

User's Guide



Integrated Library System

Virtua Record Loading

VIRTUA ILS – INTEGRATED LIBRARY SYSTEM

Record Loading User's Guide

Version 15.0

May 2015



Copyright © 1998-2015 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 540.557.1200
Fax 540.557.1210
E-mail: info@iii.com

Table of Contents

TABLE OF FIGURES	V
1. INTRODUCTION	1
1.1 OVERVIEW OF VIRTUA'S RECORD-LOADING FEATURES	1
1.2 USING THIS GUIDE	2
2. GENERAL ASPECTS OF RECORD LOADING	4
2.1 UNDERSTANDING PRIORITY AND SCOPE OF LOAD OPTIONS	4
2.1.1 PRIORITY OF LOAD OPTIONS	5
2.1.2 SCOPE OF LOAD OPTIONS	6
2.2 ABOUT THE 999 LOADOPT TAG	6
2.3 ABOUT LOADING ITEM RECORDS	7
2.3.1 ABOUT THE ROLE OF THE 949 VIRTUAITEM TAG	8
2.3.2 WHEN UPPER CASE OR NUMERIC SUBFIELDS ARE NOT SUPPORTED	9
2.3.3 PARAMETERS USED FOR ITEM LOADING	9
2.4 ABOUT DEFAULT LOAD OPTION VALUES	10
2.4.1 ABOUT DEFAULT MAPPING TO UNICODE®	10
2.4.2 DEFAULT CREATION OF PATRON 008 TAG VALUES	11
2.5 WORKING WITH LOAD OPTIONS FILES	11
2.5.1 ABOUT THE PRIORITY OF A LOAD OPTIONS FILE	11
2.5.2 CREATING A GENERAL LOAD OPTIONS FILE	12
2.5.3 CREATING A DUPLICATE TEST/MERGE LOAD OPTIONS FILE	13
2.5.4 CALLING A LOAD OPTIONS FILE	13
3. WORKING WITH VLOAD.EXE	15
3.1 RUNNING VLOAD.EXE	15
3.2 LOG FILES AND ERROR MESSAGES GENERATED BY VLOAD.EXE	16
3.2.1 VLOAD LOG FILE AND PROCESSING SUMMARIES	17
3.2.2 VLOAD STATISTICS LOG	19
3.2.3 VLOAD ERROR LOG	20
3.2.4 TRANSLATING DIAGNOSTIC CODE	20
3.3 OPTIONS AVAILABLE FOR VLOAD.EXE	24
3.3.1 USING A LOAD OPTIONS FILE (-@)	25
3.3.2 DETERMINING HOW VLOAD.EXE PROCESSES RECORDS IN THE INPUT FILE (-C)	26
3.3.3 DETERMINING WHETHER RECORD DATES ARE COMPARED (-C)	27
3.3.4 DETERMINING WHETHER THE 999 LOADOPT TAG IS USED (-9)	28
3.3.5 SPECIFYING THE STATE INTO WHICH RECORDS ARE LOADED (-B)	28

3.3.6 SPECIFYING THE CHARACTER SET OF INCOMING RECORDS (-D)	29
3.3.7 DETERMINING WHETHER VIRTUA CHECKS THE RECORD STATUS FLAG (-O)	30
3.3.8 CHOOSING A RANGE OF RECORDS IN THE INPUT FILE TO PROCESS (-S)	30
3.3.9 DISPLAYING HELP (--HELP)	31
3.3.10 DETERMINING THE CRITERIA FOR A DUPLICATE RECORD (-X)	32
3.3.11 SPECIFYING RULES FOR MERGING DUPLICATE RECORDS (-R)	33
3.3.12 SPECIFYING AN ACTION FOR LOADING NON-DUPLICATE RECORDS (-S)	33
3.3.13 MERGING EXISTING BIBLIOGRAPHIC RECORDS (-M)	34
3.3.14 DETERMINING THE DATE TO USE AS THE ENTERED DATE (-L)	35
3.3.15 SPECIFYING THAT VIRTUA DELETES THE RECORDS INDICATED IN THE FILE OF IDS (-X)	35
3.3.16 DEFINING A LIBRARY SYMBOL (-U)	36
3.3.17 DETERMINING WHETHER RECORDS ARE TO BE SPLIT INTO FRBR RECORDS (-F)	36
3.3.18 ADDING A SPECIAL 035 TAG (-3)	37
3.3.19 CREATING ITEM TAGS FOR HEBREW UNION COLLEGE (-H)	39
3.3.20 SPECIFYING A VALIDATION RULE (-V)	39
3.3.21 SPECIFYING A FILENAME (-W)	39
3.3.22 BLOCKING ITEM CREATION BASED ON THE 949 TAG (-K)	40
3.3.23 REPLACING SUBFIELD CODES IN THE 949 VIRTUAITEM TAG (-P)	40
3.3.24 REPLACING SUBFIELD CODES IN THE 949 VIRTUAITEMCALLNUMBER TAG (-Q)	41
3.3.25 REQUIRING THE VIRTUAITEM LABEL (-L)	42
3.3.26 BLOCKING RECORDS WITH A 949 SUBFIELD \$Z TAG (-Z)	42
3.3.27 SPECIFYING A SHELVING LOCATION FOR NEW ITEM RECORDS (-B)	43
3.3.28 SPECIFYING A DEFAULT ITEM CLASS (-D)	43
3.3.29 DETERMINING WHETHER BLIND REFERENCES ARE LOADED (-G)	44
3.3.30 SPECIFYING THE HEADING TYPE FOR AUTHORITY RECORDS (-H)	44
3.3.31 PRESERVING LOCAL TAGS (-T)	45
3.3.32 REMOVING THE 003 TAG RESTRICTION FOR 010 TAG CREATION (-R)	45
3.3.33 BLOCKING DELETION OF RECORDS WITH 1XX/5XX MATCHES (-N)	46
3.3.34 SPECIFYING THE DATE PATTERN OF THE 042 TAG (-O)	46
3.3.35 CHECKING THE PATRON RECORD DATE (-I)	47
3.3.36 DEFINING A DEFAULT EXPIRATION DATE (-J)	47
3.3.37 DEFINING A DEFAULT DELETE DATE (-K)	48
3.3.38 ADDING A DEFAULT PATRON PASSWORD (-P)	48
3.3.39 DETERMINING WHETHER 042 VALUES ARE RETAINED (-T)	49
3.3.40 MAINTAINING ENTITY LINKS DURING A FRBR LOAD (-U)	49
3.3.41 PROVIDING A VENDOR ID FOR MARC ORDER PROCESSING (-V)	50
3.3.42 DETERMINING THE TYPE OF MARC ORDER PROCESSING (-W)	50
3.3.43 MERGING AUTHORITY RECORDS WHEN A 4XX TAG CONFLICT OCCURS (-E)	51
3.3.44 DETERMINING WHETHER DATA IN TAG 880 IS MOVED (-8)	51
3.3.45 SPECIFYING THE MARC FORMAT OF INCOMING RECORDS (-Y)	52
3.4 OTHER VLOAD.EXE OPTIONS AVAILABLE FROM THE COMMAND LINE ONLY	52
3.4.1 OUTPUTTING RECORDS THAT FAILED TO DELETE TO A FILE (--FAILED-DELETE-RECORD-FILE)	52
3.4.2 DETERMINING WHETHER TO CREATE A 999 VTLSSORT TAG (--CREATE-SORT-TAG)	53
3.4.3 READING INPUT FROM A FILE (--INPUT-FILE)	53
3.4.4 ADDING AN XML FILE TO CONVERT TO MARC (--XML-ELEMENT-NAME)	53
3.4.5 DEFINING AN XML FILE FORMAT TO CONVERT TO MARC (--XML-FORMAT)	54
3.4.6 INDEXING AAPs ONLY (--AAP-ONLY-INDEXING)	55

3.4.7 INDEXING USER-DEFINED HEADINGS ONLY (--USERHEADING-ONLY-INDEXING)	55
3.4.8 INDEXING SUBJECT HEADINGS ONLY (--SUBJECTHEADING-ONLY-INDEXING)	56
3.4.9 DISABLING FRBR SUBJECT INDEXING (--NOFRBRSUBJECTINDEXING)	56
3.4.10 DISABLING INDEXING (--INDEXING-DISABLED)	56
3.4.11 DISABLING LOCATION INDEXING (--NO-LOCATION-FILTER)	57
3.4.12 DISABLING PATRON RECORD INDEXING (--NO-PATRON-INDEXING)	57
3.4.13 DISABLING HOLDINGS RECORD INDEXING (--NO-HOLDINGS-INDEXING)	58
3.4.14 MASKING INCOMING RECORDS (--MASK-RECORDS)	58
3.4.15 KEEPING DATE AND TIME OF INCOMING RECORDS (--USE-INCOMING-039-TAG)	59
3.4.16 ALWAYS CREATING THE 035 TAG (--ALWAYS-AUTO-GENERATE-035-TAG)	59
3.4.17 PREVENTING 4XX CONFLICTS IN AUTHORITY RECORDS (--PREVENT-AUTHORITY-4XX-CONFLICTS)	59
3.5 ABOUT DUPLICATE TESTS AND MERGE LOAD OPTIONS	60
3.5.1 ABOUT THE FORMAT OF THE DUPLICATE TEST OPTION	61
3.5.2 ABOUT MERGE OPTIONS	64
3.5.3 EXAMPLES OF DUPLICATE TESTS AND MERGE OPTIONS	69
4. SETTING LOAD OPTIONS IN THE VIRTUA CLIENT	71
4.1 ACCESSING THE LOAD OPTIONS WINDOW	71
4.2 WORKING WITH LOAD OPTIONS ON THE LOAD OPTIONS WINDOW	73
4.2.1 WORKING WITH THE GENERAL TAB	73
4.2.2 WORKING WITH THE BIBLIOGRAPHIC (AND COMMUNITY INFO) TAB	76
4.2.3 WORKING WITH THE AUTHORITY TAB	79
4.2.4 WORKING WITH THE HOLDINGS TAB	82
4.2.5 WORKING WITH THE PATRON TAB	85
4.3 WORKING WITH DUPLICATE TESTS AND MERGE LOAD OPTIONS IN THE CLIENT	88
4.3.1 SPECIFYING ONE OR MORE DUPLICATE TESTS	89
4.3.2 SPECIFYING THE MERGES OPTION	92
4.3.3 SPECIFYING THE NON-DUPLICATE ACTION	95
4.3.4 A NOTE ABOUT DUPLICATE TESTS FOR PATRON RECORDS	95
5. BATCH LOADING MARC FORMAT RECORDS IN THE VIRTUA CLIENT	97
5.1 WORKING WITH THE IMPORT RECORDS TOOL	97
5.1.1 ACCESSING THE IMPORT RECORDS TOOL	98
5.1.2 ASPECTS OF THE IMPORT RECORDS WINDOW	99
5.1.3 CONFIGURING THE IMPORT RECORDS TOOL	100
5.1.4 RUNNING THE IMPORT RECORDS TOOL	102
5.1.5 AVAILABILITY OF LOG FILES	103
5.1.6 SUMMARY LOG FILE	104
5.1.7 ERROR LOG FILE	104
5.2 WORKING WITH THE IMPORT RECORDS TO PURCHASE REQUEST LIST TOOL	106
5.3 WORKING WITH THE IMPORT MARC ORDERS/REQUESTS TOOL	109
6. APPENDIX A - AVAILABLE LOAD OPTIONS	113

7. APPENDIX B - SPECIFICATIONS FOR THE 949 VIRTUAITEM TAG	128
8. APPENDIX C - CHANGES IN THIS GUIDE	136
8.1 CHANGES FOR VERSION 15.0	136
INDEX	137

Table of Figures

<i>Figure 3-1. Export Window - Language Editor</i>	21
<i>Figure 3-2. Language Table - Language Editor</i>	23
<i>Figure 4-1. Load Options Window</i>	72
<i>Figure 4-2. Load Options Window - General Tab</i>	74
<i>Figure 4-3. Load Options Window - Bibliographic Tab</i>	77
<i>Figure 4-4. Load Options Window - Authority Tab</i>	80
<i>Figure 4-5. Load Options Window - Holdings Tab</i>	83
<i>Figure 4-6. Load Options Window - Patron Tab</i>	86
<i>Figure 4-7. Add Duplicate Test Window</i>	89
<i>Figure 4-8. Load Options Window - Displaying a Duplicate Test</i>	91
<i>Figure 4-9. Add Merge Window</i>	92
<i>Figure 4-10. Add Rule Window</i>	93
<i>Figure 4-11. Add Merge Window - Displaying a Merge Rule</i>	94
<i>Figure 4-12. Specify Default Rule Window</i>	94
<i>Figure 5-1. Import Records Window</i>	98
<i>Figure 5-2. Import Configuration Window</i>	100
<i>Figure 5-3. Import Records Tool in Action</i>	102
<i>Figure 5-4. Reset Warning - Import Records Function</i>	103
<i>Figure 5-5. Import To Purchase Requests Window</i>	107
<i>Figure 5-6. Import MARC Orders/Requests Window</i>	110

1. Introduction

This user's guide contains details about the record-loading features of the Virtua ILS – Integrated Library System. This guide describes how you can load records into the Virtua database using both the server program, **vload.exe**, and the Virtua client.

