User's Guide

virtua

Integrated Library System

System Management: OPAC

VIRTUA ILS – INTEGRATED LIBRARY SYSTEM

System Management

OPAC User's Guide

Version 16.1

October 2017

1701 Kraft Drive
Blacksburg, Virginia 24060
U. S. A.
Phone 800.858.8857
E-mail: info@iii.com

Table of Contents

Table of Figures

1. Introduction

This user's guide describes the procedures for running server executables and scripts related to the Virtua OPAC subsystem of the Virtua ILS – Integrated Library System. The information in this guide is intended for system administrators who feel comfortable working with programs that directly modify the contents of the Virtua database.

The scripts and executables documented in this guide are not programs that you need to run daily. If you are unsure of whether you need to run a script or executable, contact an Innovative customer services representative. Additionally, if you do not feel comfortable running a script or executable, or if you are unaware of the consequences of running a script or executable, contact Innovative.

Important:

- Since many of the scripts and executables documented in this guide modify data in the database, it is important that you have a recent export of your database.
- Most scripts in Virtua contain descriptive comments at the beginning of the file. Before you run a script, open it in a text editor and read ALL of the comments contained in the header.

Many of the scripts and executables discussed in this guide take a long time to run. For these programs, you may find it necessary to run them as background processes. For details, see the *Virtua System Management Reference Guide.*

This chapter covers the following topics:

1.1 Using this Guide

The *Virtua System Management: OPAC User's Guide* contains instructions for performing system administration tasks related to the OPAC subsystem of Virtua. You can use the list below and the table of contents to locate specific information in this guide.

For . . .	**See . . .**
Instructions for rebuilding the table of data that is used to create the bibliographic keyword search index	Chapter 2
Instructions for working with keyword indexing	Chapter 3
Instructions for working with holdings keyword and patron keyword indexing	Chapter 4
Instructions for indexing heading data for browse searches	Chapter 5
Instructions for rebuilding filters	Chapter 6
Instructions for loading a synonym file	Chapter 7
Instructions for loading a thesaurus	Chapter 8
Instructions for indexing full text documents	Chapter 9
Reference information about the scripts and executables discussed in this guide	Appendix A
Information about setting a default sort for the List of Titles screen	Appendix B
Information on changes in this guide since the last version	Appendix C

1.2 Recommended Knowledge for Using this Guide

This user's guide is intended primarily for system administrators who are comfortable working with programs that directly modify the contents of the Virtua database. Throughout this guide, we assume that you have experience working with your server from the command line.

Additionally, throughout this guide, we assume that you have already read the *Virtua System Management Reference Guide*, which provides . . .

- Details on the directory of Virtua and the Virtua database.
- Administrative tips for working with UNIX, such as information on . . .

    - ♦ Running scripts and executables as background processes.
    - ♦ Redirecting output.
    - ♦ Detecting and killing processes.
    - ♦ Working with cron jobs.
- Administrative tips for working with Oracle, such as information on . . .
    - ♦ Accessing SQL*Plus.
    - ♦ Starting and shutting down the database.
    - ♦ Working with the Oracle Listener.
    - ♦ Managing passwords.
    - ♦ Exporting tables.
- Information on database backups.
- Details on character maps.
- Information on working with **psdriver.exe**.

> **Important:** Information provided in the *Virtua System Management Reference Guide* is generally NOT repeated in this user's guide.

1.3 An Important Note about Environment Variables

Environment variables specify information about the working environment in the current UNIX session. Programs such as Virtua access and use environment variables when executing functions. Additionally, some of these variables are available for use from the command line.

Before you run any of the scripts or executables documented in this guide, you need to check the settings of at least three environment variables:

- **EXE_DIR** - Defines the path to the directory in which the scripts and executables for this version of Virtua are stored.
- **ORACLE_SID** - Defines the Oracle SID setting (i.e., *vtls01* or *vtls99*).
- **NLS_LANG** - Determines the character set used by Oracle.

If these variables are not set correctly, it is possible that you will run the wrong version of a program, modify the wrong database, or corrupt your data. Generally, these variables will be set for you in the **dbadmin** profile when you log in to the system, but we recommend that you double-check their setting before you run any scripts or executables. For information on setting and checking these and other environment variables such as VIRTUA_USER and VIRTUA_PASSWORD, see the *Virtua System Management Reference Guide*.

2. Rebuilding the Bibliographic Keyword Search Table

After you create or change your keyword indexing rules, you need to rebuild the bibliographic keyword search table (used by the Virtua client). The bibliographic_search table stores an XML file for each bibliographic record in the database. To update the XML files for the records currently in your database, you need to run the program **KeywordIndex.exe**. Running this utility updates your bibliographic keyword search table to reflect the current indexing rules. It is the first of two steps needed to re-index the database.

Specifically, you need to run **KeywordIndex.exe** in the following situations:

- If you load records via a process that does NOT automatically update the bibliographic keyword search table and synchronize the records with the keyword indexes.
- If the bibliographic keyword search table is damaged or corrupted.
- After you make a change in your Keyword Indexing Catalog rules (via the Virtua Profiler) and you want to apply the changes to all *existing* records in your database. If you do not run **KeywordIndex.exe**, the change in the rules will apply only to newly cataloged records, e.g., those saved in the client using the MARC Editor.

Running **KeywordIndex.exe** does not directly affect the bibliographic data. It does not index the data for searching. Instead, the program populates the data in the bibliographic_search table from which keyword indexes are created. The bibliographic keyword indexes are built on this table and reside in the Oracle interMedia table space.

After you make a change in your Keyword Indexing Catalog rules, you must not only run **KeywordIndex.exe**, but also recreate the CTX (keyword) indexes. Running **Cr_CTX_Indexes.sh** is the second step needed to re-index the database (see the section "Working with Keyword Indexes" in this user's guide).

For information on setting Keyword Indexing Catalog rules, see the *Virtua Profiler/Cataloging Parameters User's Guide.*

This chapter covers the following topics:

2.1 Running KeywordIndex.exe

To update, or rebuild, the bibliographic_search table,

1. Log in to your server as **dbadmin**.

2. At the prompt, type: **`Drop_CTX_Indexes.sh`**.

3. Press Enter. The script drops the indexes that exist against the XML records stored in the bibliographic_search table. In this way, the records are prepared for recreation.

   Note: When **Drop_CTX_Indexes.sh** is running, the keyword index and heading keyword index will be unavailable in the Virtua client.

4. After the script runs, execute **KeywordIndex.exe**, which will recreate the XML records using the Keyword Indexing Catalog rules. To run **KeywordIndex.exe**, use any of the following optional command line parameters:

   Note: To see the options available for use with **KeywordIndex.exe**, type either **`-?`** or **`--help`**.

   - **`-q[Quiet mode]`** - Specifies that no progress reports are to be printed.

   - **`-d [Overwrites data]`**- Deletes the data for a record from the keyword table before inserting the updated data. If you use this option, you MUST also use the **-b** option described below.

     Note: Always use **-d** unless the keyword table is empty or the range of bibliographic IDs has not yet been added. If you do not use **-d**, then when you process records for which data exists in the keyword table, these records will not be updated to reflect new keyword indexing rules.

   - **`-s[Starting Bib-ID]`** - Specifies the bibliographic ID of the first record that you want the program to process.

   - **`-e[Ending Bib-ID]`** - Specifies the bibliographic ID of the last record that you want the program to process.

   - **`-l[Locations]`** - Does not index item locations. If you choose this option, you will NOT be able to filter by location.

Hint: If you decide at a later time that you want to implement location filters, you can run the script **PopulateMissingLocationFilters.sh**.

- **`-f[Output filename]`** - Writes all data to a SQL Loader file. If you select this option, no changes will be made to the keyword data. This option is generally used for the initial data load of a large database.

- **`-c[Combined only]`** - Indexes FRBR records only. When you use this option, non-FRBR records are not indexed. If you do NOT use this option, both FRBR and non-FRBR records are indexed.

- **`-b[Path and input filename]`** - The path and filename of a line-delimited file of bibliographic IDs to use instead of other input options for defining bibliographic records for processing. This argument must be phrased as follows:

  ```
    KeywordIndex.exe –b < [path]/[name of input file]
  ```

  You may use this parameter with other command-line parameters.

- **`-r[Input Filename]`** - *This option is generally reserved for use by Innovative staff.*

For example, if you want to process records with a range of bibliographic IDs from 7946 to 10029 and delete the keyword data for each record from the database before inserting the updated data, type:

```
KeywordIndex.exe s7946 e10029 D
```

-OR-

If you want to process new records with a bibliographic ID that is greater than or equal to 5743, type:

```
KeywordIndex.exe s5743
```

5. Press Enter.

   KeywordIndex.exe builds your bibliographic keyword search table based on the records in your database or the specified range of bibliographic IDs. The amount of time that this process takes depends on the number of records you want to process and the available resources of your server.

Important: After the **KeywordIndex.exe** process is completed, you need to run the script **Cr_CTX_Indexes.sh** to rebuild the keyword indexes from the new data in the database tables. For information about this script, see the section "Recreating Keyword Indexes" in this user's guide.

2.2 Keeping Track of the Progress of the KeywordIndex.exe Process

To be able to view the progress of the **KeywordIndex.exe** process, before running the executable, set your log level to **2** or **3**.

To set the log level,

1. Before initiating **KeywordIndex.exe**, type: **`export V_LOG_LEVEL=[#]`**
   Where # is the logging level for this session. Typing a lower number increases the amount of messages you receive. We recommend that you set the log level to **2** or **3**.

