

**User's  
Guide**



---

Integrated Library System

System  
Management:  
Reporting

VIRTUA ILS – INTEGRATED LIBRARY SYSTEM

# **System Management**

## Reporting User's Guide

**Version 16.1**

**October 2017**

---



Copyright © 2002-2017 VTLS Inc./Innovative Interfaces, Inc. All Rights Reserved.  
Virtua and the Virtua Design marks are used under license from Sega Corporation.

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

# Table of Contents

<b>TABLE OF CONTENTS</b>	<b>I</b>
<b>1. INTRODUCTION</b>	<b>1</b>
1.1 USING THIS GUIDE	1
1.2 RECOMMENDED KNOWLEDGE FOR USING THIS GUIDE	2
1.3 AN IMPORTANT NOTE ABOUT ENVIRONMENT VARIABLES	3
<b>2. WORKING WITH CIRCREPORT.EXE</b>	<b>4</b>
2.1 USAGE FORMAT	4
2.1.1 DEFINING COMMAND OPTIONS IN A FILE	5
2.1.2 LIMITING UPDATES TO PATRONS WITH SPECIFIED COMMUNICATION PREFERENCE	6
2.2 REPORT TYPE OPTION	6
2.3 INPUT OPTIONS	8
2.3.1 OPERATOR NAME	8
2.3.2 USE ALTERNATE TITLE HEADING	8
2.3.3 NOTICE COUNT	9
2.3.4 LOCATION	9
2.3.5 PATRON LANGUAGE	10
2.3.6 PATRON TYPES	10
2.3.7 ITEM CLASSES	10
2.3.8 COLLECTION CODES	11
2.3.9 CHECK-OUT TYPES	11
2.3.10 OUTPUT MAP	11
2.3.11 REPORT DELIMITER	12
2.3.12 SORT ORDER	13
2.3.13 USE CHECK-OUT TYPE NAMES	13
2.3.14 DATE FORMAT	14
2.3.15 RUNNING MULTIPLE INSTANCES IN UPDATE MODE	14
2.3.16 OPTION FOR TRANSITION DISPUTED ITEMS REPORT ONLY	14
2.3.17 OPTIONS FOR INVENTORY CONTROL REPORT ONLY	15
2.3.17.1 Name of File of Barcodes	15
2.3.17.2 Beginning of Call Number Range	15
2.3.17.3 End of Call Number Range	16
2.3.17.4 Inventory Location	16
2.3.17.5 Report on Items without a Call Numbers	16
2.3.17.6 Invalid Shelving Statuses	17
2.3.17.7 Inventory Date	17
2.3.17.8 Error Tolerance	17
2.3.17.9 Accession Number Range	17

ii System Management: Reporting (v. 16.1)

2.3.17.10 Collection Code	18
2.3.17.11 User ID	18
2.3.18 OPTIONS FOR I-TIVA USERS	18
2.3.18.1 Option for the i-tiva Notices File	18
2.3.18.2 Option for the i-tiva Results File	19
2.3.18.3 Patron Phone Opt-in	19
2.3.18.4 Voice, Text, or Both Voice and Text Notifications	19
2.3.19 OPTIONS FOR EXPIRED REQUESTS AND EXPIRED BOOKINGS REPORTS ONLY	20
2.3.19.1 Expired Categories	20
2.3.19.2 Assess Fees for Requests “Not Picked Up”	21
2.3.20 OPTION FOR EXPIRED REQUESTS AND UNSATISFIED REQUEST CANCELLATION REPORTS ONLY	21
2.3.20.1 Delete Request Fees	21
2.3.21 OPTION FOR EXPIRED REQUESTS REPORT ONLY	21
2.3.21.1 Trapped Requests	21
2.3.22 OPTION FOR UNSATISFIED REQUESTS AND UNSATISFIED REQUEST CANCELLATION REPORTS	22
2.3.22.1 Item Statuses	22
2.3.23 OPTIONS FOR BILLING REPORTS ONLY	22
2.3.23.1 Add Billing Fee	23
2.3.23.2 Delete Overdue Fine	23
2.3.23.3 Report Bills Exceptions	24
2.3.24 OPTION FOR OVERDUES REPORT ONLY	24
2.3.24.1 Check-in Dates	24
2.3.25 OPTIONS FOR UNSATISFIED REQUESTS REPORT ONLY	24
2.3.25.1 Request Types	25
2.3.25.2 Level of Requests	25
2.3.26 OPTION FOR REQUEST EXPIRATION NOTIFICATION AND EXTENSION REPORT ONLY	25
2.3.26.1 Range of Expiration Dates	26
2.3.27 OPTION FOR REVIEW NOTICE REPORT ONLY	26
2.3.27.1 Suppress Notice for Overdue Items	26
2.3.28 OPTION FOR AVAILABILITY REMINDER NOTICE REPORT ONLY	26
2.3.28.1 Number of Days before Request Expiration	27
<b>2.4 OUTPUT OPTIONS</b>	<b>27</b>
<b>2.5 DATABASE UPDATE OPTIONS</b>	<b>31</b>
2.5.1 UPDATE PARAMETER OPTION	31
2.5.2 LIMIT UPDATE TO ADDRESS	32
<b>2.6 CIRCREPORT.EXE AND THE I-TIVA MESSAGE SYSTEM</b>	<b>33</b>
2.6.1 SCHEDULING VIRTUA AND THE I-TIVA MESSAGE SYSTEM	34
2.6.2 RUNNING AND MODIFYING THE ITIVAREPORT.SH SCRIPT	34
<b><u>3. CONVERTING CALL NUMBERS TO A NORMALIZED FORM</u></b>	<b><u>38</u></b>
<b><u>4. GENERATING A LIST OF URLS</u></b>	<b><u>39</u></b>
<b><u>5. ACCESSING TAG-SPECIFIC RECORD DATA AND CREATING CUSTOM VIEWS</u></b>	<b><u>41</u></b>

<b>5.1 ABOUT THE STORAGE OF BIBLIOGRAPHIC AND AUTHORITY RECORDS IN VIRTUA</b>	<b>41</b>
<b>5.2 RETRIEVING BIBLIOGRAPHIC AND AUTHORITY RECORD DATA</b>	<b>42</b>
5.2.1 RETRIEVING BIBLIOGRAPHIC INFORMATION FROM A VARIABLE FIELD TAG	42
5.2.2 RETRIEVING BIBLIOGRAPHIC INFORMATION FROM A FIXED FIELD TAG	43
5.2.3 RETRIEVING BIBLIOGRAPHIC INFORMATION FROM THE LEADER	44
5.2.4 DECODING BIBLIOGRAPHIC LEADER VALUES	44
5.2.5 RETRIEVING AUTHORITY INFORMATION FROM A VARIABLE FIELD TAG	47
5.2.6 RETRIEVING AUTHORITY INFORMATION FROM A FIXED FIELD TAG	47
5.2.7 RETRIEVING AUTHORITY INFORMATION FROM THE LEADER	48
<b>5.3 CREATING VIEWS FOR THE BIBLIOGRAPHIC_RECORD TABLE</b>	<b>48</b>
<b><u>6. INFOSTATION TOOL FOR EXTRACTING BIBLIOGRAPHIC DATA</u></b>	<b><u>51</u></b>
6.1 USING GET_BIBS.PL TO EXTRACT BIBLIOGRAPHIC DATA	51
6.2 EXAMPLE OF A COMMAND TO GENERATE A REPORT WITH GET_BIBS.PL	55
<b><u>7. APPENDIX A - SCRIPTS AND EXECUTABLES DISCUSSED IN THIS GUIDE</u></b>	<b><u>57</u></b>
<b><u>8. APPENDIX B - CHANGES IN THIS GUIDE</u></b>	<b><u>59</u></b>
8.1 CHANGES FOR VERSION 16.1	59
<b><u>INDEX</u></b>	<b><u>60</u></b>



# 1. Introduction

This user's guide describes the procedures for running executables and scripts related to the reporting features of the Virtua ILS – Integrated Library System. Information in this guide is intended for two groups:

- Developers who are creating custom reports for the library.
- System administrators who need to troubleshoot data generated by InfoStation reports.

**Note:** 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:

- ⇒ [Using this Guide](#)
- ⇒ [An Important Note About Environment Variables](#)

## 1.1 Using this Guide

The *System Management: Reporting User's Guide* contains instructions for using scripts and executables for reporting purposes. Use the list below and the table of contents to locate specific information in this guide.

For . . .	See . . .
Information about <b>CircReport.exe</b>	<a href="#">Chapter 2</a>
Information about using <b>callNumberKey.exe</b> to convert call numbers to a normalized form	<a href="#">Chapter 3</a>
Information about using <b>getURLs.sh</b> to report on the URLs stored in your database	<a href="#">Chapter 4</a>
Instructions for using stored procedures to access bibliographic data	<a href="#">Chapter 5</a>

Instructions for using the <code>get_bibs.pl</code> utility to extract bibliographic information	<a href="#">Chapter 6</a>
A list and brief description of the scripts and executables discussed in this guide	<a href="#">Appendix A</a>
A list of changes in the guide since the last version.	<a href="#">Appendix B</a>

## 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 . . .
  - ◆ 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 *System Management Reference Guide*.

## 2. Working with CircReport.exe

The program **CircReport.exe** is used to extract data from the database and to modify data. InfoStation uses **CircReport.exe** for the following notice-generating Circulation reports:

- Availability
- Availability Reminder
- Bills
- Expired Bookings
- Expired Requests
- Inventory Control
- Overdues
- Recalled Bills
- Recalled Overdues
- Recalls
- Request Expiration Notification and Extension
- Review Notice
- Transition Disputed Items
- Unsatisfied Request Cancellation

You can use **CircReport.exe** to develop your own reports or to troubleshoot InfoStation reports. In this chapter, we document the usage instructions for **CircReport.exe**.

This chapter covers the following topics:

- ⇒ [Usage Format](#)
- ⇒ [Report Type Options](#)
- ⇒ [Input Options](#)
- ⇒ [Output Options](#)
- ⇒ [Database Update Options](#)

### 2.1 Usage Format

You can run **CircReport.exe** from the command line using the following format:

**CircReport.exe** [report type] [input] [output] [update parameter]

Where . . .

- **[report type]** is a parameter that determines the type of report that is being run.
- **[input]** is one or more input options.
- **[output]** is one or more options that define the output of the report.
- **[update parameter]** is an optional parameter that determines whether the report can update the database.

In the following sections, we provide detailed reference information for these command-line options.

**Note:** If you want **CircReport.exe** to write information to a file in a directory other than the working directory, define the full path of that directory in the `VTLS_TEMP` environment variable. The program will write any file information to the directory you specify. If this variable is not set, the program will write file information to the current directory. For details on environment variables, see the *Virtua System Management Reference Guide*.

### 2.1.1 Defining Command Options in a File

Rather than specifying all **CircReport.exe** options on the command line, you can save them to a file and then call the file when running the program. You can call this file each time you want to run **CircReport.exe**.

The options you enter in a text file should be in the same format given in the previous section. To run **CircReport.exe** with a file of options, type:

```
CircReport.exe -@ [filename]
```

Where **[filename]** is the name of the file that contains the options.