Note: From this point on, we will refer to the Virtua ILS – Integrated Library System as simply Virtua or the Virtua system in this guide.

This chapter covers the following topics:

- ⇒ [Overview of Virtua's Record-loading Features](#)
- ⇒ [Using This Guide](#)

1.1 Overview of Virtua's Record-loading Features

In Virtua, you can load records into the database using either the server program, **vload.exe**, or the Virtua client. With either feature, you can specify a variety of options that tell Virtua how to handle incoming records.

- When you use **vload.exe**, you specify these options along with the **vload.exe** command within the UNIX operating system.
- When you use the client, you specify these options on the Load Options window (*Figure 4-1*).

Both the server- and client-side load programs provide options for . . .

- Inserting records into the database. You can insert bibliographic/community information, authority, holdings, and patron records. Additionally, you have the option of creating new item records when loading parent bibliographic or holdings record.
- Replacing records (all types except item records).
- Merging records (bibliographic, authority, and patron records only).
- Deleting records (all types except item records).

Furthermore, both the server and client programs automatically perform keyword indexing of bibliographic records at the time of processing. Both programs do this by reading the Keyword Indexing Configuration Editor file, which is saved to the database.

In general, **vload.exe** mirrors the functionality that is offered via the Virtua client. They both provide general validation of MARC records although **vload.exe** does not use Vericat as the client does. On the one hand, the client offers the advantage of a graphical user interface (GUI). On the other hand, **vload.exe** offers more load options than the client, and the program can be run as an unattended batch job.

1.2 Using This Guide

This document is meant to be a basic reference for the record-loading options that you can use with **vload.exe** and the Virtua client. This guide discusses the features of the load programs, and compares and contrasts their various aspects. In addition, the guide addresses the auxiliary 949 Item Load feature, which works in conjunction with the record-loading programs to create item records from parent bibliographic and holdings records.

You can use the list below and the [Table of Contents](#) to locate specific information in this guide. For your easy reference, this guide also offers a [Table of Figures](#), an [Index](#), and a reference section consisting of appendices. In addition, each chapter in this guide, where appropriate, begins with a list of hyperlinked topics that are discussed in the chapter.

For . . .	See . . .
Details on features of record-loading that are common to both the client and server programs	Chapter 2
Details on vload.exe	Chapter 3
Details on the load options available in the Virtua client	Chapter 4
Details on batch loading records in the Virtua client	Chapter 5
A complete list of available load options	Appendix A
Specifications for the 949 VIRTUAITEM tag	Appendix B
A description of the changes that were made to this user's guide since the last update	Appendix C

The organization of this guide does not lend itself to a linear discussion of load options. There is much that is unique to individual programs and much that overlaps. We suggest prudent use of the table of contents, table of figures, index, and appendices to find the information you need.

Note: Do NOT use the record-loading options in this guide to *create* your Virtua database. For details on setting up your database, contact your Innovative Customer Support representative.

2. General Aspects of Record Loading

To provide maximum flexibility in record loading and processing, Virtua offers two record-loading facilities: **vload.exe** and the **Virtua client**. The options provided by both are essentially equivalent. Your use of either the server program or the client interface is primarily a matter of preference rather than of functionality.

This chapter discusses the aspects of record loading in Virtua that are *common* to both **vload.exe** and the Virtua client.

- For details on **vload.exe**, see the chapter "[Working with vload.exe](#)."
- For details on the client's record-loading features, see the chapter "[Setting Load Options in the Virtua Client](#)."

We recommend that you read this chapter before reading either of the two chapters listed above. This chapter covers the following topics:

- ⇒ [Understanding Priority and Scope of Load Options](#)
- ⇒ [About the 999 LOADOPT Tag](#)
- ⇒ [About Loading Item Records](#)
- ⇒ [About Default Load Option Values](#)
- ⇒ [Working with Load Options Files](#)

2.1 Understanding Priority and Scope of Load Options

Load options tell Virtua how to process records that are loaded into the database. Because you can specify load options in several ways, during record processing, Virtua prioritizes load options using a four-tiered system. This system of priority is internal and cannot be altered.

The layers of load options are . . .

- Load options stored in the 999 *LOADOPT* tag of a record. The 999 *LOADOPT* tag contains the word *LOADOPT* in subfield \$a. Tag 999 is repeatable, so multiple 999 tags can exist in one record.

Note: When you load a record via the client, Virtua automatically creates a 999 *LOADOPT* tag, which stores as subfield values the load options that you used to load the record. You cannot create the tag manually using the MARC Editor.

- Load options supplied at the command line, in the client, or in a load options file saved to the server.
- Default load options that are set in the Virtua Profiler and stored in the Virtua database.
- Default load options that are stored in the Virtua code.

2.1.1 Priority of Load Options

When all four layers of options are available, Virtua prioritizes them, applying load options in the following order:

1. Load options in the 999 *LOADOPT* tag of an incoming record. For more information, see the section "[About the 999 *LOADOPT* Tag.](#)"
2. Load options from . . .
 - ◆ The command line when you use **vload.exe**.
 - OR-
 - ◆ The Load Options window when you use the Virtua client.
 - OR-
 - ◆ A load options file (can be used by **vload.exe** or the Virtua client).
3. Default load options that are defined in the Virtua Profiler and stored in the database (used by **vload.exe** only).
4. Default load options in the Virtua code.

If two layers provide the same load option but with different values, the value for the option with the higher priority overrides the value for the option with the lower priority. If a layer with the higher priority lacks an option supplied by a layer with a lower priority, the option with the lower priority is used.

2.1.2 Scope of Load Options

Below is a summary of the scope of load options available with the Virtua system:

- **999 LOADOPT tag options** - Apply *only* to the individual record that contains the tag. Records will contain the 999 LOADOPT tag only if . . .
 - ♦ Load options were set in the Virtua client.
 - AND -
 - ♦ The records were imported directly into the Virtua database OR were edited in the MARC Editor and then saved to the Virtua database.

Note: For more information, see the following section, "[About the 999 LOADOPT Tag](#)."

- **Command-line load options** - Apply to ALL records in the input file.
- **Load options specified in a file saved to the server** - Apply to ALL records in the input file.
- **Default load options** - Apply to ALL types of records and will be used as long as no option is duplicated in a layer with a higher priority.

2.2 About the 999 LOADOPT Tag

The 999 LOADOPT tag plays an important role in determining load option priority. Virtua creates this tag automatically when you set load options in the Virtua client and then save records to the database or load records via the Import Records function.

The 999 LOADOPT tag will contain a subfield for each load option that was specified when the record was loaded.

- Each client load option has a corresponding 999 LOADOPT subfield. For example, the Database Action option corresponds to subfield \$c of the 999 LOADOPT tag. (The subfields, incidentally, match the command-line options that you specify for **vload.exe**.)
- The value you select for the load option in the client will become the subfield value in the 999 LOADOPT tag.

For a list of load options and their corresponding subfield values, see the appendix "[Available Load Options](#)" in this guide.

Hint: The value stored in a given subfield will generally match the value that you specify for **vload.exe** rather than the value that you select in the client. For example, if you set the Database Action option to *Load Item Bib Parent* in the client, subfield \$c of the 999 LOADOPT tag will contain a **10** (ten) instead of the text *Load Item Bib Parent*. For more information, see the appendix "[Available Load Options](#)" in this guide.

When you load records using **vload.exe** or the client, Virtua *first* reads the load options that are stored in the 999 LOADOPT tag of the incoming records. If Virtua does NOT find a 999 LOADOPT tag, it uses the command-line options, the client load options, and/or the default load options as appropriate.

For a list of subfields in the 999 LOADOPT tag and a description of which load options they store, see the appendix "[Available Load Options](#)" in this guide.

Note: Subfield \$a of the 999 LOADOPT tag always contains the text *LOADOPT* in all capital letters. Starting with subfield \$b, each subfield corresponds to the load option with the same letter representation.

2.3 About Loading Item Records

When you load bibliographic or holdings records into the database, Virtua reads from the input file and creates item records based on information stored in the 949 VIRTUAITEM tag(s) of those records. If a bibliographic or holdings record contains multiple 949 VIRTUAITEM tags, Virtua will load multiple items per parent record.

Important: If you load bibliographic records and specify a record state for the incoming records as well as item creation, Virtua will place the incoming records in the special state and will NOT create items.

For bibliographic and holdings records, you can set other options related to item loading. For example . . .

- You can specify that item creation should be blocked under certain conditions.
- You can define default item class and shelf location values that should be used when those values are not available in the 949 VIRTUAITEM tag.

When loading item records, Virtua processes only those records where the value in the Type of Record position of the Leader indicates a holdings or bibliographic record. The program skips all other records. During any one run of the program, the system processes only ONE type of record, either bibliographic or holdings, which is

determined by the type of the first record selected for processing. If the program finds duplicate barcodes, it generates an error message in the log file and does not process the records.

2.3.1 About the Role of the 949 VIRTUAITEM Tag

To create items, Virtua reads the information stored in the 949 VIRTUAITEM tag(s) of the incoming bibliographic or holdings record. The subfields defined in the tag(s) determine the processing of the record. Information from the 949 tags, as well as location and copy number information from 852 tags in the holdings record, is used to create item records. Each tag contains information for exactly ONE item record. You can have as many 949 VIRTUAITEM tags in one record as you want.

An item record can be created only from a parent record that already exists in the Virtua database, even if that parent record exists only moments before associated items are loaded.

- Each item created from a 949 item tag in a bibliographic record is linked to a bibliographic record in the Virtua database.
- Each item created from a 949 item tag in a holdings record is linked to a holdings record in the Virtua database.

For details on the subfields that you can define for the 949 VIRTUAITEM tag and information on the values defined by default, see the appendix "[Specifications for the 949 VIRTUAITEM Tag](#)" in this guide.

As a general rule, the information in the 949 VIRTUAITEM tag is used only for the creation of *new* item records.

An exception to this rule occurs if the 949 tag subfield \$d (EPN Number) of an incoming bibliographic record matches the EPN Number of an existing item record. If this occurs, Virtua updates the existing item record by replacing certain fields with data from the 949 tag. [The EPN is the Bib ID of the associated record in the French Union Catalog (SUDOCs) shared by all the universities in France.]

When a bibliographic record containing a 949 tag subfield \$d (EPN Number) is loaded, Virtua checks for the existence of an item record with a matching EPN Number. If one is found, Virtua updates the item record as follows:

- The owning location is replaced by the 949 tag subfield \$D value.
- The call number is replaced by the 949 tag subfield \$a value.
- The last use date is replaced by the 949 tag subfield \$2 value.

