and lowercase allowed. 20 characters.
SAN (Standard Address Number)	In the U.S., SANs are assigned by R. R. Bowker. Most SANs are eight characters but two-character prefixes are sometimes used. For more information regarding SANs, see http://www.bowker.com/standards/.	No	Alphanumeric, punctuation, and spaces. Uppercase and lowercase allowed. 10 characters.

4. Click **Save**, **Cancel**, or **Clear**.

   Result: This saves, cancels, or clears the **Default Address** information.

Bibliographic Duplicate Detection Profile

Voyager Bulk Import functionality is used during the load and maintenance of records in the UC database. See Building and Maintaining the UC on page 3-5, and see Ongoing Updating of UC Records on page 3-30. To prevent and monitor the creation of duplicate records during the Bulk Import process, you must define a bibliographic duplicate detection profile for the UC database in the Voyager System Administration module.

The bibliographic duplicate detection profile is used by the UC system to compare incoming records with existing ones. The purpose of this process is to identify duplicate records. When a duplicate is found, it is handled based on configurations set in the duplicate detection profile.

In general, it is only necessary to create and use one bibliographic duplicate detection profile. This ensures that the same duplicate detection configurations are universally and consistently applied to imported records.

NOTE:
If multiple owning libraries are being implemented, additional duplicate detection profiles are required. See Multiple Owning Libraries on page B-1 for more information.

IMPORTANT:
Once the duplicate detection profile is established, you should not change it. Changing the bibliographic duplicate detection profile at any time throughout the life of the UC database can cause serious inconsistencies and repercussions with records (that is, multiple duplicates). The only exception to this rule is if you add a new participating library to the UC system at which point it is possible, but not required, that the new library database is added to the Quality Hierarchy of the bibliographic duplicate detection profile. See How Records from New Libraries are Loaded on page 3-37 for details. In all other cases, the profile can only be safely altered or replaced when the Universal Catalog system is completely re-built.

Creating Bibliographic Duplicate Detection Profiles

The procedure for creating bibliographic duplicate detection profiles is shown in Procedure 2-5, Creating Bibliographic Duplicate Detection Profiles, on page 2-22.

Procedure 2-5. Creating Bibliographic Duplicate Detection Profiles

Use the following to create a UC bibliographic duplicate detection profile.

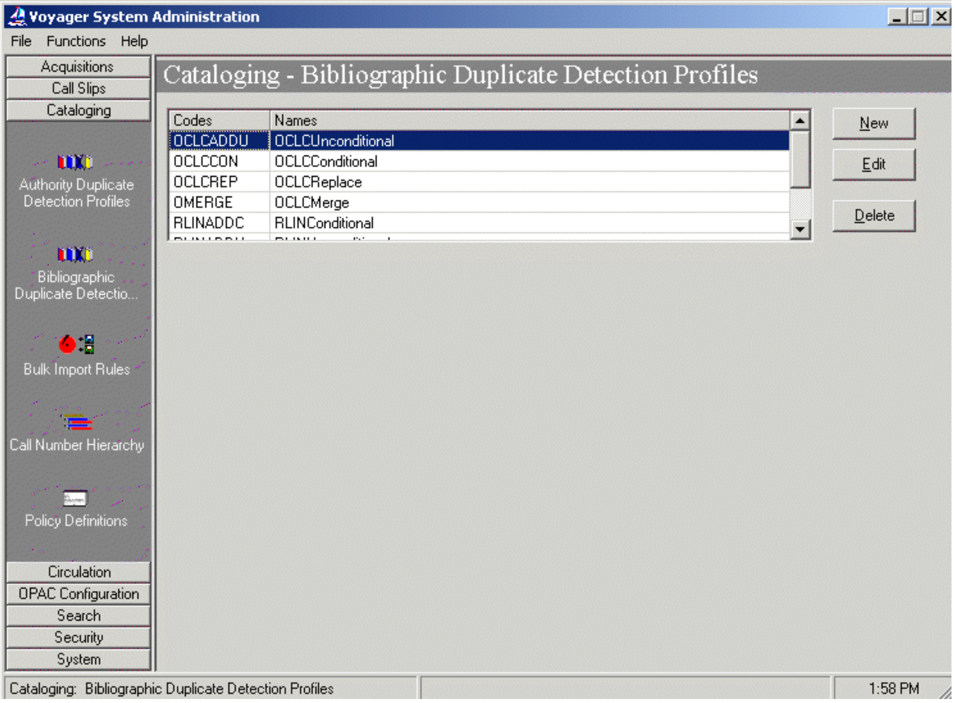
NOTE:
This procedure describes the suggested setup. Other options may be selected. Discuss your setup options with your Project Manager to determine the best choices for your site.

1. Click **Cataloging** from the vertical listbar in Voyager System Administration.

Result: The Voyager System Administration cataloging options display in the listbar.

2. Click **Bibliographic Duplicate Detection Profiles** from the listbar.

   Result: The **Cataloging - Bibliographic Duplicate Detection Profiles** dialog box opens. See Figure 2-8.

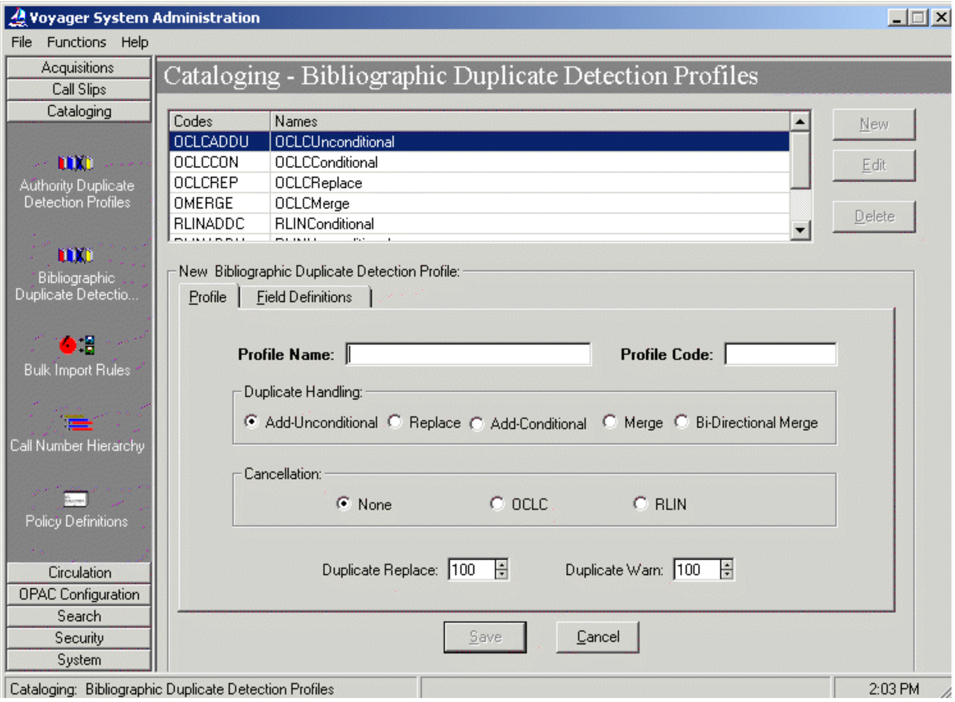

Voyager System Administration
File
Functions
Help
Acquisitions
Call Slips
Cataloging
Authority Duplicate Detection Profiles
Bibliographic Duplicate Detectio...
Bulk Import Rules
Call Number Hierarchy
Policy Definitions
Circulation
OPAC Configuration
Search
Security
System
Cataloging - Bibliographic Duplicate Detection Profiles
New
Edit
Delete
Cataloging: Bibliographic Duplicate Detection Profiles
1:58 PM

Codes	Names
OCLCADDU	OCLCUnconditional
OCLCCON	OCLCConditional
OCLCREP	OCLCReplace
OMERGE	OCLCMerge
RLINADDC	RLINConditional

Figure 2-8.   Cataloging - Bibliographic Duplicate Detection Profiles

3. Click **New.**

   Result: The **New Bib Duplicate Detection Profile** options display. See Figure 2-9.



Cataloging - Bibliographic Duplicate Detection Profiles

Codes	Names
OCLCADDU	OCLCUnconditional
OCLCCON	OCLCConditional
OCLCREP	OCLCReplace
OMERGE	OCLCMerge
RLINADDC	RLINConditional

New Bibliographic Duplicate Detection Profile:

Profile Name:

Profile Code:

Duplicate Handling: ● Add-Unconditional ○ Replace ○ Add-Conditional ○ Merge ○ Bi-Directional Merge

Cancellation: ● None ○ OCLC ○ RLIN

Duplicate Replace: 100 Duplicate Warn: 100

Cataloging: Bibliographic Duplicate Detection Profiles 2:03 PM

Figure 2-9. New Bibliographic Duplicate Detection Profile options

4. Enter the options on the **Profile** tab to match your requirements. See Table 2-9 for more information about each field on the **Profile** tab.

   Result: This completes one part of the setup for a **Bibliographic Duplicate Detection Profile**.

Table 2-9. Profile tab options - Bibliographic Duplicate Detection

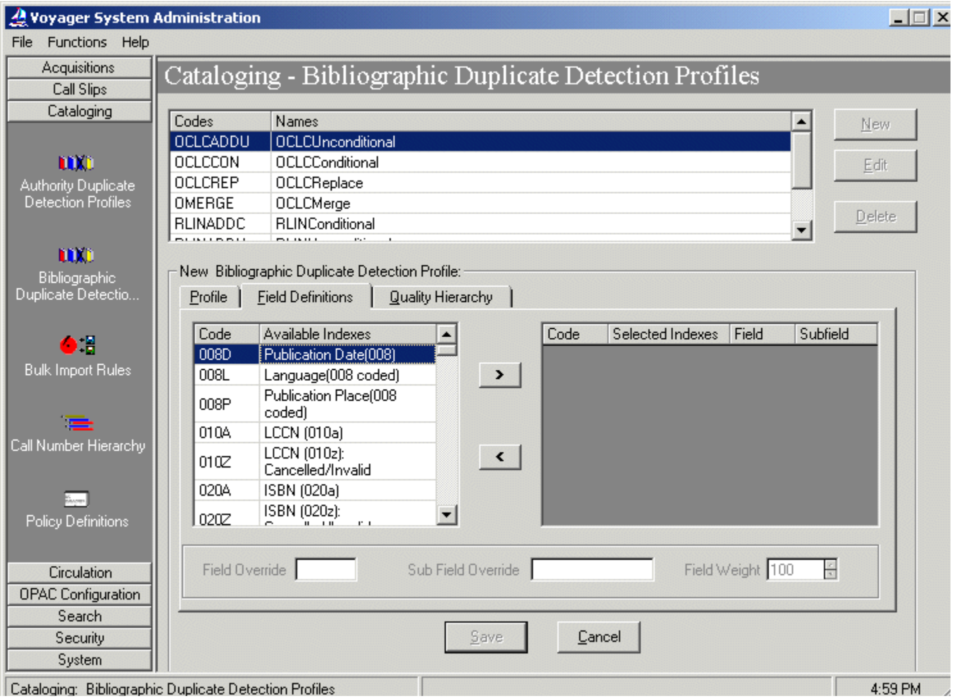
Options	Description
Profile Name	25-character maximum length (alphanumeric, punctuation, and spaces). Mixed case may be used for alpha characters.
Profile Code	8-character maximum length. Mixed case may be used for alpha characters.

Table 2-9. **Profile tab options - Bibliographic Duplicate Detection**

Options	Description
Duplicate Handling	Select Replace. Specifies that when an incoming record matches an existing Voyager record the incoming record replaces the existing Voyager record (if the quality of the incoming record meets or surpasses the values specified for Duplicate Warn/Replace and Field Weight) and retains the Voyager record number. If the system does not find a match, the incoming record is added to the database. If more than one existing record is matched (duplicate bibliographic records in the database), the system does not replace any record and the imported record is written to the discard file. **OPTIONAL:** *Bi-Directional Merge may be selected in order to create a "super" bibliographic record. If the incoming bibliographic record fails the quality hierarchy, BulkImport is able to copy protected fields from the incoming bibliographic record before it is discarded and add them to the existing bibliographic record if this option is selected.* *Or if the incoming bibliographic record becomes the preferred bibliographic record determined by the quality hierarchy, BulkImport is able to retain protected fields from the existing bibliographic record and add them to the incoming bibliographic record. Before implementing this option, talk with your Project Manager to identify any special considerations for your site.* **NOTE:** There are many components involved with the processing of records and Duplicate Handling. For more information about the logic in duplicate detection processing, see the *Voyager System Administration User's Guide*.
Cancellation	Select None.
Duplicate Replace	Set this value to 100.
Duplicate Warn	Set this value to 100.

5. Click the **Field Definitions** tab.

   Result: The Field Definitions options display. See Figure 2-10.



Cataloging - Bibliographic Duplicate Detection Profiles

Codes	Names
OCLCADDU	OCLCUnconditional
OCLCCON	OCLCConditional
OCLCREP	OCLCReplace
OMERGE	OCLCMerge
RLINADDC	RLINConditional

New Bibliographic Duplicate Detection Profile:

Code	Available Indexes
008D	Publication Date(008)
008L	Language(008 coded)
008P	Publication Place(008 coded)
010A	LCCN (010a)
010Z	LCCN (010z): Cancelled/Invalid
020A	ISBN (020a)
020Z	ISBN (020z):

Code	Selected Indexes	Field	Subfield

Field Override

Sub Field Override

Field Weight 100

Cataloging: Bibliographic Duplicate Detection Profiles

Figure 2-10. Field Definitions tab - Bibliographic Duplicate Detection

6. Select the following indexes to use to match incoming records with existing records for duplicate detection, and enter the **Field/Sub Field Override** and **Field Weight** options as indicated in Table 2-10.

   - 0350 System Number for de-duping (035a)
   - 010A LCCN (010a)
   - 022A ISSN (022a)
   - 020A ISBN (020a)

     (or 020N, 020R, or 020Z)

   The indexes should be selected in the hierarchical order the system is to use them for matching records.

   Result: Selected indexes display in the **Code/Selected Indexes** columns and the **Field/Sub Field Override** or **Field Weight** information displays on the **Field Definitions** tab.

The indexes found on the **Field Definitions** tab are used to match incoming bibliographic records with ones that already exist in the UC database. These indexes are defined in the Search Configuration component of the Voyager System Administration module.

The indexes you add should be kept simple and should not be changed at all throughout the life of the UC database.

Weights can be given to each index, indicating its importance in matching incoming bibliographic records.

TIP:

Carefully select the 020A, 020N, 020R, or 020Z indexes. The 020A index stores ISBN data with or without parenthetical information; and the 020Z index stores cancelled/invalid ISBN data with or without parenthetical information.

The 020N index stores the data from the 020A index minus any parenthetical information. The 020R stores the data from the 020Z index minus any parenthetical information.

The 020 indexes are evaluated for exact matches. If your intent is to validate an exact match on only an ISBN number (not parenthetical information), use the 020N or 020R indexes. This can be extremely useful when working with Universal Catalog databases.

RECOMMENDED:

The 020N and 020R indexes should not be selected concurrently with the 020A and 020Z indexes.

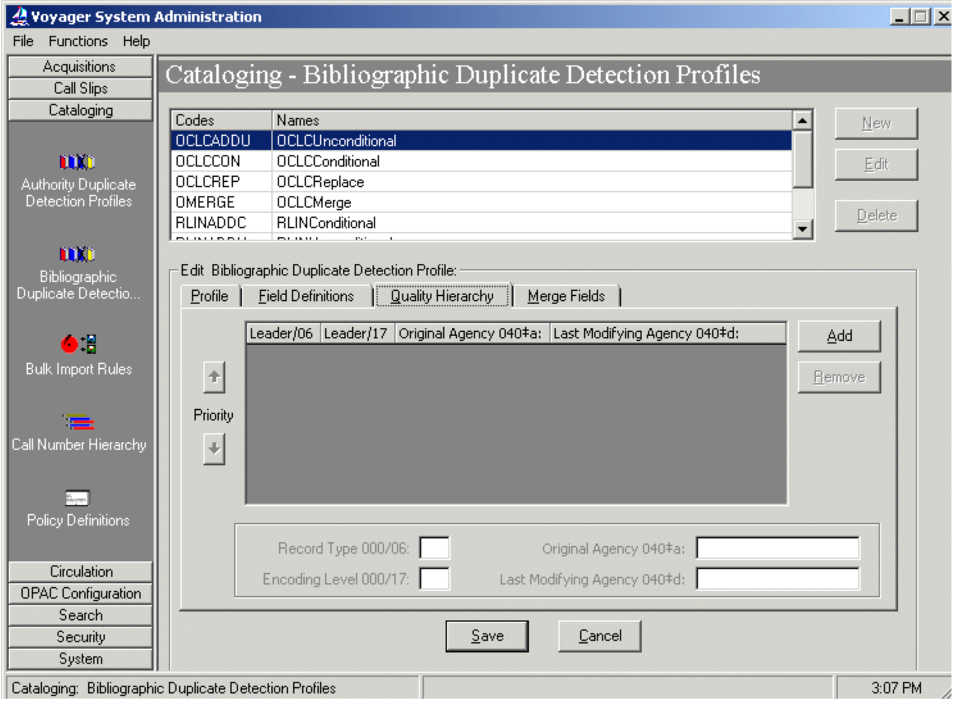
Table 2-10. Field/Sub Field Override and Field Weight options

Options	Description
Field Override	Leave this field blank.
Sub Field Override	Leave this field blank.
Field Weight	Set this value to 100 for the 0350, 022A, and 020A indexes. Set this value to 50 for the 010A index.

7. Click the **Quality Hierarchy** tab (see Figure 2-11) and enter the Quality Hierarchy options (see Table 2-11) that match your requirements of your Universal Catalog. The quality hierarchy determines whether the incoming Bibliographic record or the existing UC record should be kept in the database. The quality hierarchy allows you to specify which record formats are preferable.

   Example: You may want all records for monographs that are modified and then contributed to the UC by a specific participating library database to take precedence over an existing UC record. In this case, you set the 040‡a (NUC code) and 000/17 fields (encoding level) to wildcards, and enter the modifying agency code (040‡d) and the value such as ‘am’ for monographs for record type (000/06).

   Result: This completes the **Quality Hierarchy** tab portion of the setup for a **Bibliographic Duplicate Detection Profile**.



Cataloging - Bibliographic Duplicate Detection Profiles

Codes	Names
OCLCADDU	OCLCUnconditional
OCLCCON	OCLCConditional
OCLCREP	OCLCReplace
OMERGE	OCLCMerge
RLINADDC	RLINConditional

Edit Bibliographic Duplicate Detection Profile:

Leader/06	Leader/17	Original Agency 040‡a:	Last Modifying Agency 040‡d:

Record Type 000/06:
Original Agency 040‡a:
Encoding Level 000/17:
Last Modifying Agency 040‡d:

Cataloging: Bibliographic Duplicate Detection Profiles 3:07 PM

Figure 2-11. Quality Hierarchy tab - Bibliographic Duplicate Detection

Table 2-11. Quality Hierarchy tab options

Options	Description
Record Type 000/06	Leader 6-7 fields.
Encoding Level 000/17	The institution’s leader value.
Original Agency 040‡a	The create institution’s unique identifying code (040‡a).
Last Modifying Agency 040‡d	The modifying institution’s identifying code (040‡d).

Use the **Add** and **Remove** buttons to add and delete rows on the **Quality Hierarchy** tab.

Use the up/down arrow buttons to arrange the **Quality Hierarchy** rows in the order you prefer.

8. Click **Save** or click **Cancel**.

   Result: This saves or cancels the entire profile.

Authority Duplicate Detection Profile

RECOMMENDED:
It is our recommendation if you decide to include authority records in the UC that you load the entire LC Name and Subject Heading files. This eliminates potential confusion and bad references caused by having different types of authorities from several libraries in the UC.

Despite our recommendation, it is still necessary for you to establish an authority duplicate detection profile. This is because you need to load authority records each time an update set of the LC Authority Files (Name Authorities and Subject Headings) is released. The authority duplicate detection profile allows you to re-load the authority record files while preventing the creation of duplicate records in the UC database.

For UC purposes, it is only necessary to create and use one authority duplicate detection profile. This ensures that the same duplicate detection configurations are universally and consistently applied to imported records.

⚠ **IMPORTANT:**
Once this profile is established, you should not change it. Changing the authority duplicate detection profile at any time throughout the life of the UC database can cause serious inconsistencies and repercussions on records such as multiple duplicates. A profile can be safely altered or replaced only if you completely rebuild the Universal Catalog.

Creating Authority Duplicate Detection Profiles

The procedure for creating authority duplicate detection profiles is shown in Procedure 2-6, Creating Authority Duplicate Detection Profiles, on page 2-30.

Procedure 2-6. Creating Authority Duplicate Detection Profiles

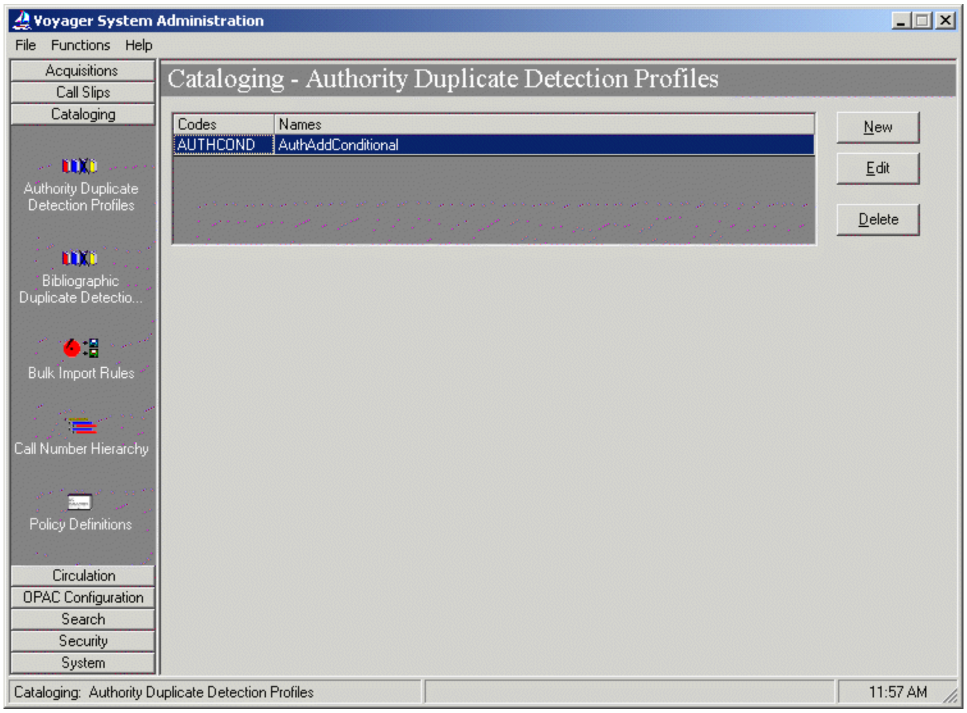
Use the following to create authority duplicate detection profiles.

1. Click **Cataloging** in the vertical listbar in Voyager System Administration.

   Result: The Voyager System Administration cataloging options display in the listbar.

2. Click **Authority Duplicate Detection Profiles**.

   Result: The **Cataloging - Authority Duplicate Detection Profiles** dialog box opens. See Figure 2-12.