For example if you named your parameter file **parameters.txt**, type:

```
CircReport.exe -@ parameters.txt
```

**Note:** You CANNOT call another file of command-line parameters from within a parameters file.

## 2.1.2 Limiting Updates to Patrons with Specified Communication Preference

**Option:** `--limit_update_to_communication_pref` string\_of\_preferences

**Description:** Limits updates to patrons whose communication preference matches one in the given string of communication preferences. Each character is a different communication preference and must be one of the following parameter values:

- 0 - paper
- 1 - e-mail
- 2 - text
- 3 - both text and e-mail

**Example:** `--limit_update_to_communication_pref 3`

## 2.2 Report Type Option

**Option:** `-t`

**Description:** Specifies the type of report that you are running.

**Parameter Values:**

- **1 – Review Notice** - Reports on patrons whose borrowed items are about to come due. Review notices are sent before overdue notices and remind patrons to return their checked-out items before the due date.
- **2 – Overdues** - Reports on transactions for which an item is overdue.
- **3 – Recalled Overdues** - Reports on transactions for which an item is overdue and recalled.

**Note:** An item is assigned the recalled overdue notice type when the item is overdue and a recall request is placed against it, AND an overdue notice has already been sent for the item.

- **5 – Dispute to Bill Conversion** - Reports on items the status of which is changed from Disputed to Overdue and for which fines are re-instated.
- **6 – Bills** - Reports on transactions for which a patron is eligible to be billed for an overdue item. If the update parameter is enabled, all items included in this report will be assigned a status of Long Overdue (4600).
- **7 – Recalled Bills** - Reports on transactions for which a patron is billed for an overdue and recalled item. If the update parameter is enabled, all items included in this report will be assigned a status of Long Overdue (4600).
- **8 – Recalls** - Reports on items that have been recalled.
- **a – Availability** - Reports on items or titles that are now available.
- **A – Availability Notice Reminder** – Reminds about items or titles that are now available.
- **e – Request Expiration Notification and Extension** – Reports on requests that are about to expire.
- **i – Inventory Control** - Using a list of barcodes from the items on a shelf or range of shelves, tracks the items on your shelves. To use this report, you must have a file containing one item barcode per line. There should be no excess spaces before or after each barcode.
- **u – Unsatisfied Request Cancellation** - Reports on items with request queues that should be cleared because the items carry a status for which requests are not allowed.
- **U – Unsatisfied Requests** – Reports on unsatisfied requests (excluding inactive requests) that could be satisfied using an available item.
- **x – Expired Requests** - Reports on requests that expired either . . .
  - ◆ Before they were filled (i.e., unavailable requests that are expired according to the request expiration date).
  - OR-
  - ◆ Before they were picked up (i.e., available requests that are expired according to the hold expiration date)
- **X – Expired Bookings** - Reports on expired bookings that meet the location and patron type criteria that you set in the report configuration.

**Example:** -t 6

**Note:** **CircReport.exe** does not return records for patrons that have an address block (043 \$a AA or 007/20 1). If **CircReport.exe** processes records for patrons that have address blocks, it writes information to a file named **[report type]blockFile**, where **[report type]** is the value entered for the **-t** option. Additionally, the record of the blocked patron will not be modified. For example, patrons with address blocks will appear in the file **8blockFile** if they are included in a Recall notice. This file is written to the directory defined by the **VTLS\_TEMP** environment variable.

## 2.3 Input Options

In this section, we list and describe each of the input options that you can use with **CircReport.exe**.

### 2.3.1 Operator Name

**Option:** -w

**Description:** Specifies the name of a Virtua user profile such as **staff**. The log-in location of the user profile is the location for which the report is run.

**Option value:** The username of a Virtua user profile.

**Example:** -w staff

**Important:** The -w option must be specified for **CircReport.exe** to run.

### 2.3.2 Use Alternate Title Heading

**Option:** -a

**Description:** If this option is included in the command, and title or author (depending on the parameter value) is chosen as an output, the program will substitute the value in the 245 tag with the title in the 880 tag.

**Option value:**

- **1xx** - To substitute for the author heading with the alternate 880 tag.

- **245** - To substitute for the title heading with the alternate 880 tag.

**Example:** -a 245

**Note:** If you want the program to output the alternate 880 tag for both author and title headings, you need to enter this option twice, specifying each tag.

### 2.3.3 Notice Count

**Option:** -n

**Description:** Restricts reports to return records that have the specified notice count.

**Option value:** An integer value indicating a notice count. If the value specified for the -t option is 1, 6, or 7, this option must specify a non-zero integer.

**Example:** -n 2

### 2.3.4 Location

**Option:** -l

**Description:** Defines the locations for which the report will run. If this option is not included in the command, the report runs for all locations.

For all report types except Expired Requests (-t x), this option defines the *shelving* locations for which the report will run. For the Expired Requests report, this option defines the *pickup* location for which the report will run.

**Option values:** A space-delimited list of location codes.

**Example:** -l 10000 20000 30000

## 2.3.5 Patron Language

**Option:** -L

**Description:** Limits the output of the report to patrons with a specific preferred language. Only patrons with no preferred language (tag 100 subfield \$l) or a preferred language that matches the provided language will be checked. The language comparison is case insensitive.

**Option values:** Language codes. (Language codes used in your patron records must match the language codes used in your InfoStation strings files.)

**Example:** -L en

## 2.3.6 Patron Types

**Option:** -p

**Description:** Defines the patron types that will be included in the report. If this option is not included in the command, the program runs for all patron types.

**Option values:** A space-delimited list of patron type codes.

**Example:** -p FC ST AD

**Note:** The `-p` option can accept patron type IDs that contain spaces as long as the entire patron type ID is enclosed in either single or double quotes. Examples:

**-p ST AA AP**

Filters on three patron classes: ST, AA and AP

**-p 'New Student' AP**

Filters on two patron classes: "New Student" and AP

**-p 'A really long patron type id'**

Filters on one patron class with the ID of "A really long patron type id"

## 2.3.7 Item Classes

**Option:** -i

**Description:** Limits the output of the report to specified item classes.

**Option values:** Item class codes as defined in the Virtua Profiler.

**Example:** -i 4

### ***2.3.8 Collection Codes***

**Option:** -x

**Description:** Limits the output of the report to specified collection codes.

**Option values:** Collection codes as defined in the Virtua Profiler.

**Example:** -x ABA

### ***2.3.9 Check-out Types***

**Option:** -c

**Description:** Defines the check-out types that will be included in the report. If this option is not included in the command, the program runs for all check-out types.

**Option values:** A space-delimited list of one or more of the following check-out type codes:

- 0 - Regular check-out
- 1 - In-house check-out
- 2 - Reserve check-out

**Example:** -c 2

### ***2.3.10 Output Map***

**Option:** -m

**Description:** Determines the character set to which the output will be mapped. If this option is not included in the command, the program outputs data in UTF-8.

**Option values:** One of the following codes:

- 2 - UTF-8
- 10 - MARC-8 (MARC 21)
- 11 - ANSI Z39.47 (ANSI-8)
- 12 - EUROPA-3

## 12 System Management: Reporting (v. 16.1)

- 13 - Windows Latin1
- 14 - PC-8
- 15 - ALA
- 16 - Windows Arabic
- 17 - Windows Hebrew
- 18 - Windows Cyrillic
- 19 - Windows Latin2
- 20 - ISO 6937-2
- 21 - CP-850 (Microsoft Code Page)
- 22 - ISO 6937-2 + Arabic
- 23 - ISO 6937-2 + Greek
- 24 - Big5 (Tamkang version)
- 25 - ANSI-8 + Hebrew
- 26 - UTF-8 character set where some separate diacritics need to be combined
- 28 - ANSI Z39.47 (ANSI-8) - Swiss version
- 30 - GBK (encoding of CJK characters)
- 31 - TIS620 Classic Thai
- 32 - ISO 5426 (International Serials Data System interchange)

**Example:** -m 20

### **2.3.11 Report Delimiter**

**Option:** -d

**Description:** Determines the delimiter that will be used in the output to separate different elements in the report. If this option is not included in the command, the program uses a single tab for the delimiter.

**Option values:** The character that you want to use as a delimiter for output.

**Example:** -d ,

## 2.3.12 Sort Order

**Option:** -s

**Description:** Determines the sort order of the output. Reports can be sorted by patron ID (001 tag), shelving location, and/or patron type. If this option is not included in the command, the program sorts output by patron ID.

**Option values:** One or more of the following values in the order or sort precedence:

- **p** - Patron ID (001 tag)
- **s** - Shelving location
- **t** - Patron type

If you are specifying more than one sort option, do not separate the values with a space.

**Example:** -s pts

## 2.3.13 Use Check-out Type Names

**Option:** -h

**Description:** Specifies that the output will include the given labels of check-out types rather than the check-out type code.

**Option values:** A space delimited list of check-out type labels. Use quotation marks for labels that include spaces. Each entry corresponds to the following check-out types:

- First entry - Regular.
- Second entry - In-house.
- Third entry - Fixed date.
- Fourth entry - Selected date.

**Example:** -h Regular In-house "Fixed Date" "Selected Date"

### 2.3.14 Date Format

**Option:** -D

**Description:** Specifies the date format that the program will use when generating output.

**Option values:** A valid Oracle date format.

**Example:** -D DD/Mon/YYYY

### 2.3.15 Running Multiple Instances in Update Mode

**Option:** --allow\_multiple\_update\_instances

**Description:** Controls whether multiple instances of **CircReport.exe** are allowed to be run in update mode at the same time.

**Option values:**

- 1 - Yes
- 0 – No (*default*)

**Example:** --allow\_multiple\_update\_instances 1

If the option is omitted or set to 0, multiple instances of **CircReport.exe** will not be permitted to run in update mode simultaneously. The default behavior will be to allow only one update instance at a time.

**Note:** A new constant has also been added to constants.cfg and the InfoStation installation script. The name of the new constant is allow\_multiple\_update\_instances. If this new constant is set to 1, InfoStation will allow multiple **CircReport.exe**-based reports to be run in update mode simultaneously.

### 2.3.16 Option for Transition Disputed Items Report Only

**Option:** -g

**Description:** Specifies the number of days an item must have the status of Disputed to be included in the output. The option is required.

**Option value:** 0 or greater (If 0, the output will include ALL items in dispute.)

**Example:** -g 15

## **2.3.17 Options for Inventory Control Report Only**

In this section, we describe the options that are specific to the Inventory Control report (-t i). You can use the Inventory Control report to keep track of the items on your shelves. Using a list of barcodes from the items on a shelf or range of shelves, the program can provide a report listing inventory or shelving problems such as:

- Barcodes that do not exist in your system.
- Items in general circulation that should be on reserve.
- Items in the wrong location.
- Items on your shelf that are reported as being checked out.
- Items on your shelf that are reported as having a status that should not be associated with a shelved item.
- Items with no call number assigned in the system.
- Items that do not belong in the range of barcodes.
- Missing items.
- Items belonging in the specified range that are out of order.

### **2.3.17.1 Name of File of Barcodes**

**Option:** --inventory\_barcode\_file

**Description:** Provides the path and the filename of the file listing the barcodes on which the program will report. This parameter is required.

**Option value:** The filename of the file of item barcodes.

**Example:** --inventory\_barcode\_file barcodes.txt

### **2.3.17.2 Beginning of Call Number Range**

**Option:** --inventory\_call\_start

**Description:** Specifies the first call number in the range of items. Type the call number in quotes. This parameter is required.

**Option value:** The first call number in the range of items. Type the call number in quotes.

**Example:** `--inventory_call_start "PS3211"`

### 2.3.17.3 End of Call Number Range

**Option:** `--inventory_call_end`

**Description:** Specifies the last call number in the range of items. Type the call number in quotes. This parameter is required.