- The public notes field is populated with the 949 tag subfield \$p value *if* the value had been empty.
- The units field is populated with the 949 tag subfield \$9 value *if* the value had been empty.

If the 949 tag subfield \$d (EPN Number) of the incoming bibliographic record does not match the EPN Number of any existing item record, a new item record is created as usual.

2.3.2 When Upper Case or Numeric Subfields Are Not Supported

The subfields in the 949 VIRTUAITEM tag are defined by lowercase, uppercase, and/or numeric characters. Some systems, such as OCLC, do not support the uppercase and numeric subfield values used in the 949 VIRTUAITEM tag. For these systems, Virtua provides a command-line load option whereby bibliographic or holdings records containing 949 subfields defined in lowercase can be loaded into the Virtua database and reprocessed. The reprocessing maps the lowercase subfield values to the uppercase or numeric subfield values required by **vload.exe**. For details, see the chapter "[Working with vload.exe](#)" in this guide.

2.3.3 Parameters Used for Item Loading

You will want to set all of your Virtua Profiler parameters before you begin loading records. For a list of available parameters and for information on the recommended order for setting them, see the *Virtua Profiler Getting Started Primer*.

Virtua uses the following parameters for loading item records:

- The Item-level Call Number Flag (Cataloging parameters)
- Item Call Number Increment (Cataloging parameters)
- Item Class Definitions (Circulation parameters)
- Location Names (Global parameters)
- Barcode Type (Cataloging parameters)

By setting these parameters, you can provide valid values in the 949 VIRTUAITEM tag for . . .

- **Item-level call number(s)** - Specify in subfield \$a and subfield \$b.
- **Location** - Specify in subfield \$D.

- **Item class** - Specify in subfield \$X.
- **Item barcode** - Specify in subfield \$6.

2.4 About Default Load Option Values

When **vload.exe** or the client finds no load options that it can read, Virtua will use default load options.

- If **vload.exe** finds no load options stored in the 999 LOADOPT tags or supplied at the command line, it will use the default value of each load option that applies to the record type being loaded.
- If the client finds that no load options are set in the Load Options window, it too will use the default value of each load option that applies to the record type being loaded.

For example, the default value for the Date Entered load option is to assign the load date as the entered date. This means that if you do not specify this load option in the client or at the command line, records will be loaded using the load date as the entered date.

For a complete list of the default load options, see the appendix "[Available Load Options](#)" in this guide.

2.4.1 About Default Mapping to Unicode®

By default, the system converts the data in each loaded record to the UTF-8 character set, which is an 8-bit transformation of the 16-bit Unicode Standard and is the internal character set of all Virtua databases.

The default character set load option is “no mapping.” Because the system will automatically use the no mapping option, you do not have to specify it *if you are loading records that are already in the UTF-8 character set.*

If your records are NOT in UTF-8, you need to specify the appropriate character set via the Character Set option in a Load Options file or the Character Set setting on the [General tab](#) of the Load Options window for the character set to be mapped correctly. If you run **vload.exe** with a character set option that is not supported by Virtua or that is not supported by the mappings available in the database, the system will display a diagnostic message describing the error.

2.4.2 Default Creation of Patron 008 Tag Values

During the loading of 2709 patron MARC records, if **vload.exe** encounters an incoming record with no 008 tag or an incomplete one, it will automatically create the tag with default values as the record is loaded. The system will do this regardless of whether there is a matching patron record already in the database that contains tag 008.

The default 008 values generated by the system are as follows:

<u>Position</u>	<u>Value</u>
00-07	Load date
08	r (read by another library)
09	a (can be used)
10	0 (use field 270)
11	0 (not a minor)
12	0 (valid)
13	0 (recalls can be placed)
14	0 (responsible person first)
15	0 (history is not maintained)
16	0 (items not allowed to be checked out again)
17	0 (name is not displayed)

2.5 Working with Load Options Files

Setting general load options and merge load options each time that you want to load a file of records can be time consuming and tedious. To make the job easier, you can load a file of load options that has been saved to the server.

In this way, if you plan to use one or more sets of options repeatedly, you do not need to reformulate the options each time you load a batch of records. Instead, you can set up different sets of load and/or merge options, save them to different files on the server, and then *call* the desired file from the command line or in the client to use those predefined options.

2.5.1 About the Priority of a Load Options File

Load options specified in a file have the same priority as options entered at the command line or defined in the Virtua client. If your load options file contains options that conflict with or match options that you specify at the command line or in the client . . .

- The client always uses the option selected on the Load Options window.
- **vload.exe** always uses whatever option was called LAST at the command line. For example, if you call the load options file at the *end* of the command and the load options file contains an option that you also called at the command line, Virtua will use the value specified in the file over the value specified on the command line.

Options specified in the record itself (i.e., in the 999 LOADOPT tag) take precedence over command-line options, client options, and options specified in the load options file.

Hint: If you want Virtua to *ignore* the 999 LOADOPT tag, you can set the Use 999 option to No.

2.5.2 Creating a General Load Options File

To create a load options file,

1. Log in to your server as the **dbadmin** user.
2. Open a new file in a text editor.
3. Type any string of options that you want to store. You can type the strings horizontally or vertically. For example, type:

```
-c10 -d1 -k1 -l1 -L0 -x001{UseNew} -x020{UseNew} -rUseNew{/I}
```

-OR-

```
-c10
-d1
-k1
-l1
-L0
-x001{UseNew}
-x020{UseNew}
-rUseNew{/I}
```

Note: Whether you plan to call the load options file from the client or from **vload.exe**, the options in the file must use the command-line format used by **vload.exe**. For a complete list of available load options, see the section "[Options Available for vload.exe](#)" or "[Appendix A - Available Load Options](#)" in this guide.

4. Save the file.

2.5.3 Creating a Duplicate Test/Merge Load Options File

Setting up tests to detect duplicate records during a load, specifying a database action, and developing options for merging duplicate records can be quite complicated. The section "[About Duplicate Tests and Merge Load Options](#)" in this guide describes how to manually create these tests. As you will see, the process involves several parts, such as determining the duplicate test tags and the corresponding database action and—in the case of merges—a merging string name, tag test(s), and merge rules.

An easier method is to use the program **DuplicateOptions.exe**, which facilitates the process of creating duplicate test and merge load options by guiding you step by step in creating a file. You can take this file, edit it in a text editor, and then call it from **vload.exe** or when loading records in the client.

To run **DuplicateOptions.exe**,

1. Log in to your server as **dbadmin**.
2. At the prompt, type: **DuplicateOptions.exe**
3. Press Enter.

The program walks you through the steps required to create a duplicate test and merge options file. Answer the questions as needed to create a file with the desired options. When the program is finished running, it will ask you to specify a filename for the file.

4. Type a filename. You will use this filename to call the duplicate test and merge options when running **vload.exe** or in the Virtua client.
5. Press Enter.

Virtua saves the file.

2.5.4 Calling a Load Options File

Once you save the load options file to the server, you can call it from the command line when using **vload.exe** or from the Load Options window in the Virtua client.

- At the command line, you must call the file with the load option **-@**. For details, see the section "[Using a Load Options File \(-@\)](#)" in this guide.
- In the client, you can simply type the filename in the field labeled Load Options File on the Load Options window. For details, see the section "[Working with Load Options on the Load Options Window](#)" in this guide.

Note: You can store the file in any location on your server.

- If you want to be able to call the file from the client by specifying only the filename, save the file to the same directory from which you run **psdriver.exe**.
- If you save the file to any other location, you will need to call the file using the full path and filename.

3. Working with vload.exe

The program **vload.exe** loads, modifies, or deletes records in the database based on an input file and the options that you specify. Depending on the log level for your session, information about each action will be directed to a log file. This chapter provides details on working with **vload.exe**, including information about its options and its output.

Note: This chapter is intended for system administrators who are comfortable working with programs on the server and are familiar with the basics of the UNIX operating system.

This chapter covers the following topics:

- ⇒ [Running vload.exe](#)
- ⇒ [Log files and Error Messages Generated by vload.exe](#)
- ⇒ [Options Available for vload.exe](#)
- ⇒ [About Duplicate Tests and Merge Load Options](#)

3.1 Running vload.exe

This section provides the basic command format for running **vload.exe**. The program offers many options that let you specify how records should be processed. You should be familiar with these options before running **vload.exe**. For details, see the section "[Options Available for vload.exe](#)" in this user's guide.

Important: Before you run **vload.exe**, check the values of the EXE_DIR and ORACLE_SID environment variables. These environment variables determine which version of **vload.exe** will be used and which database will be modified. For details, see the *Virtua System Management Reference Guide*.

To run **vload.exe**,

1. Log in to you server as the **dbadmin** user.
2. At the prompt, type:

```
vload.exe [command options] < [input file]
```

Where . . .

- **[command options]** are the parameters that you can use to specify which actions **vload.exe** will perform.
- **[input file]** is the file of records or IDs that will be used in the processing. For example, this file may contain the record that Virtua will load or the IDs of the records that will be deleted from the database.

3. Press Enter.

The program runs. After **vload.exe** finishes processing, it displays a summary of what was done, including all error messages.

Note for Systems that Use Archive Logging: Running **vload.exe** causes many transactions to be executed in the database. This means that a great deal of information will be written to the archive logs. If the partition on which the archive logs are stored runs out of space, your system will stop working. If this situation occurs, you will need to allocate more space to your archive logs. If allocating more space does not solve the problem, contact your Innovative Customer Support representative.

3.2 Log Files and Error Messages Generated by vload.exe

While executing, **vload.exe** generates processing summaries, error data, warnings, and other diagnostic entries for questionable records of any type. In addition, **vload.exe** creates Error state records.

Currently, **vload.exe** supplies diagnostic and error information via the following log files:

- **vload_<time+date>.log** (e.g., vload_hh_mm_ss_DD_MMM_YYYY.log)
- **vload_statistics_<date+time>.log[v]** (e.g., vload_statistics_YYYY/MM/DD 24hh/mm.log)
- **vload_errors_<date+time>.err[v]** (e.g., vload_errors_YYYY/MM/DD 24hh/mm.log)

Unless you specify otherwise, these log files will reside in the same directory from which you ran the load program. Alternatively, you can set the environment variable

VTLS_TEMP to specify where the program writes the log file. For details on environment variables, see the *Virtua System Management Reference Guide*.

The type of logging that **vload.exe** uses is determined by the log level that you specify. You can choose a log level by setting the V_LOG_LEVEL environment variable. For details, see the *Virtua System Management Reference Guide*.

In addition to log files, **vload.exe** creates an Error state record for each record that is processed with errors. If the error occurred during record replacement or deletion, this record is saved to the database. When you retrieve the newly loaded record and access it in the Virtua client via the MARC Editor, the Error state record is retrieved first. For additional information on working with Error state records in the Virtua client, see the *Virtua Cataloging User's Guide*.

3.2.1 Vload Log File and Processing Summaries

The log file **vload_<time+date>** provides two kinds of processing summaries:

- **General Processing Summary**, which the UNIX server displays on the screen as part of the Detailed Processing Summary.
- **Detailed Processing Summary**, which provides diagnostic code entries for errors in individual records.

3.2.1.1 General Processing Summary

The General Processing Summary that **vload.exe** creates provides statistics on the number of records that were . . .

- Read from the input file.
- Properly loaded.
- Loaded but encountered a diagnostic message during processing.
- Not properly loaded, such as records that have a character-mapping problem.

Below is an example of the General Processing Summary and the statistics provided by **vload.exe**:

Vload is processing each record, Wait for a while...