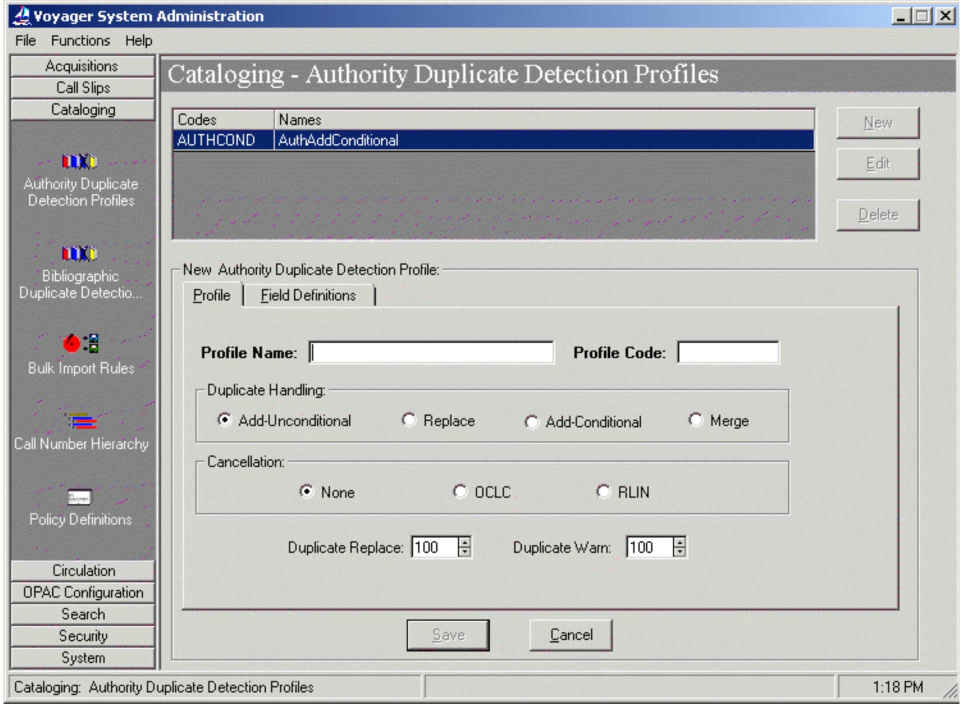

Voyager System Administration
File
Functions
Help
Acquisitions
Call Slips
Cataloging
Authority Duplicate Detection Profiles
Bibliographic Duplicate Detectio...
Bulk Import Rules
Call Number Hierarchy
Policy Definitions
Circulation
OPAC Configuration
Search
Security
System
Cataloging - Authority Duplicate Detection Profiles
New
Edit
Delete
Cataloging: Authority Duplicate Detection Profiles
11:57 AM

Codes	Names
AUTHCOND	AuthAddConditional

Figure 2-12. Cataloging - Authority Duplicate Detection Profiles

3. Click **New**.

   Result: The **New Authority Duplicate Detection Profile** options open. See Figure 2-13.



Cataloging - Authority Duplicate Detection Profiles

Codes	Names
AUTHCOND	AuthAddConditional

New Authority Duplicate Detection Profile:

Profile Name:

Profile Code:

Duplicate Handling: ● Add-Unconditional ○ Replace ○ Add-Conditional ○ Merge

Cancellation: ● None ○ OCLC ○ RLIN

Duplicate Replace: 100

Duplicate Warn: 100

Cataloging: Authority Duplicate Detection Profiles

1:18 PM

Figure 2-13. New Authority Duplicate Detection Profile options

4. Enter the options on the **Profile** tab to match your requirements. See Table 2-12 for more information about each field on the **Profile** tab.

   Result: This completes one part of the setup for an **Authority Duplicate Detection Profile**.

Table 2-12. Profile tab options - Authority Duplicate Detection

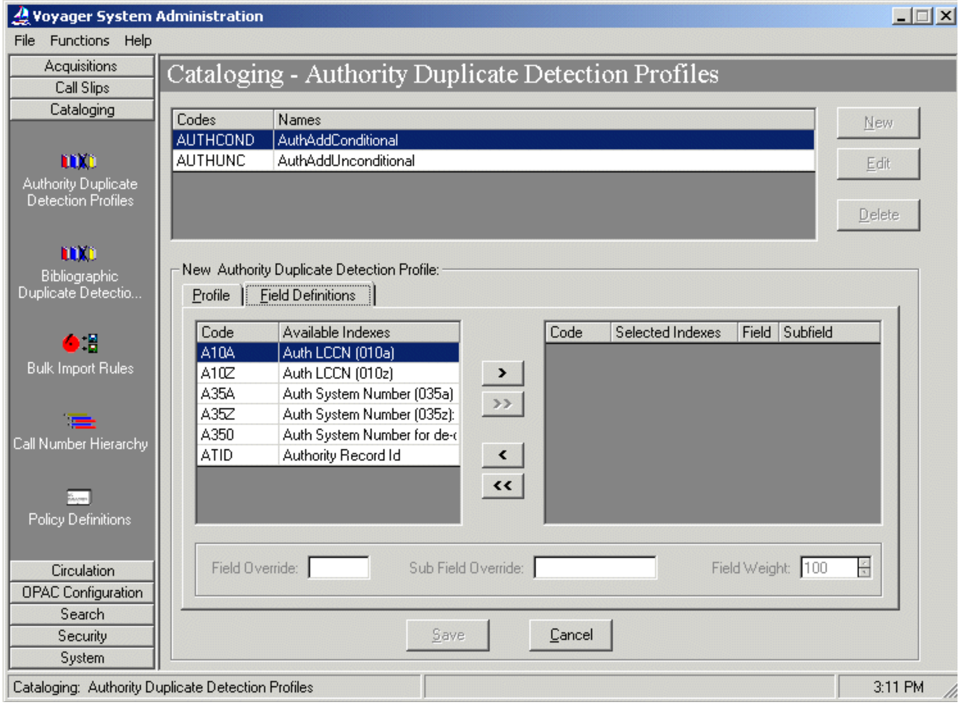
Options	Description
Profile Name	25-character maximum length (alphanumeric, punctuation, and spaces). Mixed case may be used for alpha characters.
Profile Code	8-character maximum length. Mixed case may be used for alpha characters. The Profile Code must be unique.

Table 2-12. Profile tab options - Authority Duplicate Detection

Options	Description
Duplicate Handling	Select Replace. Specifies that when an incoming record matches an existing Voyager record the incoming record replaces the existing Voyager record (if the quality of the incoming record meets or surpasses the values specified for Duplicate Warn/Replace and Field Weight) and retains the Voyager record number. If the system does not find a match, the incoming record is added to the database. If more than one existing record is matched (duplicate authority records in the database), the system does not replace any record and the imported record is written to the discard file. **NOTE:** There are many components involved in the processing of records and Duplicate Handling. For more information about the logic in duplicate detection processing, see the *Voyager System Administration User's Guide*.
Cancellation	Select None.
Duplicate Replace	Set this value to 100.
Duplicate Warn	Set this value to 100.

5. Click the **Field Definitions** tab.

   Result: The Field Definitions options display. See Figure 2-14.



Cataloging - Authority Duplicate Detection Profiles

Codes	Names
AUTHCOND	AuthAddConditional
AUTHUNC	AuthAddUnconditional

New Authority Duplicate Detection Profile:

Code	Available Indexes
A10A	Auth LCCN (010a)
A10Z	Auth LCCN (010z)
A35A	Auth System Number (035a)
A35Z	Auth System Number (035z):
A350	Auth System Number for de-(
ATID	Authority Record Id

Code	Selected Indexes	Field	Subfield

Field Override:
Sub Field Override:
Field Weight: 100

Cataloging: Authority Duplicate Detection Profiles 3:11 PM

Figure 2-14. Field Definitions tab - Authority Duplicate Detection

6. Select the following indexes to use to match incoming records with existing records for duplicate detection, and enter the **Field/Sub Field Override** and **Field Weight** options as indicated in Table 2-13.

   - A10A Auth LCCN (010a)
   - A350 Auth System Number (035a)

   The indexes should be selected in the hierarchical order the system is to use them for matching records.

   The indexes found on this tab are used to match incoming authority records with ones that already exist in the UC database. These indexes are defined in the Search Configuration component of the Voyager System Administration module.

   The indexes you add should be kept simple and should not be changed at all throughout the life of the UC database.

   Weights can be given to each index, indicating its importance in matching incoming authority records.

Result: Selected indexes display in the **Code/Selected Indexes** columns and the **Field/Sub Field Override** or **Field Weight** information displays on the **Field Definitions** tab.

Table 2-13. Field/Sub Field Override and Field Weight options

Options	Description
Field Override	Leave this field blank.
Sub Field Override	Leave this field blank.
Field Weight	Set this value to 100 for the A10A and A350 indexes.

7. Click **Save** or click **Cancel**.

   Result: This saves or cancels the entire profile.

NOTE:
Since it is our recommendation that you do not include authority records from local library databases, but rather include the entire set of LC Name and Subject Headings, it is not necessary to set value on the Quality Hierarchy tab. The Quality Hierarchy is only used if a duplicate authority record is found in a database at which point it determines whether or not an incoming authority record replaces an existing one.

If you are loading the entire set of LC Name and Subject Headings, you do not need to worry about replacing individual records in the Universal Catalog. As new versions of the LC Name and Subject Headings are released, you simply overlay all of your existing authority records.

Bulk Import Rules

Since Voyager Bulk Import functionality is used during the load and maintenance of records in the UC database, you must establish a Bulk Import Rule. For UC purposes it is only necessary to create and use one Bulk Import Rule unless you are loading authority records as well as bibliographic records and MFHDs at which point you should establish a separate rule for authority records. Having only one Bulk Import Rule ensures that the same configurations are universally and consistently applied to all the records imported into the UC database.

⚠ **IMPORTANT:**
Once the Bulk Import Rule is established, you should not change it. This ensures that records are subject to the same rules regardless of when or how they are loaded into the UC system. Failure to do so can cause serious inconsistencies and repercussions for UC records such as multiple

duplicates. The only instance in which you may safely alter the Bulk Import Rule is if you are completely rebuilding the UC database.

Creating Bulk Import Rules

The procedure for creating Bulk Import Rules is shown in Procedure 2-7, Creating Bulk Import Rules, on page 2-36.

Procedure 2-7. Creating Bulk Import Rules

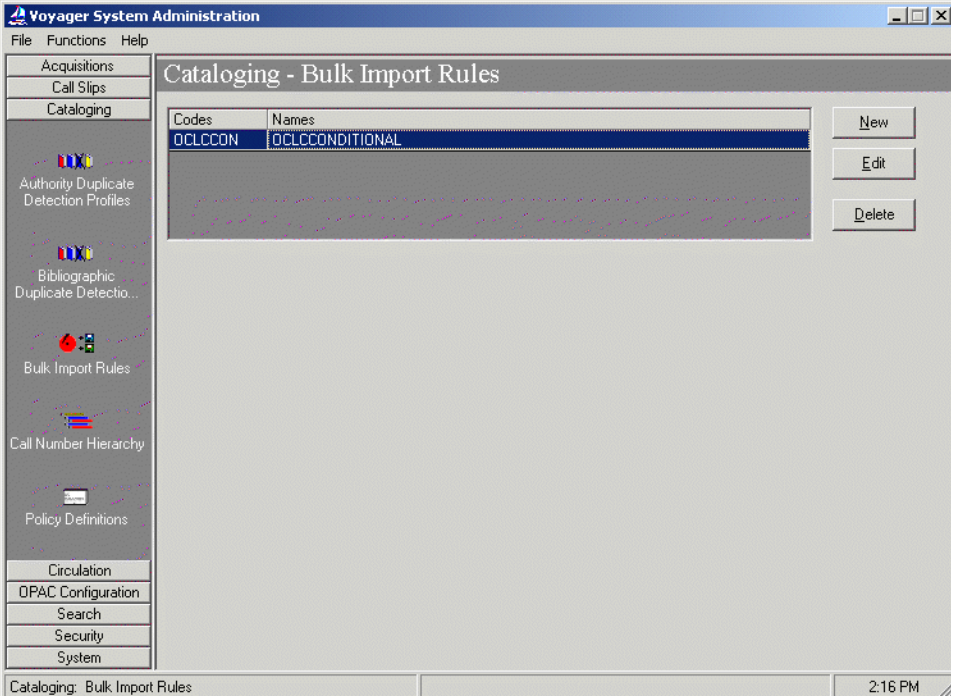
Use the following to create Bulk Import Rules.

1. Click **Cataloging** from the vertical listbar.

   Result: The Voyager System Administration cataloging options display in the listbar.

2. Click **Bulk Import Rules**.

   Result: The **Cataloging - Bulk Import Rules** dialog box opens. See Figure 2-15.

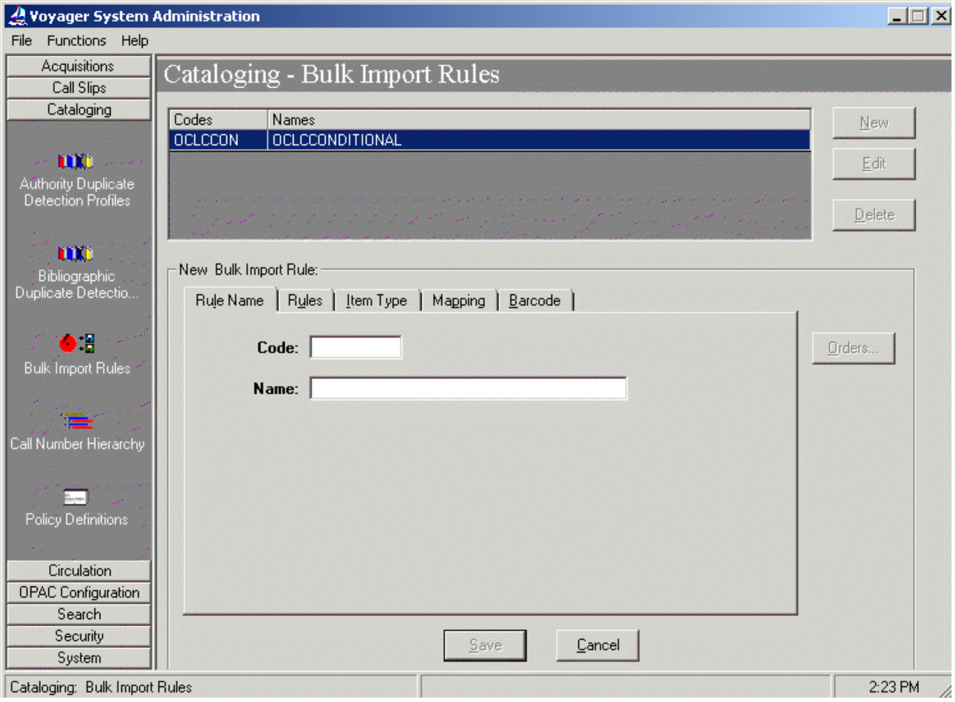


Codes	Names
OCLCCON	OCLCCONDITIONAL

Figure 2-15. Cataloging - Bulk Import Rules

3. Click **New**.

   Result: The **New Bulk Import Rule** options open. See Figure 2-16.



Codes	Names
OCLCCON	OCLCCONDITIONAL

Figure 2-16. New Bulk Import Rule options

4. Enter the **Code** and **Name** for the **Bulk Import Rule**.

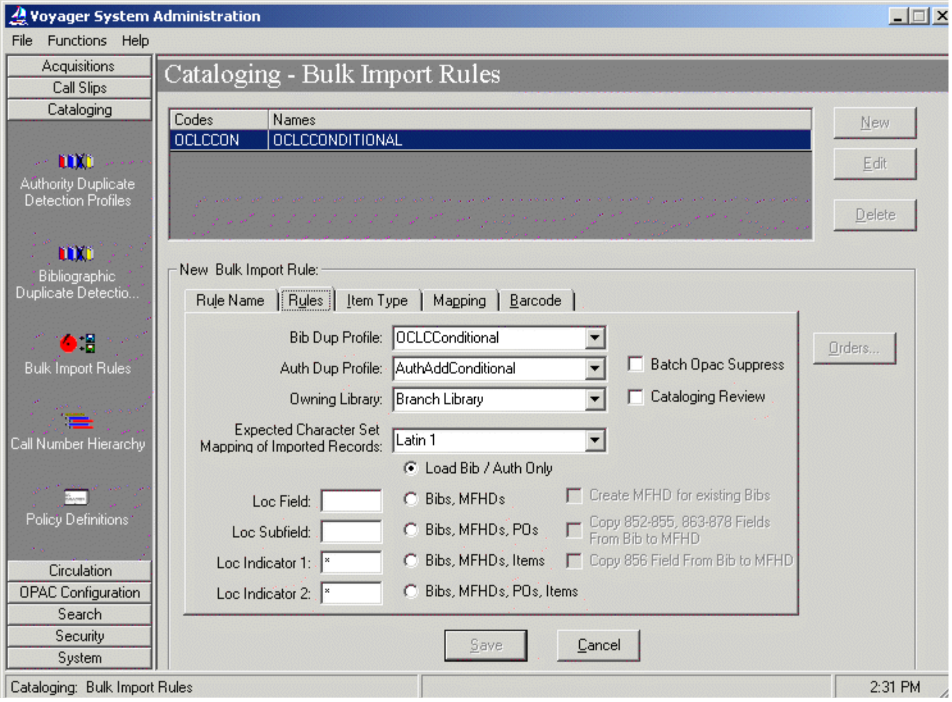
RECOMMENDED:

It is our recommendation that the name you select for the Bulk Import Rule is default. The reason for this is that during the initial load process a program is used that looks for a Bulk Import Rule code labelled “default” if one is not specified as a parameter for the program. If you call the Bulk Import Rule “default” too, you do not need to worry about specifying the rule for the program. See ucatimp.pl Program on page 3-22 for more information about the program used to bulk import records into the Universal Catalog.

Result: This identifies the new **Bulk Import Rule** that you are creating.

5. Click the **Rules** tab.

Result: The **Rules** tab options display. See Figure 2-17.



Cataloging - Bulk Import Rules

Codes	Names
OCLCCON	OCLCCONDITIONAL

New Bulk Import Rule:

Bib Dup Profile: OCLCConditional

Auth Dup Profile: AuthAddConditional

Owning Library: Branch Library

Expected Character Set Mapping of Imported Records: Latin 1

□ Batch Opac Suppress

□ Cataloging Review

● Load Bib / Auth Only

○ Bibs, MFHDs

○ Bibs, MFHDs, POs

○ Bibs, MFHDs, Items

○ Bibs, MFHDs, POs, Items

Loc Field:

Loc Subfield:

Loc Indicator 1: *

Loc Indicator 2: *

□ Create MFHD for existing Bibs

□ Copy 852-855, 863-878 Fields From Bib to MFHD

□ Copy 856 Field From Bib to MFHD

Cataloging: Bulk Import Rules

Figure 2-17. Rules tab options

6. Enter the **Rules** tab options that match your requirements for the Universal Catalog. See Table 2-14 for more information.

   Result: The **Rules** tab component of the new **Bulk Import Rule** is complete.

Table 2-14. Rules tab options (Page 1 of 2)

Option	Description
Bib Dup Profile	Select the bibliographic duplicate detection profile you created for the Universal Catalog database.
Auth Dup Profile	Select the authority duplicate detection profile you set up for the Universal Catalog database.
Owning Library	Select the owning library you set up for the Universal Catalog database.
Expected Character Set Mapping of Imported Records	Select the character set mapping option that matches the character set of the incoming records. This provides the Voyager system with the necessary information to convert incoming records to the Voyager RLIN character set.

Table 2-14. Rules tab options (Page 2 of 2)

Option	Description
Loc. Indicator 1 Loc. Indicator 2	Enter an asterisk (*) in these fields.
Load Bib/Auth Only	Select this option for the Universal Catalog database. This indicates that when bibliographic records are loaded into the database, MFHDs, items, and/or purchase orders are not created. For Universal Catalog purposes, MFHDs are created by the Prebulk program and item records and purchase orders are not included.

7. Click **Save** or click **Cancel**.

   Result: This saves or cancels the entire Bulk Import Rule.

NOTE:
The options located on the **Item Type**, **Mapping**, and **Barcode** tabs do not need to be set when the **Load Bib / Auth Only** option is selected on the **Rules** tab.

Cataloging Policy Definitions

You should establish a Cataloging Policy Definition on the UC server so that you can troubleshoot/view UC records in the Cataloging client as opposed to only in WebVoyáge. This is beneficial if you, for instance, want to compare UC records. Comparing records is easier and more detailed in the Cataloging client since you can efficiently look at two or more bibliographic records at once as well as their associated MFHD records.

IMPORTANT:
Avoid editing UC records in the UC Cataloging client. The UC database is updated, and records are changed, added, and deleted through the ongoing update process that is a part of the Universal Catalog application design. It is possible that any changes made directly to the UC database from the UC Cataloging client may be replaced by way of the ongoing update process. It is recommended, therefore, that changes be applied only to the local databases and that operators have view-only access through the UC Cataloging client.

Creating Cataloging Policy Definitions

The procedure for creating cataloging policy definitions is shown in Procedure 2-8, Creating Cataloging Policy Definitions, on page 2-41.

Procedure 2-8. Creating Cataloging Policy Definitions

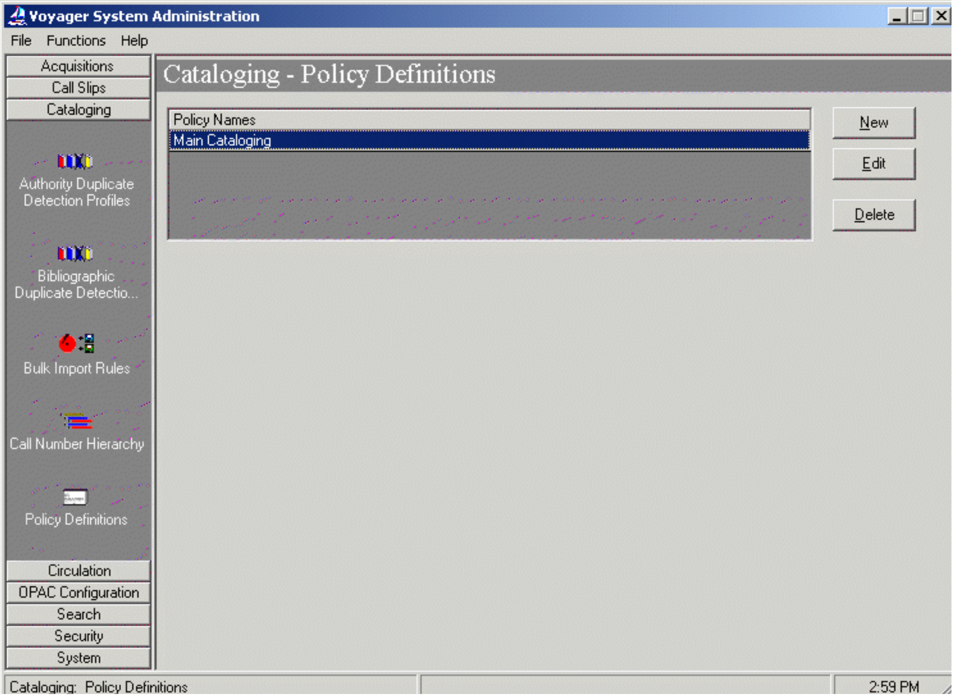
Use the following to create cataloging policy definitions.