2. Press Enter.

When you run **KeywordIndex.exe**, Virtua displays information about the process on your screen.

2.3 Using Parallel Processing for KeywordIndex.exe

When keyword indexing the entire database, you have the option of utilizing parallel processing for **KeywordIndex.exe**. Running multiple processes (i.e., multiple instances of **KeywordIndex.exe**) in parallel will significantly decrease the time needed to keyword index the database as long as sufficient system resources are available on the host server.

Usage: **`KeywordIndexParallelJob.sh [parallel_degree]`**

Where **`[parallel_degree]`** is an optional parameter that indicates the number of processes, or instances, to start for the KeywordIndex executable. The default value is 4, which means that **KeywordIndex.exe** will run four times faster than if a single instance were running.

Benchmark: **KeywordIndex.exe** can index between 300-1000 bibliographic records per process, per minute. The main determining factor is I/O and CPU speeds on the host server.

3. Working with Keyword Indexes

This chapter discusses keyword indexes in general and describes tasks that you need to perform periodically related to keyword indexing. Any change in Keyword Indexing Catalog rules via the Virtua Profiler will entail updating the bibliographic_search table by running **KeywordIndex.exe** as well as updating/recreating keyword (i.e., CTX) indexes.

This chapter covers discusses the following topics:

3.1 Overview

When you save or load a record in Virtua, the record is automatically marked for synchronization with the keyword indexes. While the regular keyword index is synchronized by Virtua as soon as the record is saved or loaded, heading keywords are synchronized by Oracle every five minutes.

When Virtua synchronizes keywords, the bibliographic and heading keyword indexes are updated only for the marked records based on the appropriate rules. While heading keyword indexing is run every five minutes, the reindexing process may take more than five minutes if there are a large number of records to synchronize.

When you make changes to your Keyword Indexing Catalog rules via the Virtua Profiler and you want these changes to apply globally to database records, you need to do two things. You need to update the bibliographic_search table, and you need to update the CTX indexes.

- To update the bibliographic_search table, run **KeywordIndex.exe** (see the section "Running KeywordIndex.exe").
- To update the CTX indexes, run **Cr_CTX_Indexes.sh**.

Specifically, you will need to update the keyword indexes if you do any of the following:

- Make changes to your Keyword Indexing Catalog rules.
- Have a significant number of records that are waiting to be synchronized.
- Modify data directly by running an executable or a server script; e.g., **MARCBibUpdate.exe** or the cataloging script **ChangeShelfLocation.ksh** (see the *System Management: Cataloging User's Guide* for information about these programs).

To reduce fragmentation of your keyword indexes, it is important that you periodically run the optimization script that we provide.

3.2 Optimizing the Keyword Indexes

Fragmentation occurs on any database that performs cataloging functions. To optimize the CTX indexes to reduce fragmentation, regularly run the optimization script **OptimizeAllIndexes.sh**.

3.2.1 About the Keyword Optimization Script

Running the script **OptimizeAllIndexes.sh** optimizes the five keyword indexes:

- Bibliographic
- Author heading
- Title heading
- Library-defined heading (Z-search)
- Subject heading

This script runs a minimum of five hours, one hour for each index, and up to a maximum of 40 hours (i.e., eight hours for each index). The duration is specified as a parameter at the command line. Depending on the size of your database and how often you catalog records, you may need to run the script once a day, once a week, and/or once a month using different parameter values.

The keyword optimization process is system-intensive. While running the script, you will notice a loss of server performance. Because of this, instead of running the script for more than five hours, we suggest that you drop the indexes altogether by running **Drop_CTX_Indexes.sh** and then recreate them by running **Cr_CTX_Indexes.sh**. Doing that has two advantages: Fragmentation will be removed in a much shorter period of time and only the keyword index and the heading keyword index will be unavailable in the period between running the two scripts.

3.2.2 Running the Keyword Optimization Script

To run the keyword optimization script,

1. Cease all cataloging activity for the duration of this process.
2. Log in to your server as **dbadmin**.
3. At the prompt, type: **`OptimizeAllIndexes.sh <x>`**

   Where **<x>** is the number of hours (1 or 8) for the script to spend optimizing each index. The default = 1

4. Press Enter.

When the parameter **<x>** is not included or when it equals 1, the optimization script will run up to five hours, one hour for each of the five indexes.

3.3 Analyzing the CTX Bibliographic Index

If the keyword or heading keyword search is running slowly in the Virtua client, it is likely that the CTX bibliographic index is fragmented. The CTX bibliographic index is used solely for keyword searches in the client. To determine the degree of fragmentation, you can run the script **AnalyzeCTXBibliographicIndex.sh**.

Note:
- **AnalyzeCTXBibliographicIndex.sh** is designed for use *only* with Oracle 11*g*.
- Running the script uses a lot of resources, so we recommend that you run it during off-peak hours to limit the impact on a production environment.

Usage:
1. Log in to your server as **dbadmin**.
2. cd $EXE_DIR
3. Type: **`./AnalyzeCTXBibliographicIndex.sh`**

The script can take anywhere from 15 minutes to up to several hours to run, depending on the server hardware.

If the script finds high fragmentation, generally over 70 percent, we recommend dropping and recreating the CTX indexes using the **Cr_CTX_Indexes.sh** scripts (see the section "Recreating Keyword Indexes").

If fragmentation is lower than 70 percent, but above 30 percent, you may be able to optimize the indexes by simply running **OptimizeAllIndexes.sh**.

3.4 Recreating Keyword Indexes

When you do a keyword search in the client, the system uses CTX indexes, which are special indexes on the XML files stored in the bibliographic_search table. The indexes are created based on the data in the Keyword Indexing Catalog rules set in the Virtua Profiler. When the rules change, the indexes must be recreated to allow the keyword search to find the new tags/subfields in the XML records.

The best way to update the CTX indexes is to drop and recreate them utilizing two scripts: **Drop_CTX_Indexes.sh** and **Cr_CTX_Indexes.sh**.

Hint: If you are loading a large batch of records, to improve performance, you may want to stop the synchronization process using **Drop_CTX_Indexes.sh** until the records are loaded. After you finish loading the records, you *must* run the **Cr_CTX_Indexes.sh** script to create keyword indexes from the loaded records and restart the synchronization process.

To drop the CTX indexes, run the script **Drop_CTX_Indexes.sh**. To do this,

1. Cease all cataloging activity for the duration of this process.
2. Log in to your server as **dbadmin**.
3. At the prompt, type: `Drop_CTX_Indexes.sh`
4. Press Enter.

To recreate the CTX indexes, run the script **Cr_CTX_Indexes.sh**. To do this,

1. Cease all cataloging activity for the duration of this process.
2. Log in to your server as **dbadmin**.
3. At the prompt, type: `Cr_CTX_Indexes.sh`
4. Press Enter.

Virtua recreates the following keyword indexes:

- Bibliographic.
- Author heading.
- Title heading.
- Library-defined heading (Z-search).
- Subject heading.

3.5 Sorting of Keyword Indexes

Virtua comes with scripts that help you to determine the sort map used for your keyword index and to change the sorting of your keyword index.

3.5.1 Determining the Sort Map for Your Database

If you do not know which sort map is used for your database, you can run the script **WhichSortMapsLoaded.sh** to print to the screen which sort map is being used. This script is usually only used in conjunction with **SetUseKeywordSortMapTableFlag.sh**.

To run WhichSortMapsLoaded.sh,

1. Log in to your server as **dbadmin**.

2. In the /usr/vtls directory, type: `WhichSortMapsLoaded.sh`

The script prints to the screen the sort map in use in the database. For information on the sort maps used in Virtua, see the *System Management Reference Guide*.

3.5.2 Changing the Sort Map Used for Keyword Indexes

The script **SetUseKeywordSortMapTableFlag.sh** can be used to load the sort2-intl sort mapping table for use by keyword searches in the Virtua client. This sort mapping table can sort non-Roman alphabets. For information on the sort2-intl sort mapping table, see the *System Management Reference Guide*.

If you are already using the sort2-intl sort mapping table in your database, you do not need to use this script. However, if you are using another sort mapping table, you may want to use this script to improve the performance of your keyword search results. If you do not know which sort mapping table is used in your database, you can run the script **WhichSortMapsLoaded.sh** to determine which sort map is in use.

To run SetUseKeywordSortMapTableFlag.sh,

1. Log in to your server as **dbadmin**.
2. In the /usr/vtls directory, type: `SetUseKeywordSortMapTableFlag.sh`
3. Press Enter.

The script loads the sort2-intl sort map for use by keyword searching. Subsequent keyword search queries run your database should return more accurate results.

4. Working with Holdings Keyword and Patron Keyword Indexes

The optional Holdings Keyword search and Patron Keyword search in the Virtua client rely upon extra underlying software. A Solr index stores the Holdings Keyword and Patron Keyword information used for these searches. Solr is a search server based on the Lucene Java search library. For more information about Solr, see http://lucene.apache.org/solr/. The interface between the Solr index and the Virtua client is handled by Innovative's Patron and Holdings Keyword Indexing software, which is based on a modified version of Chamo, Virtua's Web-based OPAC.

Important: The Patron and Holdings Keyword search functions are special Virtua features that must be activated and configured by Innovative personnel. To enable these optional search features, contact Innovative Customer Services.

In most cases, you will not need to interfere with the operation of the Patron or Holdings Keyword Indexing software. For instance, if the Patron and Holdings Keyword Indexing software goes down while cataloging activity continues, it will update its indexes automatically when the indexing software is restarted.