Vload Processing Summary will be shown at the end of processing.
Error records and/or diagnostic entries will be created in the database.
Error information for each error record will be logged into file
errormessageVload.ff

Hint: You can use <Ctrl-C> to stop vload while it is running. Vload should
finish processing the current record and then stop. A summary will
display.

```
-----
|                                     |
|               VLOAD PROCESSING SUMMARY               |
|                                     |
| INPUT:                                                 |
|                                     |
|   Total # of records scanned (read from input file)   =   1 |
|                                     |
|   Total # of idTypes of the records scanned from input =   1 |
|                                     |
|   Detail input information of each idType below:      |
|   IdType 105: # of input records done read & processing =   1 |
|                                     |
| OUTPUT:                                                |
|                                     |
|   Started processing records of input file at record   0 |
|   Ended processing records of input file at record (done) 1 |
|                                     |
|   vload processing reach end of input file flag       =   Yes |
|   vload forced to be stopped status (stopped by user) =   No  |
|                                     |
|   Total # of Skipped Records                          =   0 |
|   Total # of Processed Records                        =   1 |
|   Total # of Processed & Done Well Records            =   0 |
|   Total # of Processed & Done Diagnostics Records     =   0 |
|   Total # of Processed & Not Done Fatal Error Records =   1 |
|   Total # of Processed & Not Done Other Error Records =   0 |
|                                     |
|   Detailed Summary for records of all idTypes         |
|   -----                                             |
|   idType 105: # of records skipped (mapping failed)   =   0 |
|               # of records processed                  =   1. |
|               Detailed Processing Results              |
|               -----                                |
|               # of records processed & done Well      =   0 |
|               # of records processed & done Diag Info =   0 |
|               # of rec processed & not done FatalError =   1 |
|               # of rec processed & not done OtherError =   0 |
|                                     |
|-----
```

vload started at time: 20102011033, ended at : 20102011033

3.2.1.2 Detailed Processing Summary

When **vload.exe** encounters a problem with a record, the program writes both the file number of the record and diagnostic or error data to a flat file in the database directory. The diagnostic or error data that is generated is in code and is called the Detailed Processing Summary.

The following non-error action items may also be reported in the Detailed Processing Summary:

Action = 3 - The specified records were deleted.

Action = 5 - The specified records were merged.

Action = 7 - The specified records were put into Error state (as a result of the **-x** option).

3.2.2 Vload Statistics Log

Besides the Detailed Processing Summary, **vload.exe** produces a statistics log file, which stores additional statistics and diagnostic information received during the load process. The file extension of the log file will be **.log** for a normal load or **.logv** for a validate-only load (for details about validate-only loads, see the section [“Specifying a Validation Rule”](#)).

Format of the File:

Date/time Import Started: <date/time import started in YYYY/MM/DD 24hh/mm format>

Contents of the File:

- Global Statistics
 - ◆ Number of Records Processed:
 - ◆ Number of Records Successfully Saved to/Validated in DB:
 - ◆ Number of Records Not Saved to/Not Valid according to DB:
 - ◆ Number of Saved/Validated Records without Diagnostics:
 - ◆ Number of Records Successfully Saved to/Validated in DB but with Warnings:
- Diagnostics Received

3.2.3 Vload Error Log

The **vload.exe** program also produces an error log file, which stores diagnostic information associated with each MARC record. The file extension of the log file will be **.err** for a normal import or **.errv** for a validate-only load (for details about validate-only loads, see the section [“Specifying a Validation Rule”](#)).

Format of the File:

Date/time Import Started: <date/time import started in YYYY/MM/DD 24hh/mm format>

Contents of the File:

[For each MARC record with diagnostics]:

- ISSN: <ISSN value (022\$a) of the record>
- Record Number: <ordinal number of the record>
- List of all diagnostics, warnings, or errors received for the record in the format: <Validation ID number> <diagnostic message>
- MARC record in human-readable format <separator line>

3.2.4 Translating Diagnostic Code

The **vload.exe** program generates diagnostic and error data in code. To translate the code into a human-readable language, you need to run a utility called **AddDiagnosticText.exe**.

3.2.4.1 Running AddDiagnosticText.exe

To run **AddDiagnosticText.exe**,

1. Log in to the server as the **dbadmin** user.
2. At the prompt, type:

```
AddDiagnosticText.exe [language file] [vload.exe log file]
```

Where . . .

- **[language file]** is the language file that you obtain using the method in the following section.

- **[vload.exe log file]** is the name of the log file that is created when you run **vload.exe**.

Note: The language file will contain an unsupported diagnostic code, which serves as a placeholder for a record update failure. The code is **Diagnostic 0**. Because the code is unsupported, you will need to translate the text manually to the target language. To do this, edit the language file on the server and translate the text “Record Update Failed” as desired.

3.2.4.2 Obtaining a Language File to Be Used with AddDiagnosticText.exe

The language file that you use with **AddDiagnosticText.exe** must be exported from the Language Editor.

To obtain a language file,

1. Launch the Virtua Language Editor.
2. Open the language table for your current Virtua client.
3. On the menu bar, click File, and then choose Export.

The Export window appears (*Figure 3-1*).

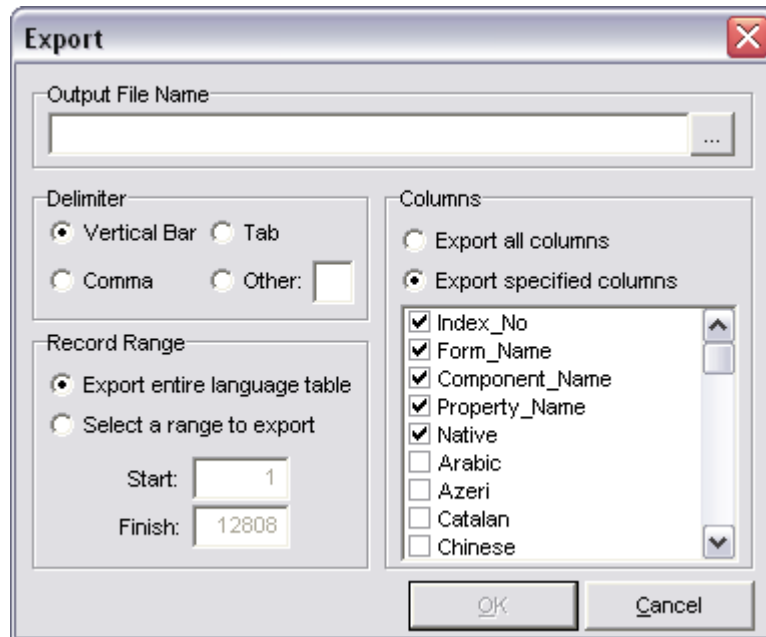
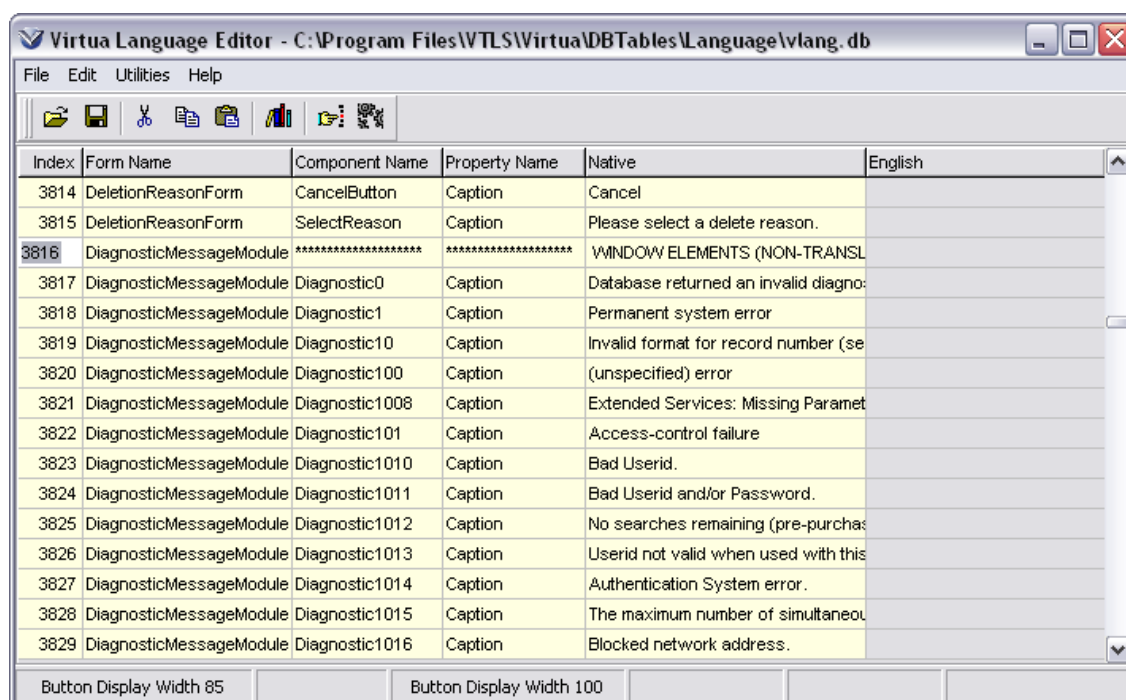


Figure 3-1. Export Window - Language Editor

4. Type the name of the output file (which will be the input file for **AddDiagnosticText.exe**).
5. In the Delimiter area, select the Tab radio button.
6. In the Columns area, select the Export Specified Columns radio button, and select the check boxes for the following columns:
 - Index_No
 - Form_Name
 - Component_Name
 - Property_Name
 - Language (You can check more than one.)
7. In the Record Range area, select the Select a Range to Export radio button.
8. In the Start field, type the Index number of the line where the Form Name *DiagnosticMessageModule* first appears in the language table.
9. In the Finish field, type the Index number of the line where the Form Name *DiagnosticMessageModule* last appears in the language table.

For example, in [Figure 3-2](#), the Index number of the line where the Form Name *DiagnosticMessageModule* first appears is 3816.



Index	Form Name	Component Name	Property Name	Native	English
3814	DeletionReasonForm	CancelButton	Caption	Cancel	
3815	DeletionReasonForm	SelectReason	Caption	Please select a delete reason.	
3816	DiagnosticMessageModule	*****	*****	*****	*****
3817	DiagnosticMessageModule	Diagnostic0	Caption	Database returned an invalid diagno:	
3818	DiagnosticMessageModule	Diagnostic1	Caption	Permanent system error	
3819	DiagnosticMessageModule	Diagnostic10	Caption	Invalid format for record number (se	
3820	DiagnosticMessageModule	Diagnostic100	Caption	(unspecified) error	
3821	DiagnosticMessageModule	Diagnostic1008	Caption	Extended Services: Missing Paramet	
3822	DiagnosticMessageModule	Diagnostic101	Caption	Access-control failure	
3823	DiagnosticMessageModule	Diagnostic1010	Caption	Bad Userid.	
3824	DiagnosticMessageModule	Diagnostic1011	Caption	Bad Userid and/or Password.	
3825	DiagnosticMessageModule	Diagnostic1012	Caption	No searches remaining (pre-purchas	
3826	DiagnosticMessageModule	Diagnostic1013	Caption	Userid not valid when used with this	
3827	DiagnosticMessageModule	Diagnostic1014	Caption	Authentication System error.	
3828	DiagnosticMessageModule	Diagnostic1015	Caption	The maximum number of simultaneou	
3829	DiagnosticMessageModule	Diagnostic1016	Caption	Blocked network address.	

Figure 3-2. Language Table - Language Editor

- After you complete all the fields on the Export window, click OK.

The Language Editor generates the language file that you specified for export. You need to ftp the language file to the server and then include the file as input for the program **AddDiagnosticText.exe**.

For additional information on using the Language Editor, see the *Virtua Translation and String Customization Guide*.

3.2.4.3 Common Error Messages and Their Meanings

As described above, you can run a utility to translate diagnostic and error code into understandable messages in an appropriate language. For your easy reference, however, below are some of the most common error codes and their meanings.