1. Click **Cataloging** from the vertical listbar in Voyager System Administration.

   Result: The Voyager System Administration cataloging options display.

2. Click **Policy Definitions**.

   Result: The **Cataloging - Policy Definitions** dialog box opens. See Figure 2-18.



Policy Names
Main Cataloging

Cataloging: Policy Definitions

Figure 2-18. Cataloging - Policy Definitions

3. Click **New**.

   Result: The **New Policy Definition** options display. See Figure 2-19.

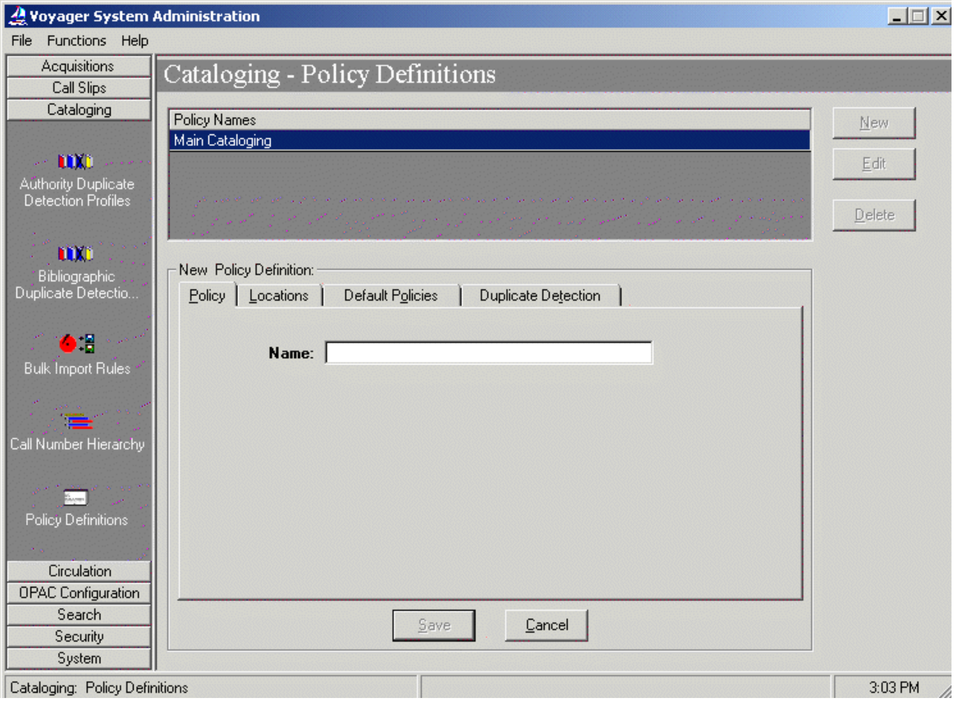


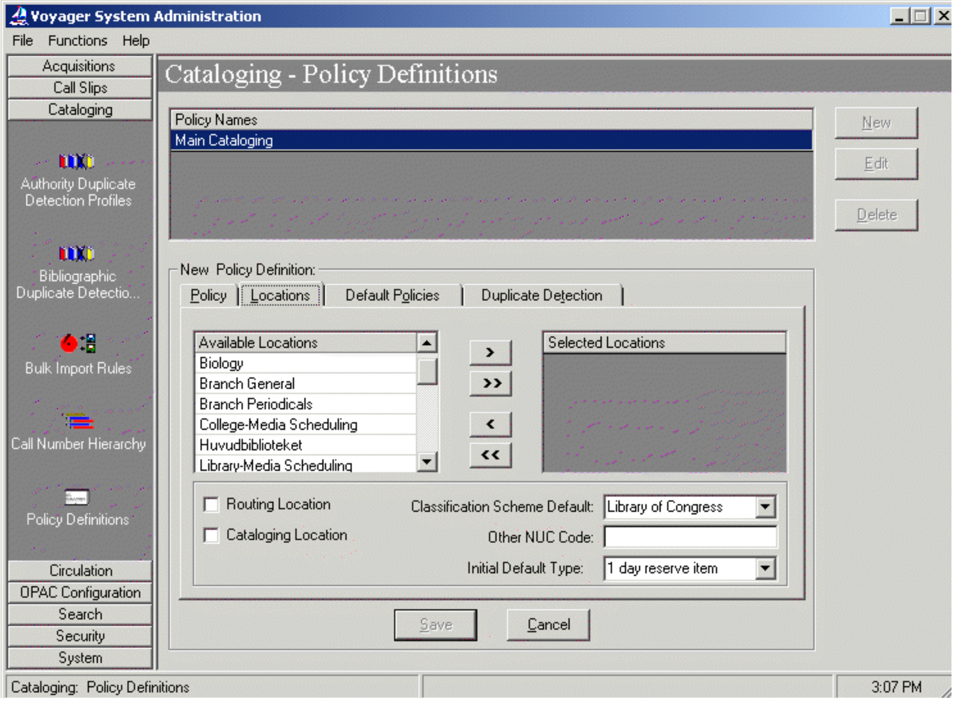
Figure 2-19. New Policy Definition options

4. Enter the **Name** on the **Policy** tab for the new policy definition. The **Name** can be up to 40 characters in length (uppercase and lowercase alphanumeric characters, punctuation, and spaces).

   Result: This name identifies the policy definition that you are creating.

5. Click the **Locations** tab.

   Result: This displays the location options for the new policy definition. See Figure 2-20.



Policy Names
Main Cataloging

Available Locations
Biology
Branch General
Branch Periodicals
College-Media Scheduling
Huvudbiblioteket
Library-Media Scheduling

Selected Locations

☐ Routing Location
☐ Cataloging Location

Classification Scheme Default: Library of Congress
Other NUC Code:
Initial Default Type: 1 day reserve item

Figure 2-20. New Cataloging - Policy Definitions Locations tab options

6. Enter the options on the **Locations** tab that match your requirements for the Universal Catalog database. See Table 2-15 for more information.

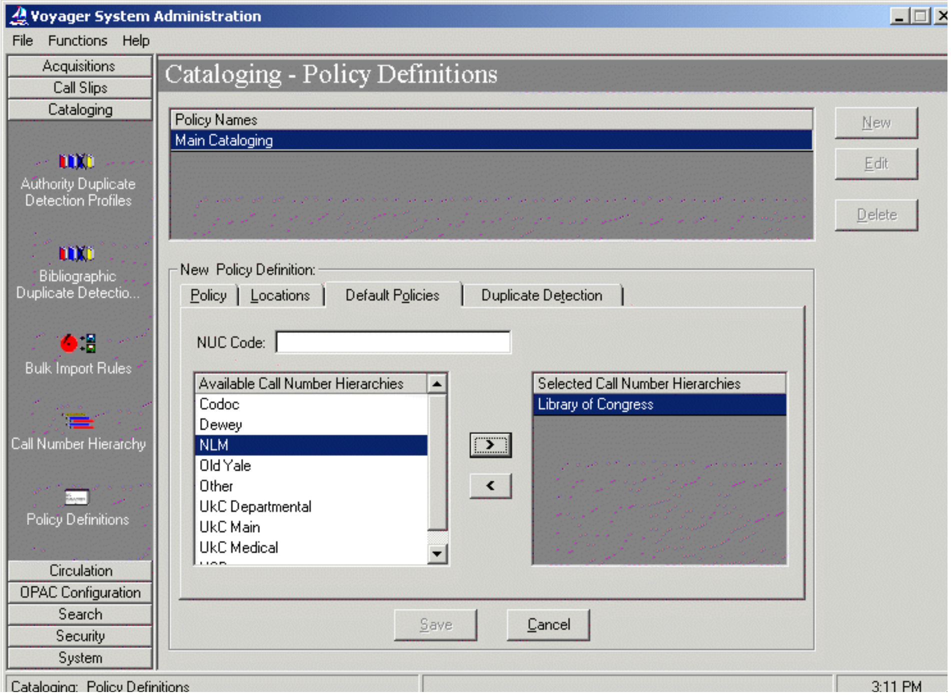
Result: This completes the locations portion of the setup for a new cataloging policy group definition.

Table 2-15. New Cataloging - Policy Definitions Locations tab options

Options	**Description**
Available/Selected Locations	Select the locations that you established for each contributing library from the Available Locations list and use the > button to move individual locations to the Selected Locations list. Use Shift + Click to highlight multiple locations in the Available Locations list and the >> button to move multiple highlighted locations to the Selected Locations list. Only locations that have not already been assigned to a cataloging policy group display in the Available Locations list. Use the < and << buttons to remove locations from the Selected Locations list. For each location selected, specify one or more of the remaining options.
Cataloging Location	Select Cataloging Location **for one of the locations in the** Selected Locations **list.** This may seem irrelevant for UC purposes since you will not be performing any cataloging operations with the UC database. However if you do not designate at least one location as a happening location, the Cataloging client does not open. See the *Voyager System Administration User’s Guide* for more information. **(c)** displays next to the location name in the **Selected Locations** list when the location is defined as a **Cataloging Location**.
Classification Scheme Default	Select a **Classification Scheme Default**.

7. Click the **Default Policies** tab.

   Result: This displays the **Default Policies** tab options for the new policy definition. See Figure 2-21.



Policy Names: Main Cataloging

NUC Code:

Available Call Number Hierarchies: Codoc, Dewey, NLM, Old Yale, Other, UkC Departmental, UkC Main, UkC Medical

Selected Call Number Hierarchies: Library of Congress

Cataloging: Policy Definitions

Figure 2-21. Default Policies tab

8. Enter the options on the **Default Policies** tab that match your requirements. See Table 2-16 for more information.

   Result: This completes the **Default Policies** tab portion of the setup for a new cataloging policy group definition.

Table 2-16. Default Policies tab options

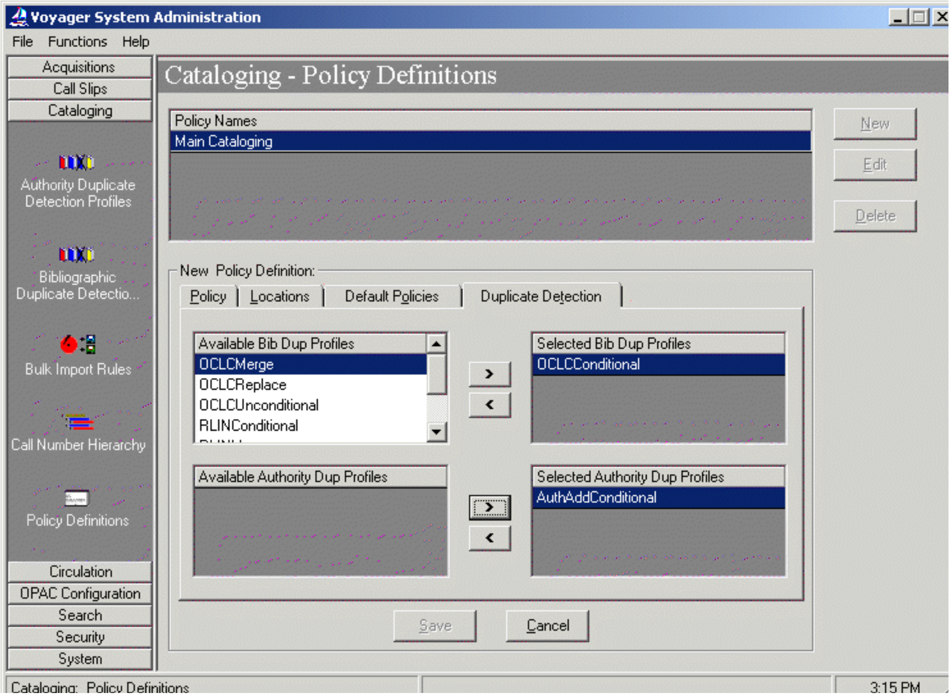
Option	Description
NUC Code	Enter the NUC Code that applies to all of the Selected Locations when exporting records. Enter an alphanumeric value. This can be an NUC code, an OCLC symbol, or any other code that is unique to the xxxdb for the UC server.

Table 2-16. Default Policies tab options

Option	Description
Available/Selected Call Number Hierarchies	Select a call number hierarchy from the Available Call Number Hierarchies list using the > button to move it to the Selected Call Number Hierarchies list. Use the < button to remove a location from the Selected Call Number Hierarchies list.

9. Click the **Duplicate Detection** tab.

   Result: This displays the **Duplicate Detection** tab options for the new policy definition. See Figure 2-22.



Policy Names: Main Cataloging

Available Bib Dup Profiles: OCLCMerge, OCLCReplace, OCLCUnconditional, RLINConditional

Selected Bib Dup Profiles: OCLCConditional

Available Authority Dup Profiles:

Selected Authority Dup Profiles: AuthAddConditional

Figure 2-22. Duplicate Detection tab

10. Enter the options on the **Duplicate Detection** tab that match your requirements. See Table 2-17 for more information.

Result: This completes the **Duplicate Detection** tab portion of the setup for a new cataloging policy group definition.

Table 2-17. Duplicate Detection tab options

Option	Description
Available/Selected Bib Dup Profiles	Select a bibliographic duplicate profile from the Available Bib Dup Profiles list using the > button to move it to the Selected Bib Dup Profiles list. Use the < button to remove a bibliographic duplicate profile from the Selected Bib Dup Profiles list. **NOTE:** Select only one bibliographic duplicate detection profile. Multiple profiles could cause inconsistencies and complications for records in the UC database.
Available/Selected Authority Dup Profiles	Select an authority duplicate detection profile from the Available Authority Dup Profiles list using the > button to move it to the Selected Authority Dup Profiles list. Use the < button to remove an authority duplicate detection profile from the Selected Authority Dup Profiles list. **NOTE:** Select only one authority duplicate detection profile. Multiple profiles could cause inconsistencies and complications for records in the UC database.

11. Click **Save** or click **Cancel**.

Result: This saves or cancels the **New Policy Definition** for **Cataloging - Policy Definitions**.

Security Setup

Security allows you to specify which functions can be performed on the Universal Catalog server, and the operators that can perform those functions. You need to establish security for the Universal Catalog System Administration module as well as the Cataloging module.

IMPORTANT:
You should establish read-only access to the Cataloging client because records should not be edited in the UC database. If you edit records in the Universal Catalog, serious inconsistencies and complications can result.

The following profiles must be setup for securing the UC database:

- Operator profiles (one for staff and one for the UC System Administrator)
- Master Profile for the UC System Administrator
- Cataloging security profile (view-only access)

Operator Profiles

Each operator or group of operators who use Voyager staff modules such as Cataloging must have an Operator Profile containing a name, operator ID, and password. Operator profiles are then associated with the appropriate security profiles for the staff modules thereby allowing operators access to the modules.

For UC purposes, you need to establish two operator profiles one for staff needing to access staff modules to look at UC records such as in Cataloging and one for the UC System Administrator.

The procedure for creating an operator profile is shown in Procedure 2-9, Creating an Operator Profile, on page 2-48.

Procedure 2-9. Creating an Operator Profile

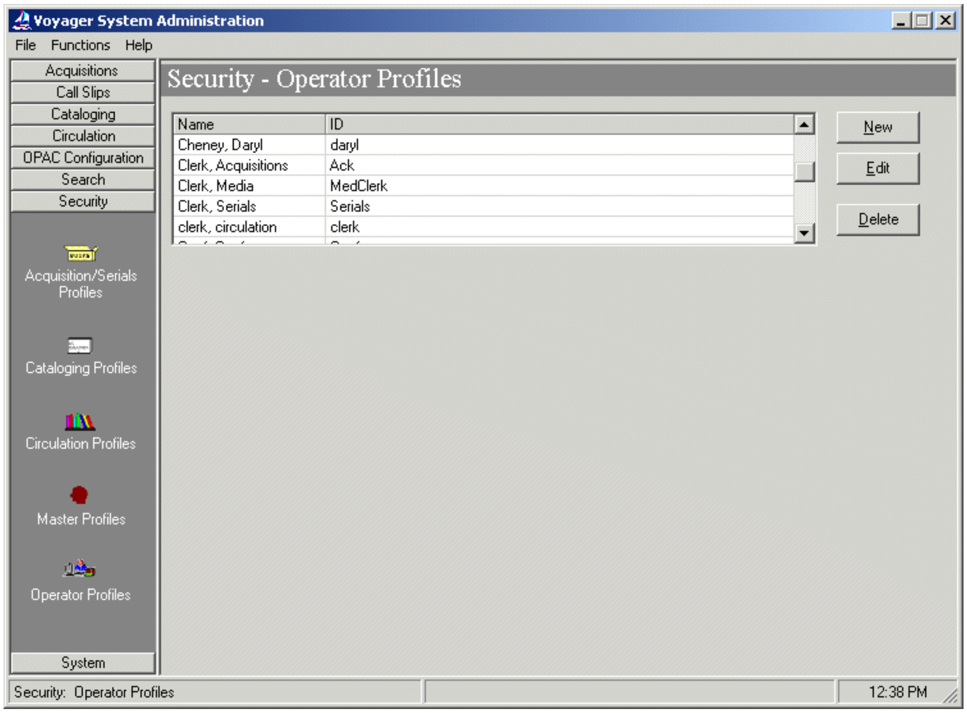
Use the following to create an operator profile.

1. Click **Security** from the vertical listbar in Voyager System Administration.

   Result: The Voyager System Administration security options display.

2. Click **Operator Profiles**.

   Result: The **Operator Profiles** dialog box opens. See Figure 2-23.

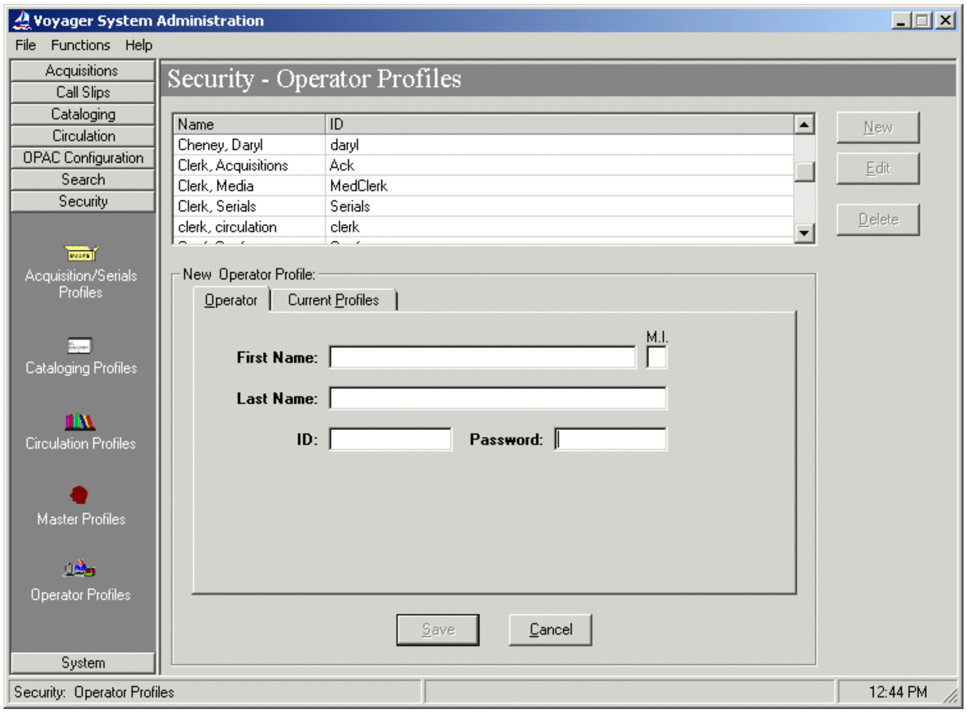


Name	ID
Cheney, Daryl	daryl
Clerk, Acquisitions	Ack
Clerk, Media	MedClerk
Clerk, Serials	Serials
clerk, circulation	clerk

Figure 2-23. Security - Operator Profiles main dialog box

3. Click **New**.

   Result: The New Operator Profile dialog box opens. See Figure 2-24.



Name	ID
Cheney, Daryl	daryl
Clerk, Acquisitions	Ack
Clerk, Media	MedClerk
Clerk, Serials	Serials
clerk, circulation	clerk

Figure 2-24. New operator profile

4. Enter the new operator ID information.

NOTE:
You may want to consider creating an operator profile for a group of staff members that have the same security access rather than one for each separate staff member.

Result: This completes the information needed to create a New Operator Profile.

5. Click **Save** or **Cancel**.

Result: This saves or cancels the new operator profile.

Repeat the steps in Procedure 2-9, Creating an Operator Profile, on page 2-48 as necessary in order to create at least one staff ID and one UC System Administrator ID.

Master Security Profile

Once you have created two Operator Profiles, one for staff to access modules and one for the System Administrator, you must setup a Master Security Profile. Essentially, this profile controls access to the Voyager System Administration module itself. In other words, it provides the authorization to authorize. Security here is imperative; and as a result, only the System Administrator has a profile established for UC purposes.

⚠ **IMPORTANT:**
To maximize security, the UC System Administrator profile should not be changed after it has been set up.

Creating a Master Security Profile

The procedure for creating a master security profile is shown in Procedure 2-10, Creating a Master Security Profile, on page 2-51.

Procedure 2-10. Creating a Master Security Profile

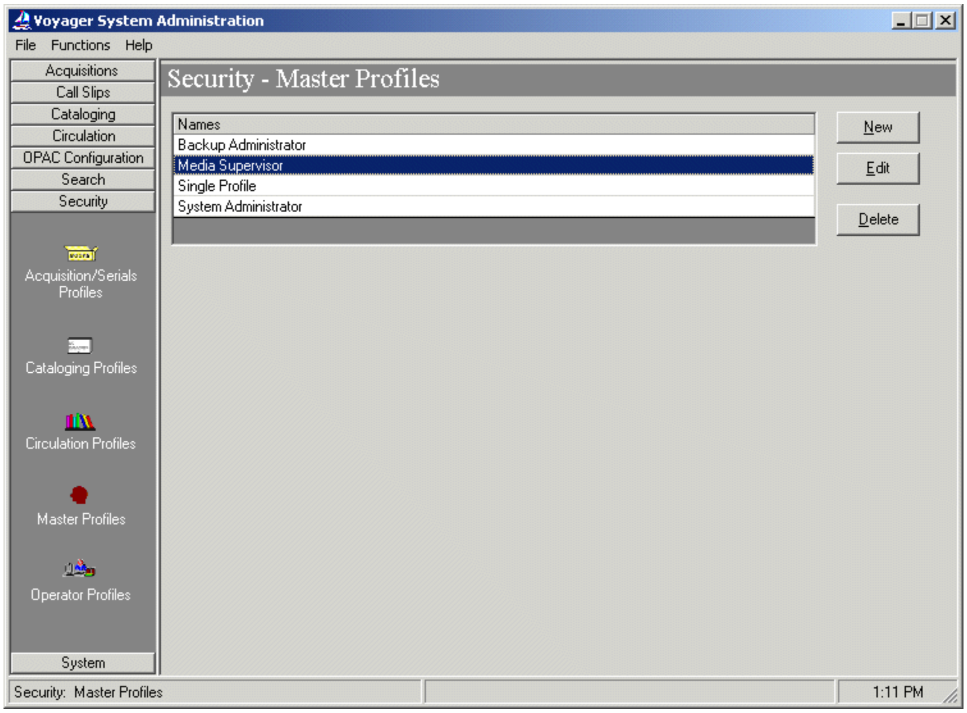
Use the following to create a master security profile.