**Option value:** The last call number in the range of items. Type the call number in quotes.

**Example:** `--inventory_call_end "PS3500"`

### 2.3.17.4 Inventory Location

**Option:** `--inventory_location`

**Description:** Specifies the location for which the inventory is being analyzed. This parameter is required.

**Option value:** A valid location code.

**Example:** `--inventory_location 10000`

### 2.3.17.5 Report on Items without a Call Numbers

**Option:** `--inventory_report_no_call`

**Description:** Specifies whether the report includes items without a call number. By default, this report does not report such items.

**Option value:**

- **1** to report items without call numbers.
- **0** to ignore items without call numbers.

**Example:** --inventory\_report\_no\_call 1

### **2.3.17.6 Invalid Shelving Statuses**

**Option:** --inventory\_invalid\_shelved\_status

**Description:** Defines a list of status codes that should not be associated with a shelved item.

**Option value:** A space-delimited list of numbers corresponding to statuses that should not be associated with a shelved item.

**Example:** --inventory\_invalid\_shelved\_status 4711 4721 2100

### **2.3.17.7 Inventory Date**

**Option:** --inventory\_date [YYYYMMDD]

**Description:** Specifies the date on which you began the inventory. If not specified, this setting defaults to the current date.

**Option value:** A valid date in YYYYMMDD format.

**Example:** --inventory\_date 20010305

### **2.3.17.8 Error Tolerance**

**Option:** --inventory\_error\_tolerance

**Description:** Specifies the number of places a book can be out of order before it is included in this report. If not specified, this setting defaults to four.

**Option value:** An integer value that indicates the number of places the item can be out of order without being reported in the Inventory Control report.

**Example:** --inventory\_error\_tolerance 2

### **2.3.17.9 Accession Number Range**

**Options:** --inventory\_accession\_start  
--inventory\_accession\_end

**Description:** Specifies the first and last valid accession number on the shelf that is being inventoried.

### 2.3.17.10 Collection Code

**Option:** --inventory\_collection\_code\_id

**Description:** Specifies the single collection code ID that is valid for the items on the shelf that is being inventoried.

### 2.3.17.11 User ID

**Option:** --inventory\_report\_user\_id

**Description:** Specifies the ID of the user who configured the Inventory Control Utility in InfoStation. When an item is modified by this report, the ID is used for the “Last Inventoried by” value on the Statistics tab of the Item Information window.

## 2.3.18 Options for i-tiva Users

The following **CircReport.exe** options are specific to users of the i-tiva Message system. The script **itivaReport.sh** specifies these options automatically when it runs **CircReport.exe**. These options are described here for informational purposes only.

**Note:** For additional information about **itivaReport.sh**, see the section “[CircReport.exe and the i-tiva Message System](#)” in this chapter.

### 2.3.18.1 Option for the i-tiva Notices File

**Option:** --i-tiva\_notices

**Description:** Specifies the filename of the notices file that will be created. The notices file is used by the i-tiva Message system for contacting patrons.

**Option values:** The filename of the i-tiva notices file.

**Example:** `--i-tiva_notices notices.itv`

### 2.3.18.2 Option for the i-tiva Results File

**Option:** `--i-tiva_results`

**Description:** Specifies the filename of the results file created by i-tiva for patrons who could not be contacted via the i-tiva Message system. The results file is used for generating paper or e-mail notices for patrons who could not be contacted by phone.

**Option values:** The filename of the i-tiva results file.

**Example:** `--i-tiva_results results.itv`

### 2.3.18.3 Patron Phone Opt-in

**Option:** `--require_phone_call_opt_in`

**Description:** Limits i-tiva to calling only those patrons who have opted in to receive phone communications according to tag 008 position 19 in the patron record. This option is disabled by default.

**Example:** `--require_phone_call_opt_in`

### 2.3.18.4 Voice, Text, or Both Voice and Text Notifications

**Option:** `--i-tiva_modules`

**Description:** Specifies the i-tiva modules (voice, text message, or both voice and text message) that are supported.

**Option values:**

- V – Each entry in the output i-tiva (.itv) file will have a media type of “V” for Voice. All patrons that qualify for notices will be updated unless the `--require_phone_call_opt_in` option is enabled, in which case patrons will only be updated if the Opt In For Phone Contact flag is enabled in the Patron Editor.
- T – Each entry in the output i-tiva (.itv) file will have a media type of “T” for Text. Only patrons that have a communication preference of Text or Email & Text will be updated. (The `--require_phone_call_opt_in` option is not considered.)

- VT – Each entry in the output i-tiva (.itv) file will either have a media type of “T” for Text or “V” for Voice, depending on how the patron should be contacted. All patrons will qualify for Voice unless the `--require_phone_call_opt_in` option is enabled, in which case patrons will only be updated if the Opt In For Phone Contact flag is enabled in the Patron Editor. Patrons will qualify for Text if they have a communication preference of Text or Email & Text. Only patrons that qualify for either Voice or Text (or both) will be updated and included in the output. If a patron qualifies for *both* Voice and Text, Voice will be used since it was the first module listed (“VT” instead of “TV”)
- TV – The software uses the same logic as for the “VT” option, except that patrons who qualify for *both* Voice and Text notifications will receive a text message instead of a phone call.

**Example:** `--i-tiva_modules VT`

### ***2.3.19 Options for Expired Requests and Expired Bookings Reports Only***

In this section we describe **CircReport.exe** options that are specific to the Expired Requests and Expired Bookings reports (`-t x` or `-t X`).

#### **2.3.19.1 Expired Categories**

**Option:** `--expired_categories`

**Description:** Specifies the report categories that will be included in the Expired Requests report. If this option is not included in the command, all report categories are included.

**Option values:** A space-delimited list of one or more of the following report category codes:

- 1 - Unsatisfied request.
- 2 - Item should be shelved at this location.
- 3 - Item should be shelved at another location.
- 4 - Item is now trapped for another request here.
- 5 - Item is now trapped for another request elsewhere.

**Example:** `--expired_categories 3 5`

### **2.3.19.2 Assess Fees for Requests “Not Picked Up”**

**Option:** --assess\_not\_picked\_up\_fees

**Description:** Assesses the Profiler-defined “Not picked up” fee for requests with a status of On Hold/Pick Up.

**Option values:** None

**Example:** --assess\_not\_picked\_up\_fees

### **2.3.20 Option for Expired Requests and Unsatisfied Request Cancellation Reports Only**

In this section we describe a **CircReport.exe** option that is specific to the Expired Requests and Unsatisfied Request Cancellation reports (**-t x** or **-t u**).

#### **2.3.20.1 Delete Request Fees**

**Option:** --delete\_request\_fees

**Description:** Indicates whether to delete request fees associated with requests that are being deleted.

**Option values:**

- 0 - No (*default*)
- 1 - Yes

**Example:** --delete\_request\_fees 1

#### **2.3.21 Option for Expired Requests Report Only**

In this section we describe a **CircReport.exe** option that is specific to the Expired Requests report (**-t x**).

##### **2.3.21.1 Trapped Requests**

**Option:** --request\_trapped

**Description:** Indicates whether the results of the report are limited to 1) untrapped expired requests, 2) trapped expired requests that are In Transit, or 3) trapped expired requests that are On Hold. If the `--request_trapped` option is omitted, all expired requests, whether trapped or untrapped, will be included in the results.

**Option values:**

- 0 - Untrapped
- 1 - In Transit
- 2 - On Hold

**Example:** `--request_trapped 02` (which would result in a report of expired requests that are either untrapped or On Hold)

## ***2.3.22 Option for Unsatisfied Requests and Unsatisfied Request Cancellation Reports***

In this section we describe a **CircReport.exe** option that is specific to the Unsatisfied Request Cancellation and Unsatisfied Requests reports (`-t u` or `-t U`).

### **2.3.22.1 Item Statuses**

**Option:** `--item statuses`

**Description:** Limits the report to requests on items at one or more item statuses.

**Option values:** Item status codes as defined in the Virtua Profiler.

**Example:** `--item statuses 3030`

**Note:** If you use a status code of 0, items with NO statuses will be included in the report output. You can use the status code of 0 with or without other status codes.

## ***2.3.23 Options for Billing Reports Only***

In this section we describe **CircReport.exe** options that are specific to the Bills and Recalled Bills reports (`-t 6` or `-t 7`).

### 2.3.23.1 Add Billing Fee

**Option:** --add\_billing\_fee

**Description:** Indicates whether the program adds the item price and processing fees to the patron record. If this option is not included in the command, the item price and processing fees are not added.

**Option values:**

- 1 - To add item price and processing fees.
- 0 - NOT add item price and processing fees.

**Example:** --add\_billing\_fee 1

### 2.3.23.2 Delete Overdue Fine

**Option:** --delete\_overdue\_fine

**Description:** Specifies whether the program deletes any checked out overdue fines that are linked to the patron and item when a billing report is run. If this option is not included in the command, checked out overdue fines are not deleted from the patron's account.

**Note:** A checked out overdue fine is a fine assessed on an overdue item that has NOT yet been checked in in Virtua. Existing checked out overdue fines are NOT assessed by Virtua unless you run the **UpdateOverdueFines.exe** executable. For details, see the *System Management: Circulation User's Guide*.

**Option values:**

- 1 - Checked out overdue fines are deleted when the report is run.
- 0 - Checked out overdue fines are NOT deleted when the report is run (*default*).

**Example:** --delete\_overdue\_fine 1

**Note:** If you use --delete\_overdue\_fine with option 1 AND the *Billed Overdue fines in Bills report?* setting is enabled in the Circulation Basic Options in the Virtua Profiler, the Virtua will delete existing checked out overdue fines that have accumulated *up until* the Bills report is run. Then when the report is run, Virtua may assess new "Billed" overdue fines.

### **2.3.23.3 Report Bills Exceptions**

**Option:** --report\_bills\_exceptions

**Description:** Specifies whether the Bills report will report on only those items that are configured (in the Location + Item Matrix) to use the Total Replacement Cost as the item replacement cost but for which the Total Replacement Cost could not be calculated.

**Option values:**

- **1** - The Bills report will report on only those items that are configured to use the Total Replacement Cost as the item replacement cost but for which the Total Replacement Cost cannot be calculated.
- **0** - The Bills report will report on all other items that qualify for the output based on the Bills report configuration.

**Example:** --report\_bills\_exceptions 1

### **2.3.24 Option for Overdues Report Only**

In this section we describe a **CircReport.exe** option that is specific to the Overdues report (-t 2).

#### **2.3.24.1 Check-in Dates**

**Option:** --checkin\_date

**Description:** A date in the form YYYYMMDDHHMI to be used as the notice date for all Overdue notices. This option is applicable/valid only when the report is run in non-update mode.

**Option values:** A date.

**Example:** --checkin\_date 20110515

### **2.3.25 Options for Unsatisfied Requests Report Only**

In this section we describe **CircReport.exe** options that are specific to the Unsatisfied Requests report (-t U).

### **2.3.25.1 Request Types**

**Option:** --request\_types <list of request type characters>

**Description:** Specifies the types of requests to be reported on.

**Option values:** One or more of the following characters:  
(H)old, (L)oan, (R)ecall and (P)age

**Example:** --request\_types HL

### **2.3.25.2 Level of Requests**

**Option:** --include\_bib\_level\_requests

**Description:** Indicates whether bib-level requests should be reported on.

**Option values:**  
0 - Do not report bib-level requests (*default*)  
1 - Report bib-level requests

**Example:** --include\_bib\_level\_requests 1

### ***2.3.26 Option for Request Expiration Notification and Extension Report Only***

In this section we describe a **CircReport.exe** option that is specific to the Request Expiration Notification and Extension report (**-t e**).