Item Records

- | | |
|------|--|
| 5062 | Item's parent bibliographic record does not exist in the database. |
| 5064 | Item's parent Bib-ID does not match the Bib-ID from subfield \$c of the 949 VIRTUAITEM tag. (<i>Item creation rejected.</i>) |

- 5083 A data format error occurred in a subfield of the 949 or 852 tag. (*Item creation rejected.*)
- 5084 Missing mandatory subfield in the 949 VIRTUAITEM tag. (*Item creation rejected.*)
- 5088 Duplicate Item-ID exists in the database. (*No database action taken.*)

Holdings Records

- 5115 No Bib-ID found for holdings record.
- 5116 No 852 tag found in holdings.
- 5117 No location found in 852 tag of holdings.
- 5125 Holdings delete failure: Associated claims exist.
- 5126 Holdings delete failure: Failed to delete associated reminders.

Patron Records

- 5300 Incoming patron record has no 270 tag (patron address).
- 5301 Incoming patron record has no 030 tag (patron type).
- 5302 Incoming patron record has no 1xx tag (patron name).
- 5303 Incoming patron record has no 015 tag (patron barcode).
- 5311 Patron barcode already exists in the database (duplicate for incoming record).
- 5333 No expiration date returned from Location + Patron matrix.

3.3 Options Available for vload.exe

This section discusses each of the options that are available for use with **vload.exe**. You can specify these options on the command line or in a text file. Before running this program, you should read and understand the contents of this section.

Hint: The command-line options available for **vload.exe** are also available for the executable **MoveStateRecords.exe**. For information about running **MoveStateRecords.exe**, see the *System Management: Cataloging User's Guide*.

For each command-line option, we provide the following information:

- **Option** - The character that you will use to call the option on the command line or in a load options file.
- **Record Types** - The type(s) of records with which you can use the option. You will want to use different options depending on the types of records you are processing.

- **Description** - A brief explanation of what the option does (i.e., how it affects the load).
- **Option Value** - The value(s) that you must specify with the option.
- **Default Value** - The default value that Virtua uses if the option is NOT specified.

Note: If you want to use the default value, you do not need to call the option when running **vload.exe**. Options that do not have a default value are considered only when specified explicitly.

- **Example** - An example of the syntax that you will use to call the option on the command line or in a load options file. The syntax for each option is either . . .
 - ♦ **-option[option value]**
 - OR-
 - ♦ **-option** (if the option does not take an option value)

3.3.1 Using a Load Options File (-@)

Option: -@

Record Types: All

Description: Lets you run **vload.exe** using options defined in a load options file.

Note:

- Unlike in the Virtua client, where you can call a different load options file for each type of record, the load options file that you call with **vload.exe** will apply to ALL records in the input file.
- Any options defined in the load options file will override any conflicting options specified in the record or at the command line.

Option Value: The path and filename of the load options file.

Note: If the file resides in the directory from which you are running **vload.exe**, you can specify just the filename.

Default Value: N/A

Example: -@loadoptions.txt

3.3.2 Determining How vload.exe Processes Records in the Input File (-c)

Option: -c

Record Types: See the Option Value description below.

Description: Specifies how **vload.exe** will process the records or record IDs in the input file.

Option Value:

- **1** (*for holdings record loading*) - If there is no 001 tag in the holdings record, will INSERT the new holdings record and will allow the addition of multiple holdings records to the same parent bibliographic record. If there is the 001 tag in the holdings record, and that holdings ID already exists in the Virtua database, will REPLACE the existing holdings record.
- **2** (*for all record types*) - Loads the records in the input file. Any incoming record that matches a record already in the database will overwrite the existing record.
- **3** (*for all record types*) - Deletes the records in the input file.
- **5** (*for authority record loading only*) - Merges incoming authority records with existing records. If a 1xx/4xx conflict is found, the record is put in Error state.
- **6** (*for authority record loading only*) - Saves incoming authority records to the database. If a 1xx/4xx conflict is found, the system overrides it so that the record is NOT put in Error state.
- **10** (*for bibliographic and FRBR record loading*) - Load the incoming bibliographic records and the items specified in the 949 VIRTUAITEM tag.
- **11** (*for holdings record loading only*) - Load the incoming holdings records and the items specified in the 949 VIRTUAITEM tag.
- **12** (*for bibliographic and FRBR record loading*) - Load only the items specified in the 949 VIRTUAITEM tag but do not load the parent bibliographic record.
- **13** (*for holdings record loading only*) - Load the incoming holdings records and the items specified in the 949 VIRTUAITEM tag AND check in issues/create items as defined in tag 949 subfield \$9.
- **16** (*for bibliographic record loading only*) - Load incoming bibliographic records and create items as specified in the 949 VIRTUAITEM tag, even if matching encoding levels (Leader position 17) are encountered. This option is used in conjunction with **-x{5}** and **-x{6}**. This makes it possible for incoming records with matching encoding levels to be treated as non-matches, and handled according to the **-s** load option.
- **22** (*for holdings record loading only*) - Load only the items specified in the 949 VIRTUAITEM tag, but do not load the parent holdings record.

- **23** (*for holdings record loading only*) - Load only the items specified in the 949 VIRTUAITEM tag (do not load the parent holdings record) AND check in issues/create items as defined in tag 949 subfield \$9.

Notes regarding values 13 and 23:

- ◆ The automated check-in will work for regular issues only.
- ◆ The *first* serials issue will have to be added manually.
- ◆ The \$9 of the 949 tag must be formatted to reflect the correct issue data.
- ◆ The automated check-in will have the same effect as a manual click of the Check-In button on the Serials Check-in window in the client.
- ◆ If the item cannot be added, the serials issue will still be checked in.

Default Value: 2

Example: -c5

3.3.3 Determining Whether Record Dates Are Compared (-C)

Option: -C

Record Types: Bibliographic and Authority

Description: Determines whether Virtua will compare the date in the 005 tag of the incoming record with the date in the 005 tag of the existing record. If the date of the incoming record is LESS recent than the date of the existing record, then the record will not be loaded.

Option Value:

- **0** - Do NOT compare the 005 tag dates.
- **1** - Compare the 005 tag dates to determine whether or not to load the record.

Default Value: 0

Example: -C0

3.3.4 Determining Whether the 999 LOADOPT Tag Is Used (-9)

Option: -9

Record Types: All

Description: Determines whether Virtua will use load options specified in the 999 LOADOPT tag of the incoming records.

Option Value:

- 0 - Do NOT use the load options specified in the 999 LOADOPT tag.
- 1 - Use the load options specified in the 999 LOADOPT tag.

Default Value: 1

Example: -90

3.3.5 Specifying the State Into Which Records Are Loaded (-b)

Option: -b

Record Types: All

Description: Specifies the state in which incoming records will be loaded.

Option Value: The state ID of the record state that you want to assign to records. To find the state ID of a given record state, you will need to query the RECORD_STATE table in the database.

Default Value: 0 (Process Immediately)

Example: -b3

3.3.6 Specifying the Character Set of Incoming Records (-d)

Option: -d

Record Types: All

Description: Specifies the character set of the records in the input file. The records will be converted into the UTF-8 encoding of Unicode as they are processed.

Option Value: The ID of the character set of the records in the input file. The codes that can be used with this parameter if you have the associated input maps loaded are:

- 2 - UTF-8 (no mapping is necessary)
- 10 - MARC-8 (MARC 21)
- 11 - ANSI extended for UCL + ANSI 8 + ANSI Z39.47
- 12 - Europa-3
- 13 - Windows Latin1
- 14 - PC-8
- 15 - Windows ALA
- 16 - Windows Arabic
- 17 - Windows Hebrew
- 18 - Windows Cyrillic
- 19 - Windows Latin 2
- 20 - ISO 6937/2
- 21 - CP 850 (Microsoft Code Page)
- 22 - ISO 6937/2 + Arabic
- 23 - ISO 6937/2 + Greek
- 24 - Big5 (variant used by Tamkang University)
- 25 - ANSI-8 + Hebrew
- 26 - UTF-8 character sets in which some separate diacritics need to be combined.
- 28 - ANSI-8 (Swiss version)
- 30 - GBK (encoding of CJK characters)
- 31 - TIS620 Classic Thai
- 32 - ISO 5426 (International Serials Data System interchange character set)
- 33 - CCCII (Chinese Character Code for Information Interchange)
- 34 - GB 18030 (Chinese National Standard GB 18030-2000)
- 35 - Big5-HKSCS (Hong Kong Special Character Set)

Default Value: 2

Example: -d10

Note: Do NOT specify a character set for which no input map exists in the database. For information about loading input maps, see the *System Management Reference Guide*.

3.3.7 Determining Whether Virtua Checks the Record Status Flag (-O)

Option: -O

Record Types: All

Description: Determines whether Virtua checks the fifth position of the Leader (Record Status Flag) of each record to determine whether duplicate records will be deleted.

Option Value:

- **0** - Ignores the value of the Record Status Flag.
- **1** - If the Record Status Flag of an incoming record is set to **d**, an existing duplicate record in the database will be deleted. If there is no existing duplicate record, no action will be taken (the record will not be loaded).
- **2** - If the Record Status Flag of an incoming record is set to **d**, an existing duplicate record in the database will be deleted. If the incoming record's Record Status Flag is set to **x**, the incoming record will be discarded. Note that this option value applies ONLY to authority records.

Default Value: 1

Example: -O0

3.3.8 Choosing a Range of Records in the Input File to Process (-S)

Option: -S OR --start

Record Types: All

Description: Defines a range of records in the input file to process. You can use this option to choose a beginning point and/or an ending point or the number of records to process.

Option Value: [x]E[y]L[z]

Where . . .

- **[x]** is the number of the record in the input file at which Virtua will *begin* processing.
- **E[y]** (*optional*) is the number of the record in the input file at which Virtua will *end* processing.
- **L[z]** (*optional*) is the number of *total* number records to process.

Default Value: None. By default, **vload.exe** will process all records in the input file.

Examples:

- To begin processing at the fiftieth record, type: **-S50**
- To begin processing at the one hundredth record in the input file and end processing at the two hundredth record in the input file, type: **-S100E200**
- To begin processing at the one hundredth record in the input file and process 50 records before stopping, type **-S100L50**

3.3.9 Displaying Help (--help)

Option: --help OR -?

Record Types: N/A

Description: Shows available command-line options for running **vload.exe**.

Option Value: This option does not accept parameters. Additionally, this option is not intended to be used with an input file. It simply displays help for **vload.exe** and should not be specified when you are actually loading records.

Default Value: N/A

3.3.10 Determining the Criteria for a Duplicate Record (-x)

Option: -x

Record Types: Bibliographic, Authority, and Patron

Description: Defines the criteria for identifying a duplicate record. Virtua will examine the tag or tags that you define for this option. If the incoming record has data in the specified tags that matches the data in the same tags, the record will be considered a duplicate record. You can use the **-r** option to determine how Virtua merges duplicate records.

Option Value: [tag]{n}

Where . . .

- **[tag]** is the tag on which you want to perform the duplicate test. For a list of tags that you can use for performing duplicate tests, see the section "[Tags to Use for Duplicate Testing](#)" in this chapter.
- **{n}** is a value which indicates which action should be taken when a duplicate record is found. For details, see the section "[Specifying What Virtua Should Do When a Duplicate Record is Found](#)" in this chapter.

For additional information about specifying duplicate test options, see the section "[About Duplicate Tests and Merge Load Options](#)" in this chapter.

Default Values:

- **Authority Records:** -x001{1}, -x010{1}, and -x1xx{1}
- **Bibliographic records:** -x001{1} and -x035{1}, and -xMRG{1}
- **Patron records:** -x001{1}, -x015{1}, and -x016{1}

Note: -xMRG is a special test and is ignored unless an -m value is also provided.

According to these default duplicate tests, whenever a record in the input file matches a record in the database based on any of the tags listed above, the *existing record will be replaced with the incoming record*.

If you do NOT want records in the database to be overwritten automatically based on these rules, you must explicitly define alternate duplicate tests. You must define an alternate duplicate test for *each* tag for which you do not want to use the

default. For example, if you are loading bibliographic records and do not want to use either of the default duplicate tests, you must define alternate duplicate tests for both the 001 and 035 tags.