However, you may need to manually start the indexing software in the event of a problem, or manually re-index your data when you update Virtua. These situations are discussed in this chapter.

Note: The Holdings Keyword and Patron Keyword searches use the same index. In this chapter, we give instructions for maintaining both searches, but even if you use only one of these searches, the same instructions are applicable.

This chapter provides details on performing the following tasks:

4.1 Accessing the Patron and Holdings Keyword Indexing Web Interface

Innovative offers a Web interface for the Patron and Holdings Keyword Indexing software. This interface lets you view the status of and reindex the patron and holdings keyword indexes.

Once the Patron and Holdings Keyword Indexing software is installed and configured, the Web interface can be accessed using a Web browser at the address **http://host:port/auth/login**, where **host** is the host on which the software is installed, and **port** is the port on which Solr is running. You should log in as the **admin** user.

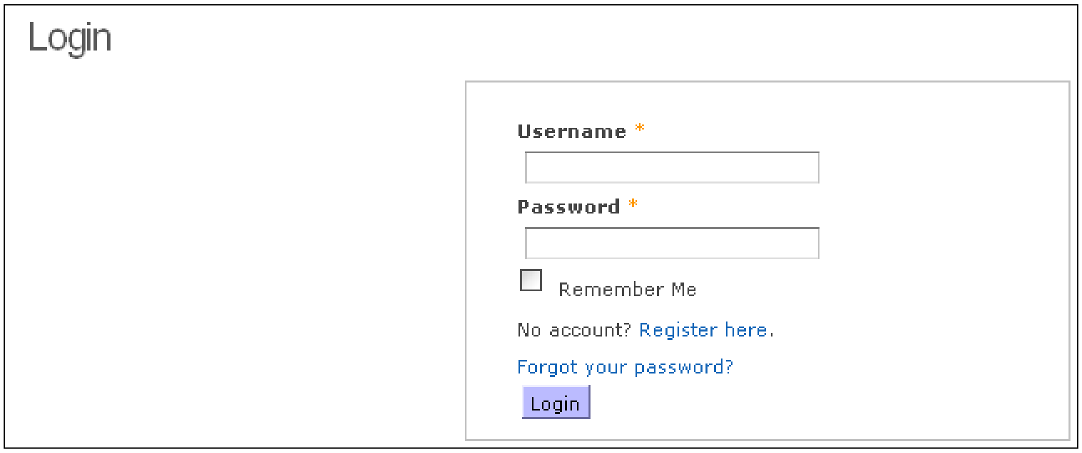


Figure 4-1. Patron and Holdings Keyword Indexing Software Login Screen

In addition to viewing the status of these indexes and reindexing them, this Web interface lets you change the password used to access the interface, and provides a means to translate the strings that appear in the interface.

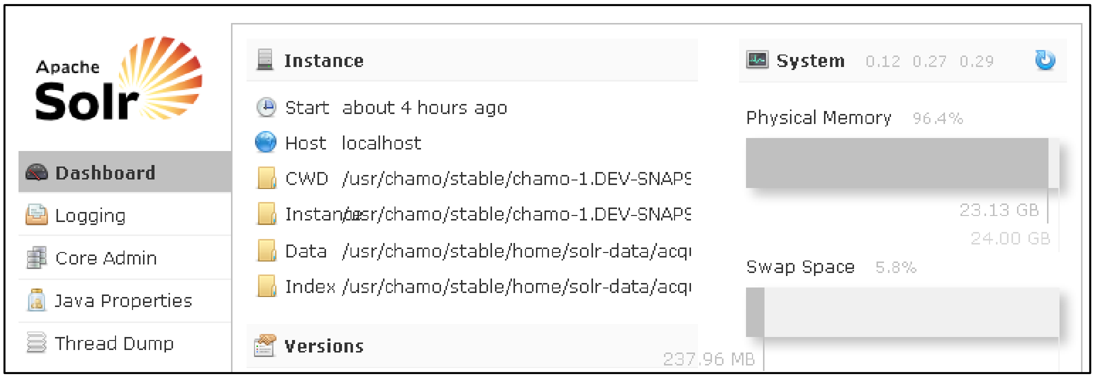
4.2 Restarting the Patron and Holdings Keyword Indexing Software

If the Holdings Keyword and/or Patron Keyword searches do not work, Solr or the Patron and Holdings Keyword Indexing software may have stopped running.

To see whether Solr is running,

- Open a Web browser and view the page **http://host:port/solr/admin**, where **host** is the server name, and **port** is the port on which Solr is running.

If Solr is running, you will see a Solr Administration console (*Figure 4-2*).



Start about 4 hours ago
Host localhost
CWD /usr/chamo/stable/chamo-1.DEV-SNAPS
Instance /usr/chamo/stable/chamo-1.DEV-SNAPS
Data /usr/chamo/stable/home/solr-data/acqu
Index /usr/chamo/stable/home/solr-data/acqu

Physical Memory 96.4%
23.13 GB
24.00 GB
Swap Space 5.8%
237.96 MB

Figure 4-2. Solr Administration Console

To see whether the Patron and Holdings Keyword Indexing software is running,

- Open a Web browser and view the page **http://host:port/auth/login**, where **host** is the server on which the Patron and Holdings Keyword Indexing software is installed, and **port** is the port on which Solr is running. You should see a Login screen (*Figure 4-3*).

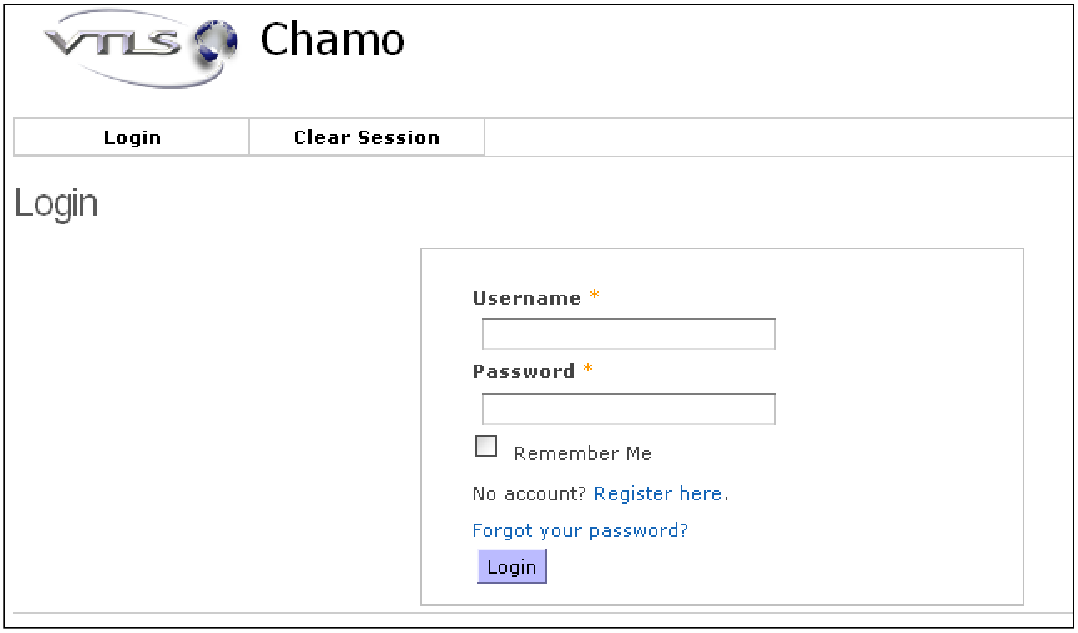


Figure 4-3. Patron and Holdings Keyword Indexing Software Login Screen

If you do not see a Solr administrative console or if you do not see a Login screen, you should restart the Patron and Holdings Keyword Indexing software.

To restart the Patron and Holdings Keyword Indexing software on your server,

1. Log in to your server as **dbadmin**.
2. Navigate to the **/usr/vtls/chamo** directory.
3. At the prompt, type: **`bin/shutdown.sh`**
4. Press Enter.
5. At the prompt, type: **`bin/startup.sh`**
6. Press Enter.

The Patron and Holdings Keyword Indexing software is restarted.

If Holdings Keyword cataloging activity has taken place while the Patron and Holdings Keyword Indexing software was down, the indexing software will receive this information from the Virtua database and will update the Patron and Holdings Keyword search indexes. However, in certain cases, you may wish to manually re-index your data. These cases are discussed in the section "Reindexing the Patron and Holdings Keyword Search Indexes" below.

Note: If restarting the Patron and Holdings Keyword Indexing software does not solve the problems you are having with the Holdings Keyword or Patron searches, contact Innovative Customer Services for assistance.