1. Click **Security** from the vertical listbar in Voyager System Administration.

   Result: The Voyager System Administration security options display.

2. Click **Master Profiles**.

   Result: The **Master Profiles** dialog box opens. See Figure 2-25.



Names
Backup Administrator
Media Supervisor
Single Profile
System Administrator

Figure 2-25. Security - Master Profiles main dialog box

3. Click **New**.

Result: The **New Master Profile** dialog box opens. See Figure 2-26.

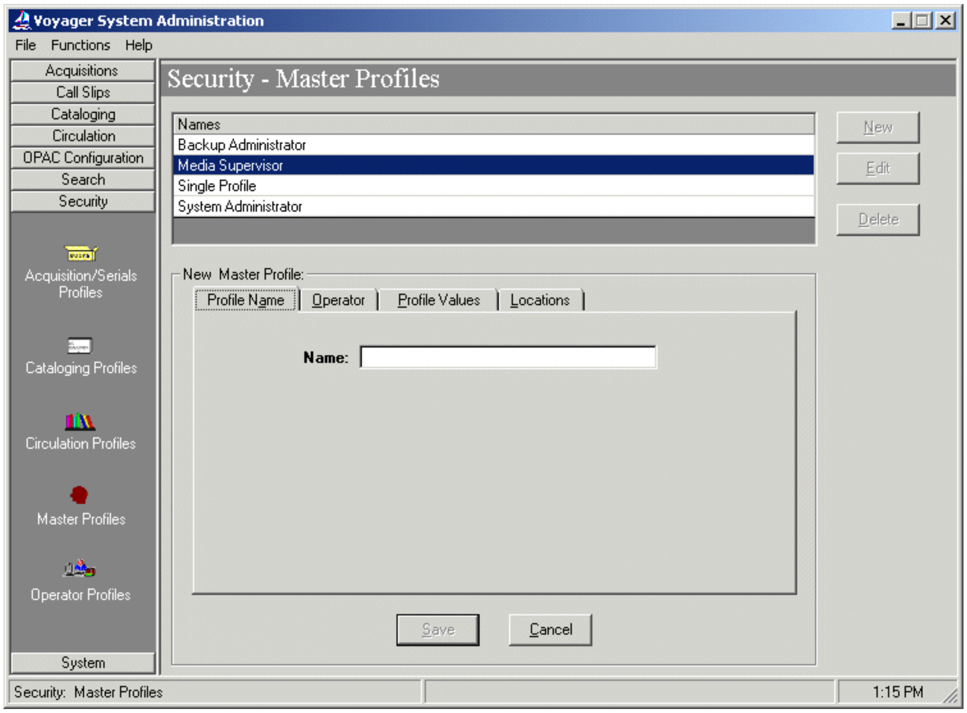


Figure 2-26. New Master Profile dialog box

4. Enter the master profile name up to 25 alphanumeric characters.

   You may want to consider choosing a name that reflects the scope of the authority being conferred such as System Administrator.

   Result: The master profile has an identifying name.

5. Click the **Operator** tab and select the System Administrator operator profile to link to this master profile.

   Result: The System Administrator operator is moved to the **Selected Operators** list.

6. Click the **Profile Values** tab and select **Security**, **System-Wide Configuration**, and **Cataloging Policy Groups**.

   Result: The profile values are associated with the System Administrator operator profile.

7. Click the **Locations** tab and verify that the selected locations are correct.

Result: The locations are set for this master profile definition.

8. Click **Save** or **Cancel**.

   Result: This saves or cancels the new master profile.

Cataloging Security Profile

Since staff should be able to view records in the Voyager Cataloging module for troubleshooting purposes, you need to establish a Cataloging Security Profile for the Universal Catalog. This profile controls the functions an operator can perform in Cataloging.

Creating a Cataloging Security Profile

The procedure for creating a cataloging security profile is shown in Procedure 2-11, Creating a Cataloging Security Profile, on page 2-54.

Procedure 2-11. Creating a Cataloging Security Profile

Use the following to create a cataloging security profile.

1. Click **Security** from the vertical listbar in Voyager System Administration.

   Result: The Voyager System Administration security options display.

2. Click **Cataloging Profiles**.

   Result: The **Cataloging Profiles** dialog box opens. See Figure 2-27.

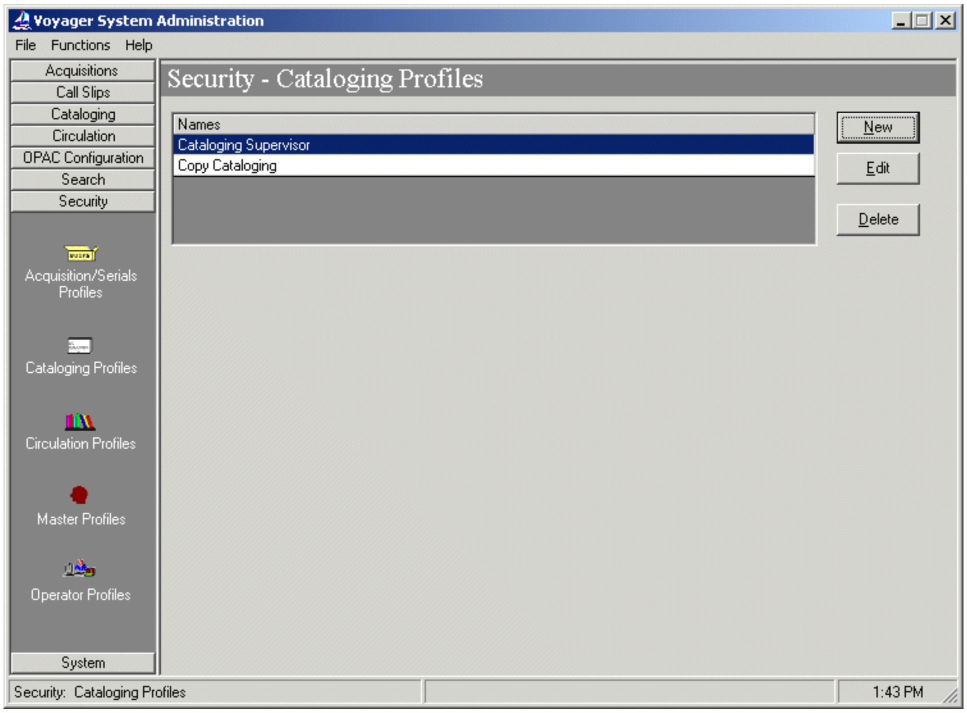
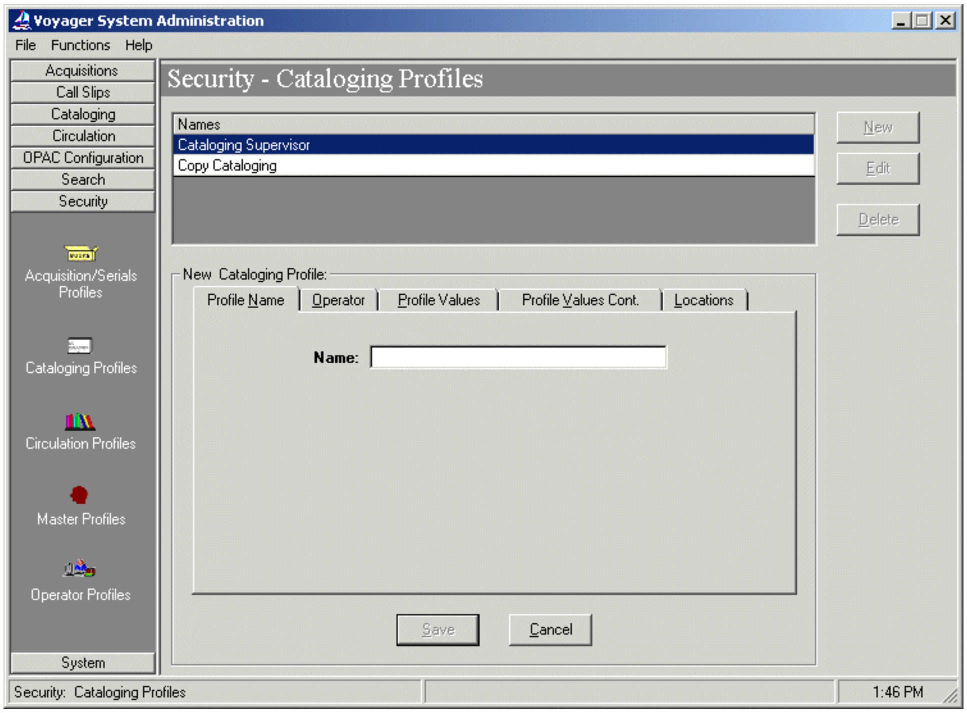


Figure 2-27. Security - Cataloging Profiles main dialog box

3. Click **New**.

   Result: The New Cataloging Profile dialog box opens. See Figure 2-28.



Names
Cataloging Supervisor
Copy Cataloging

Figure 2-28. New Cataloging Profile dialog box

4. Enter a name up to 25 alphanumeric characters in length.

   Result: This identifies the Cataloging security profile.

5. Click the **Operator** tab, select the names of the UC operators you created previously and move them to the **Selected Operators** list.

   Result: This associates the selected UC operators with this cataloging profile.

6. Click the **Profile Values** tab and select the following.

   - **View** for **Bibliographic Record**
   - **View** for **Holding Record**
   - **View** for **Authority Record**

IMPORTANT:
If you select check boxes on this tab that allow operators to add, delete, or update UC records, serious inconsistencies and complications can occur in

the UC database. Therefore, it is crucial that you only select the check boxes that allow operators to view records.

Result: This identifies the authorization values for the Cataloging profile (and the selected operators within this profile).

7. Click the **Locations** tab and select all the locations that you created in Cataloging > Policy Definitions for the UC database.

   Result: This associates these locations with the Cataloging profile.

8. Click **Save** or **Cancel**.

   Result: This saves or cancels the new master profile.

Contributing Database System Setup: Pre-Initial Load

Dynamic retrieval and display of detailed holdings and item information requires the UC database server and contributing library database servers to “communicate.” In order for this exchange of data to occur, the contributing library server needs to have database definitions and related connection information set up in Voyager System Administration. See the *Voyager System Administration User’s Guide* for details regarding the set up of database definitions for connecting to another Voyager database.

Additionally, another contributing library consideration may be policy issues related to excluding certain records from the UC database load. See Excluding Records by Location on page 3-3 for more information about the exclude option and policy issues that may require Voyager System Administration setup on the contributing libraries’ database systems.

UC Directory Structure

Figure 2-29 illustrates the directory structure on both the Voyager servers of participating libraries and on the UC server as shipped. Your location may have a different directory structure. A description of each directory is located in Table 2-18 on page 2-58.

NOTE:
This section has been added here as a precursor to the more technical components used when working with the Universal Catalog. Knowledge of the

scripts used to load and maintain the Universal Catalog facilitates a full understanding of this directory structure. See Building and Maintaining the UC on page 3-5 for information on how to load and maintain the UC database.

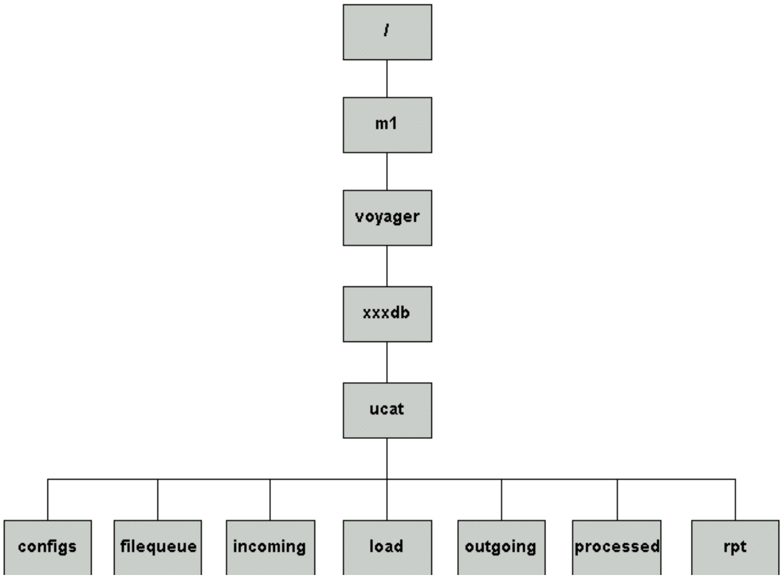


Figure 2-29. UC directory structure

Table 2-18. UC directory structure descriptions

DIRECTORY	DESCRIPTION
configs	Where all Prebulk configuration files from participating databases should reside. If Prebulk is done on local databases, these are basically duplicates of those used in the local database. If you are excluding locations from export, the export exclude locations file should reside here.
filequeue	When FTP of records from local databases is complete, newly uploaded files are moved here from the incoming directory. This is where the Bulk Import script (`ucatimp.pl`) on the UC server expects to find files to process.

Table 2-18. UC directory structure descriptions

DIRECTORY	DESCRIPTION
incoming	Where the MARC Export script (`ucatexp.pl`) FTPs its files to on the UC server. Once all files are FTPed, they are automatically moved into the filequeue directory.
load	Where the `LOAD.ucat` script resides and runs on the UC server.
outgoing	Where the MARC Export script (`ucatexp.pl`) assembles the files it uploads to the UC server.
processed	Both the MARC Export and Bulk Import scripts place the files they have processed here. There is at least one file in this location on the servers of participating libraries.
rpt	Where the log files created by the scripts as well as those created by MARC Export and Bulk Import reside. On the UC server only, the Bulk Import program `ucatimp.pl` automatically creates subdirectories (named after the modifying agency whose records are being imported), off of this directory. In these subdirectories, `ucatimp.pl` places its log files. For example, if the UC server is importing records from a local library database with a modifying agency code of dev2000db, `ucatimp.pl` writes its log files to `/m1/voyager/`*`xxx`*`db/ucat/rpt/dev2000db` on the UC server.

Loading Records into and Maintaining the Universal Catalog

3

Overview

To insure that the Universal Catalog contains meaningful data, some forethought needs to be given to which records in the local databases should be exported and shared with the Universal Catalog database and which should remain as local records only.

As a result, this chapter describes considerations and methodology for excluding records from export to the Universal Catalog database as well as the process for initially loading, maintaining and updating the records in the Universal Catalog database.

Excluding Records from the UC

As a first step, it is important to understand the process for excluding records. Decisions about excluding records need to be made and implemented before the first record is initially loaded into the Universal Catalog database.

Policy decisions may be required regarding locations identified in the Voyager System Administration module, and additional procedures may be implemented to utilize the Suppress from OPAC feature.

Example: What options would you implement if you want the bibliographic records of professors’ course reserve materials to display in the local library database but not be exported to the Universal Catalog database?

See Methods to Exclude Records from the UC on page 3-2 for answers to this and other questions.

Methods to Exclude Records from the UC

You can identify records for exclusion from the Univeral Catalog database in one of the following ways.

- Checking the **Suppress from OPAC** check box for individual bibliographic records
- Creating a file of locations that identifies bibliographic records for exclusion

NOTE:
The **OK to export** check box on the **System** tab (see Figure 3-30) of a given bibliographic record in the Cataloging module has nothing to do with making records eligible or ineligible for inclusion in the UC database. If this is left unchecked, records are still able to be loaded into the UC database.

Suppress from OPAC

If you check the **Suppress from OPAC** check box for a bibliographic record in a local database, it is not loaded into the UC database. If the check box is left unchecked, the record is eligible for load into the UC database.

The procedure for checking the Suppress from OPAC check box is shown in Procedure 3-12, Checking the Suppress from OPAC Check Box, on page 3-2.

Procedure 3-12. Checking the Suppress from OPAC Check Box

Use the following to select **Suppress from OPAC for a bibliographic record in Cataloging**.

1. Perform a search in the Cataloging module for the bibliographic record to be suppressed.

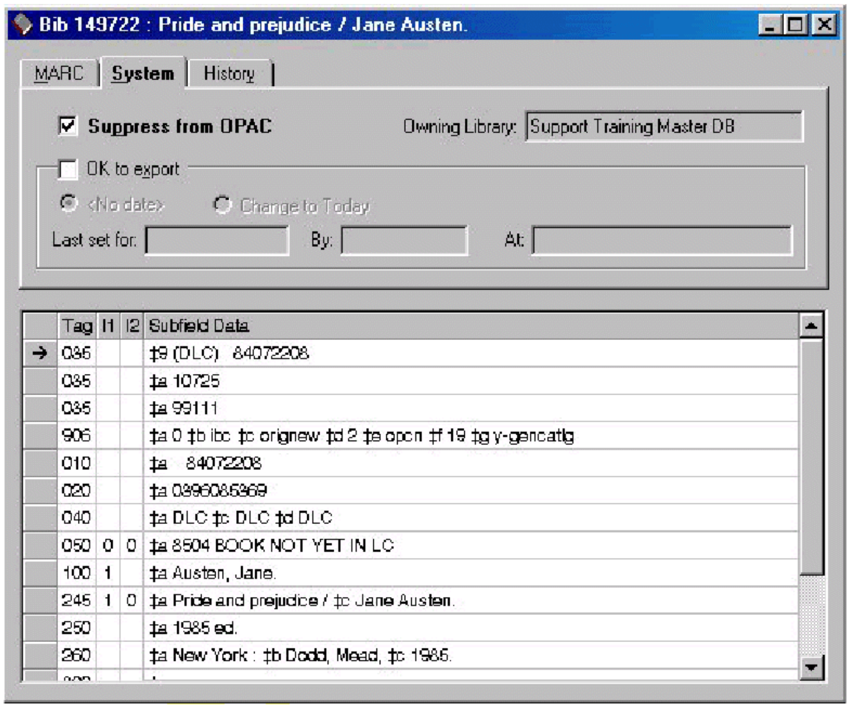
   Result: The bibliographic record contents on the **MARC** tab displays.

2. Click the **System** tab.

   Result: The System tab options and bibliographic record contents display. See Figure 3-30.

3. Check the **Suppress from OPAC** check box, or press **ALT+P**.

   Result: The bibliographic record is now ineligible for loading into the UC database.



☑ Suppress from OPAC

Owning Library: Support Training Master DB

□ OK to export

● <No date> ○ Change to Today

Last set for: By: At:

	Tag	I1	I2	Subfield Data
→	035			‡9 (DLC) 84072208
	035			‡a 10725
	035			‡a 99111
	906			‡a 0 ‡b ibc ‡c orignew ‡d 2 ‡e opcn ‡f 19 ‡g y-gencatlg
	010			‡a 84072208
	020			‡a 0396085369
	040			‡a DLC ‡c DLC ‡d DLC
	050	0	0	‡a 8504 BOOK NOT YET IN LC
	100	1		‡a Austen, Jane.
	245	1	0	‡a Pride and prejudice / ‡c Jane Austen.
	250			‡a 1985 ed.
	260			‡a New York : ‡b Dodd, Mead, ‡c 1985.

Figure 3-30. Checking the Suppress from OPAC Check Box

Excluding Records by Location

To exclude records by location (from being exported to the UC database) requires the following.

- Exclude locations file
- Bibliographic record location history

With this information, the Voyager bulk export software is able to identify the records eligible for exclusion when `ucatexp.pl` is run.

The exclude locations file(s) is one that you create that contains the location code for each location that identifies bibliographic records for exclusion (preventing export of records based on a processing location). Each location code is entered on a separate line in the exclude locations file. The location codes entered in the exclude locations file are the same as the ones established in Voyager System Administration.

⚠ **IMPORTANT:**
Location codes in the exclude locations file are case-sensitive and must appear exactly as they do in the Voyager System Administration module.

RECOMMENDED:
It is recommended that the exclude locations file be stored in the following directory on the local server: `/m1/voyager/`*`xxx`*`db/ucat/configs`*. See UC directory structure descriptions on page 2-58 for more details on this directory.*

The exclude locations file(s) is specified in the `voyager.env` file using the following parameters.

- `UCATLOCEXCLUDEFILE`
- `UCATLOCEXCLUDELASTFILE`

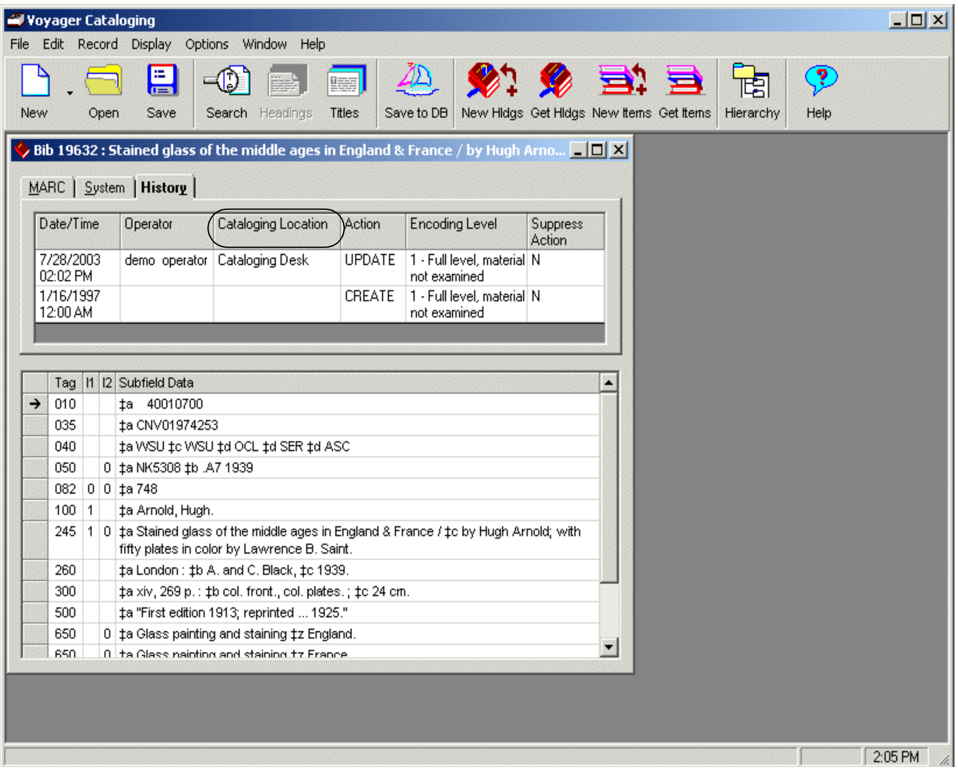
The `UCATLOCEXCLUDEFILE` and the `UCATLOCEXCLUDELASTFILE` specify exclude locations files that have different logic applied for identifying bibliographic records for exclusion.

Exclude records processing uses one of the following methods.

- Bibliographic records are identified as eligible for exclusion based on the create location or the most recent activity location. The exclude locations file specified with the `UCATLOCEXCLUDEFILE` parameter uses this method.
- Bibliographic records are identified as eligible for exclusion based on the most recent activity location only. The exclude locations file specified with the `UCATLOCEXCLUDELASTFILE` parameter uses this method.

The bulk export program uses the bibliographic record history location information as viewable on the bibliographic record **History** tab. See Figure 3-31, "Bibliographic record History tab," on page 5.

NOTE:
Any activity (create or update) performed against a bibliographic record is retained for the life of the record. This complete history for a bibliographic record provides the capability to have multiple methods of processing the exclude locations file(s).