### **2.3.26.1 Range of Expiration Dates**

**Option:** --extension\_notify\_days <number of days>

**Description:** Indicates the range of expiration dates to look for. Only requests with an expiration date that falls within the specified number of days from the current date will be included in the report output.

**Option values:** A number.

**Example:** --extension\_notify\_days 7

### **2.3.27 Option for Review Notice Report Only**

In this section we describe a **CircReport.exe** option that is specific to the Review Notice report (-t 1).

#### **2.3.27.1 Suppress Notice for Overdue Items**

**Option:** --suppress\_overdue\_items

**Description:** Indicates via a flag whether overdue items will be reported in the Review Notice report. If set to 1, overdue items will still be advanced to the next notice type, but they will NOT be listed in the output and no review notices will be sent.

**Option values:**

0 - Do NOT suppress overdue item notices (*default*)

1 - Suppress overdue item notices

**Example:** --suppress\_overdue\_items 0

### **2.3.28 Option for Availability Reminder Notice Report Only**

In this section we describe a **CircReport.exe** option that is specific to the Availability Reminder Notice report (-t A).

### 2.3.28.1 Number of Days before Request Expiration

**Option:** --availability\_reminder\_days <number of days>

**Description:** Specifies the number of days before a request expires that an availability reminder notice needs to be sent.

**Option values:** A number.

**Example:** --availability\_reminder\_days 7

## 2.4 Output Options

In this section, we list the options that you can use to define the output generated by **CircReport.exe**. You can specify as many output options as needed by separating each code with a single space. **CircReport.exe** will print the chosen data in the order in which you list the options, separating each output field with a single tab or with the delimiter you specified with the **-d** option.

**Tip:** You can indicate a block of output options by specifying the first option in the block and the last option in the following format: **[first option]-[second option]**. For example, to specify options 12, 13, 14, and 15, type:

**12-15**

**Option:** -o

**Description:** Specifies the output fields for the report.

**Option values:** One or more of the following output fields:

- 1 - Report date.
- 2 - Due date.
- 3 - Notice number.
- 4 - Notice type.
- 5 - **REQUESTED** - If the item is requested.  
-OR-  
**NOT-REQUESTED** if the item is NOT requested.
- 6 - Check-out type code **Tip:** If you used the **-h** option, the labels you defined will appear for this output rather than the numeric codes.
- 7 - Operator ID.

- **8** - Item ID
- **9** - Item barcode.
- **10** - Item class code.
- **11** - Item class name.
- **12** - Item shelf location code.
- **13** - Item shelf location name.
- **14** - Main shelf location name for item.
- **15** - Units field from item record.
- **16** - Copy number from item record.
- **17** - Item price, which could be the Default Item Price from the Location + Item Matrix, the Price in the item record, or the Total Replacement Cost calculated on the Lost/Damaged window.
- **18** - Call number.
- **19** - Sorting form of call number.
- **20** - Patron ID (001).
- **21** - Patron barcode (015 \$a).
- **22** - Address preference flag (008/10 tag).
- **23** - Patron type code (030 \$a).
- **24** - Patron type (245 \$a).
- **25** - Patron name (1xx \$a \$b \$c).
- **26** - Patron personal name(1xx \$a).
- **27** - Patron name prefix (100 \$e).
- **28** - Patron name suffix (100 \$f).
- **29** - Patron primary C/O name (or ATTENTION OF) (270 \$h).
- **30** - Patron primary street address (*first occurrence of tag 270 subfield \$a with a first indicator of 1*).
- **31** - Second part of primary street address (*second occurrence of tag 270 subfield \$a with a first indicator of 1*).
- **32** - Third part of primary street address (*third occurrence of tag 270 subfield \$a with a first indicator of 1*).
- **103** - Fourth part of primary street address (*fourth occurrence of tag 270 subfield \$a with a first indicator of 1*).
- **104** - Fifth part of primary street address (*fifth occurrence of tag 270 subfield \$a with a first indicator of 1*).
- **33** - Patron primary postal code (270 \$e with a first indicator of 1).
- **34** - Patron primary city (270 \$b with a first indicator of 1).
- **35** - Patron primary county (270 \$c with a first indicator of 1)..
- **36** - Patron primary state or province (270 \$d with a first indicator of 1).
- **37** - Patron primary country (270 \$f with a first indicator of 1).
- **38** - Patron primary e-mail address (271 \$a with a first indicator of 1).

- 39 - Patron secondary e-mail address (271 \$a *with a first indicator of 2*).
- 40 - Patron preferred language (100 \$l or 110 \$l).
- 41 - Patron home library (247 \$a).
- 42 - Patron institution name (301 \$b).
- 43 - Patron college or school (301 \$c).
- 44 - Patron department (301 \$d).
- 45 - Patron primary telephone (270 \$k *with a first indicator of 1*).
- 46 - Patron secondary telephone (270 \$k *with a first indicator of 2*).
- 47 - Patron primary specialized telephone (270 \$j *with a first indicator of 1*).
- 48 - Patron secondary specialized telephone (270 \$j *with a first indicator of 2*).
- 49 - Patron primary fax (270 \$l *with a first indicator of 1*).
- 50 - Patron secondary fax (270 \$l *with a first indicator of 2*).
- 51 - Full patron record.
- 52 - Processing fee.
- 53 - Holdings ID (001 of holdings record).
- 54 - Holdings units (866 \$a of holdings record).
- 55 - Request type . . .
  - ◆ H - Hold
  - ◆ L - Loan
  - ◆ R - Recall
  - ◆ P - Page
- 56 - Pickup location code.
- 57 - Pickup location name.
- 58 - Hold expiration date, or on shelf until date.
- 59 - System name of user who ran report (for example dbadmin).
- 60 - Bibliographic ID (001).
- 61 - Parts title(s) (245 \$n \$p).
- 62 - Author (all tags and subfields indexed for a browse search; for a complete list, see the *Cataloging Reference Guide*, “Bibliographic Heading Tags.”)
- 63 - Title (all tags and subfields indexed for a browse search; for a complete list, see the *Cataloging Reference Guide*, “Bibliographic Heading Tags.”)
- 64 - Patron class (245 \$a).
- 65 - Category number for the output line.
- 66 - Error type for the inventory control report - number.
- 67 - Error type for the inventory control report - text. Below we provide a description of each error type code that may be returned in the output:
  - ◆ **BAR** - Barcode is not found in the system.
  - ◆ **RESR** - Item is supposed to be on reserve.
  - ◆ **LOC** - Item belongs to another location.
  - ◆ **CKDOUT** - Item is supposed to be checked out.
  - ◆ **STAT** - Item has a status that should not be attached to a shelved item.

- ◆ **NOCALL** - Item has no call number.
- ◆ **RANGE** - Item's call number is out of the specified range.
- ◆ **KWNMSS** - Item has previously been identified as missing (and still is).
- ◆ **MISS** - Item is missing (the first time the item is identified as such).
- ◆ **CALL** - Call number is out of order.
- ◆ **ACCNUM** – Item's accession number is not in the specified range.
- ◆ **COLCODE** – Item is missing the specified collection code.

**Note:** Even if an inventoried item possesses multiple error conditions, Virtua will report on only one. The one it reports on is based on the priority listing of error types as shown above. In this way, for example, if an inventoried item has two error conditions, 1) NOCALL and 2) RESR, Virtua will report on the RESR error condition only because it has the higher priority.

- **68** - Nearest correct call number to a misshelved book.
- **69** - Owning location of the book.
- **70** - Billing period.
- **71** - Maximum number of notices.
- **72** - Recallable date.
- **73** - Mail allowance period.
- **74** - Notice period.
- **75** - Due date.
- **76** - First notice period.
- **77** - Notice date.
- **78** - Unformatted item price value.
- **79** - Unformatted processing fee.
- **80** - MARC 21 parts title(s).
- **81** - Patron address block.
- **82** - Due date in YYYYMMDDHHMISS format.
- **83** - Second item level call number.
- **84** - Sorting form of second item level call number.
- **85** - Recall period.
- **86** - Patron course.
- **87** - Patron tutor.
- **89** - Patron health care status.
- **90** - Patron university card category.
- **91** - Patron isidore.
- **92** - Patron first name (1xx tag subfield h).
- **93** - Patron middle name (1xx tag subfield m).
- **94** - Patron last name (1xx tag subfield a).
- **95** - Current item status code.

- **96** - Current item status name.
- **97** - Item accession number.
- **98** - Patron other name.
- **99** - Patron other name suffix.
- **100** - Transaction reference number.
- **101** - Billed overdue fine. This is the overdue fine that Virtua would assess if the item were checked in at the time the report was run. The “Billed Overdue fines in Bills report?” setting in the Circulation Basic Options in the Virtua Profiler determines if Virtua will actually assess this fine for the Bills report.
- **102** - Alternate patron barcode (The alternate barcode used will be based on the "Use 017 as alternate patron barcode" setting in the Circulation Basic Options in the Virtua Profiler.)
- **105** - Item collection code
- **106** - Item collection code description
- **107** - Notice library name
- **108** - Notice library alternate name
- **109** - Notice library address 1
- **110** - Notice library address 2
- **111** - Notice library address 3
- **112** - Notice library address 4
- **113** - Notice library address 5
- **114** - Notice library telephone
- **115** - Item owning location name
- **116** – Request control number

## 2.5 Database Update Options

This section lists and describes each of the database update options that you can use with **CircReport.exe**.

### 2.5.1 Update Parameter Option

**Option:** -u

**Description:** Determines whether the program updates the database after the report runs. If you set this value to **1** . . .

- When you run reports that have a type of **1, 2, 3, 6, 7, or 8**, the program updates values in the database such as the notice count and notice type.

- When you run a report with the type of **i**, any item found on the shelf that has been reported as missing is given a status of Available.

If you do not include this option in the command, the program will not update the database.

**Parameter value:**

- 1 - To update the database.
- 0 - To NOT update the database.

**Example:** -u 1

**Note:** This parameter does not affect the `--add_billing_fee` option.

## 2.5.2 Limit Update to Address

**Option:** `--limit_update_to_address_pref`

**Description:** Specifies whether the program limits database updates and report results returned depending on the patron's address preference (patron record tag 008, position 10).

**Note:** This option takes effect ONLY when the [update parameter option](#) is set to 1.

**Option values:**

- 1 - Database updates are only made for those patrons whose address preference is set to e-mail AND for whom a valid e-mail address exists in the 271 tag of the patron record.
- 0 - Database updates are only made for those patrons whose address preference is set to paper address in the patron record.

**Example:** `--limit_update_to_address_pref 1`

## 2.6 CircReport.exe and the i-tiva Message System

For customers who use the i-tiva Message system to contact patrons about overdue items, billed items, available items, and expired requests, Virtua offers the script **itivaReport.sh**. The script runs **CircReport.exe** multiple times (once for each notice type) and generates an input file that the i-tiva Message system uses to contact patrons.