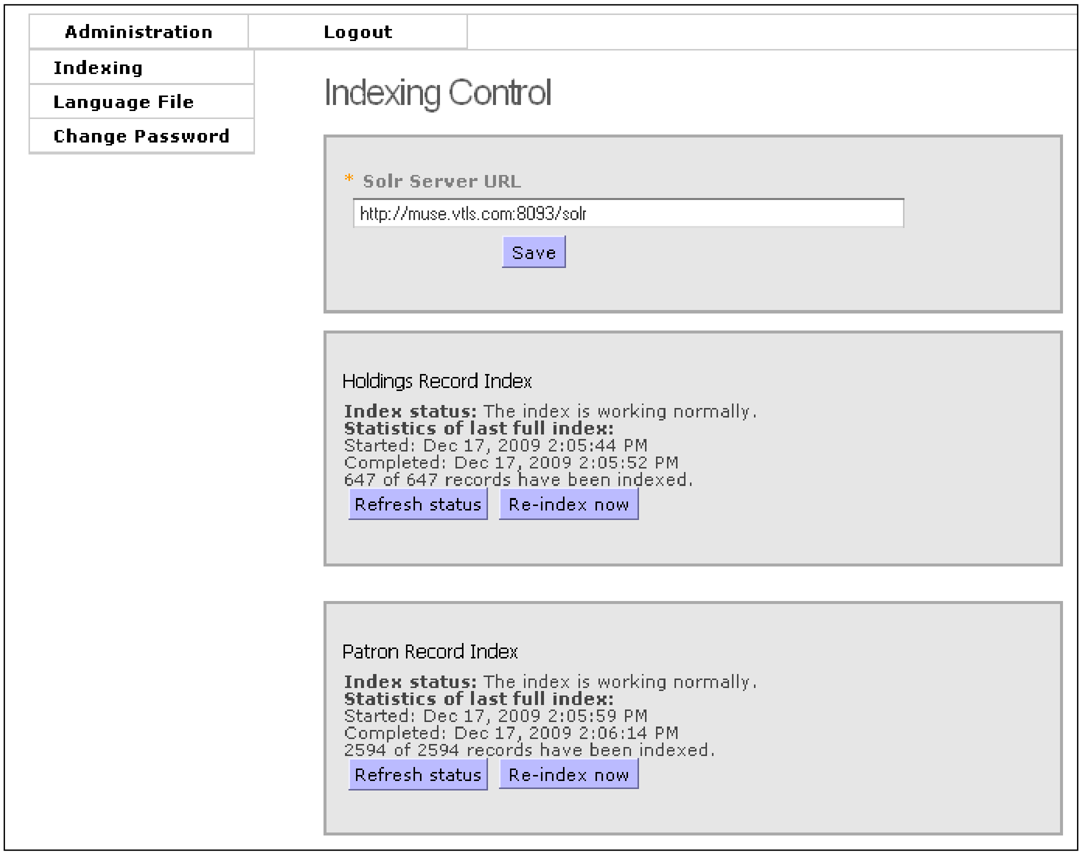
4.3 Reindexing the Patron and Holdings Keyword Search Indexes

In general, the Innovative Patron and Holdings Keyword Indexing software does not need administration. However, you may need to reindex your data when . . .

- You update Virtua to a new version.
- You have added a new sort map to your database (see the *Virtua System Management Reference Guide*).
- You make changes to your keyword indexing rules (see the *Virtua Profiler/Cataloging Parameters User's Guide*).
- An index shows a status other than "*The index is working normally"* on the Indexing Control screen.

To manually reindex your patron index and/or holdings keyword index,

1. Log in to the Innovative Patron and Holdings Keyword Indexing software at the Login screen (*Figure 4-1*). You will see the Indexing Control screen (*Figure 4-4*).



Indexing Control

* Solr Server URL

http://muse.vtls.com:8093/solr

Holdings Record Index

Index status: The index is working normally.
Statistics of last full index:
Started: Dec 17, 2009 2:05:44 PM
Completed: Dec 17, 2009 2:05:52 PM
647 of 647 records have been indexed.

Patron Record Index

Index status: The index is working normally.
Statistics of last full index:
Started: Dec 17, 2009 2:05:59 PM
Completed: Dec 17, 2009 2:06:14 PM
2594 of 2594 records have been indexed.

Figure 4-4. Patron and Holdings Keyword Indexing Software Indexing Control Screen

2. Click the *Re-index now* button for the index you want to reindex. Reindexing begins.

   The status for the index will be updated to reflect the status of the indexing process (*Figure 4-5*). Reindexing an index can take several hours.

> **Important:** You should reindex only ONE index at a time. Reindexing more than one index at a time may cause your system to become unstable. It is also best to avoid significant cataloging activity while an index is being reindexed.

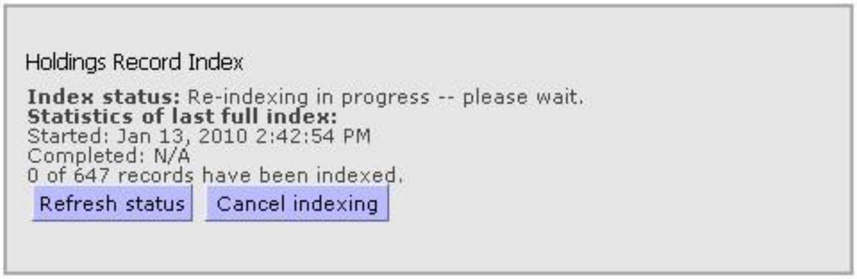
To update the status of the indexing process,

- Click the *Refresh status* button.

To cancel the reindexing process,

- Click the *Cancel indexing* button (*Figure 4-5*).

Note: If you cancel reindexing, the un-indexed data will not be available via the Patron or Holdings Keyword search. Therefore, we suggest that you cancel the reindexing process only if absolutely necessary.



Holdings Record Index

Index status: Re-indexing in progress -- please wait.
Statistics of last full index:
Started: Jan 13, 2010 2:42:54 PM
Completed: N/A
0 of 647 records have been indexed.

Figure 4-5. Patron and Holdings Keyword Indexing Software Indexing Process

To resume the indexing process after canceling,
Click the *Re-index now* or *Resume Indexing* button.

You can tell that an index is completely indexed when there is a Completed date and time, and the Index status says that all records are indexed; e.g., “647 of 647 records have been indexed.”

You may close the Indexing Control window at any time during the indexing process. As long as the server is running and the indexing software is not stopped, the indexing process will continue until the index is completely reindexed.

4.4 Translating Strings in the Patron and Holdings Keyword Indexing Software Web Interface

The Patron and Holdings Keyword Indexing software Web interface has a tool that lets you download an XLS file (spreadsheet) of strings to translate and then upload the XLS file of the translated strings. The strings you entered in the XLS file are then applied to the interface.

Note: To view the Patron and Holdings Keyword Indexing Software Web interface in a language other than the default language, you must select that language from the drop-down list box at the top of the screen (*Figure 4-6*).

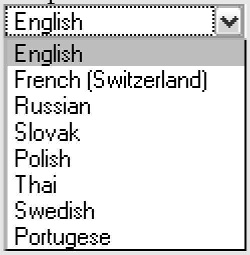


Figure 4-6. Language Selection Menu

To translate strings in the Patron and Holdings Keyword Indexing software,

1. Log in to the Innovative Patron and Holdings Keyword Indexing software at the Login screen (*Figure 4-1*).

2. Click the Language File link to see the Language String Management screen (*Figure 4-7*).

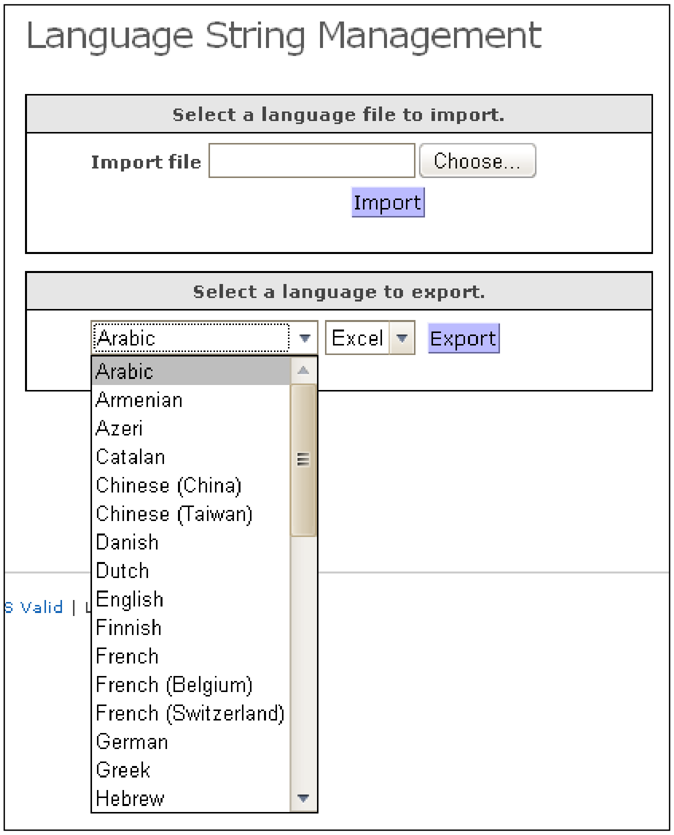
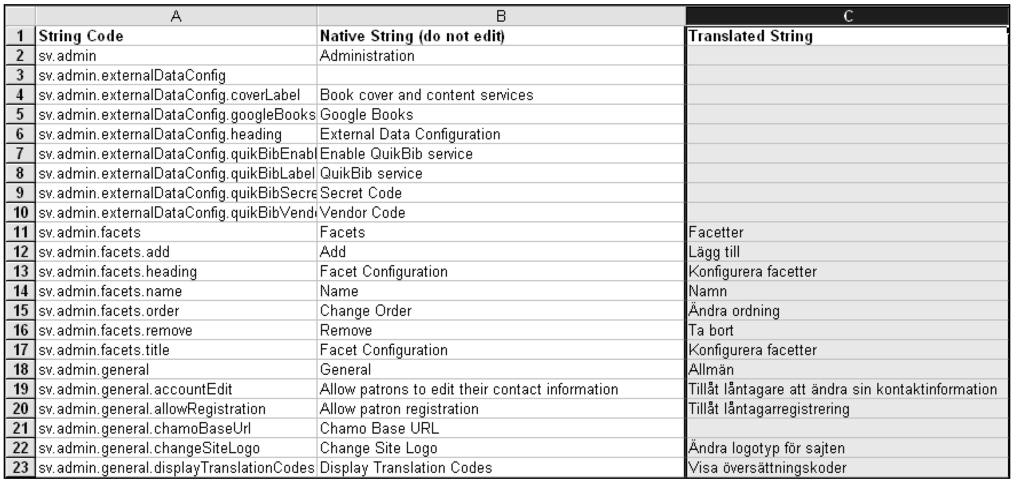


Figure 4-7. Language String Management Screen

3. In the *Select a language to export* area of the screen, select the language you wish to translate from the drop-down list box, and click the Export button. Save the XLS file to your PC.