Date/Time	Operator	Cataloging Location	Action	Encoding Level	Suppress Action
7/28/2003 02:02 PM	demo operator	Cataloging Desk	UPDATE	1 - Full level, material not examined	N
1/16/1997 12:00 AM			CREATE	1 - Full level, material not examined	N

Tag	I1	I2	Subfield Data
010			‡a 40010700
035			‡a CNV01974253
040			‡a WSU ‡c WSU ‡d OCL ‡d SER ‡d ASC
050		0	‡a NK5308 ‡b .A7 1939
082	0	0	‡a 748
100	1		‡a Arnold, Hugh.
245	1	0	‡a Stained glass of the middle ages in England & France / ‡c by Hugh Arnold; with fifty plates in color by Lawrence B. Saint.
260			‡a London : ‡b A. and C. Black, ‡c 1939.
300			‡a xiv, 269 p. : ‡b col. front., col. plates. ; ‡c 24 cm.
500			‡a "First edition 1913; reprinted ... 1925."
650		0	‡a Glass painting and staining ‡z England.
650		0	‡a Glass painting and staining ‡z France.

Figure 3-31. Bibliographic record History tab

More details about the UCATLOCEXCLUDEFILE and the UCATLOCEXCLUDELASTFILE parameters are provided in Default Parameters - Voyager.env on page 3-15.

Building and Maintaining the UC

The Universal Catalog database is built and maintained in the following ways.

- Initial load of records into the UC database

- Ongoing update of the UC records and database which includes the addition of new records and the replacement and deletion of existing records
- Load of records from new participating library databases

Initially Loading Records into the UC

Once configurations are set for the UC database, you can load all the eligible records of contributing libraries. The initial load process uses the following programs:

- `ucatexp.pl`
- `LOAD.ucat`

ucatexp.pl

The `ucatexp.pl` file resides in the `/m1/voyager/bin` directory on the Voyager servers of participating libraries. It is invoked by a shell script located in the `/m1/voyager/`*`xxx`*`db/sbin` directory called `Pucatexp`.

The `ucatexp.pl` program does the following.

- Performs a bulk export of MARC records from the databases of contributing libraries.
- Runs Prebulk on bibliographic records if local libraries elect to do so.

Bulk Export of MARC Records

When executed, `ucatexp.pl` performs a bulk export of MARC records of created, updated, and suppressed bibliographic records from contributing databases.

Bulk export of MARC records has been enhanced for the purpose of all UC exports to put the NUC code for each individual library into the 040‡d field of bibliographic records. It is then possible to identify to which contributing library database a UC bibliographic record belongs.

The result of the bulk export of MARC records is the generation of two files.

- `upd.YYYYMMDDMISS.NUC.bib` or `upd.YYYYMMDDMISS.NUC.bibmfhd (update file)`
- `del.YYYYMMDDMISS.NUC.bib` or `del.YYYYMMDDMISS.NUC.bibmfhd (delete file)`

The names of these files differ depending on where Prebulk is run. If Prebulk is not run on the Voyager servers of contributing libraries before records are sent to the UC, the files have the `.bib` extension. If Prebulk is run on the Voyager servers of contributing libraries, the file names have the `.bibmfhd` file extension. Notice that the actual NUC code for contributing library databases is included in the file name formats.

NOTE:
Bulk export of MARC records does not generate export files that are greater than 1 gigabyte. This is to prevent loss of records from larger sites that do large exports such as an export of every record from a database. In the event that the output file becomes greater than 1 gigabyte while bulk export of MARC records is writing to a file, bulk export of MARC records closes its current output file, adds a sequence number to the filename, and opens up a new file to which it can write.

The following are examples of file names generated with sequence numbers.

```
upd.20010320092159.GUA.1.bib
upd.20010320092159.GUA.2.bib
upd.20010320092159.GUA.3.bib
```

You should never have to touch or deal with these files. These files containing bibliographic records or bibliographic records and their associated MFHDs are automatically FTPed to the UC server for further processing.

Update File

The update file contains bibliographic records of created and updated records. These records are stored in `upd.YYYYMMDDMISS.NUC.bib` or `upd.YYYYMMDDMISS.NUC.bibmfhd`.

Delete File

The delete file contains one or more of the following components.

- Bibliographic records identified for deletion and stored in `deleted.bibs.marc`
- Bibliographic records marked Suppress from OPAC stored in a file of suppressed records.
- Bibliographic records identified for exclusion stored in a file of records eligible for exclusion

By default, the file called `deleted.bibs.marc` and the file containing suppressed records are concatenated into `del.YYYYMMDDMISS.NUC.bib` or `del.YYYYMMDDMISS.NUC.bibmfhd`.

To concatenate all components (`deleted.bibs.marc`, Suppress from OPAC records, and records identified for exclusion) into the `del.YYYYMMDDMISS.NUC.bib` or `del.YYYYMMDDMISS.NUC.bibmfhd`, the `-a` parameter needs to be set on the command line with Pucatexp. See Table 3-25, "Valid command line parameters for Pucatexp," on page 14 for more information about the `-a` parameter.

Prebulk

Contributing libraries can elect to run Prebulk on the files generated by `ucatexp.pl`. The files are then FTPed to the UC server. FTP is invoked by the `ucatexp.pl` program. If participating libraries do not elect to perform Prebulk on their local servers, it automatically runs on the UC server.

IMPORTANT:
Where Prebulk is performed depends on whether or not the -p parameter is used during the execution of the ucatexp.pl program. If the -p parameter is used on local servers to run ucatexp.pl, Prebulk is performed locally. If the -p switch is not used during the execution of ucatexp.pl, Prebulk is performed on the UC server. See Command Line Parameters on page 3-14 for details regarding command line parameters.

Regardless of where Prebulk is performed, it serves to pre-process bibliographic records and generate basic holdings information. For UC purposes, the MFHDs generated by Prebulk serve as pointer records.

The UC MFHD records contain the following fields and subfields.

- 014‡a
- 852‡b

The 014‡a contains the bibliographic ID that links the MFHD to its associated bibliographic record in the database of the contributing library.

The 852‡b contains the location code identifying to which contributing library database the bibliographic record belongs. This location code is defined in the UC System Administration module. See Locations for Contributing Libraries on page 2-2 for more information.

Call number information is typically not stored in the UC MFHD records. However, in an effort to provide Universal Catalog implementers with as much flexibility as possible, call numbers can optionally be stored in the UC MFHD records during the initial load and/or during ongoing maintainance of the UC database. There are special considerations though when this is done. For additional information regarding these considerations and the steps required to implement call numbers in the UC MFHD records, see Call Numbers in MFHDs on page A-1.

Prebulk Configuration File (ucatexp.pl)

If Prebulk is run on the servers of participating libraries, a configuration file must exist locally. The path to this file is specified by using the `-p` switch during the execution of `ucatexp.pl`.

NOTE:
Since Prebulk needs to run on the UC server if the `-p` is not used, a Prebulk configuration file must also reside on the UC server.

Prebulk configuration files consist of stanzas. The Prebulk configuration file must be tab-delimited. This means that for any entries in the file in which multiple pieces of information must be specified on the same line, you must separate each piece of information with a tab. Prebulk does not recognize spaces or other characters as separators between pieces of information.

For more information on Prebulk configuration files, see the *Voyager Technical User's Guide*.

⚠ **IMPORTANT:**
When creating and editing a Prebulk configuration file, you must use the UNIX vi editor. Configuration files created and/or edited using other editors such as Notepad cause Prebulk to fail.

Example: Figure 3-32 shows a sample Prebulk configuration file. The configuration file you generate for your local library database is likely to be very similar, if not identical to this one. A description of each stanza follows.

NOTE:
This sample configuration file is appropriate whether Prebulk is run on the Voyager servers of contributing libraries or on the UC server.

```
[OVERRIDES]
CREATEMFHD=YES
DEFAULTCALLNO=
DEFAULTCALLIND=8
USE001FOR014=YES
USE003FORLOC=NO

[STRIP]

[MFHDTAG]
XXX

[CALLTYPES]
050     0

[LOCATIONS]
default dev20001db      999

[MAPPING]
x       852b

[008]
000     6       am      9905280u    0   0
uuund|990528
```

Figure 3-32. Sample Prebulk configuration file

[OVERRIDES] Stanza

```
[OVERRIDES]
CREATEMFHD=YES
DEFAULTCALLNO=
DEFAULTCALLIND=
USE001FOR014=YES
```

```
USE003FORLOC=NO
```

Table 3-19 contains an explanation of each stanza found in the [OVERRIDES] stanza.

Table 3-19. Sample Prebulk Configuration File: [OVERRIDES] stanza

CREATEMFHD=YES	This field (not case-sensitive) indicates whether or not MFHDs are to be created by Prebulk. This is always set to YES for UC purposes. NO indicates that Prebulk will be run only to strip tags.
DEFAULTCALLNO=	This field is left blank because call numbers are not being added to the MFHDs for the UC. Under different circumstances, you could enter a call number here to be placed in the 852‡h field if no call number is found in the [MAPPING] stanza. See Call Numbers in MFHDs on page A-1 for more information about call numbers in the UC database.
DEFAULTCALLIND=8	This field is only used if a value is found in the [DEFAULTCALLNO] stanza. It indicates the value for indicator one of the 852 field. Only numbers between 0 and 8 are valid. Pipes and blanks are not acceptable. You can input any number as a value.
USE001FOR014=YES	Entering YES indicates that the 014 field will be created with $a containing the value of the 001 tag. For UC purposes, this should always be set to YES. This field is not case sensitive.
USE003FORLOC=NO	Entering YES indicates that the incoming 003 tag will be the location used in the 852‡b field of the MFHDs. For UC purposes, this field is always set to NO because the 852‡b field is not being built off of the 003 tag. This field is not case sensitive.

[STRIP] Stanza

```
[STRIP]
```

Nothing is set in this stanza for the Universal Catalog, because fields of incoming bibliographic records are not being stripped during Prebulk.

[MFHDTAG] Stanza

```
[MFHDTAG]
XXX
```

The [MFHDTAG] stanza identifies which MARC tag contains the data to be used to create MFHDs. See Table 3-20.

Table 3-20. Sample Prebulk Configuration File: [MFHDTAG] stanza

XXX	XXX indicates that no tag contains holdings information and the default location specified in the [LOCATIONS] stanza should be used in the 852‡b. With a value of XXX, the program does not try to create item information and detailed holdings.

[CALLTYPES] Stanza

```
[CALLTYPES]

099      8
```

This stanza specifies what indicator is to be used when creating an 852 in a MFHD. See Table 3-21.

Table 3-21. Sample Prebulk Configuration File: [CALLTYPES] stanza

099 8	This example indicates that if a call number is found in the 099 field, it should be marked with an indicator of 8 which means other. Even though call numbers are not usually included in UC MFHDs, there must be at least one call number field listed here. See Call Numbers in MFHDs on page A-1 for more information about call numbers in the UC database.

[LOCATIONS] Stanza

```
[LOCATIONS]

default dev2001db      999
```

This stanza determines what location is to be used in MFHDs. The first line is the default setting which is the only line to be entered for UC purposes. See Table 3-22.

Table 3-22. Sample Prebulk Configuration File: [LOCATIONS] stanza

default dev2001db 999	In this example, Prebulk puts the default location code `dev2001db` (which is the location code for the modifying agency) in the MFHD 852‡bs. The location code in the prebulk configuration file and the location code as defined in Voyager System Administration must match exactly including uppercase and lowercase characters. 999 is a nonexistent field that is used as a placeholder in this stanza for call number information. This causes the program to reference the `DEFAULTCALLNO=` in the `[OVERRIDES]` stanza which has “blank” as a specified value; and therefore, no call number is generated during prebulk. See Call Numbers in MFHDs on page A-1 for more information about call numbers in the UC database.

[MAPPING] Stanza

```
[MAPPING]
x      852b
```

This stanza indicates the subfields in the tag specified in the `[MFHDTAG]` stanza which contains specific data and where that data goes in MFHDs.   See Table 3-23.

Table 3-23. Sample Prebulk Configuration File: [MAPPING] stanza

x 852b	What this means is that if the tag specified in the `[MFHDTAG]` stanza is XXX which for UC purposes it is, then no mapping occurs to the 852‡b field.

[008] Stanza

```
[008]
```

```
    000     6     am     9905280u     0     0     uuund |
    990528
```

This stanza specifies how to create the 008 field in new MFHDs. You can specify a location in a field in the bibliographic record. If the field matches a string that you specify, you can designate a string that is to become the 008. See Table 3-24.

Table 3-24. Sample Prebulk Configuration File:[008] stanza

000 6 am 9905280u 0 0 uuund | 990528	For UC purposes, we are not building a 008 field. However, something must be entered here. What the example indicates is that the 008 field is built off of what is in offset 6 of the leader field for a Prebulk run on May 28, 1999. This field must be 41 characters in length including blanks and/or spaces.

For more information on Prebulk configuration files and related stanzas, see the *Voyager Technical User’s Guide*.

Command Line Parameters

Table 3-25 describes the command line parameters available for use with the `Pucatexp` program.

Table 3-25. Valid command line parameters for Pucatexp

PARAMETER	DESCRIPTION
-a	Concatenates the deleted.bibs.marc file, the suppressed bibliographic records file, and the file of bibliographic records eligible for exclusion.
-d	Destination for the export file on the UC server, the `/m1/voyager/`*`xxx`*`db/ucat/incoming` directory on the UC server.
-e	Export interval in minutes if not running as a cron job.
-i	IP/hostname of UC server.
-m	Modifying agency code, the 040‡d field.
-p	Path to Prebulk configuration file on the Voyager servers of local libraries. If this switch and its appropriate value is used, Prebulk is run locally. If not, Prebulk is run on the UC server.

Table 3-25. Valid command line parameters for Pucatexp

PARAMETER	DESCRIPTION
-L	Load mode. This causes the export of all records with the exception of those suppressed from the OPAC and those excluded by location using the `-X` switch. Suppressed or deleted record files are not generated in this mode.
-h	Provides a listing and brief description of all the valid parameters for the script.

Additional Parameters (Automatically Generated)

The -r and -X parameters (see Table 3-26) are automatically generated for Pucatexp processing when the UCATLOCEXCLUDEFILE or the UCATLOCEXCLUDELASTFILE parameters are specified in `voyager.env` (see Default Parameters - Voyager.env on page 3-15).

Table 3-26. Parameters automatically generated for Pucatexp

PARAMETER	DESCRIPTION
-r	Setting a value for “export UCATLOCEXCLUDELASTFILE=” (see Table 3-33, “Default voyager.env file parameters,” on page 16) causes Pucatexp to automatically generate and use this parameter. Path to exclude locations file for bulk export of MARC records processing.
-X	Setting a value for “export UCATLOCEXCLUDEFILE=” (see Table 3-33, “Default voyager.env file parameters,” on page 16) causes Pucatexp to automatically generate and use this parameter. Path to exclude locations file for bulk export of MARC records processing.

Default Parameters - Voyager.env

Setting default parameters in the `voyager.env` file eliminates the need to specify command line parameters (see Table 3-25) when running Pucatexp.

The `voyager.env` file resides in the `/m1/voyager/`*`xxx`*`db/ini` directory on the local libraries’ Voyager servers. The **Universal catalog parms** comment line provides a heading that identifies where to set the parameters in the `voyager.env` file. See Figure 3-33.

```
##      Universal catalog parms
export UCATSERVERADDR=
export UCATVOYAGER=
export UCATDATABASE=
export UCATMODIFYAGENCY=
export UCATUPDATEINTERVAL=
export UCATLOCEXCLUDEFILE=
export UCATLOCEXCLUDELASTFILE=
export UCATTRANSFER=
```

Figure 3-33. Default voyager.env file parameters

The following information provides an example of possible values to set for each parameter, a description of what each value means, and the switch each value replaces at the command line.

export UCATSERVERADDR=xxx.xxx.xxx.xxx

This parameter indicates the IP address of the UC server. If a value is entered here, you do not need to use the `-i` switch at the command line.

export UCATVOYAGER=/m1/voyager/

This parameter indicates the destination location on the UC server for the export file generated by `Pucatexp`. If a value is entered here, you do not need to use the -d switch at the command line.

export UCATDATBASE=xxxdb

This parameter indicates the name of the destination database (the UC server) for the export file generated by `Pucatexp`. Used in combination with the `export UCATVOYAGER=` line, you do not need to use the `-d` switch at the command line.

export UCATMODIFYAGENCY=

This parameter indicates the name of the modifying agency such as UGA or what is found in the 040‡d field of the bibliographic records being exported. If you enter a value here, you do not need to use the `-m` switch at the command line.

export UCATUPDATEINTERVAL=30

For initial load purposes, you do not need to enter a value for this parameter. This parameter is only used during the ongoing update process. See *Ongoing Updating of UC Records* on page 3-30 for more details about ongoing updates. If a parameter is set here for ongoing updates of the UC, you do not need to use the `-e` switch.

Any value between 5 (minutes) and 1440 (one day) represents optimum values for this interval. A value of 0 causes the program to continuously check for updated files in the filequeue. If the value is left blank, the program checks the filequeue for files to process; and when processing is complete, it quits and does not check again.

export UCATLOCEXCLUDEFILE= /m1/voyager/xxxdb/ucat/configs/filename

This parameter indicates the path to an exclude locations file that is to be used with the bulk export of MARC records.

The locations identified in /m1/voyager/xxxdb/ucat/configs/filename, specify bibliographic records eligible for exclusion based on the create location or the most recent activity location per the record’s stored bibliographic history location information. See Excluding Records by Location on page 3-3.

When a value for this parameter is set in the `voyager.env` file, the `-X` switch is automatically generated along with the full path and filename.

RECOMMENDED:
It is recommended that the exclude locations file be stored in the following directory on the local server: `/m1/voyager/xxxdb/ucat/configs`*. See UC directory structure descriptions on page 2-58 for details on this directory.*

IMPORTANT:
`UCATLOCEXCLUDELASTFILE` *and* `UCATLOCEXCLUDEFILE` *are mutually exclusive. You must choose to use one or the other but not both when identifying bibliographic records for exclusion.*

export UCATLOCEXCLUDELASTFILE= /m1/voyager/xxxdb/ucat/configs/filename

This parameter indicates the path to an exclude locations file that is to be used for bulk export of MARC records.

The locations identified in /m1/voyager/xxxdb/ucat/configs/filename, specify bibliographic records eligible for exclusion based solely on the most recent activity location per the record’s stored bibliographic history location information. See Excluding Records by Location on page 3-3.

NOTE:
If no location is associated with the most recent activity, a search in reverse chronological order is performed to a find a location. If no location is found, the record is exported.

When a value for this parameter is set in the `voyager.env` file, the `-r` switch is automatically generated along with the full path and filename.

RECOMMENDED:
It is recommended that the exclude locations file be stored in the following directory on the local server: `/m1/voyager/xxxdb/ucat/configs`*. See UC directory structure descriptions on page 2-58 for details on this directory.*

IMPORTANT:
`UCATLOCEXCLUDELASTFILE` and `UCATLOCEXCLUDEFILE` are mutually exclusive. You must choose to use one or the other but not both when identifying bibliographic records for exclusion.

export UCATTRANSFER=

This parameter specifies the path (`/m1/voyager/xxxdb/sbin/transfer.ftp.ksh` is suggested) for the script file that is executed when files are transferred to the Universal Catalog server. You may create your own script file or use one similar to the example in Figure 3-34. This file works in combination with the `.netrc` file (see Setting Permissions on page 3-21) for logging into the UC server for file transfer.

```
#!/bin/ksh
#
# A transfer script for UC
#
# This is invoked by ucexp.pl with file host srcdir
        tempdir finaldir
# It will transfer the file and rename it
#
# Note that this is dependent on ~/.netrc, if not found
        this will prompt
# interactively for the login and password.
#
FL=$1
HOST=$2
SRCDIR=$3
TEMPDIR=$4
FINALDIR=$5
echo transferring $FL from $SRCDIR to $TEMPDIR
ftp -v $HOST <<EOF | grep ^226
binary
cd $TEMPDIR
delete $FL
put $SRCDIR/$FL $FL
EOF
RC=$?
if [[ $RC != 0 ]]
then
  echo transfer failed with $RC
  exit $RC
fi
echo renaming $TEMPDIR/$FL $FINALDIR/$FL
```

Figure 3-34. Example script for file transfer

```
ftp -v $HOST <<EOF | grep ^250
cd $TEMPDIR
rename $FL $FINALDIR/$FL
EOF
exit $RC
```

Figure 3-34. Example script for file transfer (Continued)

If your UC database is located on the same server as the contributing library database, you could create a script that simply copies data instead of doing a file transfer. See the example script in Figure 3-35.

```
cp -f $3|$1 $5
```

Figure 3-35. Example script for copying data

Running ucatexp.pl

The following is an example of a typical command line for running the `ucatexp.pl` program to export all records from a database. Keep in mind, you can set defaults for certain parameters in the `/m1/voyager/`*xxx*`db/ini/voyager.env` directory so that you do not need to use all the switches.

When logged in as voyager in the `/m1/voyager/`*xxx*`db/sbin` directory, enter the following.

```
nohup Pucatexp -L &
```

Using nohup indicates that there is no hangup if a terminal session is terminated at some point throughout the running of `Pucatexp`. Using & indicates that the process should be run in the background, returning control to command prompt.

Log File

The `ucatexp.pl` program generates a log file containing activities and problems the program encounters during its runs such as number of records processed, deletions of bibliographic records, and so on. The file includes information whether `ucatexp.pl` processed zero or many records. The log file is named with a timestamp in the following format.

```
log.ucexp.20000319.PID
```

`PID` is the Process ID.

You can check the log file in the `/m1/voyager/`*`xxx`*`db/ucat/rpt` directory of the local server after `ucatexp.pl` has been run, using standard UNIX commands.

Setting Permissions

In order for the `ucatexp.pl` script to FTP files from the servers of participating libraries to the Universal Catalog, it must be logged on to the UC server as voyager. To do this, a file named `.netrc` should be created in the Voyager home directory on the Voyager servers of contributing libraries.

The `.netrc` file contains one line in the following format.

```
default login voyager password voypass
```

Substitute the voyager password on the UC server for voypass.

The permissions must be set on the `.netrc` file so that only voyager can read, write, or otherwise access the file. To set permissions, use the following chmod command.

```
chmod 0600 .netrc
```

⚠ **IMPORTANT:**
If the permissions are not set correctly, for example, set to 755, there is be a security issue, and `ucatexp.pl` *does not attempt to FTP to the UC server.*

LOAD.ucat Program

This program resides and is run in the `/m1/voyager/xxxdb/ucat/load` directory on the Universal Catalog server. It performs the following functions.

- Invokes `ucatimp.pl`

- Prompts index regeneration

Subdirectories

`LOAD.ucat` creates two subdirectories in the `/m1/voyager/xxxdb/ucat/load` directory.

- One for the load component
- One for the index regeneration component

You can set the subdirectories to be created elsewhere, for example, if your work space is limited using the WORKDIR parameter. See WORKDIR= on page 3-27 for details. All work files associated with the load and regeneration of indexes end up in subdirectories which are named according to the following syntax.

```
load.MMDDYY.HHMISS.work
regen.MMDDYYY.HHMISS.work
```

From within these subdirectories, `LOAD.ucat` links to the load scripts and executables, leaving the original load directory untouched. It knows where to find the scripts and executables from `/m1/voyager/`*`xxx`*`db/ini/voyager.env`.

ucatimp.pl Program

After the files generated by bulk export of MARC records using the `ucatexp.pl` program are FTPed to the UC database, the `LOAD.ucat` program invokes the `ucatimp.pl` program. The purpose of the `ucatimp.pl` program is twofold.