## 2.6.1 Scheduling Virtua and the i-tiva Message System

In order for the Virtua system and i-tiva Message system to work together to ensure that all patrons are contacted, special care must be taken in scheduling

**CircReport.exe**, the i-tiva Message system, and InfoStation (i-tiva) reports. Below, we describe the sequence of events that must occur:

- Step 1** The script **itivaReport.sh** is run via a scheduled job, producing an input file for i-tiva to use to contact patrons. The Virtua database is automatically updated to indicate that these patrons have received notices.
- Step 2** The i-tiva system retrieves the resulting input file from the Virtua server and uses it to call patrons during the designated call hours.
- Step 3** The i-tiva system generates a report of those patrons it could **not** contact that day and FTPs the file to the Virtua server.
- Step 4** The scheduled InfoStation (i-tiva) reports run **after** the i-tiva report of patrons it could not contact is FTPed to the Virtua server. The InfoStation (i-tiva) reports use this file to produce circulation notices and bills for all patrons who could not be contacted.
- Step 5** The process begins again -- The script **itivaReport.sh** is run as a scheduled job, producing an input file for i-tiva to use to contact patrons. The Virtua database is updated to indicate that these patrons have received notices.

## 2.6.2 Running and Modifying the itivaReport.sh Script

The **itivaReport.sh** script uses the following command format. Note that all options supported by the script must be specified.

```
itivaReport.sh [base_filename] [report_user] [update_flag]
[require_phone_call_opt_In] [supported_modules]
```

**Example:** `itivaReport.sh output staff 0 1 VT`

Where ...

- **[base\_filename]** - Is the base name used for the files that are generated. Virtua will create the following files using this name:
  - ◆ **[base\_filename].itv** - A file that contains the information that will be submitted to the i-tiva Message system.
  - ◆ **[base\_filename].[report number].brp** - A file that contains backup report information. The script generates one of these files for *each* circulation notice type. The numbers range from 30 to 36 for i-tiva-related reports (see the *InfoStation Reference Guide*). InfoStation will use these files, which will reside in the **/temp** directory of the InfoStation installation, to generate notices for patrons who could not be contacted via the i-tiva Message system.
  - ◆ **[base\_filename].log** - A log file.
- **[report\_user]** - Specifies the username of the user that will be used to run CircReport.exe.
- **[update\_flag]** - Specifies whether you want to update the database. A value of **1** indicates that the database will be updated, while a value of **0** indicates that the database will NOT be updated. If you do not specify an update option, the database will be updated by default.

**Note:** This is the same update option that is used by **CircReport.exe**. For details, see the section “[Update Parameter Option](#)” in this chapter.

- **[require\_phone\_call\_Opt\_In]** - Restricts the script to produce output only for patrons who have the phone\_call\_opt\_in flag turned ON in the 008 tag position19 of the patron record. This parameter is optional. A value of **1** indicates that the script will respect this flag, while a value of **0** indicates that the script will NOT respect this flag. The default behavior is to ignore the phone\_call\_opt\_in flag.
- **[supported\_modules]** - Specifies the i-tiva modules (voice, text message, or both voice and text message) that are supported. The following options may be used:
  - ◆ V – Each entry in the output i-tiva (.itv) file will have a media type of “V” for Voice. All patrons that qualify for notices will be updated unless the --require\_phone\_call\_opt\_in option is enabled, in which case patrons will only be updated if the Opt In For Phone Contact flag is enabled in the Patron Editor.
  - ◆ T – Each entry in the output i-tiva (.itv) file will have a media type of “T” for Text. Only patrons that have a communication preference of Text or Email & Text will be updated. (The --require\_phone\_call\_opt\_in option is not considered.)
  - ◆ VT – Each entry in the output i-tiva (.itv) file will either have a media type of “T” for Text or “V” for Voice, depending on how the patron should be contacted. All patrons will qualify for Voice unless the --require\_phone\_call\_opt\_in option is enabled, in which case patrons will only

be updated if the Opt In For Phone Contact flag is enabled in the Patron Editor. Patrons will qualify for Text if they have a communication preference of Text or Email & Text. Only patrons that qualify for either Voice or Text (or both) will be updated and included in the output. If a patron qualifies for *both* Voice and Text, Voice will be used since it was the first module listed (“VT” instead of “TV”)

- ◆ TV – The software uses the same logic as for the “VT” option, except that patrons who qualify for *both* Voice and Text notifications will receive a text message instead of a phone call.

By default, when you run the script **itivaReport.sh**, Virtua generates **base\_filename.\*** files for the following notice types (i.e., all the notice types supported by the i-tiva Message system):

- Overdues
- Recalled Overdues
- Bills
- Recalled Bills
- Availability
- Recalls
- Expired Requests

If you want **itivaReport.sh** to generate results for a different set of notices (for example, if you want results for only three of the notice types instead of all seven), you can modify the **noticesToGenerate** string in the **itivaReport.sh** script. To do so, open the script in a text editor, and modify the string according to the instructions provided in the script.

After the i-tiva system uses **base\_filename.itv** to contact patrons, i-tiva...

- Generates a file containing information about patrons who could not be contacted.  
-AND-
- FTPs this file to the Virtua server where InfoStation can use it to produce reports (for details, see the *InfoStation/Circulation Reports Reference Guide*).

### Example:

To run **itivaReport.sh** so that it has a base filename of “itivareport,” **CircReport.exe** is run with the user “staff,” the database is updated, only those patrons who have opted in receive calls from i-tiva, and the voice notification module is enabled, you would type at the command line:

```
itivaReport.sh itivareport staff 1 1 v
```

The script runs, and...

- Creates files named **itivareport.itv**, **itivareport.brp**, and **itivareport.log**.
- The user “staff” is used to run **CircReport.exe**.
- The i-tiva system contacts only those patrons whose patron records indicate that they have opted in to receive telephone communications.
- The database is updated to indicate that the patrons have been contacted.
- The Voice module is supported, which means that each entry in the output i-tiva (.itv) file will have a media type of “V” for Voice. All patrons that qualify for notices will be updated unless the `--require_phone_call_opt_in` option is enabled, in which case patrons will only be updated if the Opt In For Phone Contact flag is enabled in the Patron Editor.
- Any patrons who could not be contacted are kept to a list that InfoStation can use to run the i-tiva versions of the Overdues, Recalled Overdues, Bills, Recalled Bills, Availability, and Expired Requests reports (for details, see the *InfoStation/Circulation Reports Reference Guide*).

## 3. Converting Call Numbers to a Normalized Form

Several InfoStation reports use the program **callNumberKey.exe** to convert call numbers into a normalized sort form. InfoStation uses the normalized sort form to . . .

- Sort call numbers.
- Filter results by a call number range.

If you are having problems with either of these aspects of an InfoStation report, you can run **callNumberKey.exe** from the command line to troubleshoot.

Below, we display the basic format for running **callNumberKey.exe**:

```
callNumberKey.exe < [input file] > [output file]
```

Where . . .

- **[input file]** is the name of the file which lists, line-by-line, the call numbers you want to normalize.
- **[output file]** is the file to which you want **callNumberKey.exe** to generate normalized call numbers.

Instead of using an input file of call numbers, you can enter individual call numbers at the command line. To do this,

1. At the prompt, type: **callNumberKey.exe**
  2. Press Enter.
  3. Type a call number, and press Enter.  
The program displays the sort form of the call number.
  4. Type another call number, and press Enter to see the sort form displayed.
- OR-
- If you have finished entering call numbers, signal the program that input is complete by pressing **Ctrl+D**.

## 4. Generating a List of URLs

The script **getURLs.sh** reports on the URLs in bibliographic, holdings, and item records in your database. Using this script, you can create a report that outputs the ID of each record that contains a URL and/or the actual URL.

**Hint:** After you have generated the file of URLs, you can use InfoStation's URL Checker report to check for broken URLs. For details, see the *InfoStation/Cataloging Reports Reference Guide*.

The basic format for running **getURLs.sh** is:

```
getURLs.sh [record type] [output content] [output file]
```

Where . . .

- **[record]** type is one of the following codes that specify the type of record on which this script will report:
  - ◆ **b** - Bibliographic record.
  - ◆ **h** - Holdings record.
  - ◆ **i** - Item record.
- **[output content]** is one of the following codes that specify the output of this report:
  - ◆ **1** - Generates one file listing only the ID of each qualifying record.
  - ◆ **2** - Generates one file listing only the URL of each qualifying record.
  - ◆ **3** - Generates one file listing both the ID and the URL of each qualifying record.
  - ◆ **0** - Generates all three of the above outputs in three separate files.

**Note:** InfoStation's URL Checker report requires an input file that lists *both* the record ID and the URL. If you are running **getURLs.sh** to create a file for use with InfoStation, you need to run the executable using option 3 or 0.

- **[output file]** is the base filename to which the script will generate output. If for the **[output content]** option you choose . . .
  - ◆ **1, 2, or 3** - The script will write the output to the file you specify in this parameter.
  - ◆ **0** - The script will output three files using the base filename you specify here and the following extensions:

- **\_1\_ids** - For the file containing the list of record IDs.
- **\_2\_URLs** - For the file containing the list of URLs.
- **\_3\_idsURLs** - For the file containing both the record IDs and the URLs.

For example . . .

- If you want to generate to a file named **item\_urls**, a list of the URLs in the item records in your database, type:

```
getURLs.sh i 2 item_urls
```

This command generates a file named **item\_urls** that lists, line-by-line, the item URLs stored in your database.

- If you want to generate to three files with a base filename of **bibliographic\_urls** . . .
  - ◆ A list of bibliographic IDs with 856 \$u tags.
  - ◆ A list of URLs contained in bibliographic records.
  - ◆ A list of the both the bibliographic IDs and the URLs.

Type:

```
getURLs.sh b 0 bibliographic_urls
```

This command generates . . .

- ◆ A file named **bibliographic\_urls\_1\_ids** that lists, line-by-line, the IDs of the bibliographic records in your database that contain URLs.
- ◆ A file named **bibliographic\_urls\_2\_URLs** that lists, line-by-line, the URLs contained in the bibliographic records in your database.
- ◆ A file named **bibliographic\_urls\_3\_idsURLs** that lists, line-by-line, the IDs and the associated URLs of the bibliographic records in your database that contain URLs.

## 5. Accessing Tag-specific Record Data and Creating Custom Views

This chapter describes how to access tag-specific record data using the MARC package and the AUTHORITYMARC package, which are distributed for use with the Ad Hoc Reporting System, and how to create custom views for specific bibliographic tags. This chapter is intended for libraries that have purchased and installed the Ad Hoc Reporting System. If you do not have the Ad Hoc Reporting System, you will not be able to use the procedures documented in this chapter. For information about obtaining the Ad Hoc Reporting System, contact Innovative Customer Support.

This chapter covers the following topics:

- ⇒ [About the Storage of Bibliographic and Authority Records in Virtua](#)
- ⇒ [Retrieving Bibliographic and Authority Record Data](#)
- ⇒ [Creating Views for the Bibliographic\\_Record Table](#)

### 5.1 About the Storage of Bibliographic and Authority Records in Virtua

Bibliographic records are stored in MARC format in the `bibliographic_record` table in the Virtua database. Authority records are stored in MARC format in the `authority_note_record` database table.

The `bibliographic_record` table consists of a number of fields, two of which are:

- **Bib\_ID** - The unique identifier for the MARC record. This field is populated with the value in the 001 tag of each MARC record.
- **MARC\_Record** - The entire MARC record stored as a Binary Large Object (BLOB).

Likewise, the `authority_note_record` table also consists of a number of fields, two of which are:

- **Auth\_ID** - The unique identifier for the MARC record. This field is populated with the value in the 001 tag of each MARC record.

- **Note\_Record** - The entire MARC record stored as a Binary Large Object (BLOB).

In order to efficiently accommodate large records, the Virtua database stores each bibliographic and authority record as a Binary Large Object (BLOB). The BLOB data type gives Virtua the ability to . . .

- Make efficient use of available disk space.
- Index records of virtually any size.
- Easily modify records in the database.