> **Note:** The language file that you download for translation has many strings that are not relevant to the Patron and Holdings Keyword Indexing software. To find the strings that appear in the Web interface, find the strings in the String Code column of the spreadsheet that begin with the following prefixes:
> **xx.application.login**
> **xx.admin.indexing**
> **xx.admin.languageEdit**
> **xx.admin.password**
> where **xx** is the two-letter language code for the language you are translating. You may also search for the text of the string you wish to translate or edit.

4. Once you save the file to your PC, use Microsoft® Excel or another spreadsheet-editing program to type translations in the Translated String column of the spreadsheet.



	A	B	C
1	**String Code**	**Native String (do not edit)**	**Translated String**
2	sv.admin	Administration	
3	sv.admin.externalDataConfig		
4	sv.admin.externalDataConfig.coverLabel	Book cover and content services	
5	sv.admin.externalDataConfig.googleBooks	Google Books	
6	sv.admin.externalDataConfig.heading	External Data Configuration	
7	sv.admin.externalDataConfig.quikBibEnabl	Enable QuikBib service	
8	sv.admin.externalDataConfig.quikBibLabel	QuikBib service	
9	sv.admin.externalDataConfig.quikBibSecre	Secret Code	
10	sv.admin.externalDataConfig.quikBibVend	Vendor Code	
11	sv.admin.facets	Facets	Facetter
12	sv.admin.facets.add	Add	Lägg till
13	sv.admin.facets.heading	Facet Configuration	Konfigurera facetter
14	sv.admin.facets.name	Name	Namn
15	sv.admin.facets.order	Change Order	Ändra ordning
16	sv.admin.facets.remove	Remove	Ta bort
17	sv.admin.facets.title	Facet Configuration	Konfigurera facetter
18	sv.admin.general	General	Allmän
19	sv.admin.general.accountEdit	Allow patrons to edit their contact information	Tillåt låntagare att ändra sin kontaktinformation
20	sv.admin.general.allowRegistration	Allow patron registration	Tillåt låntagarregistrering
21	sv.admin.general.chamoBaseUrl	Chamo Base URL	
22	sv.admin.general.changeSiteLogo	Change Site Logo	Ändra logotyp för sajten
23	sv.admin.general.displayTranslationCodes	Display Translation Codes	Visa översättningskoder

Figure 4-8. Sample Language Translation File - Viewed in Microsoft Excel

5. When you are finished typing the translations for the strings in Chamo, save the spreadsheet to your PC.

6. To upload the file of translated strings, in the *Select a language to import* area of the Language String Management screen (*Figure 4-7*), click the Browse button to locate your file, and click the Import button.

4.5 Changing the Admin Password

To log in to the Web interface for the Patron and Holdings Keyword Indexing software, you must use the **admin** user. You can change the password for this user using the following steps.

To change the password for the admin user,

1. Log in to the Innovative Patron and Holdings Keyword Indexing software (*Figure 4-1*).
2. Click the Change Password link to see the Change Password screen (*Figure 4-9*).

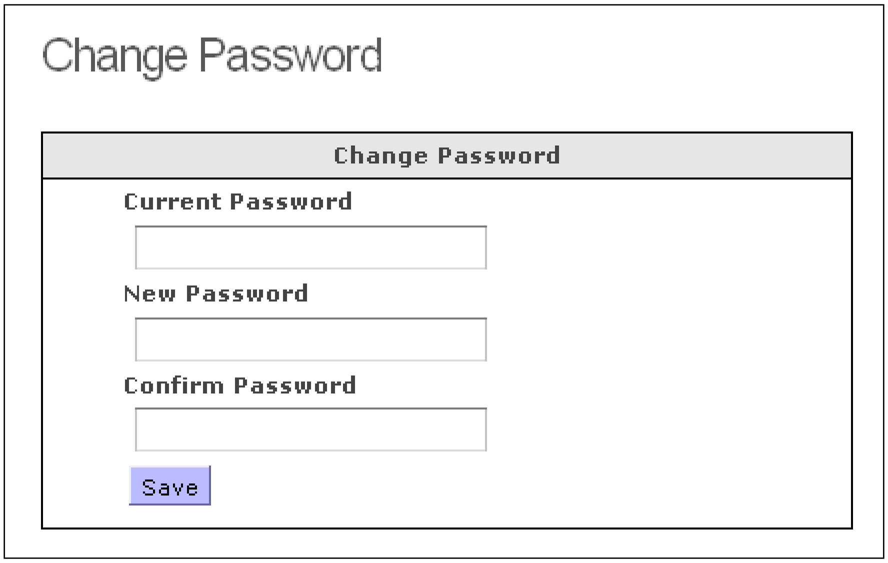


Figure 4-9. Change Password Screen

3. Type the current password in the Current Password field.
4. Type your new password in the New Password and Confirm Password fields.
5. Click the Save button.

Chamo changes the **admin** password to the new password you selected.

5. Indexing Browse Headings

To index browse headings in the database, Virtua offers the following programs:

⇒ Re_CreateHeadingSort.sh
⇒ CallIndexForm.sh
⇒ CallIndexFormItemsOnly.sh
⇒ ReIndexPatrons.sh
⇒ ReIndexUserHeadings.sh
⇒ SetPreventBrowseOf2ndItemLevelCallNumber.sh

This chapter describes when and how to run each program.

5.1 Re_CreateHeadingSort.sh

The script **Re_CreateHeadingSort.sh** re-creates the heading_sort, heading_display, and heading_search columns for the author, title, and subject browse tables. Situations that require you to run this script, such as when there is a change in your sort form, are not everyday occurrences. For example, you might need to run this script after you have loaded a new sort map to your database.

Normally, you do not need to run this script. If you are unsure about whether you should run it, contact a Innovative customer services representative.

Important: Prior to running this script, you should cease all cataloging activity. Cataloging records while running this script can result in faulty data processing.

To run Re_CreateHeadingSort.sh,

1. Ensure that all cataloging activity has ceased and will not resume until the script has successfully completed indexing.
2. Log in to your server as **dbadmin**.
3. Navigate to the **/usr/vtls/clasXX** directory, where **XX** is the number of the database for which you want to run this script.
4. At the prompt, type:

```
nohup Re_CreateHeadingSort.sh 2> Re_CreateHeadingSort.log &
```

This command initiates the indexing process. The process runs in the background, writing diagnostic messages to the file **Re_CreateHeadingSort.log**. The type of messages logged is determined by the logging level that is set for your server. For information about determining the logging level, see the *Virtua Installation User's Guide*.

5.2 CallIndexForm.sh

The script **CallIndexForm.sh** indexes call numbers for the call number browse list.

You might need to run **CallIndexForm.sh** after you have…

- Loaded a new sort map to your database.
- Changed values for the following parameters in the Virtua Profiler:
  - ♦ Call Number Index and Priority
  - ♦ Call Number Indexing Type

Notice that these situations are not everyday occurrences. Normally, you do not need to run this script. If you are unsure about whether you should run it, contact an Innovative customer services representative.

To run CallIndexForm.sh,

1. Log in to your server as **dbadmin**.
2. Navigate to the **/usr/vtls/clasXX** directory, where **XX** is the number of the database for which you want to run this program.
3. At the prompt, type: `CallIndexForm.sh`

This command initiates the indexing process. The process runs in the background, writing diagnostic messages to the file **CallIndexForm.log**. The type of messages logged is determined by the logging level that is set for your server. For information about determining the logging level, see the *Virtua Installation User's Guide*.

Tip: You can stop and restart this process at any time. This script is designed to remember the last field it indexed, and, when restarted, to begin again at that place.

5.3 CallIndexFormItemsOnly.sh

The script **CallIndexFormItemsOnly.sh** indexes only *item-level* call numbers for the call number browse list. This script allows you to re-index item-level call numbers after running the **SetPreventBrowseOf2ndItemLevelCallNumber.sh** script without recreating the entire call number index.

To run CallIndexFormItemsOnly.sh,

1. Log in to your server as **dbadmin**.
2. Navigate to the **/usr/vtls/clasXX** directory, where **XX** is the number of the database for which you want to run this program.
3. At the prompt, type **`CallIndexFormItemsOnly.sh`**
4. Press Enter.

5.4 ReIndexPatrons.sh

The script **ReIndexPatrons.sh** indexes patron names for the patron name browse list. Normally, you do not need to run this script.

> **Important:** Prior to running this script, you *must* cease all cataloging activity related to patron records. Cataloging patron records while running this script can result in faulty data processing.

The script **ReIndexPatrons.sh** uses a three-part process to index the patron name browse.

1. The script generates patron data file from the database.
2. The script deletes the old index information from the database.
3. The file created in step 1 are loaded into the database.

Using the **ReIndexPatrons.sh** script, you can choose to . . .

- Perform step 1 only, which extracts patron data to a file. You will then index the patron browse list at a later time.
- Perform steps 2 and 3 only, provided that step 1 was previously completed.
- Perform all three steps.