- It performs Prebulk on the UC server if participating libraries did not elect to run it locally.
- It performs Bulk Import of records into the UC database.

Prebulk and Bulk Import

For the purpose of ongoing updates of UC records, the `ucatimp.pl` program performs Bulk Import, which allows many records to be imported at one time from the databases of local libraries. The files of records to be imported are picked up from the `/m1/voyager/`*`xxx`*`db/ucat/filequeue` directory on the UC server. This is the directory in which records are automatically placed for import. Records are de-duped based on configurations you set in the Voyager System Administration module. See *Universal Catalog System Setup: Pre-Initial Load* on page 2-1 for details.

However, the `ucatimp.pl` program first performs Prebulk of bibliographic records on the UC server if participating libraries elected not to run it locally. In other words, if local libraries do not use the `-p` parameter when running `ucatexp.pl`, Prebulk needs to be run on the UC server.

NOTE:
The `ucatimp.pl` program knows that Prebulk has not been run locally, because it looks for incoming MFHDs on import. Specifically, it looks for files created by bulk export of MARC records with the `.bibmfhd` file extension, indicating that MFHDs have already been generated. When it doesn't find any files with the `.bibmfhd` file extension, it runs Prebulk.

Prebulk Configuration Files (ucatimp.pl)

Since the `ucatimp.pl` program performs Prebulk if it was not run on local servers, each participating database needs to have a Prebulk configuration file on the UC server. This configuration file should have a standardized name, using the following format.

```
prebulk.config.MODAGENCY
```

`MODAGENCY` is the same code used during the export process.

The configuration file for each contributing database should be placed in the `/m1/voyager/`*`xxx`*`db/ucat/configs` directory on the UC server. The `ucatimp.pl` program looks for a Prebulk configuration file for the library based on the modifying agency code in the configuration filename.

If a Prebulk configuration file cannot be found for the participating library, a generic one is used. Hence, a generic Prebulk configuration file should also be placed in the `/m1/voyager/`*`xxx`*`db/ucat/configs` directory on the UC server. The file should be called `prebulk.config.generic`. See *UC Directory Structure* on page 2-57 for more information regarding the UC directory structure.

NOTE:
If the generic file is used during Prebulk, the modifying agency code for contributing libraries is used in the -a switch on Bulk Import. This ensures that the MFHDs have their proper location information.

For detailed information about Prebulk configuration files used for UC purposes, see *Prebulk Configuration File (ucatexp.pl)* on page 3-9.

Index Regeneration

After the `LOAD.ucat` program runs the `ucatimp.pl` script, it automatically prompts index regeneration on the UC server.

It is possible that in specialized circumstances you may need to perform a manual index regeneration. See LOAD.ucat Header on page 3-24 for more information.

CAUTION:
You must be Voyager certified to perform a manual index regeneration. If you are not Voyager certified, you must contact Ex Libris for assistance with performing the index regeneration. If you are certified, remember to contact Support for the most current instructions for performing a manual index regeneration.

LOAD.ucat Header

The LOAD.ucat contains important information as show in Figure 3-36.

```
#  It is expected that the auth, bib, and mfhd readonly is not set at the
#  start of this script.  If they are it will interfere with the running
#  of this script.
#  This script should NOT be killed.
#  If this script is killed you should run manually
#  sqlplus $USERPASS<$WORKDIR/load.$RUNDATE.work/indexes.all.script
```

Figure 3-36. LOAD.ucat header

If for any reason the `LOAD.ucat` is killed, check the `dbcompare.xxxx.x.log` file in the `/regen.datetime.work` directory after the regen completes. If this log file indicates that any indices are missing, follow the instructions in the `LOAD.ucat` header for running the `indexes.all.script`.

CAUTION:
You must be Voyager certified to perform a manual index regeneration. If you are not Voyager certified, you must contact Ex Libris for assistance with performing the index regeneration. If you are certified, remember to contact Support for the most current instructions for performing a manual index regeneration.

Parameters

The valid parameters for the `LOAD.ucat` program are set at the top of the script file itself, located in the `/m1/voyager/`*`xxx`*`db/ucat/load` directory on the UC server. The parameters are not entered by way of switches at the command line. `LOAD.ucat` works differently than other programs.

NOTE:
You may want to make a copy of the `LOAD.ucat` script file, for example, call it LOAD.ucdb so that you maintain a copy of the original file without values set for each parameter. This provides you with a copy of the file you can modify while still maintaining the original on the UC server.

When you execute the `LOAD.ucat` program, you see the various parameters at the top of the script. See the sample in Figure 3-37. A description of each parameter follows.

```
RUNREGEN=N              ## run necessary regens when done (Y/N)

LOADSCHEMAONLY=N        ## only setup initial database structure

IMPORTRULE=             ## specify import rule - else defaults to
                        ## appropriate based off mod agency

CLEARBIBMARCDATA=NO     ## truncate necessary tables for full reload

CLEARAUTHMARCDATA=NO    ## must specify YES or NO - use with care!

PARALLELIZATION=        ## simultaneous bulk imports to run (1-9)
                        ## or blank for no simultaneous bulkimports

WORKDIR='pwd'           ## where to do actual load (Do not use relative path)
```

Figure 3-37. Sample LOAD.ucat parameters

NOTE:
The `LOAD.ucat` program gets the Voyager root directory and the UC database name from the `voyager.env` file located in the `/m1/voyager/xxxdb/ini/` directory.

RUNREGEN=

This parameter can be set to `Y` or `N`. This tells the `LOAD.ucat` program whether or not to regenerate indexes once records have been imported into the UC. This most commonly is set to Y for production databases, unless for some reason index regeneration is being run separately.

If this is set to Y, `LOAD.ucat` does the following.

- Creates a `regen.`*`MMDDYY`*`.`*`HHMISS`*`.` work subdirectory at the same level as the load working directory.
- Links to utility exes/scripts by way of the `VOYAGER_UTIL` in the `voyager.env` file, leaving the originals untouched.
- Creates `REGEN.indexes.`*`xxx`*`db` with the appropriate regen parameters set.
- Invokes the `REGEN.indexes.`*`xxx`*`db`, doing all the necessary index regenerations.

LOADSCHEMAONLY=

This parameter can be set to `Y` or `N`. If it is set to `Y`, `LOAD.ucat` sets up a basic database structure such as table space without loading any records. If it is set to `N`, `LOAD.ucat` sets up the database structure as well as import records. It is usually set to `N`.

IMPORTRULE=

With `IMPORTRULE=` you specify the Bulk Import rule you defined in the Voyager System Administration module that is used for importing records into the UC. If no rule is entered here, `LOAD.ucat` looks for a rule entitled, DEFAULT. That is why DEFAULT is suggested as the name for the Bulk Import rule you define in Voyager System Administration. Then you do not have to worry about specifying a rule for this parameter. Remember, the Bulk Import rule should not be changed during the life of the UC database.

CLEARBIBMARCDATA=

This parameter can be set to `YES` or `NO` (both words must be spelled out in full). If it is set to `YES`, `LOAD.ucat` empties out any existing bibliographic records and MFHDs already in the database before records are loaded. In addition, the mfd_count by location_id entry is set to zero to enable a count by library at each restart of the load of how many bibliographic and holdings records are being contributed.

⚠ **IMPORTANT:**
The `YES` *option is appropriate during the initial load, but the parameter must be set to* `NO` *when new contributing libraries are added to the UC, or the UC database is stripped of existing bibliographic records.*

CLEARAUTHMARCDATA=

This parameter can be set to `YES` or `NO` (both words must be spelled out in full). If set to `YES`, `LOAD.ucat` empties out any existing authority records already in the database before records are loaded. If set to `NO`, authority records are retained.

⚠ **IMPORTANT:**
Setting this parameter to `YES` *is appropriate during the first initial load of the Universal Catalog. However, the parameter must be set to* `NO` *when new libraries are added to the UC, or if you are engaging in a second initial load during which you do not want to lose all your authority records. Otherwise, the UC database is stripped of all existing authority records.*

PARALLELIZATION=

This parameter is used to identify the number of simultaneous bulk imports to run. The valid range is 1-9.

RECOMMENDED:
As a rule, if you have the whole UC server dedicated to the load, we suggest that you do not input a value that exceeds the total number of processes you want to perform concurrently.

WORKDIR=

With `WORKDIR=` you can specify a default working directory on the UC server in which `LOAD.ucat` creates its subdirectories and in which it places its files if you do not want to use the default directory. You may want to use this option if you do not have enough space in the default directory for the subdirectories.

The default `'pwd'` option extracts the current working directory. An option must be specified for this parameter.

NOTE:
When specifying an alternate default directory path use a specific (not relative) path such as `/m1/voyager/tmp/load`.

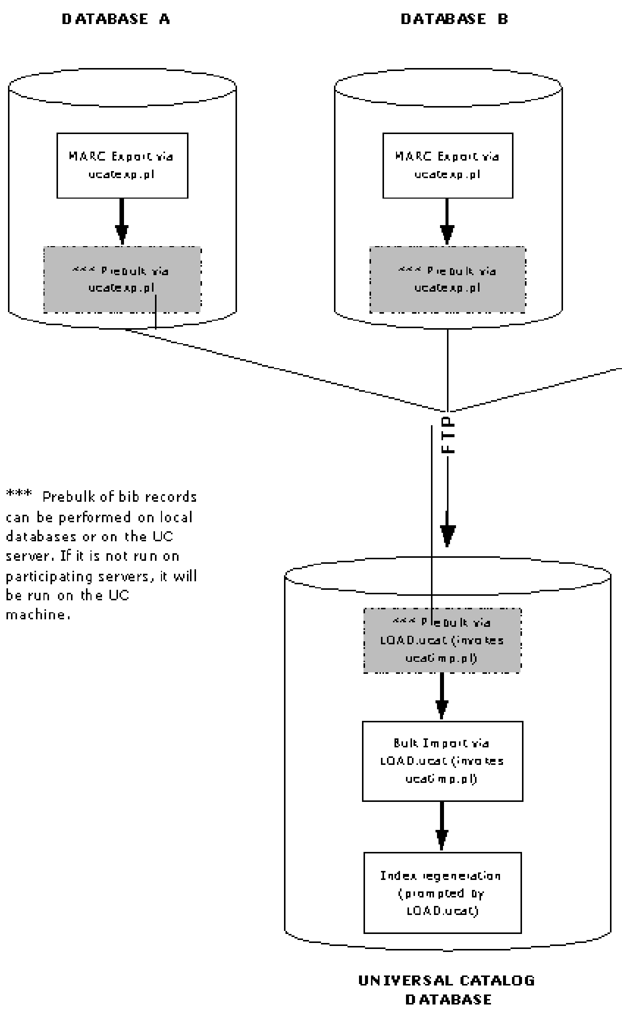
Running the LOAD.ucat Program

The following is a typical command line for running the `LOAD.ucat` program. In the `/m1/voyager/`*xxx*`db/ucat/load` directory, enter the following.

```
nohup ./LOAD.ucdb &
```

Using nohup indicates that there is no hangup if a terminal session is terminated at some point throughout the running of `Pucatexp`. Using & indicates that the process should be run in the background, returning control to command prompt.

Figure 3-38 shows the entire process of loading records into the Universal Catalog database.


DATABASE A
DATABASE B
DATABASE C etc.
MARC Export via ucatexp.pl
*** Prebulk via ucatexp.pl
MARC Export via ucatexp.pl
*** Prebulk via ucatexp.pl
MARC Export via ucatexp.pl
*** Prebulk via ucatexp.pl
FTP
*** Prebulk via LOAD.ucat (invokes ucatImp.pl)
Bulk Import via LOAD.ucat (invokes ucatImp.pl)
Index regeneration (prompted by LOAD.ucat)
UNIVERSAL CATALOG DATABASE

*** Prebulk of bib records can be performed on local databases or on the UC server. If it is not run on participating servers, it will be run on the UC machine.

Figure 3-38. Initial load of records from contributing library databases into the Universal Catalog

Loading Authority Records

Authority records are loaded into the Universal Catalog using the same methodology as bibliographic records and MFHDs. However since it is our recommendation that authority records are not contributed by participating library databases but rather that the entire LC Authority Files are loaded, exporting authority records from local library databases is not necessary.

ucatimp.pl Program

Importing the LC Authority Files into the UC is performed by using the `ucatimp.pl` script (which during the initial load, is invoked by `LOAD.ucat`). This script Bulk Imports authority records in the same way in which bibliographic records and MFHDs are imported. See ucatimp.pl Program on page 3-22 and see Pucatimp on page 3-34 for details on `ucatimp.pl`.

In order for the LC Authority Files to be imported by using `ucatimp.pl`, they must be placed in the following directory on the UC server.

```
/m1/voyager/xxxdb/ucat/filequeue
```

This is the same directory in which bibliographic records and MFHDs are automatically placed for import. Once the `ucatimp.pl` program is invoked by `LOAD.ucat`, the files of authority records are picked up at the same time as the bibliographic records and MFHDs.

For the UC system to know in which tables to place authority records, the files being placed in `/m1/voyager/`*`xxx`*`db/ucat/filequeue` on the UC server must have the `.auths` file extension, not the `.bibmfhd` file extension. The `.auths` file extension tells the system that the records should be placed into the authority record tables. In addition, files of authority records should be named following the *`prefix.timestamp.agency.suffix`* convention, (where prefix is upd, timestamp follows the YYYYMMDDHRMISS format, agency is the NUC code of the modifying agency, and suffix is auths).

NOTE:
Including authority records in the UC requires the creation of an Authority Duplicate Detection Profile and a Bulk Import Rule specifically for authority records. See Authority Duplicate Detection Profile on page 2-29 and see Bulk Import Rules on page 2-35 respectively for details.

Ongoing Updating of UC Records

Ongoing updates to the UC database are imperative because records are continually added, deleted, and modified in the databases of contributing libraries. The following scenarios shown in Table 3-27 illustrate how additions, deletions, and changes to records in local databases (including record exclusion or suppression) are handled in the UC after the UC database has been initially loaded.

Table 3-27. Processing logic for UC updates

Bibliographic Record Activity on the Local Database of Contributing Libraries	Result in the Universal Catalog Database
Bibliographic record not existing in the UC is added in the database of one participating library	Bibliographic record and MFHD are added
Bibliographic record that has been contributed by several contributing library databases is added to the database of one library	MFHD is added, and bibliographic record is modified depending on bibliographic duplicate detection profile
Bibliographic record is modified in one contributing library database	Bibliographic record is modified (depending on bibliographic duplicate detection profiles)
Sole contributing database for a title deletes its bibliographic record	Bibliographic record and corresponding MFHD are deleted
Bibliographic record with holdings in several local libraries is deleted from database of only one contributing library	MFHD for the library that deleted its bibliographic record is deleted
Bibliographic record with holdings in only one local library database is suppressed from the database of that library	Bibliographic record and corresponding MFHD are deleted
Bibliographic record with holdings in several local library databases is suppressed from the database of one local library	MFHD for the library that suppressed its bibliographic record is deleted
All MFHDs are deleted from the database of the only library to hold the bibliographic record	No result (the bibliographic record still exists at the contributing location)

Table 3-27. Processing logic for UC updates

Bibliographic Record Activity on the Local Database of Contributing Libraries	Result in the Universal Catalog Database
Bibliographic record with holdings in several local library databases is identified as a record eligible for exclusion from export in the database of one local library (`-a` is specified when Pucatexp is run)	MFHD for the library that excluded the bibliographic record is deleted from the UC database (Assumption: Bibliographic record has been previously exported to the UC)
Bibliographic record with holdings in several local library databases has its create location specified in `export UCATLOCEXCLUDEFILE=/m1/voyager/xxxdb/ucat/configs/filename`	MFHD for the local library is withheld from export
Bibliographic record with holdings in several local library databases has its most recent activity location specified in `export UCATLOCEXCLUDEFILE=/m1/voyager/xxxdb/ucat/configs/filename`	MFHD for the local library is withheld from export
Bibliographic record with holdings in several local library databases has its most recent activity location specified in `export UCATLOCEXCLUDEFILE=/m1/voyager/xxxdb/ucat/configs/filename` and record exists in the UC	MFHD remains in the UC
Bibliographic record with holdings in several local library databases has its most recent activity location specified in `export UCATLOCEXCLUDELASTFILE=/m1/voyager/xxxdb/ucat/configs/filename`	MFHD for the local library is withheld from export
Bibliographic record with holdings in several local library databases has its most recent activity location specified in `export UCATLOCEXCLUDELASTFILE=/m1/voyager/xxxdb/ucat/configs/filename` and record exists in the UC (`-a` is specified when Pucatexp is run)	MFHD is deleted from the UC

Table 3-27. Processing logic for UC updates

Bibliographic Record Activity on the Local Database of Contributing Libraries	Result in the Universal Catalog Database
Bibliographic record with holdings in several local library databases doesn't have any locations associated with actions taken (as, for example, when initially loaded) and `UCATLOCEXCLUDELASTFILE=/m1/voyager/xxxdb/ucat/configs/filename` is specified with Pucatexp processing causing program to look for exclude locations See the Note on page 3-18.	MFHD is exported to the UC

How Updates are Performed

The update process itself is similar to the initial load of records into the UC. It is still necessary for bulk export of MARC records, Prebulk, and Bulk Import to be performed; and scripts are still invoked in order to do so. However for the purpose of ongoing updates, the two programs used to invoke bulk export of MARC records, Prebulk, and Bulk Import are `Pucatexp` and `Pucatimp`. The `LOAD.ucat` program is not used for ongoing updates. The `Pucatexp` and `Pucatimp` programs are intended to be run as dæmons or cronjobs.

IMPORTANT:
The same Voyager System Administration settings defined prior to the initial load are used for ongoing updates. Changing the settings at any time throughout the life of the UC database can cause serious inconsistencies and repercussions on records such as multiple duplicates. The only time Voyager System Administration settings can be safely altered or replaced is when the Universal Catalog is being completely rebuilt. For details on configurations set in the Voyager System Administration module, see Universal Catalog System Setup: Pre-Initial Load on page 2-1.

Pucatexp Program

Pucatexp calls ucatexp.pl, the same program used during the initial load of records into the UC. See ucatexp.pl on page 3-6 for additional information regarding ucatexp.pl. The primary difference between processing database updates and the initial database load is that ongoing updates should be run by way of a dæmon or a **cronjob** due to the repetitive nature of the ongoing updates.

Setting Up the Pucatexp Program as a Dæmon

The Pucatexp program was created with the intention of it being run as a dæmon or a cronjob. Running the program as a dæmon requires the use of the -e (intervals) switch which is the only difference between setting up the program as a dæmon and running it manually. This causes the program to run periodically to update your database independent of human intervention. The -m (modifying agency) switch then gets passed on to bulk export of MARC records.

Intervals at which the program should run vary depending on the needs of participating libraries. Some contributing libraries may catalog a large number of records daily requiring records to be exported to the UC often. Other contributing libraries may catalog a small number of records daily, thus requiring records to be exported to the UC database less frequently. Constraints to be considered in deciding on intervals include the number of records cataloged in each participating library database and the speed with which updates need to reach the Universal Catalog.

The following is an example of a typical command line for setting up the Pucatexp as a dæmon running every 30 minutes. When logged in as Voyager in the /m1/voyager/*xxx*db/sbin directory, enter the following (the parameters between the square brackets are optional).

```
nohup ./Pucatexp -i207.87.90.34
-d/m1/voyager/UCAdb/ucat/incoming -mEISI
-e30[-p/m1/voyager/ucat/config
-X/m1/voyager/UCAdb/ucat/config] &
```

Using nohup indicates that there is no hangup if a terminal session is terminated at some point throughout the running of Pucatexp. Using & indicates that the process should be run in the background, returning control to command prompt.

NOTE:
If you wish to set up the Pucatexp to run as a cronjob, the use of the -e switch is not required. Rather, let the cronjob settings take care of the export intervals. Depending on what software you have, your system may perform cronjobs

differently. For information on setting up `Pucatexp` as a cronjob, talk to your Server Administrator or see your UNIX documentation.

Timestamp

Participating libraries must decide the date and time the bulk export of MARC records process should begin. This is known as a timestamp which follows the *`YYYYMMDDHRMISS`* format.

To indicate this to `ucatexp.pl`, use an empty file in the `/m1/voyager/`*`xxx`*`db/ucat/processed` directory, and give it a name that follows the file formats discussed in the previous section of this user’s guide. The `ucatexp.pl` program always leaves the last file it processed in the `/m1/voyager/`*`xxx`*`db/ucat/processed` directory for future reference. If `ucatexp.pl` cannot find such a file in this location, it begins the export from midnight of the current day *`YYYYMMDD`*`000000`.

Pucatimp

`Pucatimp` calls `ucatimp.pl`, the same program used during the initial load of records into the UC. For the purpose of ongoing updates of the UC database, the `ucatimp.pl` program performs the same functions as during the initial load process. It performs Prebulk if it was not run locally and Bulk Import of records into the UC. The `ucatimp.pl` program can tell if Prebulk was run locally by looking at the file extensions of incoming records. If incoming records have the `.bib` file extension, `ucatimp.pl` runs Prebulk. If incoming records have the `.bibmfhd` file extension, `ucatimp.pl` does not run Prebulk. The files with records to be imported are picked up by `ucatimp.pl` from the `/m1/voyager/`*`xxx`*`db/ucat/filequeue` directory on the UC server. This is the directory in which records are automatically placed for import.

There is one major difference between how the `ucatimp.pl` program works for ongoing updates compared to the initial load. For ongoing updates, it is invoked by a shell script `Pucatimp` located in the `/m1/voyager/`*`xxx`*`db/sbin` directory.

Command Line Parameters

The Pucatimp program offers the valid command line parameters shown in Table 3-28 on page 3-35.

Table 3-28. Valid command line parameters for ucatimp.pl

PARAMETER	DESCRIPTION
-I	Import intervals in minutes (if not running as a cronjob).
-i	Bulk Import rule code. If you specify a Bulk Import rule using this switch, that same rule is used for importing records from all participating library databases. If the ucatimp.pl program is run without this switch, it looks up a Bulk Import rule for the local database based on the modifying agency code embedded in the import filename. In this way, a different import rule can be used for each participating library database. If ucatimp.pl fails to find an import rule match at this time, ucatimp.pl uses a rule of default.
-o	Operator code for Bulk Import.
-l	Location code for Bulk Import.
-a	MFHD location code for Bulk Import. Using this switch causes all the locations in the Universal Catalog to have the same, single location code. As a result, we recommend that you do not use this switch.
-k	OK to export for Bulk Import.

Setting Up the Pucatimp Program as a Dæmon

The Pucatimp program, which resides in the /m1/voyager/bin directory of the UC server, is intended to be run for ongoing updates as a dæmon or a cronjob**.**

Running the program as a dæmon requires the use of the -I (intervals) switch, which is the only difference between setting up the program as a dæmon and running it manually. Invoking the program as a dæmon causes the Pucatimp program to run periodically and independent of human intervention which is much more efficient for updating UC records.