Note: This rule applies for merge rules as well. If you want to merge duplicate records instead of overwriting them based on the default duplicate tests, you must define a **MergeString** with the desired merge rules and then call that **MergeString** for *each* tag that has a default duplicate test defined. For details on merge rules, see the following section and the section "[About Merge Options](#)" in this chapter.

3.3.11 Specifying Rules for Merging Duplicate Records (-r)

Option: -r

Record Types: Bibliographic, Authority, and Patron

Description: Supports record merging if requested by a **-x** option setting.

Option Value: For information about specifying rules for merging duplicate records, see the section "[Determine the Criteria for a Duplicate Record \(-x\)](#)" and "[About Duplicate Tests and Merge Load Options](#)" in this chapter.

Default Value: N/A

3.3.12 Specifying an Action for Loading Non-duplicate Records (-s)

Option: -s

Record Types: Bibliographic, Authority, and Patron

Description: Defines the action that Virtua will take on non-duplicate records.

Option Value:

- 0 - Do nothing (the record is not loaded).
- 1 - Load the record normally.
- 2 - Load the record into Error state.

Default Value: 1

Example: -s2

Note: You can use this option only when you also use the **-x** option to define criteria for duplicate records.

3.3.13 Merging Existing Bibliographic Records (-m)

Option: -m

Record Types: Bibliographic

Description: Specifies a filename containing the Innovative ID(s) of any existing record(s) to be absorbed by another existing record(s). The file should contain two IDs per line, separated by a tab. For each line,

- On the left, type the Innovative ID of the incoming record to be absorbed.
- On the right, type the Innovative ID of the record that will absorb all of the incoming record's items and other dependencies.

Note: *Absorbing* means attaching all of the items and other dependencies of the incoming record - specified on the left of each line of the file - to the existing record - specified on the right of each line of the file - and deleting the ID of the incoming record.

Option Value:

- A filename

Default Value: N/A

Note: To merge, rather than absorb, a record, you can include an `-xMRG{}` option to specify merge rules in the same manner used for merging new records with existing records, as described for `-x[tag] {n}`.

3.3.14 Determining the Date to Use as the Entered Date (-l)

Option: -l (lowercase L)

Record Types: All

Description: Determines whether incoming records retain the date specified in their 008 tags or if this tag is modified to reflect the current date.

Option Value:

- **0** - Replace the date specified in positions 0-5 of the 008 tag with the current date (specified by the server).
- **1** - Retain the date specified in positions 0-5 of the 008 tag.

Default Value: 0

Example: -l1

3.3.15 Specifying that Virtua Deletes the Records Indicated in the File of IDs (-X)

Option: -X

Record Types: All

Description: Specifies that the records associated with the IDs listed in the input file will be deleted from the database.

Important: When you use this option, records will be permanently deleted from the database.

Option Value:

- **101** - Specifies that the IDs in the input file correspond to bibliographic records. The bibliographic records in your database that have an ID specified in the input file will be permanently deleted.
- **102** - Specifies that the IDs in the input file correspond to authority records. The authority records in your database that have an ID specified in the input file will be permanently deleted.

- **104** - Specifies that the IDs in the input file correspond to holdings records. The holdings records in your database that have an ID specified in the input file will be permanently deleted.
- **105** - Specifies that the IDs in the input file correspond to patron records. The patron records in your database that have an ID specified in the input file will be permanently deleted.

Default Value: N/A

Example: -X101

3.3.16 Defining a Library Symbol (-U)

Option: -U

Record Types: Bibliographic

Description: Specifies a library symbol that will be added to each bibliographic record. This option is for Union databases only.

Option Value: Any library symbol that you have defined in the Library Members parameter in the Virtua Profiler.

Default Value: N/A

Example: -UDR

3.3.17 Determining Whether Records are to be Split into FRBR Records (-f)

Option: -f

Record Types: Bibliographic

Description: Determines whether the incoming bibliographic record will be split into a FRBR record. For information about FRBR records, see the *FRBR Cataloging User's Guide*.

Option Value:

- **0** - Do not split incoming bibliographic records into FRBR records.

- 1 - Split incoming bibliographic records into FRBR records.

Default Value: 0

Example: -f1

3.3.18 Adding a Special 035 Tag (-3)

Option: -3

Record Types: Bibliographic

Description: Determines whether to add a special 035 tag to use for duplicate checking.

Option Value:

- 0 - Do not add special 035
- 1 - Add special 035 (TMUA option).
- 2 - Add special 035 (UCL option).
- 3 - Add special 035 (NLW option).
- 4 - Add special 035 (BRO option)

Default Value: 0

Example: -31

Note: This option is for use by the following customers only: Thai Union Catalog (TMUA), Université catholique de Louvain (UCL), National Library of Wales (NLW), and Hong Kong Public Library's Book Registration Office (BRO). The option uses a special duplicate checking algorithm to generate an additional 035 tag in an incoming bibliographic record when the incoming record does not have either an Innovative Bib-ID or a valid 035 tag. All data is normalized for duplicate checking, and is based on the contents of the following tags:

Option 1 (TMUA):	Option 2 (UCL):
• 100 subfield \$a	• 100 subfield \$a
• 110 subfield \$b	• 110 subfield \$b
• 111 subfield \$a	• 111 subfield \$a
• 130 subfield \$a	• 130 subfield \$a
• 240 or 245 all subfields except \$c	• 240 or 245 all subfields except \$c
• 250 subfield \$a	• 250 subfield \$a
	• 008 positions 07-10
Option 3 (NLW):	Option 4 (HKPL - BRO):
• 245 subfield \$a	• Leader position 07
• 852 subfield \$h	• 020 subfield \$a
	• 245 subfields \$a, \$b, \$c, and \$n
	• 260 subfields \$a, \$b, and \$c
	• 970 subfield \$a
<p>Instead of using vload.exe to add a special 035 tag, you can run AddSpecial035.exe directly. A usage example for AddSpecial035.exe when run as an executable is provided below (where the -s option corresponds to -3 in vload.exe):</p> <p>Usage: AddSpecial035.exe [-I --input-file-name <Input Name>] [-O --output-file-name <Output Name>] [-s --special_option <Special Option >] [-? --help] [--usage]</p> <p>By default, AddSpecial035.exe runs with -s1.</p>	

3.3.19 Creating Item Tags for Hebrew Union College (-h)

Option: -h

Record Types: Bibliographic

Description: Determines whether 949 item tags are created in *bibliographic* records using the 950 and 998 tags.

Option Value:

- 0 - 949 tags are NOT constructed from 950 and 998 tags.
- 1 - 949 tags are constructed.
- 2 - 949 tags are constructed from 852 tags

Example: -h1

3.3.20 Specifying a Validation Rule (-v)

Option: -v

Record Types: Bibliographic

Description: Specifies whether incoming bibliographic records are validated and saved to the database, validated *only*, or saved to the database without validation.

Option Value:

- 0 - Validate and save
- 1 - Validate only
- 2 - Save but do not validate

Default Value: 0

Example: -v1

3.3.21 Specifying a Filename (-w)

Option: -w

Record Types: Bibliographic (*ISSN databases only*)

Description: Provides a filename for the source of records loaded by vload.exe. The filename is stored in subfield \$w of the 039 tag. If the record goes into Error state, its position in the file will be stored in subfield \$x of the 039 tag.

Option Value: A filename

Default Value: No filename, so subfield \$w and subfield \$x in the 039 tag are not created.

3.3.22 Blocking Item Creation Based on the 949 Tag (-k)

Option: -k

Record Types: Bibliographic and Holdings

Description: Indicates whether to block the creation of item records based on information in the 949 tag incoming bibliographic and holdings records.

Option Value:

- 0 - Do not block item creation.
- 1 - Block item creation if items already exist with a location that matches the *main part* of the location code specified in tag 949 \$D of the record being processed.
- 2 - Block item creation if items already exist with a location that matches the *entire* location code specified in tag 949 \$D of the record being processed.

Default Value: 0

Example: -k2

3.3.23 Replacing Subfield Codes in the 949 VIRTUAITEM Tag (-p)

Option: -p

Record Types: Bibliographic and Holdings

Description: Defines subfield codes in the VIRTUAITEM 949 tag of incoming bibliographic and holdings records that will be replaced by other subfield codes before the item record is loaded.

Option Value: One or more subfield code pairs. The first code in the pair represents the subfield code in the VIRTUAITEM 949 tag of incoming records that is to be replaced. The second code represents the subfield code that will be used as a replacement in the loaded record.

Default Value: N/A

Example: To specify that the subfield code \$a of the VIRTUAITEM 949 tag of incoming bibliographic and holdings records will be replaced by the subfield code \$A, type:

-paA

Note: You can specify multiple subfield replacements in a single parameter by entering them one after another with no spaces. For example: **-paAdD**

3.3.24 Replacing Subfield Codes in the 949 VIRTUAITEMCALLNUMBER Tag (-q)

Option: -q

Record Types: Bibliographic and Holdings

Description: Defines subfield codes in the VIRTUAITEMCALLNUMBER 949 tag of incoming bibliographic and holdings records that will be replaced by other subfield codes before the item record is loaded.

Option Value: One or more subfield code pairs. The first code in the pair represents the subfield code in the VIRTUAITEMCALLNUMBER 949 tag of incoming records that is to be replaced. The second code represents the subfield code that will be used as a replacement in the loaded record.

Default Value: N/A

Example: To specify that the subfield code \$a of the VIRTUAITEMCALLNUMBER 949 tag of incoming bibliographic and holdings records will be replaced by the subfield code \$A, type:

-qaA

Note: You can specify multiple subfield replacements in a single parameter by entering them one after another with no spaces. For example: **-qaAdD**

3.3.25 Requiring the *VIRTUAITEM LABEL* (-L)

Option: -L

Description: Specifies whether the label *VIRTUAITEM* is required in subfield \$A of the 949 tag of incoming bibliographic and holdings records for which item records will be loaded.

Option Value:

- **0** - The label *VIRTUAITEM* is NOT required. (This option is used for backwards compatibility, and is not used in most cases.)
- **1** - The label *VIRTUAITEM* is required.

Default Value: 1

Example: -L0

3.3.26 Blocking Records with a 949 Subfield \$z Tag (-z)

Option: -z

Record Types: Bibliographic and Holdings

Description: Determines whether item records are created from *VIRTUAITEM* 949 tags that have an existing \$z. When **vload.exe** loads item records from 949 tags it adds a subfield \$z to each 949 tag containing the time and date that the record was loaded.

If a record has a 949 tag with a subfield \$z, the item record has been loaded into a Virtua database at some point. You can use this option to specify that **vload.exe** ignore an existing subfield \$z and load the item record anyway.

Option Value:

- **0** - Virtua will NOT create item records from *VIRTUAITEM* 949 tags that have a subfield \$z.
- **1** - Virtua will ignore an existing subfield \$z in *VIRTUAITEM* 949 tags and load the item record.

Default Value: 0

Example: -z1

3.3.27 Specifying a Shelving Location for New Item Records (-B)

Option: -B

Record Types: Bibliographic and Holdings

Description: Defines a default shelving location that Virtua will use when loading item records for which either no shelving location or a non-existent shelving location has been defined in the 949 tag, or, for Holdings records, in the 852 tag subfield \$b.

Option Value: A numeric location code. This code must match the location code of an existing location in your database.

Default Value: N/A

Example: -B10000

3.3.28 Specifying a Default Item Class (-D)

Option: -D

Record Types: Bibliographic and Holdings

Description: Defines a default item class that Virtua will use when loading item records for which either no item class or a non-existent item class has been defined in the 949 tag.

Option Value: An item class code. This code must match an item class code of an item class in the database.