If the patron browse data fails to load properly, **ReIndexPatrons.sh** will automatically call **ReProcessPatrons.sh**, which will reprocess any bad records. You can also run

ReProcessPatrons.sh as a standalone script (see the *System Management: Cataloging User's Guide*).

5.4.1 Running ReIndexPatrons.sh

To run ReIndexPatrons.sh,

1. Log in to your server as **dbadmin**.
2. Navigate to the **/usr/vtls/clasXX** directory, where **XX** is the number of the database for which you want to run this script.
3. At the prompt, type: **`ReIndexPatrons.sh`**

   The script displays the following options:

   ```
   1) Only create the patron browse data file
      to load at a later time.
   2) Load previously created patron browse
      data file.
   3) Create patron browse data file and load
      into database immediately after creation.
   ```

4. Type the number of the option you want to utilize.

5. Press Enter.

6. Follow the interactive prompts and wait for the script to complete.

5.5 ReIndexUserHeadings.sh

The script **ReIndexUserHeadings.sh** indexes user-defined authority headings for browse lists. It lets you reprocess bibliographic records from a file of bib IDs to index user-defined headings only without changing the bibliographic record. **ReIndexUserHeadings.sh** uses an input file of bibliographic record IDs. The input file must be a text file that contains one bib ID per line and that resides in the database directory on the server.

To run ReIndexUserHeadings.sh,

1. Log in to your server as **dbadmin.**

2. At the prompt, type:

```
ReIndexUserHeadings.sh [bibIdsfilename]
```

Where **[bibIdsfilename]** is the name of the file of bib IDs to be reprocessed.

3. Press Enter.

5.6 SetPreventBrowseOf2ndItemLevelCallNumber.sh

When you enable the use of two item-level call numbers, Virtua's default behavior is to index *both* the first and the second item-level call number for browse searching.

Note: The Item Call Number Flag parameter allows you to enable or disable the use of multiple item-level call numbers. See the *Virtua Profiler/Cataloging Parameters User's Guide* for more information.

The script **SetPreventBrowseOf2ndItemLevelCallNumber.sh** allows you to specify that Virtua index the first item-level call number, but *not* index the second item-level call number.

To run SetPreventBrowseOf2ndItemLevelCallNumber.sh,

1. Log in to your server as **dbadmin.**

2. At the prompt, type:

```
SetPreventBrowseOf2ndItemLevelCallNumber.sh [value]
```

Where **[value]** specifies whether or not second item-level call numbers are indexed:

- **0** - Second item-level call numbers WILL be indexed for the call number browse.
- **1** - Second item-level call numbers WILL NOT be indexed for the call number browse.

4. Press Enter.

Note: After running **SetPreventBrowseOf2ndItemLevelCallNumber.sh**, you must immediately run the script **CallIndexFormItemsOnly.sh** to apply the change to the existing entries in the call number browse index. See the section "CallIndexFormItemsOnly.sh" for more information.

6. Rebuilding Filters

In most cases, Virtua maintains updates to search filters without intervention. However, there may be cases in which it is necessary to update some filters to ensure that search results return correct results.

This section covers the following topics:

6.1 Rebuilding the Location Filters

The location filters in a database may be incomplete if indexing is interrupted or if the locations in the system have been modified. Running **KeywordIndex.exe** with the **-l** command-line option can also cause the location filters to not be in sync with the locations in the database.

The script **PopulateMissingLocationFilters.sh** repopulates the location filters so that they match the locations in the bibliographic_location table in the database. This script will also remove any invalid locations from the bibliographic record table.

To rebuild the location filter,

1. Log in to your server as **dbadmin.**

2. At the prompt, type:

```
PopulateMissingLocationFilters.sh
```

3. Press Enter.

The script updates the bibliographic_location table to include any missing locations for items and holdings, and removes extra data that does not exist for items or holdings to correctly synchronize the table with the locations in the database.

6.2 Rebuilding the Format Filters

The script **PopulateMarc21FormatFilter.sh** refreshes the format filters that appear on the Format Filters window in the Virtua Profiler, which in turn affect the appearance of the Format Filters available on the Search Filters window in the Virtua client.

See the *Virtua Profiler/OPAC Parameters User's Guide* for more information on the Profiler parameter, and the *OPAC Workflow-based Reference Guide* for information on format filters in the Virtua client.

If you have a **.dat** file of format filters, you can load the file using this script. If you do not have a file, the script loads Virtua's default format filters.

To reload the Format Filters,

1. Log in to your server as **dbadmin.**

2. At the prompt, type:

   ```
   PopulateMarc21FormatFilters.sh [filenameprefix]
   ```

   Where `[filenameprefix]` is the beginning name of a **.dat** file of formats. For instance, if your file is named **formats.dat**, you would type:

   ```
   PopulateMarc21FormatFilters.sh formats
   ```

3. Press Enter.

The script populates the format filter list with the values specified in your file, or if no file is specified, the default format filters are loaded.

6.3 Rebuilding Item Class Filters

The OPAC Basic Options setting "Allow collection group filters for browse/heading" lets you enable or disable item class search filtering in the Virtua client. When this setting is enabled, you can filter Virtua client Browse and Heading searches using the item class by clicking on the Item Class button on the Search window.

If you choose to enable item class search filtering, you MUST enable the "Allow item class filters for browse/heading" setting, then run the script **PopulateItemClassFilters.sh**, which rebuilds item class filter data in the database.

Important: Before you run **PopulateItemClassFilters.sh**, you must enable the OPAC Basic Options setting "Allow collection group filters for browse/heading." See the *Virtua Profiler/OPAC Parameters User's Guide* for information on this setting.

To rebuild the Item Class Filters,

1. Log in to your server as **dbadmin.**

2. At the prompt, type:

   ```
   PopulateItemClassFilters.sh
   ```

3. Press Enter.

   The script rebuilds item class filter data, making it possible for you to use the item class search filtering functionality in the Virtua client.

6.4 Rebuilding Collection Group Filters

The script **PopulateCollectionGroupFilters.sh** adds existing collection group filters to any search tables that may be missing them. This will let you successfully filter searches for bibliographic or item records by collection group.

Important: Before you run **PopulateCollectionGroupFilters.sh**, you must enable the OPAC Basic Options setting "Allow collection group filters for browse/heading." See the *Virtua Profiler/OPAC Parameters User's Guide* for information on this setting.

To reload the Collection Group Filters,

4. Log in to your server as **dbadmin.**

5. At the prompt, type:

   ```
   PopulateCollectionGroupFilters.sh
   ```

6. Press Enter.

The script populates the collection group filter list with the values specified in the Collection Code Groups parameter in the Virtua Profiler. See the *Virtua Profiler/Cataloging Parameters User's Guide* for information on this parameter.

7. Loading a Synonym File

Virtua and Chamo support synonym searching, which means that words like “color” and “colour” can be treated as the same word in keyword searching. In other words, when a user searches for “color,” the records retrieved will also include records containing the word “colour,” and vice versa. Non-Roman characters (such as Chinese or Arabic) can also be used as synonyms.

You can create/edit the synonym list in a text file called, for example, **Synonym.txt**. The synonym file must be in Unicode-compliant text format. Both Virtua and Chamo need to use the same synonym file. The file must contain one line for each set of synonyms with a comma between each synonym; e.g., syn1,syn2,syn3.

For Chamo, the text file can be uploaded via the Indexing Control screen in the Administrative interface of Chamo. When the synonym file has been modified, Chamo will respond with a message, indicating keyword indexing must take place before the modifications will be applied to the database (similar to the message the user gets when modifying the stop word list).

For Virtua, the text file needs to reside on the server and be uploaded to the database using the following script: **LoadSynonymFile.sh**.

Each time you edit **Synonym.txt**, you will need to upload it again, and, in Virtua’s case, run **LoadSynonymFile.sh** again. After running the script, you will need to keyword index your database again (see the section “Rebuilding the Bibliographic Keyword Table”).

To run the script,

1. Navigate to the directory on the server in which you created the synonym file.

2. At the prompt, type: `LoadSynonymFile.sh` followed by the name (and location) of the text file. For example, type: `LoadSynonymFile.sh Synonym.txt`

3. Press Enter.

   Virtua loads the synonym file to the database.

Sample entries in a synonym list text file:

```
ABCENSE,ABSENCE
ABRIDGEMENT,ABRIDGMENT
ACCOMODATE,ACCOMMODATE
ACKNOWLEDGEMENT,ACKNOWLEDGMENT
AFTERWARD,AFTERWARDS
ALUMINUM,ALUMINIUM
AMEBA,AMOEBA
ANALOG,ANALOGUE
ANALYZE,ANALYSE
```

8. Loading a Thesaurus

Before you can use Virtua's thesaurus features, you need to load a subject thesaurus file to your database. This chapter includes instructions for loading a thesaurus. Once a thesaurus is loaded, you can use the thesaurus operators in a precision search.

To load a subject thesaurus file for use with precision searching and the Thesaurus Search Interface, you need to complete the following steps:

- Use the custom thesaurus loading script that we created for your thesaurus format to load the entries in your thesaurus file to the tables you created.
- Extract the data from the thesaurus tables in a format that is supported by Oracle interMedia.
- Load the extracted data to interMedia.

After you complete these steps you will be able to use the Thesaurus Search Interface and include thesaurus operators in your precision search queries.

This chapter covers the following topics:

8.1 Loading a Thesaurus to the Thesaurus Tables

For your thesaurus, we create a custom script that loads the data to the tables. Before you run the script to load the data, make sure that the text file that contains your thesaurus entries exists in the **gw_xx/thesaurus/admin** directory.