Intervals at which the program should run vary depending on the needs of participating libraries. Some contributing libraries may catalog a large number of records daily requiring records to be imported into the UC often. Other contributing libraries may catalog a small number of records daily requiring records to be imported into the UC database less frequently. Constraints to be considered in deciding on intervals include the volume of records cataloged in each local library database and the desire to make the UC database seem like it is operating in real time (in other words, make records cataloged by participating libraries seem like they go into the UC database at the same time as they are being cataloged). The following is an example of a typical command line for setting up the `Pucatimp` as a dæmon running every 30 minutes. When logged in as voyager in the `/m1/voyager/xxxdb/sbin` directory, enter the following.

```
nohup ./Pucatimp -I30 &
```

Using nohup indicates that there is no hangup if a terminal session is terminated at some point throughout the running of `Pucatexp`. Using & indicates that the process should be run in the background, returning control to command prompt.

NOTE:
If you wish to set up the `Pucatimp` program to run as a cronjob, the use of the `-I` switch is not required. Rather, let the cronjob settings take care of the import intervals. Depending on what software you have, your system may perform cronjobs differently. For information on setting up `Pucatimp` as a cronjob, talk to your Server Administrator or refer to your UNIX documentation.

Log File

The `Pucatimp` program generates a log file, containing activities and problems the program encountered during its last run such as number of records processed or deletions of bibliographic records. The log file includes information regardless of the number of records processed by `ucatimp.pl` (zero or many records). The log file is named with a date and time stamp in the following format.

```
log.ucimp.YYYYMMDDHHMMSS.PID
```

The log file is placed in subdirectories that are created automatically by the program on the UC server under the `/m1/voyager/xxxdb/ucat/rpt` **d**irectory. These subdirectories are named for the local database (actually the modifying agency code) whose records are being imported by `Pucatimp`.

For example, if the UC server is importing records from a local library database with a modifying agency code of dev2000db, `Pucatimp` writes its logs into the `/m1/voyager/xxxdb/ucat/rpt/dev2000db` directory on the UC server.

The ucatutils.pm file

This module file works behind the scenes during the ongoing update process. For purposes here, it is only necessary to know that you see this file in the `/m1/voyager/bin` directory.

Adding Records from New Participating Libraries

Records from new participating libraries can be merged into an existing Universal Catalog database at any time without requiring a complete reload of existing data. This process may also be referred to as “top-up loads.”

NOTE:
During the load of records from new participating libraries, the UC database itself remains available for searching.

How Records from New Libraries are Loaded

The process of adding records from new contributing libraries is identical to the initial load process. For detailed instructions on the process, see *Initially Loading Records into the UC* on page 3-6.

Loading records from new participating libraries requires some additional Voyager System Administration setup on the UC machine, including the following.

- Defining a location for the new participating library database. See Locations for Contributing Libraries on page 2-2 for details.
- Associating the new location with the UC Owning Library. See UC Owning Library on page 2-7 for details.

The following Voyager System Administration setup is possible but not required for adding new participating libraries to the UC.

- Associating the location for the new library database with the Cataloging Policy Group established for the UC. See Operator Profiles on page 2-48 for details.
- Associating the location for the new database with a Cataloging Security Profile, in order to access the Cataloging module for troubleshooting purposes (with view-only access). See Cataloging Security Profile on page 2-54 for details.
- Adding the new participating library database to the Quality Hierarchy for the UC bibliographic duplicate detection profile. For example, you may want to add a new library to the Quality Hierarchy if it is universally agreed upon that the library will be contributing the best bibliographic records for a certain record type like serials. In that case, adding the new library to the Quality Hierarchy ensures that the serials records contributed by the database of that library are used ahead of others.

Adding a new database to the Quality Hierarchy only affects records that are added, modified, or deleted to the UC after the Quality Hierarchy is adjusted. It does not affect existing UC records.

See Universal Catalog System Setup: Pre-Initial Load on page 2-1” for details on pre-initial load configurations.

UC to Local

UC to Local, an optional function within the Universal Catalog environment, provides the capability to do a reverse export. That is, bibliographic records stored in the Universal Catalog may be exported/extracted so that contributing local databases may import these records.

NOTE:
Activation of the UC to Local function needs to be implemented by Ex Libris staff. Contact your Project Manager to communicate your requirements.

Why might you consider importing bibliographic records from the UC? The result of exporting bibliographic records from several local databases to the UC database is that one, “best” bibliographic record is stored in the UC. The UC to Local reverse export function, therefore, provides the capability to share the “best” bibliographic record with all the local databases that hold this bibliographic record but were not the originating source of the “best” bibliographic record as stored in the UC.

To accomplish this, there should be an extract (reverse export) from the UC for each contributing library database. If a higher quality bibliographic record is extracted, each local library that has a stub holdings record associated with the updated bibliographic record receives the updated, “best” bibliographic record. The UC to Local extraction process is based on the holding location and modify date information.

The remainder of this section, “UC to Local,” focuses on the setup steps and the new script options required to enable reverse export.

UC Setup

On the UC database system, create directories for each local library using the following `/m1/voyager/ucdb/ucat` directory. For local libraries 1, 2, and 3, the directory setup may be set up as follows.

```
/m1/voyager/ucdb/ucat/locallib1/
/m1/voyager/ucdb/ucat/locallib2/
```

```
/m1/voyager/ucdb/ucat/locallib3/
```

As part of the process of exporting records from the UC, the software dynamically appends “outgoing” and “processed” directories to the path which results in the following path names.

```
/m1/voyager/ucdb/ucat/locallib1/outgoing/
/m1/voyager/ucdb/ucat/locallib1/processed/
/m1/voyager/ucdb/ucat/locallib2/outgoing/
/m1/voyager/ucdb/ucat/locallib2/processed/
/m1/voyager/ucdb/ucat/locallib3/outgoing/
/m1/voyager/ucdb/ucat/locallib3/processed/
```

The “outgoing” directory path contains the bibliographic records extracted from the UC database. The “processed” directory path contains the records that have been processed via FTP or via copying if the local database and UC database reside on the same server.

Contributing System Setup

On each local library database system, you need to perform the following steps.

- Create a default bulk import rule (see Bulk Import Rule on page 3-39).
- Create a new, limited-use operator ID (see Operator ID on page 3-40).
- Create directory path (seeDirectory Path on page 3-40).

Bulk Import Rule

Create a new bulk import rule using the parameter settings shown in Table 3-29 on page 39. The steps for creating a bulk import rule are located in Creating Bulk Import Rules on page 2-36.

Table 3-29. Bulk Import Rule Settings for UC to Local

Parameter	Setting
Rule Name	default
Rule Code	default
Bib Dup Profile	OCLCReplace
Auth Dup Profile	AuthorityReplace
MARC Character Set Mapping	RLIN

Table 3-29. Bulk Import Rule Settings for UC to Local

Parameter	Setting
What to Import	Load Bibs/Auths Only

Operator ID

The local database system requires an operator ID unique to the UC to Local process. It is not to be used by staff for any other purpose.

The purpose of this ID is to identify bibliographic records on the local database that have been imported from the UC database using the UC to Local function and have not been modified by local staff. When bibliographic records are imported to the local database from the UC database, the bib_history table is updated with the “unique” operator ID attached to these records. As a result, these records can later be identified for exclusion when database exports are created from the local database to be imported into the UC database.

Create the new operator ID on the local database system with the following characteristics.

- Unique name easily identified as associated with the UC to Local function, perhaps “UC.”
- Privileges for add/update bibliographic records and authority records.

Directory Path

On the local system, a basic directory path is created as part of the standard implementation process. Library 1, for example, may have a path setup as follows.

```
/m1/voyager/locallib1/ucat/
```

As part of the process of importing records from the UC, the software dynamically appends `/reverse/processed/` directories to the path which results in the following.

```
/m1/voyager/locallib1/ucat/reverse/processed/
```

The “processed” directory contains the bibliographic records successfully imported on the local library server.

Script Options

There are two scripts used with the UC to Local function, `Pucatexp` and `Pucatimp`. Parameters specific to the UC to Local process are available for use with `Pucatexp` and `Pucatimp`. See Table 3-30 and Table 3-31 for more information.

`Pucatexp` is used to extract/export bibliographic records from the UC database for each local database. `Pucatimp` is used to import these bibliographic records to each local database.

The script options available for use with `Pucatexp` are shown in Table 3-30 on page 41.

Table 3-30. Script parameters for Pucatexp

Parameter	**Description**
-U <location> [<ip address>] <destination directory>	Indicates that you are doing a reverse export. Identifies the local library for whom the extract is being performed. Specifies the IP address for the local library database. When the UC database and the local library database reside on the same server, use 127.0.0.1 for the IP address. The copy function is used instead of FTP when both databases reside on the same system. Specifies the path in which to store the extracted/exported bibliographic records on the local library system. See UC Database Export Example on page 3-42. Use a colon to separate the different components of this command string. See UC Database Export Example on page 3-42.
-E <operator id>	Used to exclude records from an extract. Operator ID is required. Specifies operator ID that is associated with bibliographic records to exclude from the local library export.

The script options available for use with `Pucatimp` are shown in Table 3-31 on page 3-42.

Table 3-31. Script parameters for Pucatimp

Parameter	Description
-U	Indicates that you are doing a reverse import.
<operator id>	Specifies the operator ID to associate with each imported “best” bibliographic record. The same ID is used for `Pucatexp` to exclude these records from the extract that have not been modified on the local library database.

UC Database Export Example

The following script example is for environments where the UC database and the local library database reside on different systems. FTP is used to move records to the defined destination.

```
Pucatexp -Ulocallib1:130.58.22.0:/m1/voyager/2001.1/
xxxdb/ucat/incoming
```

Specifying “incoming” as in this example is required for environments where the UC database and the local library database reside on separate servers.

The following script example is for environments where the UC database and the local library database reside on the same system. The copy function is used to move records to the defined destination.

```
Pucatexp -Ulocallib1:127.0.0.1:/m1/voyager/2001.1/
xxxdb/ucat/filequeue
```

Specifying “filequeue” as in this example is required for environments where the UC database and the local library database reside on the same server.

Local Library Database Export Example

The following script example is used for exporting bibliographic records from a local library and excluding from the extract those bibliographic records previously imported from the UC database and identified by the unique operator ID created for the UC to Local function. See Operator ID on page 3-40 for more information.

```
Pucatexp -EUC
```

UC is the unique operator ID created for the UC to Local function.

Local Library Database Import Example

The following script example is used for importing bibliographic records into the local library database and flagging each imported record with the unique operator ID created for the UC to Local function. See Operator ID on page 3-40 for more information.

```
Pucatimp -UUC
```

UC is the unique operator ID in this example.

Searching the Universal Catalog

4

Searching the UC Database

Once the database is set up, you can search for records stored in the Universal Catalog using the Voyager WebVoyáge client. Records in the UC can be searched in one of two ways, locally or remotely. A local search occurs when the UC database is searched directly while a remote search occurs when the UC database is searched indirectly by way of simultaneous searching or by way of search redirect.

Search Setup

Before you can search a UC database, you must provide information about the type of connection to be made to the UC server. Configuring a UC database is exactly the same as configuring any other Voyager database. This setup involves both WebVoyáge configuration files and the Voyager System Administration module for both the UC database and the contributing library database systems.

See Before You Load Records on page 2-1 for System Administration setup information and the *Voyager WebVoyáge User's Guide* for additional WebVoyáge setup information.

WebVoyáge Configuration Files

WebVoyáge configuration files focus on how WebVoyáge displays various pieces of information, including search results from the UC server. For example, in the `connect.ini` configuration file, you can specify how all of the searchable databases display on the Database page in WebVoyáge; and in the `displayn.cfg` configuration file, you can specify the types of information that appear in the record display.

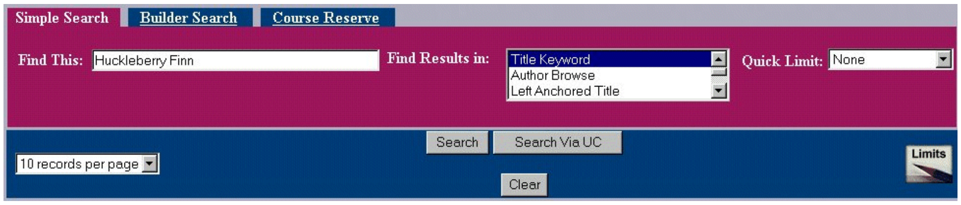
Opac.ini - [NoHits_Redirect] Function

WebVoyáge configuration files are located in the following directory and its subdirectories on the UC server: `/m1/voyager/`*`xxx`*`db/etc/webvoyage`. Configuration files have either `.ini` or `.cfg` file extensions. Of particular interest is the `opac.ini` file that contains a stanza for the search redirect functionality.

```
[NoHits_Redirect]

RedirectEnabled= {Y or N; N is the default}

ReSearchSubmitButton=Search Another Database

DatabaseToSearch= {no default}
```

When a search by a patron concludes with the message, “Your search resulted in no hits!,” WebVoyáge can dynamically update the display with a search redirect button. For purposes of the Universal Catalog system, this button can point to another database to search and for the local libraries it can point to the UC database. The display maintains the original search criteria so that the patron can simply click the search redirect button to search another database.

Figure 4-39 shows how search redirect on a local library system quickly enables searching on the Universal Catalog database.



Find This: Huckleberry Finn

Find Results in: Title Keyword

Quick Limit: None

10 records per page

Figure 4-39. Search Redirect Button ... “Search Via UC”

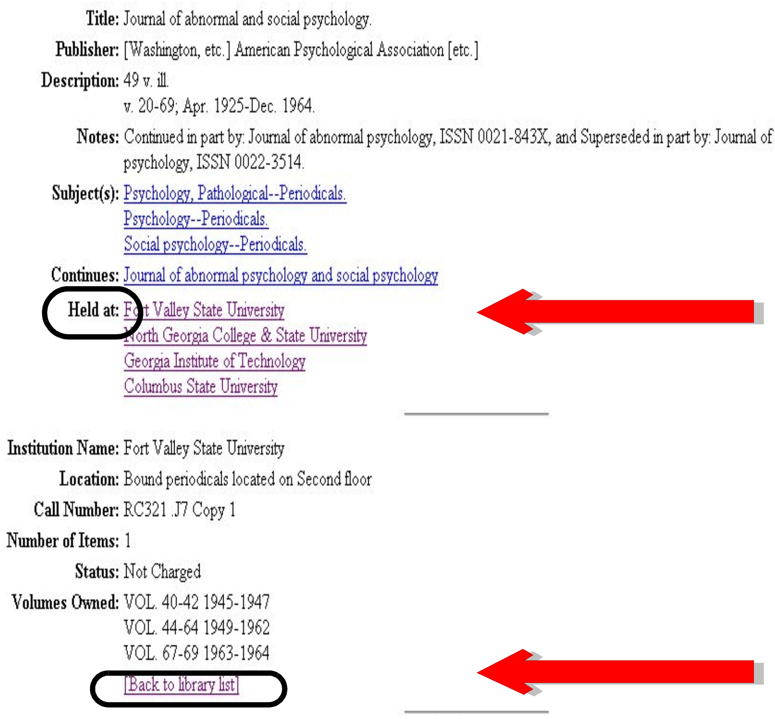
The `[NoHits_Redirect]` stanza needs to be customized to match your preferences. To invoke this new function, **Y** (yes) needs to be entered for the `RedirectEnabled=` option. The text you wish to appear on the redirect button should be entered in the `ReSearchSubmitButton=` option. Use the `DatabaseToSearch=` option to point to another database or the UC database from a local library.

Opac.ini - Jump Bar Function

Within the `opac.ini` file is the “Jump Bar” function specified in the `[View_Record_Page]` stanza.

```
[View_Record_Page]
...
DBHoldingsJumpBar=Y
DBHoldingsJumpBarText=Held at:
DBHoldingsJumpBarBack=Back to library list
```

Implementing this function generates the following search results viewing capability. See Figure 4-40.



Title: Journal of abnormal and social psychology.
Publisher: [Washington, etc.] American Psychological Association [etc.]
Description: 49 v. ill.
v. 20-69; Apr. 1925-Dec. 1964.
Notes: Continued in part by: Journal of abnormal psychology, ISSN 0021-843X, and Superseded in part by: Journal of personality and social psychology, ISSN 0022-3514.
Subject(s): Psychology, Pathological--Periodicals.
Psychology--Periodicals.
Social psychology--Periodicals.
Continues: Journal of abnormal psychology and social psychology
Held at: Fort Valley State University
North Georgia College & State University
Georgia Institute of Technology
Columbus State University

Institution Name: Fort Valley State University
Location: Bound periodicals located on Second floor
Call Number: RC321 .J7 Copy 1
Number of Items: 1
Status: Not Charged
Volumes Owned: VOL. 40-42 1945-1947
VOL. 44-64 1949-1962
VOL. 67-69 1963-1964
[Back to library list]

Figure 4-40. Holdings Jump Bar Feature

This feature enables the patron to quickly move through a long holdings list by providing hyperlinks to the holding libraries. Clicking one of the **Held at:** links takes the patron to a specific library’s holdings in the list. Clicking the **Back to library list** hot link returns the patron to the list of holding libraries.

If there is only one holding library in the search results, no Jump Bar capability is provided.

This function can be tailored to local preferences. `DBHoldingsJumpBar=` can be set to `Y` (yes) or `N` (no). To invoke the “Jump Bar” function, this needs to be set to `Y` which is also the default. `DBHoldingsJumpBarText=` specifies the text that precedes the list of holding databases. “Held at:” is the default text. It can be changed to match local preferences.

NOTE:
The order of the resulting displayed “Held at:” locations is determined by weightings set in database definitions.

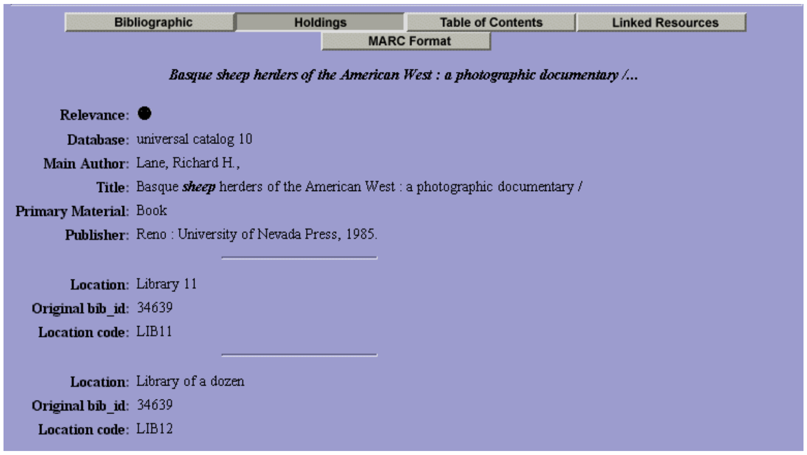
Similarly, `DBHoldingsJumpBarBack=` specifies the text that displays for the end-user to click to “jump back” to the list of holding databases. This text, too, can be modified to match local preferences.

For detailed information on setting up WebVoyáge configuration files, see the *Voyager WebVoyáge User’s Guide*.

Performing Searches

You search the UC database locally or remotely in the same way as you search any other OPAC using WebVoyáge. However, remember the following.

- Records you retrieve from the UC database come from many library databases as compared to just one.
- Typically, MFHD records in the UC serve as pointer records and dynamically retrieve detailed holdings and item information when a selection is made.
- Optionally, MFHDs in the UC can have call number information stored in the 852‡h and 852‡i fields. As a result, call number searching can be performed. See “Call Numbers in MFHDs” starting on A-1 for special considerations when call numbers are stored in the UC MFHD records.



Basque sheep herders of the American West : a photographic documentary /...

Relevance: ●
Database: universal catalog 10
Main Author: Lane, Richard H.,
Title: Basque ***sheep*** herders of the American West : a photographic documentary /
Primary Material: Book
Publisher: Reno : University of Nevada Press, 1985.

Location: Library 11
Original bib_id: 34639
Location code: LIB11

Location: Library of a dozen
Original bib_id: 34639
Location code: LIB12

Figure 4-41. WebVoyáge record display for a UC record

Aside from these considerations, you can search the Universal Catalog as you do any other Voyager OPAC. Standard features can be used, such as search limits and simultaneous searching.

For detailed information and instructions on searching for records, see the *Voyager WebVoyáge User's Guide*.

Universal Borrowing and the Universal Catalog

Voyager's Universal Borrowing (UB) provides a structure for reciprocal borrowing in a Universal Catalog setting. With Universal Borrowing, a patron can request, borrow, and return materials from any Universal Catalog (UC) participating library. Borrowed items are returnable to any Universal Borrowing library and all internal tracking is noted in real time for efficiency and accuracy of records in all libraries. Especially essential in consortia settings, Universal Borrowing provides the structure needed for libraries to manage the many aspects of patron borrowing to include the requirements of distance education programs.

For information about Universal Borrowing and its setup, the *Voyager Universal Borrowing User's Guide* is your primary source of information. Some considerations specific to the Universal Catalog environment with the addition of Universal Borrowing are highlighted in this section.

Making Requests - Considerations

In general, making requests in a combined UC/UB environment is quite similar to making requests in a single database environment. The differences that may occur are the result of customizing differences between the local server and the Universal Catalog server.

If a patron accesses the Universal Catalog database directly, the system setup customized for users on that system determines the services (such as system features as specified in Voyager System Administration, the `opac.ini`, or the `connect.ini`) delivered to that user.

However, it is also possible that a patron may initially access a local library database and subsequently access the Universal Catalog by way of Search Redirect. In this case, the services delivered to a user are determined by the local library database server.

See Figure 4-42 on page 4-7 for a diagram highlighting the results from different methods of accessing the Universal Catalog database.