## 5.2 Retrieving Bibliographic and Authority Record Data

Unlike data stored as data types such as VARCHAR2, the binary data in a BLOB in the bibliographic\_record and the authority\_note\_record table cannot be extracted with conventional tools. To give you a way to access and retrieve bibliographic and authority data from your database, Innovative provides a set of scripts to be used to...

- [Retrieve information from a variable-field tag](#)
- [Retrieve information from a fixed field tag](#)
- [Retrieve information from the Leader](#)
- [Decode bibliographic Leader data](#)
- [Retrieve authority information from a variable field tag](#)
- [Retrieve authority information from a fixed field tag](#)
- [Retrieve authority information from the Leader](#)

The scripts are discussed in detail, below.

### 5.2.1 Retrieving Bibliographic Information from a Variable Field Tag

The script **RetrieveBibVariableTagData.sh** produces a comma-delimited file of Bib IDs and tag data stored in variable field tags within bibliographic records.

**Usage:**

```
RetrieveBibVariableTagData.sh bib_id Tag_Number [Tag_Occurrence]
[Subfield] [Subfield_Occurrence] [Validation]
```

Where . . .

- **bib\_id** (*required*) is the specific bibliographic ID of the record you want to retrieve tag data from. If you want to retrieve ALL bibliographic records, type **bib\_id** for this parameter.
- **Tag\_Number** (*required*) is the bibliographic record tag number you want to retrieve data from. This must be a variable field tag number.
- **[Tag\_Occurrence]** (*optional*) specifies the occurrence of the tag you want to return. You can specify only one occurrence within a single command. Default value is 1
- **[Subfield]** (*optional*) specifies the subfield you want to return. If you do not specify a subfield, all subfields in the tag are returned.
- **[Subfield\_Occurrence]** (*optional*) specifies the occurrence of the subfield you want to return. You can specify only one subfield occurrence within a single command. Default value is 1.
- **[Validation]** (*optional*) determines whether the record is validated. Enter 1 to validate the record or 0 to skip validation. Default value is 1.

## 5.2.2 Retrieving Bibliographic Information from a Fixed Field Tag

The script **RetrieveBibFixedTagData.sh** produces a comma-delimited file of Bib IDs and tag data stored in fixed field tags within bibliographic records

Usage:

```
RetrieveBibFixedTagData.sh bib_id tag_number start_position
[length] [validation]
```

Where . . .

- **bib\_id** (*required*) is the specific bibliographic ID of the record you want to retrieve tag data from. If you want to retrieve ALL bibliographic records, type **bib\_id** for this parameter.
- **tag\_number** (*required*) is the bibliographic record tag number you want to retrieve data from. This must be a fixed field tag number 001-009.
- **start\_position** (*required*) is the first position of the fixed field tag that you want to return. MARC starts with position 0. Default is 0.
- **[length]** (*optional*) specifies the number of fields to return, starting from the position you specified in the start\_position parameter. Default value is 1.
- **[validation]** (*optional*) determines whether the record is validated. Enter 1 to validate the record or 0 to skip validation. Default value is 1.

### 5.2.3 Retrieving Bibliographic Information from the Leader

The script **RetrieveBibLeaderData.sh** produces a comma-delimited file of Bib IDs and data stored in the Leader within bibliographic records.

**Usage:**

```
RetrieveBibLeaderData.sh bib_id [start_position] [length]
```

Where . . .

- **bib\_id** (*required*) is the specific bibliographic ID of the record you want to retrieve Leader data from. If you want to retrieve ALL bibliographic records, type **bib\_id** for this parameter.
- **[start\_position]** (*optional*) is the first position of the Leader that you want to return. MARC starts with position 0. Default is 0.
- **[length]** (*optional*) specifies the number of fields to return, starting from the position you specified in the start\_position parameter. Default value is 24.

### 5.2.4 Decoding Bibliographic Leader Values

The script **DecodeBibLeaderData.sh** displays decoded Leader data for a specified Bib ID and Leader value.

**Usage:**

```
DecodeBibLeaderData.sh Decode_Function Leader_Position Bib-ID
```

Where . . .

**Decode\_Function** is one of the elements listed below, which is specified along with the corresponding **Leader\_Position** and the **Bib ID** of the associated record:

- **Decode\_BibLdr\_RecStatus** - Decodes the Record Status specified in position 5 of the Leader. This function returns the following values:
  - ◆ Increase in encoding level
  - ◆ Corrected or revised
  - ◆ Deleted
  - ◆ New

- ◆ Increase in encoding level from prepublication
- ◆ NULL
- **Decode\_BibLdr\_RecType** - Decodes the Record Type specified in position 6 of the Leader. This function returns the following values:
  - ◆ Language material
  - ◆ Printed music
  - ◆ Manuscript music
  - ◆ Cartographic material
  - ◆ Manuscript cartographic material
  - ◆ Projected medium
  - ◆ Nonmusical sound recording
  - ◆ Musical sound recording
  - ◆ Two-dimensional nonprojectable graphic
  - ◆ Computer file
  - ◆ Kit
  - ◆ Mixed material
  - ◆ Three-dimensional artifact or naturally occurring object
  - ◆ Manuscript language material
  - ◆ NULL
- **Decode\_BibLdr\_Level** - Decodes the Bibliographic Level specified in position 7 of the Leader. This function returns the following values:
  - ◆ Serial component part
  - ◆ Collection
  - ◆ Subunit
  - ◆ Monograph/item
  - ◆ Serial
  - ◆ Subject
  - ◆ Author
  - ◆ Series Title
  - ◆ Uniform Title
  - ◆ Author/Title Series
  - ◆ User Defined
  - ◆ Subdivision
  - ◆ Genre/Form

- **Decode\_BibLdr\_ControlType** - Decodes the Control Type specified in position 8 of the Leader. This function returns the following values:
  - ◆ No specific type of control
  - ◆ Archival
  
- **Decode\_BibLdr\_Coding** - Decodes the Character Coding Scheme specified in position 9 of the Leader. This function returns the following values:
  - ◆ MARC-8
  - ◆ UCS/Unicode
  
- **Decode\_BibLdr\_EncodeLvl** - Decodes the Encoding Level specified in position 17 of the Leader. This function returns the following values:
  - ◆ Full level
  - ◆ Full level, material not examined
  - ◆ Less-than-full level, material not examined
  - ◆ Abbreviated level
  - ◆ Core level
  - ◆ Partial (preliminary) level
  - ◆ Minimal level
  - ◆ Prepublication level
  - ◆ Unknown
  - ◆ Not applicable
  
- **Decode\_BibLdr\_CatForm** - Decodes the Descriptive Cataloging Form specified in position 18 of the Leader. This function returns the following values:
  - ◆ Non-ISBD
  - ◆ AACR
  - ◆ ISBD
  - ◆ Unknown
  
- **Decode\_BibLdr\_LinkRec** - Decodes the Linked Record Requirement specified in position 19 of the Leader. This function returns the following values:
  - ◆ Related record not required
  - ◆ Related record required

**Note:** The decode functions were designed and tested for databases storing bibliographic records in MARC 21 format. These functions may not return accurate information on databases using other MARC formats.

## 5.2.5 Retrieving Authority Information from a Variable Field Tag

The script **RetrieveAuthVariableTagData.sh** produces a comma-delimited file of Auth IDs and tag data stored in variable field tags within permanent authority records. This script extracts only *non-indexed* authority data.

### Usage:

```
RetrieveAuthVariableTagData.sh auth_id Tag_Number [Tag_Occurrence]
[Subfield] [Subfield_Occurrence] [Validation]
```

Where . . .

- **auth\_id** (*required*) is the specific authority ID of the record you want to retrieve tag data from. If you want to retrieve ALL permanent authority records, type **auth\_id** for this parameter.
- **Tag\_Number** (*required*) is the authority record tag number you want to retrieve data from. This must be a non-indexed, variable field tag number.
- **[Tag\_Occurrence]** (*optional*) specifies the occurrence of the tag you want to return. You can specify only one occurrence within a single command. Default value is 1
- **[Subfield]** (*optional*) specifies the subfield you want to return. If you do not specify a subfield, all subfields in the tag are returned.
- **[Subfield\_Occurrence]** (*optional*) specifies the occurrence of the subfield you want to return. You can specify only one subfield occurrence within a single command. Default value is 1.
- **[Validation]** (*optional*) determines whether the record is validated. Enter 1 to validate the record or 0 to skip validation. Default value is 1.

## 5.2.6 Retrieving Authority Information from a Fixed Field Tag

The script **RetrieveAuth FixedTagData.sh** produces a comma-delimited file of Auth IDs and tag data stored in fixed field tags within permanent authority records

### Usage:

```
RetrieveAuthFixedTagData.sh auth_id tag_number start_position
[length] [validation]
```

Where . . .

- **auth\_id** (*required*) is the specific authority ID of the record you want to retrieve tag data from. If you want to retrieve ALL permanent authority records, type **auth\_id** for this parameter.
- **tag\_number** (*required*) is the bibliographic record tag number you want to retrieve data from. This must be a fixed field tag number 001-009.
- **start\_position** (*required*) is the first position of the fixed field tag that you want to return. MARC starts with position 0. Default is 0.
- **[length]** (*optional*) specifies the number of fields to return, starting from the position you specified in the start\_position parameter. Default value is 1.
- **[validation]** (*optional*) determines whether the record is validated. Enter 1 to validate the record or 0 to skip validation. Default value is 1.

### 5.2.7 Retrieving Authority Information from the Leader

The script **RetrieveAuthLeaderData.sh** produces a comma-delimited file of Auth IDs and data stored in the Leader within permanent authority records.

#### Usage:

```
RetrieveAuthLeaderData.sh auth_id [start_position] [length]
```

Where . . .

- **auth\_id** (*required*) is the specific authority ID of the record you want to retrieve Leader data from. If you want to retrieve ALL permanent authority records, type **auth\_id** for this parameter.
- **[start\_position]** (*optional*) is the first position of the Leader that you want to return. MARC starts with position 0. Default is 0.
- **[length]** (*optional*) specifies the number of fields to return, starting from the position you specified in the start\_position parameter. Default value is 24.

## 5.3 Creating Views for the Bibliographic\_Record Table

While the functions discussed in the section “[Retrieving Bibliographic and Authority Record Data](#)” are useful tools for retrieving information from the bibliographic\_record table, they are limited in that they can return information for only one tag at a time.

To provide you with an additional means of retrieving information from the bibliographic\_record table, the Ad Hoc Reporting package includes a script that creates custom Ad Hoc views for specific bibliographic tags. The script **CreateCustomAdHocView.sh** lets you create views that provide access to one or more tags stored in the bibliographic\_record table.

Views created with **CreateCustomAdHocView.sh** are accessible via any report writing tool that accesses an Oracle database.

### Usage:

```
CreateCustomAdHocView.sh View_Name \"[Tag Definitions]\" [880 Flag]
```

Where...

- **View Name** is the name of the view you are creating. The MARC package automatically appends **\_mv** to the end of the view name to distinguish it as a custom view.
- **[Tag Definitions]** is the space-delimited list of tags for this view. Unless you specify otherwise, each tag you choose returns all subfields in the first occurrence of the tag. You can choose:
  - ◆ Subsequent occurrences of the tag by typing:

```
[tag]#[occurrence number]
```

For example, **650#3** specifies the third occurrence of the 650 tag.