Note: Virtua supports ONE thesaurus per database. On a database with an existing thesaurus, loading another thesaurus will overwrite the existing thesaurus.

To load the entries in your thesaurus file to the thesaurus tables,

1. Navigate to the **gw_XX_x/thesaurus/admin** directory.

2. Type: **`./[script name]`**

   Where **[script name]** is the name of the custom script we created for your thesaurus file.

This command loads the data in your thesaurus file to the thesaurus tables. After you successfully run this script, the thesaurus entries should appear in the Thesaurus Search Interface.

8.2 Extracting Thesaurus Data

After you load the data into the thesaurus tables, you need to extract that data to load to interMedia so that you can initiate OPAC searches from the Thesaurus Search Interface and use thesaurus search operators in your precision search queries. The script **create_intermedia.pl** extracts the data from the thesaurus tables and writes it to a file that can be loaded to interMedia.

To extract thesaurus data into a file,

1. Navigate to the **gw_XX_x/thesaurus/admin** directory.
2. Type: **`./create_intermedia.pl > intermedia_data.txt`**

This command creates a file named **intermedia_data.txt** that contains all of your thesaurus entries formatted in a way that can be read by interMedia.

8.3 Loading Thesaurus Data to interMedia

To load thesaurus data to interMedia,

1. Copy or FTP the file **intermedia_data.txt** to the **/usr/vtls/clasxx** directory of the database to which you want to load the thesaurus, where **xx** is a database number such as 99.

2. From the directory to which you copied the **intermedia_data.txt** file, type:

```
/usr/vtls/virtua/r_xx_x/src/LoadThesaurus.ksh intermedia_data.txt
```

Where **xx_x** is the version of the Virtua server installation. For example, for the 2014.2 version of the Virtua server software, type:

```
/usr/vtls/virtua/r_2014_2/src/LoadThesaurus.ksh intermedia_data.txt
```

This command loads the thesaurus data in **intermedia_data.txt** to interMedia. After you successfully run this script, you can initiate OPAC searches from the Thesaurus Search Interface and use thesaurus search operators in your precision search queries.

9. Loading and Indexing Full Text Documents

Virtua lets you load documents individually or in batch into the Virtua database, index the full text of the documents, and associate them with a bibliographic record. The text within any loaded document can then be keyword searched in the Virtua client and the bibliographic record associated with the document retrieved. For each document, the full text indexing program, **DocumentLoader.sh** or **DocumentBatchLoader.sh**, creates entries in the full text search index. You can search the full text index using the full text search category (**8900**).

Note: By adding a document to the full text search index, you are not appending the document to a bibliographic record. To make the document accessible from the record, you need to create a link via the Item URL of an attached item or 856 tag of the record.

This chapter covers the following topics:

9.1 Supported Document Formats

The full text indexing programs **DocumentLoader.sh** and **DocumentBatchLoader.sh**, work with only three formats: plain text files, PDF files, and Microsoft Word files. The scripts convert PDF or Word documents to UTF-8 and then load the resulting document text into the database, associated with a particular bibliographic record. A plain text file is loaded as is with no modifications.

The scripts support these file extensions:

DOC
CSV
PPT
TXT
RTF

XML
XSLT
PDF

DocumentLoader.sh and **DocumentBatchLoader.sh** use PDF-to-text and Antiword utilities to extract text from PDF and Word files. Both utilities are distributed with the Virtua server software and must reside in the /usr/vtls/virtua directory.

Note: The Xpdf utility in **DocumentBatchLoader.sh**, which is used to extract text from PDF, supports PDF version 1.4 only.

Be aware that for any document you choose to index, you must remove ALL password protection. However, this does not mean that the document to which you provide a link in the record cannot have password protection.

9.2 Steps for Loading and Indexing Documents

Loading and indexing the full text of documents requires two steps:

1. Create a file that contains information about each document you want to load and index (for details, see below).
2. Run the script **DocumentLoader.sh** (to load a single document) or **DocumentBatchLoader.sh** (to load a batch of documents).

9.2.1 Creating a File of Document Information

The first step in loading and indexing the full text of one or more documents is creating a tab-delimited file that contains information about each document.

To create a tab-delimited file of document information,

1. Log in to your server as **dbadmin**.
2. Navigate to the directory in which you have saved the documents you want to index.
3. Create and open a file named **fulltext.txt**
4. On the first line of the file, type:

```
[document filename][tab][bib ID][tab][text|not_text][tab][doc_ID]
```

Where . . .

- **[document filename]** is the name of the file to load.
- **[bib ID]** is the ID of the bibliographic record with which you want to associate the document.
- **[text|not_text]** is a flag that indicates whether the document is in text format or another format such as PDF or Word, which needs to be converted.
- **[document ID]** (*optional*) is the alphanumeric ID of the document. This ID must be unique for the bibliographic record you specified. If this option is not specified, <document filename> will be used as the document ID.

For example, if you are indexing a document where . . .

- The filename is **faulkner_bio.pdf**.
- The bibliographic ID of the record to which you want to associate this document is **10043**.
- The file format is PDF.
- The document ID is **bio_01**.

Type: **`faulkner_bio.pdf    10043      not_text     bio_01`**

5. For each document you want to index, repeat step 4 on a new line.
6. Save the file.

9.2.2 Running DocumentLoader.sh or DocumentBatchLoader.sh

After you create a file of document information, you can run either **DocumentLoader.sh** or **DocumentBatchLoader.sh**.

To run DocumentLoader.sh or DocumentBatchLoader.sh,

1. Log in to your server as **dbadmin**.

2. Navigate to the directory in which you created the file **fulltext.txt** (the file containing information about the documents).

3. At the prompt, type:

```
DocumentLoader.sh fulltext.txt
-OR-
DocumentBatchLoader.sh fulltext.txt
```

4. Press Enter.

The specified document(s) is loaded and indexed.

Note: Entries in the full text index are automatically removed if you delete the associated bibliographic record.

9.2.3 Linking a Record to the Document

After you load and index a document for full text searching, the bibliographic record that you specified in the file of document information becomes associated with the document. Any full text search that matches text in the indexed document will return the associated bibliographic record as a result.

There is no automatic linking between the bibliographic record and the document. To create a link from the bibliographic record to a document, you must . . .

- Make the document available on a Web server.
- Create an item- or bibliographic-level link to the document.

9.3 Removing Documents from the Full Text Index

Removing documents from the full text index requires two steps:

1. Create a file that contains information about each document you want to remove from the index.
2. Run the script **DocumentBatchDeleter.sh**.

9.3.1 Creating a File of Document Information

The first step in removing a document from the full text index is creating a tab-delimited file that contains the following information about each document you want to remove from the full text index:

- **Bibliographic ID** - The bibliographic ID of the record associated with the document you want to remove from the index.
- **Document ID** - The alphanumeric ID of the document you want to remove. This information is optional. If you do not specify a document ID, the program will remove from the index all documents associated with the bibliographic ID you specified.

To create a tab-delimited file of document information,

1. Log in to your server as **dbadmin**.
2. Create and open a file named **remove_fulltext.txt**
3. On the first line of the file, type:

   ```
   [bibliographic ID][tab][document ID]
   ```

   Where . . .

   - **[bibliographic ID]** is the bibliographic ID of the record associated with the document you want to remove from the index.
   - **[document ID]** (*optional*) is the ID of the document. This ID must be unique for the bibliographic record you specified. If you do not specify a document ID, the program will remove from the index all documents associated with the bibliographic ID you specified.

   For example, if you are removing a document where . . .

   - The bibliographic ID of the record associated with this document is **10165**.
   - The document ID is **edit_02**.

   Type: `10165	edit_02`

4. For each document you want to remove, repeat step 3 on a new line.
5. Save the file.

9.3.2 Running DocumentBatchDeleter.sh

After you create a file of document information, you can run **DocumentBatchDeleter.sh**.

To run DocumentBatchDeleter.sh,

1. Log in to your server as **dbadmin**.
2. Navigate to the directory in which you created the file **remove_fulltext.txt** (the file containing information about the documents).
3. At the prompt, type **`DocumentBatchDeleter.sh remove_fulltext.txt`**
4. Press Enter.

The **DocumentBatchDeleter.sh** script removes the documents specified in the file **remove_fulltext.txt** from the full text search index.

10. Indexing Invalid ISBNs

ISBNs that contain nine digits are invalid, and as such, they are stored in tag 010 subfield $z (UNIMARC) or tag 020 subfield $z (MARC 21) of the bibliographic record. The script **SetIndexInvalidISBNFlag.sh** gives you the means to index these invalid ISBNs so that you can conveniently retrieve them via the ISBN search option on the Control Number tab of the Search window in the Virtua client. The ISBN search option looks for the ISBN in tag 010 $a and $z in UNIMARC databases and in tag 020 $a and $z in MARC 21 databases.

To run SetIndexInvalidISBNFlag.sh,

1. Log in to your server as **dbadmin**.

2. Navigate to the **/usr/vtls/clasXX** directory, where **XX** is the number of the database for which you want to run this program.

3. At the prompt, type: **`SetIndexInvalidISBNFlag.sh <value>`**

   Where **<value>** is either 0 or 1:

   0 - Do not index invalid ISBNs
   1 - Index invalid ISBNs

4. Press Enter.

11. Normalizing Patron Phone Numbers

The **NormalizePatronTelephone.sh** script works in conjunction with the *Format patron phone number for display, normalize searches* setting in the Virtua Profiler's Patron Cataloging Options parameter. Virtua runs the script automatically during the migration to the current release. If *Format patron phone number for display, normalize searches* is enabled at the time your database is migrated, you do not need to run the script again. However, if *Format patron phone number for display, normalize searches* is turned OFF at the time your database is migrated and then you turn the setting ON, we recommend that you run the script.