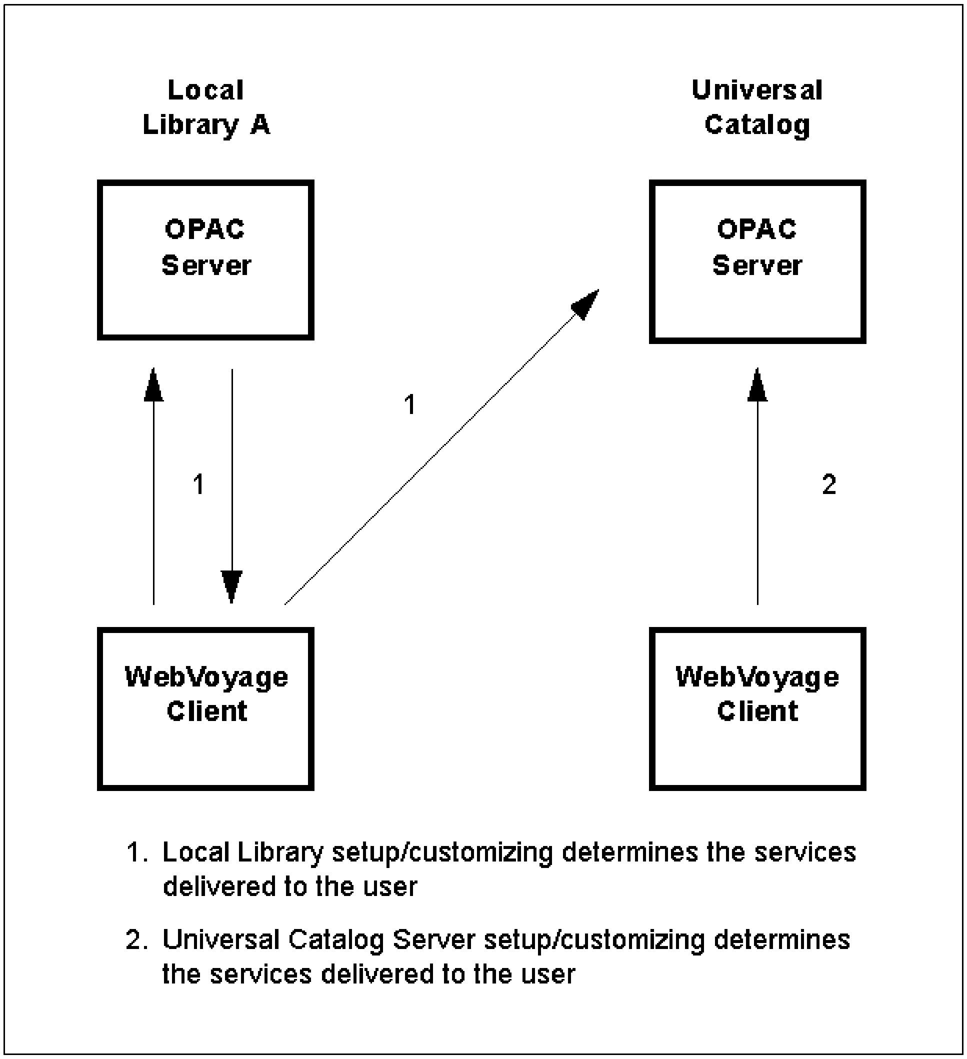

Local
Library A
Universal
Catalog
OPAC
Server
OPAC
Server
1
1
2
WebVoyage
Client
WebVoyage
Client
1. Local Library setup/customizing determines the services delivered to the user
2. Universal Catalog Server setup/customizing determines the services delivered to the user

Figure 4-42. Access Method Affects Services Delivered to the User

The initial point of entry of the patron into the system determines what services (such as system features as specified in Voyager System Administration, the `opac.ini`,or the `connect.ini`) are invoked.

Call Slip Considerations

For call slip information to be routed through the Universal Catalog system, default Call Slip definitions need to be configured. Procedure 4-13, Configuring Call Slips - Queues provides the steps and values required to set up the default Call Slip configuration. For more information, see the *Voyager System Administration User’s Guide*.

The procedure for configuring Call Slips - Queues is shown in Procedure 4-13, Configuring Call Slips - Queues, on page 4-8.

Procedure 4-13. Configuring Call Slips - Queues

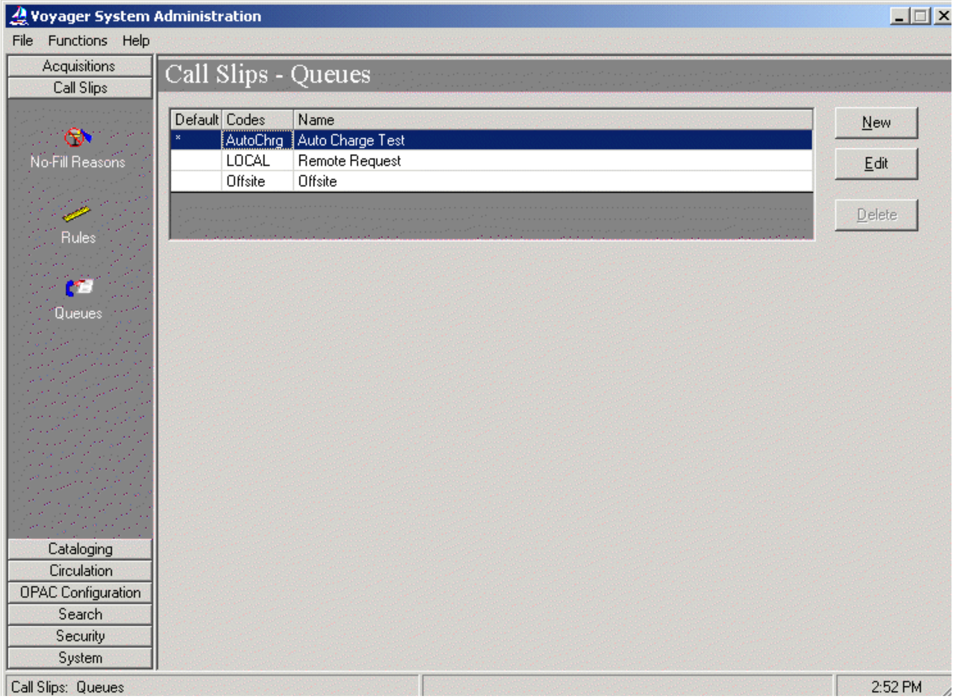
Use the following to configure the **Call Slips - Queues** options in Voyager System Administration for the Universal Catalog/Universal Borrowing environment.

1. Click **Call Slips** from the vertical listbar in Voyager System Administration.

   Result: The Voyager System Administration **Call Slips** main dialog box opens.

2. Click **Queues**.

   Result: The **Call Slips - Queues** main dialog box opens. See Figure 4-43.

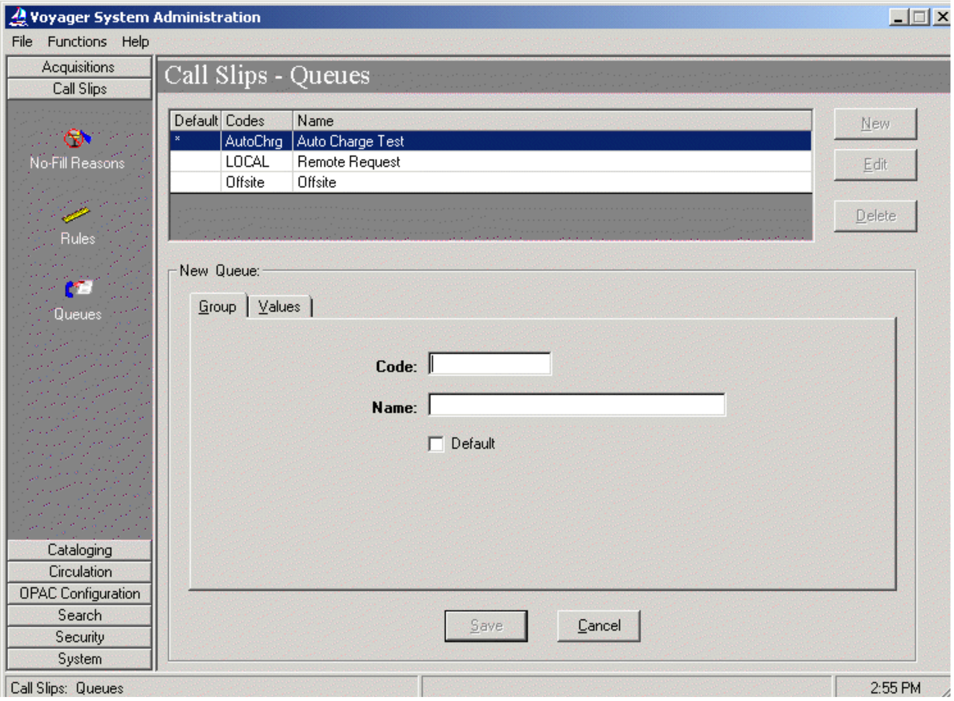


Default	Codes	Name
*	AutoChrg	Auto Charge Test
	LOCAL	Remote Request
	Offsite	Offsite

Figure 4-43. Call Slips - Queues main dialog box

3. Click **New**.

Result: The **Call Slips - Queues Group** tab options open for **New Queue**. See Figure 4-44.



Default	Codes	Name
*	AutoChrg	Auto Charge Test
	LOCAL	Remote Request
	Offsite	Offsite

Code:

Name:

□ Default

Figure 4-44. Call Slips - Queues Group tab

4. Enter the **Code**, **Name**, and **Default** options to match your preferences. See Table 4-32 for more information about these options.

   Result: This completes the **Group** tab component of the **Call Slips - Queues** definition.

Table 4-32. Call Slips - Queues Group tab options

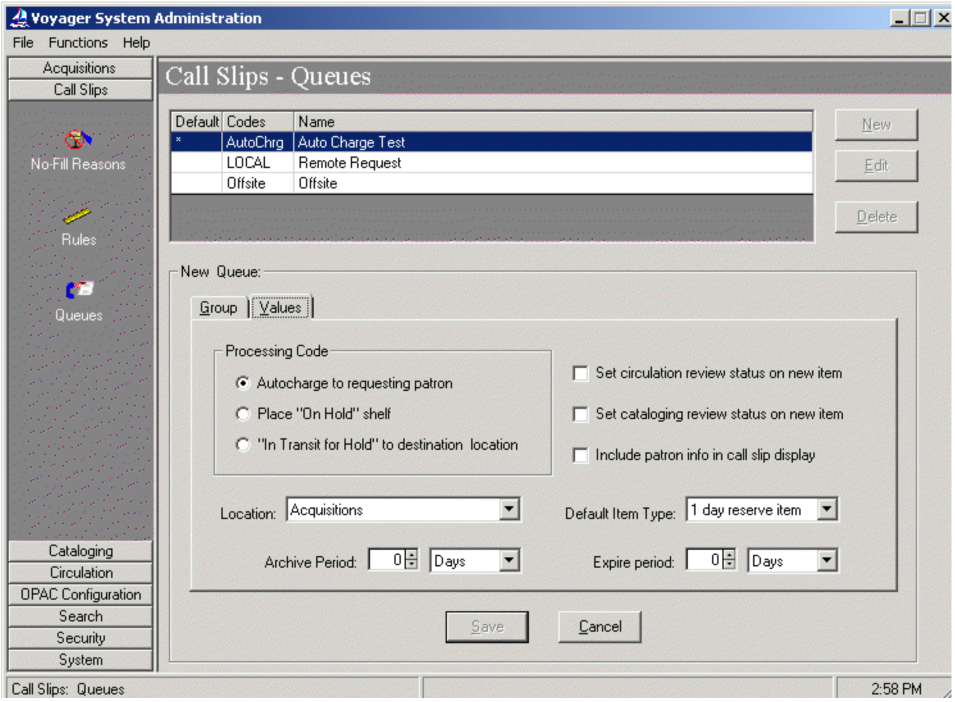
Option	Description
Code	Enter the Code for the call slip queue, up to 10 alphanumeric characters in length. Since only one Call Slips - Queues definition is required for the Universal Catalog, you may want to consider “default” as the naming convention for this field.

Table 4-32. Call Slips - Queues Group tab options

Option	Description
Name	Enter the **Name** for the call slip queue, up to 25 alphanumeric characters in length. Since only one Call Slips - Queues definition is required for the Universal Catalog, you may want to consider “default” as the naming convention for this field.
Default	Check the Default box to make this call slip queue the default. **NOTE:** Any call slip requests that do not match any of the other call slip queue definitions are sent to the queue which is identified as the Default.

5. Click the **Values** tab.

   Result: This displays the **Values** tab options. See Figure 4-45.



Call Slips - Queues

Default	Codes	Name
*	AutoChrg	Auto Charge Test
	LOCAL	Remote Request
	Offsite	Offsite

New Queue:

Processing Code

- ● Autocharge to requesting patron
- ○ Place "On Hold" shelf
- ○ "In Transit for Hold" to destination location

□ Set circulation review status on new item

□ Set cataloging review status on new item

□ Include patron info in call slip display

Location: Acquisitions

Default Item Type: 1 day reserve item

Archive Period: 0 Days

Expire period: 0 Days

Figure 4-45. Call Slips - Queues Values tab

6. Enter the **values** to match your requirements for the Universal Catalog. Set the **Processing Code** value to **“In Transit for Hold” to destination location**. See the Voyager System Administration User’s Guide for a complete description of all the **Values** tab options.

   Result: This completes the definition of values for the **New Queue**.

7. Click **Save** or click **Cancel**.

   Result: This saves or cancels the **New Queue** definition.

Call Numbers in MFHDs

A

Why Include Call Numbers In The UC MFHD Records?

Two main reasons are cited for including call numbers in the Universal Catalog MFHD records.

- The ability to search the Universal Catalog database using call number browse as search criteria.
- The ability to display a default call number when a dynamic connection cannot be made to the server of a participating library to pull detailed MFHD/Item information.

Considerations for Including Call Numbers in MFHDs

The call numbers stored in the stub MFHD records of the Universal Catalog database are pulled from bibliographic records owned by the contributing library. Depending on the method at the contributing library for creating and storing call numbers in the contributing database, the call numbers pulled from the bibliographic records on the database of the contributing library may or may not provide the desired results when retrieved from the MFHD records stored in the Universal Catalog database.

Are the call numbers stored in the bibliographic records and the MFHD records of UC contributing library databases the same? If the answer to this question is yes, the call numbers stored in the Universal Catalog MFHD records should provide the desired results when call number-specific information is being sought from the UC database. If the answer to this question is no, you may not want to store call number data in the Universal Catalog database.

There are many different places where call number information may be stored in a bibliographic record such as in 050, 090, or 082. Table A-1 on page A-2 highlights an example of how the 852‡h and 852‡i fields in the UC MFHD records may be sourced.

Table A-1. Call Number Retrieval/Storage Example

“Pulled From” **(Local Library Database** **Bibliographic Record)**	**“Stored In”** **(Universal Catalog** **MFHD Record)**
050 ‡a ===>	852 ‡h
050 ‡b ===>	852 ‡i

There are many variables that may affect what call number information is retrieved during a search when call numbers are stored in the stub MFHD records of the UC database. As a result, each group installing a Universal Catalog database needs to evaluate its circumstances and the benefits of incorporating call numbers into the MFHD records.

Steps to Include Call Numbers in UC MFHD Records

Customizing or tailoring the contents of the UC MFHD records to include call numbers is done through the Prebulk program. Specifically the Prebulk program looks to the Prebulk configuration file for instructions regarding how to manipulate the data stored in the files generated by `ucatexp.pl`. (See Initially Loading Records into the UC on page 3-6 for more information about `ucatexp.pl`.) In order for call numbers to be included in the UC MFHD records, the Prebulk Configuration file needs to direct the Prebulk program to pull the call number fields from the bibliographic record and store them in the UC MFHD record.

Prebulk Configuration File

The Prebulk configuration file consists of stanzas indicated by names in square brackets and followed by named fields. The file must be tab-delimited. This means that for any entries in the file in which multiple pieces of information must be specified on the same line, you must separate each piece of information with a tab. Prebulk does not recognize spaces or other characters as separators between pieces of information.

For more information on Prebulk configuration files, see the *Voyager Technical User's Guide*.

IMPORTANT:
When creating and editing a Prebulk configuration file, you must use the UNIX vi editor. Configuration files created and/or edited using other editors like Notepad cause Prebulk to fail.

Prebulk Configuration File Example

Figure A-1 on page A-3 illustrates a sample Prebulk configuration file. An explanation of each stanza in the Prebulk configuration file and how to tailor the options to generate call numbers in the UC MFHD records is provided in Table A-2 on page A-4.

NOTE:
This sample configuration file generates the same results whether Prebulk is run on the Voyager server of a contributing library or on the Universal Catalog server.

```
[OVERRIDES]
CREATEMFHD=YES
DEFAULTCALLNO=
DEFAULTCALLIND=
USE001FOR014=YES
USE003FORLOC=NO

[STRIP]
```

Figure A-1. Prebulk configuration file - Call Number example

```
[MFHDTAG]
999

[CALLTYPES]
050     0

[LOCATIONS]
default dev20001db      050, 090, 082

[MAPPING]
x       852b

[008]
000     6       am      9905280u    0   0
uuund|990528
```

Figure A-1. Prebulk configuration file - Call Number example (Continued)

[OVERRIDES] Stanza

```
[OVERRIDES]
CREATEMFHD=YES
DEFAULTCALLNO=
DEFAULTCALLIND=
USE001FOR014=YES
USE003FORLOC=NO
```

Table A-2. Prebulk Configuration File: [OVERRIDES] stanza

CREATEMFHD=YES	This field (not case-sensitive) indicates whether or not MFHDs are to be created by Prebulk. This is always set to YES for UC purposes. NO indicates that Prebulk will be run only to strip tags.

Table A-2. Prebulk Configuration File: [OVERRIDES] stanza

DEFAULTCALLNO=	This field should be left blank if you do not wish to use call numbers in the MFHDs for the UC. Optionally, you may enter a call number here to be placed in the 852‡h field if no call number is found in the `[MAPPING]` stanza.
DEFAULTCALLIND=	This field is only used if a value is found in the `[DEFAULTCALLNO]` stanza. It indicates the value for indicator one of the 852 field. Only a number between 0 and 8 are valid. Pipes and blanks are not acceptable. You can input any number as a value.
USE001FOR014=YES	Entering YES indicates that the 014 field will be created with a $a containing the value of the 001 tag. For UC purposes, this should always be set to YES. This field is not case sensitive.
USE003FORLOC=NO	Entering YES indicates that the incoming 003 tag will be the location used in the 852‡b field of the MFHDs. For UC purposes, this field is always set to `NO` because the 852‡b field is not being built off of the 003 tag. This field is not case sensitive.

`[STRIP]` Stanza

```
[STRIP]
```

Nothing is set in this stanza for the Universal Catalog, because fields of incoming bibliographic records are not being stripped during Prebulk.

`[MFHDTAG]` Stanza

```
[MFHDTAG]

999
```

The `[MFHDTAG]` stanza identifies which MARC tag contains the data to be used to create MFHDs. See Table A-3 on page A-6.

Table A-3. Prebulk Configuration File: [MFHDTAG] stanza

010 - 999, XXX	This field replaces the usual 9XX field used for mapping the 852. XXX is a valid field (tag) in the `[MFHDTAG]`. If XXX is used, the Call Number will not be created for the MFHD**.** XXX means that no holdings information is found and the default location in the `[LOCATIONS]` stanza should be used in the 852‡b**.** To add call numbers, the data line must include a 3 digit numeric value. [MFHDTAG] data must be 3 digits in length or XXX.

`[CALLTYPES]` Stanza

```
[CALLTYPES]

999     8
```

This stanza specifies what indicator is to be used when creating an 852 in a MFHD. See Table A-4 on page A-6.

Table A-4. Prebulk Configuration File: [CALLTYPES] stanza

099 8
050 0
090 0
082 1
086 2
	The `[CALLTYPES]` stanza should include all valid call types that can and may be found in the incoming bibliographic record.
	The 099 is a local field; and therefore, it must exist within the bibliographic record.
	NOTE:
	There should be a single tab between the call type and the first indicator. The first indicator should be a valid value 0-8.

`[LOCATIONS]` Stanza

```
[LOCATIONS]

default dev2001db      999
```

This stanza determines what location is to be used in MFHDs. The first line is the default setting, which is the only line to be entered for UC purposes. See Table A-5 on page A-7.

Table A-5. Prebulk Configuration File: [LOCATIONS] stanza

(1) default	**(1)** The first field location code represents a location in the incoming bibliographic record.
(2) DBNAME	**(2)** The second field location code which should be a valid SYSADMIN-defined location code is the location to be placed in the 852‡b of the MFHD. If the incoming data does not find a matching location code under the `[LOCATIONS]` stanza, the first entry that is used is the default entry.

(3) 090, 050(0), T	**(3)** The third field containing call types is placed in the 852‡h and 852‡i call number types for the specified location.
	The numeric value found in parenthesis in the third (3) field is the override value for the first indicator of the 852. It overrides whatever value is specified in the `[CALLTYPES]` stanza.
	Call types must be represented both in the `[CALLTYPES]` and `[LOCATION]` stanzas to be true. Prebulk generates a valid error message in the I/O display if a call type appears only in the `[LOCATION]` stanza.
	Example 1:
	`[LOCATIONS]`
	`default    DBNAME   050(0), 090, 082, 086`
	Example 2:
	(No call number created in this example.)
	`[LOCATIONS]`
	`ACQ     ACQ     T`

`[MAPPING]` Stanza

```
[MAPPING]
x     852b
```

This stanza indicates the subfields in the tag specified in the [MFHDTAG] stanza which contains specific data and where that data goes in MFHDs. See Table A-6 on page A-8.

Table A-6. Prebulk Configuration File: [MAPPING] stanza

x 852b	The x represents that no mapping for a 9XX field has taken place and the second field of the [LOCATION] stanza should be used for the 852‡b.

[008] Stanza

```
[008]

000      6      am      9905280u      0      0      uuund |
990528
```

This stanza specifies how to create the 008 field in new MFHDs. You can specify a location in a field in the bibliographic record. If the field matches a string that you specify, you can designate a string that is to become the 008. See Table A-7 on page A-8.

Table A-7. Prebulk Configuration File:[008] stanza

000 6 am 9905280u 0 0 uuund | 990528	For UC purposes, we are not building a 008 field, however something must be entered here. What the example indicates is that the 008 field will be built off of what is in offset 6 of the leader field for a Prebulk run on May 28, 1999. This field must be 32 characters in length including blanks and/or spaces.

For more information on Prebulk configuration files and related stanzas, see the "Prebulk" section of the *Voyager Technical User's Guide*.

Multiple Owning Libraries

B

Why Have More Than One Owning Library?

The main reason sited for creating multiple owning libraries in the UC database is to keep certain bibliographic records intact from a particular local/feeder database.

For example, a consortia may decide that a particular law library or medical library at one university may be the authority on subject heading information that they want to maintain in the UC database. Specifically, a consortium with a medical library may want MSH headings stored in bibliographic records in the UC database. In order for this information to be maintained in the UC database, multiple owning libraries need to be defined, one for the main UC database and another for the information at the specialty library.

Without multiple owning library definitions, it is possible that information from the specialty library may be replaced or deleted that the consortia does not want replaced or deleted. This could occur as a result of the ongoing maintenance process for the UC database.

Considerations

The following information highlights the special considerations and possible risks associated with implementing multiple owning libraries in the UC database:

- Increased complexity from an implementation and ongoing system administration perspective.
- Increased complexity from a usability perspective for end users.
- Increased number of setup tasks.
- Confusion for end users when search results list presents duplicate bibliographic records.
- Confusion among end users when attempting to select/identify item for Universal Borrowing when duplicate bibliographic records are presented in the search results list.
- Increased ongoing maintenance tasks, separate processing of records imported for the different owning libraries.

Additional Steps Required for Multiple Owning Libraries

The following are the additional steps/processes required for implementing multiple owning libraries in the UC database:

- A separate owning library definition needs to be created. Contributing library locations are associated with the Owning Library as a part of this definition.
- A separate bulk import rule needs to be set up. See Bulk Import Rules on page 2-35 for more information.
- A separate bibliographic record duplicate detection replace profile is required. See Bibliographic Duplicate Detection Profile on page 2-21 for more information.
- A separate cataloging policy definition needs to be set up and associated with the appropriate owning library.
- A quality hierarchy needs to be set up for the additional owning library records.
- If authority records are stored in the UC database, a separate authority records duplicate detection replace profile needs to be set up.
- Additional operator security profiles need to be set up. See Security Setup on page 2-47 for additional information regarding operator security profiles.
- Exception handling needs to be done for the automated processes affected by the inclusion of multiple owning libraries. For example, new or edited scripts need to be created for the ongoing import/maintenance of records from the additional owning library(s). See Initially Loading Records into the UC on page 3-6 for more information regarding the

`upd.YYYYMMDDMISS.NUC.bib` files as described in the Initially Loading Records into the UCsection. Also, see Ongoing Updating of UC Records on page 3-30.

Index