Default Value: N/A

Example: -D3

3.3.29 Determining Whether Blind References Are Loaded (-G)

Option: -G

Record Types: Authority

Description: Specifies whether Virtua loads authority records that have no associated bibliographic records in the database (blind references).

Option Value:

- 0 - Blind references will be loaded.
- 1 - Blind references will NOT be loaded.

Default Value: 0

Example: -G1

3.3.30 Specifying the Heading Type for Authority Records (-H)

Note: You no longer have to use this option to designate the heading type of incoming authority records as *subject* because when Virtua processes a name authority, the software automatically creates both a name and subject authority record, or “single authority,” without the need for a second load.

Option: -H

Record Types: Authority

Description: Specifies the heading type of incoming authority records.

Option Value:

- 0 - The heading type of incoming authority records is determined by the contents of the record.
- 1 - The heading type of incoming authority records is *subject*.

Default Value: 0

Example: -H1

3.3.31 Preserving Local Tags (-t)

Option: -t

Record Types: Authority

Description: Specifies that an existing record's local tags (tags with subfield \$5 containing the current library symbol) should be preserved when the record is replaced by a matching incoming record.

Option Value:

- 0 - Local tags in existing records are not preserved.
- 1 - Local tags in existing records are preserved.

Default Value: 0

Example: -t1

3.3.32 Removing the 003 Tag Restriction for 010 Tag Creation (-R)

Option: -R

Record Types: Authority

Description: Specifies whether Virtua requires that incoming authority records have a DLC 003 tag before an 010 tag is automatically created.

Option Value:

- 0 - Virtua will not create an 010 tag for incoming records unless a DLC 003 tag already exists.
- 1 - Virtua does not require that a DLC 003 tag exists to create an 010 tag.

Default Value: 0

Example: -R1

3.3.33 Blocking Deletion of Records with 1xx/5xx Matches (-N)

Option: -N

Record Types: Authority

Description: Specifies whether or not Virtua will block the deletion of a record if a 1xx value in the authority record to be deleted matches a 5xx value in another authority record.

Option Value:

- 0 - Deletion of the record is allowed.
- 1 - Deletion of the record is blocked.

Default Value: 0

Example: -N1

3.3.34 Specifying the Date Pattern of the 042 Tag (-o)

Option: -o

Record Types: Patron

Description: Specifies the date pattern that will be used by Virtua to interpret dates found in the 042 tag subfields \$a, \$b, and \$c and 247 tag subfield \$b of incoming patron records.

Option Value:

- 0 - YYYYMMDDHH24MI (e.g., 201301301735)
- 1 - MMDDYYYYHHMI (e.g., 013120131735)

Default Value: 0

Example: -o1

3.3.35 Checking the Patron Record Date (-I)

Option: -I

Record Types: Patron

Description: Specifies whether Virtua compares the 005 tag of duplicate patron records (records are considered duplicates if they have the same 015 tag subfield \$a) before overwriting the existing patron record. The 005 tag specifies the last transaction date of the patron.

Option Value:

- **0** - The 005 tags are NOT compared. Virtua overwrites existing patron records with matching incoming patron records.
- **2** - The 005 tags are compared. If the 005 tag of an incoming patron record specifies a date that is *earlier* than the date in the 005 tag of the existing patron record, the record will NOT be loaded. If the 005 tag of the incoming patron is *later* than the date in the 005 tag of the existing patron, Virtua will overwrite the existing record with the incoming record.

Default Value: 0

Example: -I2

3.3.36 Defining a Default Expiration Date (-J)

Option: -J

Record Types: Patron

Description: Defines a default expiration date that Virtua will use when loading patron records that do not have an 042 subfield \$b tag.

Option Value: A date in the following format: YYYYMMDDHHMM.

Default Value: N/A

Example: -J200412312359

3.3.37 Defining a Default Delete Date (-K)

Option: -K

Record Types: Patron

Description: Defines a default delete date that Virtua will use when loading patron records that do not have a tag 042 subfield \$c.

Option Value: : A date in the format specified by load option -o; the default format is YYYYMMDDHHMM.

Default Value: N/A

Example: -K201812312359

3.3.38 Adding a Default Patron Password (-P)

Option: -P

Record Types: Patron

Description: Specifies whether a default password is added to the incoming patron record.

Note: Virtua uses the date of birth specified in the patron record 100 tag subfield \$d for the password value. If no date of birth is specified in the patron record, the system uses the patron barcode from the 015 tag subfield \$a.

Option Value:

- 0 - Do not add default password.
- 1 - Add default password.

Default Value: 0

Example: -P1

3.3.39 Determining Whether 042 Values Are Retained (-T)

Option: -T

Record Types: Patron

Description: Specifies whether the dates specified in the 042 tag of incoming patron records are retained or if they are recalculated based on parameters set in the Virtua Profiler.

Option Value:

- **0** - Existing 042 tags are retained.
- **1** - Existing 042 tags are NOT retained. Dates in the 042 tag are recalculated based on Circulation parameters set in the Virtua Profiler. For information about setting Circulation parameters, see the *Virtua Profiler/Circulation User's Guide*.

Default Value: 0

Example: -T1

3.3.40 Maintaining Entity Links During a FRBR Load (-u)

Option: -u

Record Types: Bibliographic

Description: When loading FRBR records from another database, specifies whether Virtua will load the records as FRBR records and maintain the links between Works, Expressions, and Manifestations.

Option Value:

- **0** - Do not use Special FRBR load
- **1** - Use special FRBR load

Default Value: 0

Example: -u1

3.3.41 Providing a Vendor ID for MARC Order Processing (-V)

Option: -V

Record Types: Bibliographic

Description: Provides the vendor ID for MARC order processing of embedded ordering information in bibliographic records.

Option Value:

- 0 – Disable MARC order processing
- >0 – Value must match a Innovative database vendor ID

Default Value: 0

Example: -V565

3.3.42 Determining the Type of MARC Order Processing (-W)

Option: -W

Record Types: Bibliographic

Description: Determines the type of processing for MARC order information embedded in a bibliographic record. For this setting to take effect, the -V option must be set to a value greater than 0.

Option Value:

- 1 – Process MARC order information as an order
- 2 – Process MARC order information as a request

Default Value: 1

Example: -W2

3.3.43 Merging Authority Records When a 4xx Tag Conflict Occurs (-E)

Option: -E

Record Types: Authority

Description: Specifies whether to merge records being loaded with local or provisional authority records when there is a 1xx/4xx tag conflict.

Option Value:

- 0 – No action
- 1 – Merge records accordingly

Default Value: 0

Example: -E1

3.3.44 Determining Whether Data in Tag 880 Is Moved (-8)

Option: -8

Record Types: Bibliographic

Description: Determines whether Virtua will move 880 tag data to associated tags that have equivalent data.

Option Value:

- 0 - Do NOT move 880 tag data to associated tags.
- 1 - Move 880 tag data to associated tags.

Default Value: 0

Example: -81

3.3.45 Specifying the MARC Format of Incoming Records (-Y)

Option: -Y

Record Types: Bibliographic

Description: Determines whether incoming records will be treated as MARC21 records or as UNIMARC records. This setting is only relevant when the value of the MARC Format setting on the Cataloging Basic Options window in the Virtua Profiler is set to Multiple; see the *Virtua Profiler/Cataloging Parameters User's Guide* for more information.

Option Value:

- 0 - Incoming records are in MARC21 format.
- 1 - Incoming records are in UNIMARC format.

Default Value: 0

Example: -Y1

3.4 Other vload.exe Options Available from the Command Line Only

3.4.1 Outputting Records that Failed to Delete to a File (--failed-delete-record-file)

Option: --failed-delete-record-file

Record Types: Authority

Description: Specifies a file into which records that cannot be deleted are written.

Option Value: A filename

Default Value: None

Example: --failed-delete-record-file [filename]

Note: In the situation where you are loading authority records with Leader/05=d, this option is useful for catching records that cannot be deleted because, for example, they have dependencies.

3.4.2 Determining Whether to Create a 999 VTLSSORT Tag (--create-sort-tag)

Option: --create-sort-tag

Record Types: All

Description: Determines whether a 999 VTLSSORT tag will be created for those incoming records that do not already have a 999 VTLSSORT tag. This new 999 VTLSSORT tag uses the tag order of the incoming record.

Example: --create-sort-tag

3.4.3 Reading Input from a File (--input-file)

Option: --input-file

Record Types: All

Description: Specifies a file instead of standard input device from which to read input.

Option Value: A filename

Default Value: None

Example: --input-file [filename]

3.4.4 Adding an XML File to Convert to MARC (--xml-element-name)

Option: --xml-element-name [XML element] <[filename]

Record Types: XML

Description: Lets you load a file of XML records into the database and create corresponding MARC records. The argument for this option must be the name of the beginning XML element. After the records are converted to MARC and loaded into the database, the XML is available to view and edit from the XML tab of the View Bibliographic Record window in the Virtua client.

Note: If you are using any format other than MARCXML, you **MUST** use this argument in conjunction with the **--xml-format** command-line option.

Option Value: First XML element

Default Value: None

Example: --xml-element-name record <marc.xml

3.4.5 Defining an XML File Format to Convert to MARC (**--xml-format**)

Option: --xml-format [XML format name] <[filename]

Record Types: XML

Description: Defines the format of a file of incoming XML records. This argument **MUST** be used in conjunction with **--xml-element-name** to effectively convert and load XML records into the database.

Option Value: XML type; currently only MODS is supported

Note: If you are using MODS, you **MUST** move the stylesheet and schema **MODS2MARCXMLStyleSheet.xml** and **MODSSchema.xsd** from the installation directory to the local working directory. To do this, use the following command:

```
> LoadXMLFormat.sh MODS MODS2MARCXMLStyleSheet.xml MODSSchema.xsd
```

Default Value: None

Example: --xml-element-name mods --xml-format MODS <modsXMLrecords

3.4.6 Indexing AAPs Only (--AAP-only-indexing)

Option: --AAP-only-indexing

Record Types: Bibliographic

Description: Indexes author and title headings only; does not support modification of the bibliographic record.

Default Value: None

Example: --AAP-only-indexing

Note:

- This option is useful for processing records when options in the Additional Access Points parameter have been changed.
- If the HEADING_INDEXING_OFF environment variable is activated, Virtua will not index any incoming headings. See the *System Management Reference Guide* for information about environment variables.

3.4.7 Indexing User-defined Headings Only (--UserHeading-only-indexing)

Option: --UserHeading-only-indexing

Record Types: Bibliographic

Description: Indexes user-defined headings only; does not support modification of the bibliographic record.

Default Value: None

Example: --UserHeading-only-indexing

Notes:

- This option is useful for processing records when options in the Tags Indexed for User Defined Search parameter have been changed.
- If the HEADING_INDEXING_OFF environment variable is activated, Virtua will not index any incoming headings. See the *System Management Reference Guide* for information about environment variables.

3.4.8 Indexing Subject Headings Only (--SubjectHeading-only-indexing)

Option: --SubjectHeading-only-indexing

Record Type: Bibliographic

Description: Indexes subject headings only; does not support modification of the bibliographic record.

Default Value: None

Example: --SubjectHeading-only-indexing

Note: If the HEADING_INDEXING_OFF environment variable is activated, Virtua will not index any incoming headings. See the *System Management Reference Guide* for information about environment variables.

3.4.9 Disabling FRBR Subject Indexing (--NoFRBRSubjectIndexing)

Option: --NoFRBRSubjectIndexing

Record Type: Bibliographic

Description: Prevents subject indexing of FRBR records during the FRBR splitting process, but the bibliographic record will be modified.

Default Value: None

Example: --NoFRBRSubjectIndexing

3.4.10 Disabling Indexing (--indexing-disabled)

Option: --indexing-disabled

Record Types: Bibliographic

Description: Allows modification of the bibliographic record but performs no indexing.

Default Value: None

Example: --indexing-disabled

Note: This option can serve to speed up the cataloging process when you only want to merge tags that are not generally indexed.