NormalizePatronTelephone.sh checks the *Format patron phone number for display, normalize searches* setting to see if it is ON. If it is, the script normalizes (i.e., removes all non-numeric characters from) the primary and secondary patron phone numbers (tag 270 $k and $j, indicator 1 or 2) in the patron record and stores them in the database. In addition, the Virtua server will normalize phone numbers used as search terms.

To run NormalizePatronTelephone.sh,

1. Log in to your server as **dbadmin**.
2. Navigate to the **/usr/vtls/clasXX** directory, where **XX** is the number of the database for which you want to run this script.
3. At the prompt, type: **`NormalizePatronTelephone.sh <value>`**

   Where **<value>** is either 0 or 1:

   0 - Do not update patron telephone number data in the database, but do output tab delimited text files (patron_telephone_primary_special.txt, patron_telephone_secondary_special.txt, patron_telephone_primary.txt, and patron_telephone_secondary.txt) that contain the patron IDs and telephone numbers of patrons whose telephone numbers are over 10 digits long.

   1 - Normalize patron telephone numbers in the database and also output the text files described above.

4. Press Enter.

The script runs, checks the Patron Cataloging Options setting, and if appropriate, normalizes the patron phone number data. If the setting is OFF, the script takes no other action.

12. Deleting Entries from the Operation Log

The script **Delete_Operation_Log.sh** lets you delete all entries in a specified date range from the database operation log. To control the growth rate of the database, Innovative recommends purging this historic log on a systematic basis.

> **Important:** Before running **Delete_Operation_Log.sh**, shut down all circulation, cataloging, and OPAC-based activity.

12.1 Running Delete_Operation_Log.sh

To delete entries from the operation log,

1. Log in to your server as the **dbadmin** user.

2. At the prompt, type: **`Delete_Operation_Log.sh`**

3. Press Enter.

   Via a series of prompts, the script asks you to specify a start date and an end date within which all entries will be deleted from the operation log.

4. Enter the dates as requested.

   The script displays the start date and end date you entered and the number of rows of operations that it will delete from the log.

   > **Note:** The script deletes operations in an inclusive manner, meaning that all operations from the start date to the end date, including the start and end dates themselves, are deleted from the log. You can set the start date and end date to be the same day so that information from only that single date is deleted from the log.

5. Type:
   - **`Y`** to continue with the deletion.
   - **`N`** to exit the program without deleting information from the log.

6. Press Enter.

If you choose to continue with the deletion, the script will . . .

- Export the entire operation log to a file named **exp_operation_log.dat**.

  Note: The current time and date will be appended to the filename (before the extension).

- Delete the information that falls within the date range you specified.

13. Appendix A - OPAC Scripts and Executables

In this appendix, we list and briefly describe each of the scripts and executables discussed in this guide. You can click the hyperlink in the Filename column to view additional information about each script or executable.

Filename	Purpose	Command-line Parameters	Notes
AnalyzeCTXBibliographicIndex.sh	Analyzes the degree of fragmentation in the CTX Bibliographic Index.	None	For use only with Oracle 11*g*. Run during off-peak hours.
CallIndexForm.sh	Indexes headings for the call number browse.	None	Before you run this script, cease all cataloging activity. You can start and stop this process at any time.
CallIndexFormItemsOnly.sh	Indexes item-level call numbers for the call number browse. This script allows you to re-index item-level call numbers after running the **SetPreventBrowseOf2ndItemLevelCallNumber.sh** script without recreating the entire call number index.	None	Before you run this script, cease all cataloging activity. You can start and stop this process at any time.
Cr_CTX_Indexes.sh	Recreates the five keyword search indexes to take into account Keyword Indexing Catalog changes.	N/A	Before you run this script, cease all cataloging activity.

Filename	Purpose	Command-line Parameters	Notes
create_intermedia.pl	Extracts data from the thesaurus tables and writes it to a file that can be loaded to interMedia.	None	Use this script as part of the process described in the chapter "Loading a Thesaurus."
Delete_Operation_Log.sh	Deletes entries within a specified date range from the database operation log.	None	Before running, all circulation, cataloging, and OPAC activities must cease.
DocumentBatchDeleter.sh	Removes documents from the full text index.	This script accepts a filename of a text file with document information.	
DocumentBatchLoader.sh	Loads and indexes a batch of documents for full text searching.	This script accepts the filename of a text file containing document information.	
DocumentLoader.sh	Loads and indexes a single document for full text searching.	This script accepts the filename of a text file containing document information.	

Filename	Purpose	Command-line Parameters	Notes
Drop_CTX_Indexes.sh	Drops current keyword search indexes to enable re-creation of the search indexes to take into account Keyword Indexing Catalog changes.	N/A	

Filename	Purpose	Command-line Parameters	Notes
KeywordIndex.exe	Updates the files in the bibliographic_search table with all the rules in the Keyword Indexing Catalog configuration.	This executable accepts the following parameters: • **-q** • **-d** • **-s[Bib-ID]** • **-e[Bib-ID]** • **-l** • **-f[Filename]** • **-r[Filename]** • **-c** • **-b** For detailed information about each of these options, see the section "Running KeywordIndex.exe."	Before you run this executable, read the entire chapter "Rebuilding the Bibliographic Keyword Table".

Filename	Purpose	Command-line Parameters	Notes
KeywordIndexParallelJob.sh	Keyword indexes the entire database utilizing parallel processing for **KeywordIndex.exe**.	This script accepts an optional parameter to indicate the number of parallel processes to be started.	Running multiple processes in parallel can significantly decrease the overall time needed to keyword index the database.
LoadSynonymFile.sh	Loads a synonym file to the database, allowing for synonym searching.	This script accepts the filename of a synonym file.	The file should have one line for each set of synonyms with a comma between each synonym.
LoadThesaurus.ksh	Loads a thesaurus file to the database.	This script accepts a filename of a thesaurus file.	Use this script as part of the process described in the chapter "Loading a Thesaurus."
NormalizePatronTelephone.sh	Normalizes patron phone number data for display and searches	None	The script works in conjunction with the *Format patron phone number for display, normalize searches* setting in the Virtua Profiler's Patron Cataloging Options parameter.

Filename	Purpose	Command-line Parameters	Notes
OptimizeAllIndexes.sh	Optimizes the five keyword indexes.	Optional index name	Running this process causes a significant loss of server performance.
PopulateCollectionGroupFilters.sh	Populates the collection group filters.	None	You must enable the OPAC Basic Option “Allow collection group filters for browse/ heading” before running this script.
PopulateItemClassFilters.sh	Populates the item class filters.	None.	You must enable the OPAC Basic Option “Allow item class filters for browse/heading” before running this script.
PopulateMarc21FormatFilter.sh	Populates the format filters.	Optional filename prefix	Can be used to load a file of format filters or to load default format filters.
PopulateMissingLocationFilters.sh	Updates the Bibliographic_Location table to include any missing locations for items and holdings.	None	Also removes non-existent location data from items and holdings records to correctly synchronize the table with the locations in the database

Filename	Purpose	Command-line Parameters	Notes
Re_CreateHeadingSort.sh	Indexes headings for the author, title, subject, and user-defined browses.	None	Before you run this script, cease all cataloging activity.
ReIndexPatrons.sh	Indexes headings for the patron browse.	None	Before you run this script, cease all patron cataloging activity.
SetIndexInvalidISBNFlag.sh	Indexes invalid ISBNs.	This script accepts a value of 0 or 1.	Available for UNIMARC and MARC 21 databases.
SetPreventBrowseOf2ndItemLevelCallNumber.sh	Allows you to specify that Virtua index the first item-level call number, but *not* index the second item-level call number.	This script accepts a value of 0 or 1.	After running this script, you must run the script **CallIndexFormItemsOnly.sh** to update the call number browser index.
SetUseKeywordSortMapTableFlag.sh	Changes the sort order for your keyword index to sort2-intl	None	
WhichSortMapsLoaded.sh	Prints the sort order used in your database.	None	

14. Appendix B - Setting a Default Sort for the List of Titles Window

When you view the results of your bibliographic keyword search on the List of Titles window, the hits for your search are initially sorted by the default sort type that you have set for your system.

You can select from the following default sort options in the Virtua Profiler > OPAC Basic Options parameter. For details, see the *Virtua Profiler/OPAC Parameters User's Guide.*

- **1 - No sort:** Records are displayed in the order in which they appear in the database.
- **2 - Call number:** Records are sorted in ascending order by call number. The call number sort is based on call number parameters that you set on the Cataloging tab of the Virtua Profiler.
- **3 - Author:** Records are sorted in ascending order by authoritative author with a subsort by title.
- **4 - Title:** Records are sorted in ascending order by title with a subsort by author.
- **6 - Publication Date/Author:** Records are sorted in ascending order by the publication date defined in the 008 tag with a subsort on author.
- **7 - Publication Date/Title:** Records are sorted in ascending order by the publication date defined in the 008 tag with a subsort on title.

15. Appendix C - Changes in this Guide

15.1 Changes for Version 16.1

No changes were made.

15.2 Changes for Version 15.2

"Rebuilding Item Class Filters" - Updated to correct an error in the description of the **PopulateItemClassFilters.sh** script.

Index