- ◆ A specific subfield to return by typing:

```
[tag]s[subfield letter]#[occurrence number]
```

For example, **650sc#2** specifies the second occurrence of subfield \$c in the 650 tag. **Tip:** If you want the first occurrence of the subfield, you can omit the **#[occurrence number]** part of the string.

- ◆ Subsequent occurrences of the 650 tag and a specific subfield by typing:

```
[tag]#[occurrence number]s[subfield letter]#[occurrence number]
```

For example, **650#3sc#2** specifies the second occurrence of subfield \$c in the third occurrence of the 650 tag.

- **[880 Flag]** is a flag that determines whether information in an associated 880 tag will display in place of the tags specified for the view. To set this flag, type:
  - ◆ **0** - To ignore associated 880 tags.
  - ◆ **1** - To display associated 880 tags in place of the original tag.

**Example:** Let's say that you want to create a view named `Bib_View_mv` that stores the following tags and ignores associated 880 tags:

- 100 \$a \$d
- 245 \$a \$b \$c
- 250 \$a
- 260 (all subfields)
- 300 (all subfields)
- 440 (all subfields)
- 500 \$a (first tag occurrence)
- 500 \$a (second tag occurrence)
- 650 \$v (first subfield occurrence)
- 650 \$v (second subfield occurrence)

At the command line, type:

```
CreateCustomAdHocView.sh Bib_View "100sa 100sd 245sa 245sb 245sc 250sa  
260 300 440 500#1sa 500#2sa 650sv#1 650sv#2" 0
```

This command creates a view named `Bib_View_mv` with the following fields:

- Bib\_ID
- T100#1SA#1
- T100#1SD#1
- T245#1SA#1
- T245#1SB#1
- T245#1SC#1
- T250#1SA#1
- T260#1
- T300#1
- T440#1
- T500#1SA#1
- T500#2SA#1
- T650#1SV#1
- T650#1SV#2

## 6. InfoStation Tool for Extracting Bibliographic Data

The script **get\_bibs.pl** uses a list of bibliographic IDs to extract information from the bibliographic records stored in your database. This method of extracting data from the BLOBs lets you view the desired data in a readable format without creating a view or running a shell script.

This chapter covers the following topics:

- ⇒ [Using get\\_bibs.pl](#)
- ⇒ [Example of a Command to Generate Reports with get\\_bibs.pl](#)

### 6.1 Using get\_bibs.pl to Extract Bibliographic Data

From a list of bibliographic IDs that you provide in a text file, the program **get\_bibs.pl** extracts bibliographic data from the BLOBs. Using **get\_bibs.pl**, you can . . .

- Specify a list of records from which you want to extract data.
- Choose which tags you want to display for the records you are extracting.
- Choose a format in which you want to display your results.
- Specify a file to save the output of your query.

**Note:** The **get\_bibs.pl** script is not designed to work with the Ad Hoc views or third-party report writer tools. It will not extract data stored outside of the `bibliographic_record` table.

You can run **get\_bibs.pl** from the UNIX command line. To do this . . .

1. Log in as **dbadmin** on the server on which **get\_bibs.pl** is located.
2. Navigate to the `cgi-bin` directory of your InfoStation installation. If you installed InfoStation to the default directory, the path will be . . .

`/usr/vtls/virtua/webrpt_XX_X/cgi-bin`

... where XX\_X is the version number of your InfoStation installation.

3. Type the following string to set the REPORTS\_HOME directory as an environment variable:

`export REPORTS_HOME=/usr/vtls/virtua/webrpt_XX_X/`

... where XX\_X is the version number of your InfoStation installation.

**Note:** This environment variable needs to be set once each session. After you set this environment variable the first time, you will not need to do so again until you end your UNIX session.

4. At the prompt type:

`./get_bibs.pl`

-AND-

Type the following command line options:

**Note:** Do not press the Enter key until you have finished typing all options *in the order in which they are listed below*.

- **-bibfile [filename]** (*Required*) - The filename of the list of bibliographic IDs of the records from which you want to extract data. This file must be a text file that lists each bibliographic number on a separate line. For information on a method to extract bibliographic IDs from your database, see the section [“Generating a List of Bibliographic IDs”](#) in this guide.
- **-ldr 1** (*Optional*) - Displays the record’s Leader in the output.
- **-tag "[tag information]"** (*Required for at least one tag*) - Specifies the tags that you want to appear in the output. Enter each tag that you want to appear in the following format:

`-tag "[Occurrence] [Tag Number] [Subfield]"`

where ...

- ◆ **Occurrence** specifies which tag in a repeating series is displayed. You can specify as many tags as you wish. Tags will be displayed in the order in which you enter them on the command line. Type:
  - **first** - To display the first occurrence of a tag.
  - **first: [#]** - To display the first number of occurrences of a tag. For example, to display the first three occurrences of a tag, type:
 

```
first:3
```
  - **all** - To display all occurrences of a tag.
  - **last** - To display the last occurrence of a tag.
  - **last: [#]** - To display the last number of occurrences of a tag. For example, to display the last three occurrences of a tag, type:
 

```
last:3
```
- ◆ **Tag** specifies the three-digit tag you want to display.
- ◆ **Subfield** specifies the subfields you want to display for the given tag.
  - If you want to display more than one subfield, separate each subfield with a single space.
  - If you want to display all subfields for a tag, type **\***

For example, if you want to display all occurrences of the 100 tag, subfields \$a, \$b, and \$c, type:

```
-tag "all 100 a b c"
```

- **-sort "[tag information]"** - Specifies the tags by which you want to sort the records in your query. You can specify as many sorts as you wish. Sorts will be applied in the order you enter them on the command line. Enter each tag that you want to use for sorting in the following format:

```
-sort "[Occurrence] [Tag Number] [Subfield]"
```

where . . .

- ◆ **Occurrence** specifies which tag in a repeating series is used. Type:
  - **first** - To use the first occurrence of a tag.
  - **first: [#]** - To use the first number of occurrences of a tag. For example, to use the first three occurrences of a tag, type:
 

```
first:3
```
  - **all** - To use all occurrences of a tag.
  - **last** - To use the last occurrence of a tag.
  - **last: [#]** - To display the last number of occurrences of a tag. For example, to use the last three occurrences of a tag, type:
 

```
last:3
```
- ◆ **Tag** specifies the three-digit tag you want to use for sorting.
- ◆ **Subfield** specifies the subfields of the given tag that you want to use for sorting.

- If you want to use more than one subfield, separate each subfield with a single space.
- If you want to use all subfields for a tag, type **\***

For example, if you want to sort by the first occurrence of the 245 tag, subfields \$a, \$b, and \$c, type:

```
-sort "first 245 a b c"
```

**Note:** Choosing a sort will slow down the time it takes for **get\_bibs.pl** to return results. The more sorts you choose, the longer **get\_bibs.pl** will take. On some servers, attempting too many sorts can cause a system crash.

- **-output [output code]** (*Required*) - Specifies the format in which the report is displayed. You can choose any one of the following options:
  - ◆ **a** - Plain text. Below, we display sample output in this format.

```
Record 15425
100  Aul D. Kengshourses
245  Egg Assembly in Six Easy Steps
500  Applies only to initial assembly.
```

**d** - Delimited format. The default delimiter is a single tab. You can specify a different delimiter by using the **-delimiter** option. If you choose this option, we recommend that you direct the output to a text file. Below, we display sample output in this format.

```
Bib ID      100    245    260 b
65744      McClanahan, Ed      Natural Man      Farrar, Straus, and Giroux
```

- ◆ **l** - Plain text with the subfield codes and indicator numbers appearing. Below, we display sample output in this format.

```
Record 95249
100 10 \a Faulkner, William,\d 1897-1962.\w cn
245 10 \a As I lay dying.
260 0  \a New York,\b Random House
300  \a 250 p.\c 21 cm.
```

- **-host [server host name]** (*Required*) - The host name of the database you are querying.
- **-sid [SID for the database]** (*Required*) - The Oracle System Identifier (SID) of the database you are querying.

- **-username [username for the database]** (*Required*) - The username for the database you are querying.
- **-password [password for given username]** (*Required*) - The password for the username you specified for the **-username** option.
- **-charset [language].AL32UTF8** (*Required*) - This option specifies the language and character set of the database. For language, type the Oracle language setting for your database. For information about the language setting, see your Oracle documentation. ALL Virtua databases use the AL32UTF8 character set.
- **-delimiter "[delimiter]"** (*Optional*) - If your output option is set to delimited text and you want to use a delimiter other than a tab, you can specify a delimiter with this option. For example, if you want to generate comma-delimited output, type:

**-delimiter ", "**

- **> [Name of output file]** (*Optional*) - Specifies a filename to which you want to save the output. If you choose this option, the results of your query will NOT display on your screen.

5. Press the Enter key on your keyboard.

The **get\_bibs.pl** program extracts the data you requested and displays or saves it in the format you specified.

## 6.2 Example of a Command to Generate a Report with **get\_bibs.pl**

Here is an example of a command using **get\_bibs.pl** to generate a report with the following specifications:

- Extracts information from records with bibliographic IDs that are listed in a file named **bibids.txt**.
- Extracts records from a database with . . .
  - ◆ A host name of **maginot.vtls.com**.
  - ◆ A System Identification (SID) of **vtls01**.

- ◆ A username of **syscli**.
- ◆ A password of **adhoc01**.
- ◆ A language setting of American.
- For each record, displays:
  - ◆ The Leader
  - ◆ The first occurrence of the 082 \$a tag.
  - ◆ All occurrences of the 100 tag, including all subfields.
  - ◆ All occurrences of the 245 tag, including all subfields.
  - ◆ The first two occurrences of the 500 \$a \$3 tag.
- Sorts records by the first occurrence of the 245 \$a tag.
- Displays the output in plain text.

Type the following command:

```
./get_bibs.pl -bibfile bibids.txt -ldr 1 -tag "first 082 a" -tag  
"all 100 *" -tag "all 245 *" -tag "first:2 500 a 3" -sort "first  
245 a" -output a -host maginot.vtls.com -sid vtls01 -username  
syscli -password adhoc01 -charset AMERICAN.AL32UTF8
```

## 7. Appendix A - Scripts and Executables Discussed in this Guide

In this appendix, we list and briefly describe the scripts and executables that are documented in this user's guide. You can click the hyperlink in the Filename column to access additional information about that program.

Filename	Purpose	Notes
<a href="#">callNumberKey.exe</a>	Converts call numbers into a normalized sort form.	Used by InfoStation.
<a href="#">CircReport.exe</a>	Generates information for Circulation reports and notices.	Can be used to develop custom reports or to troubleshoot InfoStation reports.
<a href="#">CreateCustomAdHocView.sh</a>	Creates a database view for the specified tag definitions.	This script is installed with the Ad Hoc Reporting System.
<a href="#">DecodeBibLeaderData.sh</a>	Displays decoded Leader data for a specified Bib ID and Leader value.	This script is installed with the Ad Hoc Reporting System.
<a href="#">get_bibs.pl</a>	Extracts bibliographic data from records in your database.	This script is installed with InfoStation.
<a href="#">getURLs.sh</a>	Reports on the records in your database that include URLs.	Can be used in conjunction with InfoStation's URL Checker report to check for broken URLs.
<a href="#">itivaReport.sh</a>	Runs <b>CircReport.exe</b> and generates a file that can be submitted to the i-tiva Message system.	This script takes three input parameters.

Filename	Purpose	Notes
<a href="#">RetrieveAuthFixedTagData.sh</a>	Returns the specified fixed field tag information from one or more permanent authority records.	This script is installed with the Ad Hoc Reporting System.
<a href="#">RetrieveAuthLeaderData.sh</a>	Returns the specified Leader values from one or more permanent authority records.	This script is installed with the Ad Hoc Reporting System.
<a href="#">RetrieveAuthVariableTagData.sh</a>	Returns the specified variable field tag information from one or more permanent authority records.	This script is installed with the Ad Hoc Reporting System.
<a href="#">RetrieveBibFixedTagData.sh</a>	Returns the specified fixed field tag information from one or more bibliographic records.	This script is installed with the Ad Hoc Reporting System.
<a href="#">RetrieveBibLeaderData.sh</a>	Returns the specified Leader values from one or more bibliographic records.	This script is installed with the Ad Hoc Reporting System.
<a href="#">RetrieveBibVariableTagData.sh</a>	Returns the specified variable field tag information from one or more bibliographic records.	This script is installed with the Ad Hoc Reporting System.

## **8. Appendix B - Changes in this Guide**

### **8.1 Changes for Version 16.1**

No changes were made.

# Index

---

.brp file · 35  
.itv file · 35

---

## A

Ad Hoc Reporting System  
  accessing data for use with · 41  
address blocks · 8  
AL32UTF8 character set · 55  
authority\_note\_record table  
  about · 41  
Availability Reminder report · 4  
Availability report · 4

---

## B

-bibfile option · 52  
bibliographic\_record table · 51  
  about · 41  
  creating a view for · 48  
  extracting data from · 51  
Bills report · 4  
BLOBs  
  advantages of · 42  
  extracting bibliographic data from · 51  
  extracting data from · 42  
  used in the Virtua database · 41

---

## C

call numbers, converting to sort form · 38  
callNumberKey.exe · 38, 57  
  usage format · 38  
changes in the guide · 59  
character set mapping · 11  
-charset option · 55  
CircReport.exe · 4, 57  
  address block · 8  
  Availability report · 6  
  choosing an output map · 11  
  choosing the report type · 6  
  command options  
    @ · 5  
    report type · 6  
    specifying in a file · 5

t · 6  
command-line options · 31  
database update options · 31  
defining a date format · 14  
error types for inventory control reports · 30  
Expired Bookings report · 6  
Expired Requests report · 6  
input options · 8  
  a · 8  
    --add\_billing\_fee · 23  
      not affected by the -u option · 32  
  allowing multiple update instances · 14  
  --assess\_not\_picked\_up\_fees · 21  
  --availability\_reminder\_days · 27  
  c · 11  
    --checkin\_date · 24  
  check-out types · 11  
  collection codes · 11  
  d · 12  
  D · 14  
  date format · 14  
  --delete\_overdue\_fine · 23  
  --delete\_request\_fees · 21  
  --expired\_categories · 20  
  --extension\_notify\_days · 26  
  for Availability Reminder report · 26  
  for Bills report · 22  
  for Expired Bookings report · 20  
  for Expired Requests report · 20, 21  
  for Inventory Control report · 15  
  for Overdues report · 24  
  for Recalled Bills report · 22  
  for Request Expiration Notification and  
    Extension report · 25  
  for Review Notice report · 26  
  for Unsatisfied Request Cancellation report ·  
    21, 22  
  for Unsatisfied Requests report · 22, 24  
  g · 14  
  h · 13  
  i · 10  
    --include\_bib\_level\_requests · 25  
    --inventory\_accession\_end · 17  
    --inventory\_accession\_start · 17  
    --inventory\_barcode\_file · 15  
    --inventory\_call\_end · 16  
    --inventory\_call\_start · 15  
    --inventory\_collection\_code\_id · 18  
    --inventory\_date · 17  
    --inventory\_error\_tolerance · 17  
    --inventory\_invalid\_shelved\_status · 17  
    --inventory\_location · 16  
    --inventory\_report\_no\_call · 16  
    --inventory\_report\_user\_id · 18

- item classes · 10
- item\_statuses · 22
- l · 9
- L · 10
- m · 11
- n · 9
- notice count · 9
- operator name · 8
- output map · 11
- p · 10
- patron language · 10
- patron types · 10
- report delimiter · 12
- report\_bills\_exceptions · 24
- request\_trapped · 21
- request\_types · 25
- s · 13
- shelf location · 9
- sort order · 13
- suppress\_overdue\_items · 36
- Transition Disputed Items report · 14
- use alternate title heading · 8
- use check-out type names · 13
- w · 8
- x · 11
- Inventory Control report · 6
- i-tiva options · 18
- limit update to address option · 32
- limit update to patrons who opt in · 19
- output options · 27
  - for inventory control reports · 29
  - o · 27
  - specifying a range · 27
  - values · 27
- Overdue Bills report · 6
- Overdues report · 6
- Recalled Bills report · 6
- Recalled Overdues report · 6
- Recalls report · 6
- Request Expiration Notification and Extension report · 6
- Review notice · 6
- running to generate an i-tiva input file · 33
- scheduling · 34
- specifying an operation ID · 8
- specifying an output delimiter · 12
- Transition Disputed Items report · 6
- Unsatisfied Request Cancellation report · 6
- Unsatisfied Requests report · 6
- update parameter option · 31
  - does not affect the --add\_billing\_fee option · 32
- usage · 4
- Circulation reports · 4
- CreateCustomAdHocView
  - example · 50
  - sample table generated by · 50
- CreateCustomAdHocView.sh · 49, 57
- creating, views for the bibliographic\_record table · 48
- custom views, creating · 48

---

## D

- database
  - backing up before running a script or executable · 1
  - modified by scripts · 1
- date range · 38
- dbadmin · 3
- decode functions
  - Decode\_BibLdr\_CatForm · 46
  - Decode\_BibLdr\_Coding · 46
  - Decode\_BibLdr\_ControlType · 46
  - Decode\_BibLdr\_EncodeLvl · 46
  - Decode\_BibLdr\_Level · 45
  - Decode\_BibLdr\_LinkRec · 46
  - Decode\_BibLdr\_RecStatus · 44
  - Decode\_BibLdr\_RecType · 45
  - using on non-MARC 21 databases · 46
- Decode\_BibLdr\_CatForm · 46
- Decode\_BibLdr\_Coding · 46
- Decode\_BibLdr\_ControlType · 46
- Decode\_BibLdr\_EncodeLvl · 46
- Decode\_BibLdr\_Level · 45
- Decode\_BibLdr\_LinkRec · 46
- Decode\_BibLdr\_RecStatus · 44
- Decode\_BibLdr\_RecType · 45
- DecodeBibLeaderData.sh · 44, 57
- delimited output
  - choosing in get\_bibs.pl · 54
  - default delimiter · 54
- delimiter
  - default for get\_bibs.pl · 55
  - specifying for get\_bibs.pl · 55
- delimiter option · 55

---

## E

- environment variables · 3
  - ORACLE\_SID · 5
- EXE\_DIR · 3
- executables, list of · 57
- Expired Bookings report · 4
- Expired Requests report · 4
- extracting
  - authority data · 42
  - bibliographic data · 42
  - data from bibliographic records · 51
  - tag-specific record data · 41

---

## G

- get\_bibs.pl · 51
  - choosing
    - an output delimiter · 55
    - tags to display in the output · 52

- the subfields by which you want to sort · 53
- the subfields you want to display · 53
- the tags by which you want to sort · 53
- the tags you want to display · 53
- command line parameters · 52
  - bibfile · 52
  - charset · 55
  - delimiter · 55
  - host · 54
  - ldr 1 · 52
  - output · 54
  - password · 55
  - sid · 54
  - sort · 53
  - tag · 52
  - username · 55
- default delimiter · 55
- displaying
  - a delimited output · 54
  - all occurrences of a tag · 53
  - all subfields of a tag · 53
  - results in plain text · 54
  - results in plain text with subfields and indicators · 54
  - the first occurrence(s) of a tag · 53
  - the last occurrence(s) of a tag · 53
  - the record's Leader · 52
- example of a command · 55
- features of · 51
- finding · 51
- running · 51
- sorting
  - by all occurrence of a tag · 53
  - by all subfields of a tag · 54
  - by the first occurrence(s) of a tag · 53
  - by the last occurrence(s) of a tag · 53
  - results · 53
- specifying
  - database password · 55
  - database username · 55
  - file of bibliographic IDs · 52
  - host of the database · 54
  - Oracle System Identifier · 54
- using · 51
- writing results to a file · 55
- getURLs.sh · 39, 57

---

## H

- host option · 54
- host, specifying in get\_bibs.pl · 54

---

## I

- indicators, displaying in the output of get\_bibs.pl · 54
- InfoStation

- Circulation reports · 4
  - get\_bibs.pl and · 51
  - running get\_bibs.pl from the cgi-bin directory · 51
- installing, get\_bibs.pl · 51
- inventory control error types, CircReport.exe and · 29
- Inventory Control report · 4
- inventory control, CircReport.exe and · 15
- i-tiva input file, generating · 33
- i-tiva Message system
  - overview · 33
  - scheduling · 34
- i-tiva options for CircReport.exe · 18
- itivaReport.sh · 18, 33, 57
  - example syntax · 36
  - running · 34

---

## L

- ldr 1 option · 52
- Leader, decoding values · 44

---

## M

- MARC record data, extracting · 42
- MARC records, storage of · 41

---

## N

- NLS\_LANG · 3

---

## O

- Oracle System Identifier
  - specifying in get\_bibs.pl · 54
- ORACLE\_SID · 3
- output map, CircReport.exe and · 11
- output option · 54
- Overdues report · 4

---

## P

- password option · 55
- password, specifying for get\_bibs.pl · 55
- plain text, displaying results from get\_bibs.pl in · 54

---

## R

- Recalled Bills report · 4
- Recalled Overdues report · 4
- Recalls report · 4

report types, CircReport.exe and · 6  
 reports · 4  
 REPORTS\_HOME  
   setting · 52  
 Request Expiration Notification and Extension report · 4  
 RetrieveAuthFixedTagData.sh · 47, 58  
 RetrieveAuthLeaderData.sh · 48, 58  
 RetrieveAuthVariableTagData.sh · 47, 58  
 RetrieveBibFixedTagData.sh · 43, 58  
 RetrieveBibLeaderData.sh · 44, 58  
 RetrieveBibVariableTagData.sh · 42, 58  
 Review Notice report · 4  
 running  
   callNumberKey.exe · 38  
   CircReport.exe · 4  
   get\_bibs.pl · 51  
   getURLs.sh · 39

---

## S

scripts, list of · 57  
 -sid option · 54  
 -sort option · 53  
 sorting  
   call numbers · 38  
   results from get\_bibs.pl · 53  
 storage of records in the database · 41  
 subfields, displaying in the output of get\_bibs.pl · 54

---

## T

-tag option · 52  
 Transition Disputed Items report · 4

---

## U

Unsatisfied Request Cancellation report · 4  
 UpdateOverdueFines.exe · 23  
 URLs, reporting on · 39  
 -username option · 55  
 username, specifying for get\_bibs.pl · 55

---

## V

views  
   created by the CreateCustomAdHocView.sh · 50  
   creating, for the bibliographic\_record table · 48  
 Virtua database  
   AL32UTF8 character set · 55  
   bibliographic\_record table · 51  
   data types used · 41  
   extracting record data from · 41, 42  
   record storage and · 41  
   specifying  
     character set in get\_bibs.pl · 55  
     Oracle System Identifier in get\_bibs.pl · 54  
     password in get\_bibs.pl · 55  
     username in get\_bibs.pl · 55  
 VIRTUA\_PASSWORD · 3  
 VIRTUA\_USER · 3