3.4.11 Disabling Location Indexing (--no-location-filter)

Option: --no-location-filter

Record Types: Bibliographic

Description: Allows loading of the bibliographic record without updating the bibliographic_location table.

Default Value: None

Example: --no-location-filter

Note: In most cases, this option should NOT be used.

3.4.12 Disabling Patron Record Indexing (--no-patron-indexing)

Note: This option will only be of use to those users who have the optional enhanced Patron Search. For more information about the enhanced Patron search, see the *Virtua OPAC User's Guide*.

Option: --no-patron-indexing

Record Types: Patron

Description: When loading a file of patron records, prohibits patron records from being indexed for the Patron search.

Default Value: None

Example: --no-patron-indexing < PatronRecords.rec

3.4.13 Disabling Holdings Record Indexing (--no-holdings-indexing)

Note: This option will only be of use to those users who have the optional Holdings Keyword Search. For more information about the Holdings Keyword search, see the *Virtua OPAC User's Guide*.

Option: --no-holdings-indexing

Record Types: Holdings

Description: When loading a file of holdings records, prohibits holdings records from being indexed for the Holdings Keyword search.

Default Value: None

Example: --no-holdings-indexing < HoldingsRecords.rec

3.4.14 Masking Incoming Records (--mask-records)

Option: --mask-records

Record Types: Bibliographic

Description: Determines whether incoming MARC records will be masked.

Default Value: None

Example: --mask records

3.4.15 Keeping Date and Time of Incoming Records (--use-incoming-039-tag)

Option: --use-incoming-039-tag

Record Types: Bibliographic

Description: Maintains existing last modified and create dates (subfields \$a and \$y) in the 039 tags of incoming records.

Default Value: None

Example: --use-incoming-039-tag

3.4.16 Always creating the 035 tag (--always-auto-generate-035-tag)

Option: --always-auto-generate-035-tag

Record Types: Authority

Description: Uses the prefix (plus the value from the 001 tag) specified in the Profiler Cataloging Basic Options parameter Automatic Generation of 035 to always create the 035 tag whether or not other 035 tags exist in the record.

Default Value: None

Example: --always-auto-generate-035-tag

3.4.17 Preventing 4xx Conflicts in Authority Records (--prevent-authority-4xx-conflicts)

Option: --prevent-authority-4xx-conflicts '[prefix]'

Record Types: Authority

Description: Prevents 4xx conflicts in incoming authority records with the defined prefix by adding subfield \$z to the 4xx tag causing the conflict. The new subfield \$z will contain the 035 control number.

If ANY is given as the prefix, the first 035 tag will be used regardless of its prefix.

Example: --prevent-authority-4xx-conflicts '(DE-588)'

The 035 control number must begin with (DE-588) in order to prevent the 4xx conflict by adding \$z containing the control number to the 4xx tag.

3.5 About Duplicate Tests and Merge Load Options

From the command line, **vload.exe** lets you specify complex load options that tell Virtua to perform testing to find duplicate records, take a specific action if duplicates are found, and, if desired, merge the two records in a specific way. Currently, both the duplicate test and merge load options are available for bibliographic, authority, and patron records.

The duplicate test load option (**-x**) lets you specify . . .

- Tags that Virtua will use to test for duplicate records.
-AND-
- The action that Virtua will take if it finds duplicates.

Additionally, the merge string load option (**-r**) works in conjunction with the **-x** option to let you specify criteria by which two records will be merged to create a new record in the database. All the tags in this new record come from the two source records:

- The **incoming record**, which comes from an input file.
- The **existing record**, which is the one currently in the database.

Important: The order of the **-x** options controls the behavior of duplicate testing. Duplicate testing responds to the **-x** options *from left to right*. This means that if a record finds a duplicate for one of the **-x** tests, the system will use the action or merging referenced by that **-x** option, and it will NOT use the **-x** options to the right of it. For example, for **-x001{0} -x1xx{0} -x010{Test} -rTest{T035/E//I}** etc., if the system detects a duplicate with the **-x001** test, it will ignore the **-x** options to the right of it and will not perform the specified merge.

The following sections provide details on how to construct duplicate test and merge options, along with examples that you can use as guidelines for constructing your own rules.

Note: If you prefer, instead of defining your duplicate test and merge options at the command line, you can store the options in a file, which you can call from the command line when you run **vload.exe**. For details, see the section "[Working with Load Options Files](#)" in this user's guide.

3.5.1 About the Format of the Duplicate Test Option

The duplicate test option (**-x**) uses the following general format:

[tag]{n}

Where . . .

- **[tag]** is the tag on which you want to perform the duplicate test. For a list of tags that you can use for performing duplicate tests, see the section "[Tags to Use for Duplicate Testing](#)" in this chapter.
- **{n}** is a value which indicates which action should be taken when a duplicate record is found. For details, see the section "[Specifying What Virtua Should Do When a Duplicate Record is Found](#)" in this chapter.

3.5.1.1 Tags to Use for Duplicate Testing

Virtua lets you use a particular set of MARC tags to perform duplicate testing. Below, we describe the tags in different types of records that are available for use in a duplicate test.

Note: If you want Virtua to perform duplicate tests on more than one tag, specify multiple **-x** options at the command line. By default, the system always does duplicate tests 001 and 035 for bibliographic records; 001, 010, and 1xx for authority records; and 001 and 015 for patron records.

Tags for Duplicate Tests for Bibliographic Records

- **001** - The 001 tag of *any* incoming record must begin with *vtls* followed by a record ID number. The 003 tag of the incoming record must match the value stored in

the database. If the 001 tag starts with *vtls*, the 003 tag contains database code, and matches a record already in the database, then 001 duplication exists.

- **010** - Subfield \$a in the 010 tag of an incoming bibliographic record must match the 010 tag, subfield \$a of an existing bibliographic record.
- **020** - Subfield \$a in the 020 tag of an incoming bibliographic record must match the 020 tag, subfield \$a of an existing bibliographic record.
- **022** - Subfield \$a in the 022 tag of an incoming bibliographic record must match the 022 tag, subfield \$a of an existing bibliographic record.
- **024** - Subfield \$a and the 1st indicator of the 024 tag of an incoming bibliographic record must match the 024 tag, subfield \$a and the 1st indicator of an existing bibliographic record.
- **028** - Subfield \$a and the 1st indicator of the 028 tag of an incoming bibliographic record must match the 028 tag, subfield \$a and the 1st indicator of an existing bibliographic record.
- **035** - Subfield \$a in the 035 tag of an incoming bibliographic record must match the 035 tag, subfield \$a of an existing bibliographic record. Also, 035 tags are usually constructed from 001 and 003 tags. If the 035 tag that would be constructed from the 001 and 003 tags of an incoming record matches the 035 tag of an existing record, the record will be an 035 duplication.
- **222** - The 222 tag of the incoming bibliographic record must match the 222 title heading from a 222 tag in an existing bibliographic record.

Tags for Duplicate Tests for Authority Records

- **001** - The 001 tag of *any* incoming record must begin with *vtls* followed by a record ID number. The 003 tag of the incoming record must match the value stored in the database. If the 001 tag starts with *vtls*, the 003 tag contains database code, and matches a record already in the database, then 001 duplication exists.
- **010** - Subfield \$a in the 010 tag of an incoming authority record must match the 010 tag, subfield \$a of an existing authority record.
- **035** - Subfield \$a in the 035 tag of an incoming authority record must match the 035 tag, subfield \$a of an existing authority record.
- **1xx** - Subfield \$a in the 1xx tag of an incoming *authority* record must match an existing authority heading in an authority record in the database.

Tags for Duplicate Tests for Patron Records

- **001** - The 001 tag of *any* incoming record must begin with *vtls* followed by a record ID number. The 003 tag of the incoming record must match the value stored in the database. If the 001 tag starts with *vtls*, the 003 tag contains database code, and matches a record already in the database, then 001 duplication exists.
- **015** - Subfield \$a in the 015 tag of an incoming patron record must match the 015 tag, subfield \$a of an existing patron record.

- **016** - Subfield \$a in the 016 tag of an incoming patron record must match the 016 tag, subfield \$a of an existing patron record.
- **035** - Subfield \$a in the 035 tag of an incoming patron record must match the 035 tag, subfield \$a of an existing patron record.

3.5.1.2 Specifying What Virtua Should Do When a Duplicate Record Is Found

For each tag that you want to use for a duplicate test, you can specify one of the following options, which indicates what Virtua should do when it detects a duplicate record:

- **0** - Do nothing with the new (incoming) record.
- **1** - Replace the database record with the new record.

Note: If a new MARC bibliographic or authority record is added that contains a heading that differs from an existing permanent authority heading in *ending punctuation only*, the new heading will be considered identical to the existing heading, and the new record will replace the existing record in the database.

- **2** - Save the new record in Error state.
- **3** - Discard the new record and delete the database record.
- **4** - Keep the database record and add the library symbol specified with **-U**.
- **5** - Keep the new record and delete the database record IF the new record has a less thorough or weaker level of cataloging; otherwise, discard the new record.
- **6** - Keep the new record and delete the database record IF the new record has a more thorough or stronger level of cataloging; otherwise, discard the new record.

Note: For both 5 and 6, a cataloging level (position 17 of the Leader) with higher values is *weaker* than a cataloging level with lower values. If the level is the same, the duplicate test will fail. The **-s** option, which applies only to bibliographic records, can be used to determine what to do if no other duplicate tests are given.

- **MergeString** - Merge the incoming record with the database record using the specified merge options. The value for **MergeString** should match the string that you specify for the **-r** option. For details on merge options, see the section "[About Merge Options](#)" in this chapter.

3.5.1.3 Examples and Format Variations of the Duplicate Test Option

As described in the following sections, the duplicate test option uses the following general format:

-x[tag]{n}

For example, to specify that a duplicate test should be performed on the 001 tag and that when a duplicate record is found the existing record should be replaced with the new record, use this option:

-x001{1}

If you want to perform duplicate tests on more than one tag but perform the *same* action when a duplicate record is found, use the following shortened format:

-x[tag]&[tag]&[tag]{n}

instead of **-x[tag]{n} -x[tag]{n} -x[tag]{n}**

For example, to perform duplicate tests on the 001, 015, and 016 tags and specify that the incoming record should be saved in Error state when a duplicate is found, use the following option:

-x001&015&016{2}

You can specify as many tags as you like, separating each with an ampersand (&).

Important: When entered via the command line, the ampersand (&) can cause errors from the UNIX interpreter. To prevent errors from occurring, escape the & character as shown in the following example: **-x020\&022{2}**

3.5.2 About Merge Options

The merging process prevents any two tags existing in the new, merged record from having both identical tags and field data. Once a merge is complete, the incoming record is discarded, and the identical record will be replaced in the database by the new record created by the merge process.

For each duplicate test that you define, you have the option of specifying that duplicate records should be merged according to particular rules. For example, to specify that duplicate tests should be performed on the 001 tag and that duplicate records should be merged, type:

-x001{MergeString} -rMergeString[merge rules]

Where **{MergeString}** specifies that the records should be merged according to the rules defined in the **-r** option, and **[merge rules]** specifies the merge rules. In a merge, Virtua discards the incoming record and replaces the existing record in the database with a *new* record created by the merge process. The new record will contain the control number (001 tag) of the existing record.

The merge load option involves several components that the system reads from left to right. On the command line, you specify one or more merge options *after* the duplicate tests option.

- The merge load option starts with the **-r** command, which is followed by the merging string name.
- The merge string name links the merge rule to the preceding duplicate test (**-x**).
- The merge string name is followed by the tag test, which determines which tags will be merged.
- The final component of the merge load option is the merge rule, which tells Virtua what action to take.

The general format of a merge rules load option is as follows:

-rMergeString{TagTest/MergeRule//DefaultMergeRule}

Where . . .

- **TagTest** specifies the tags to which a given **MergeRule** applies.
 - **MergeRule** is the rule applied to a given **TagTest**.
- Note:** You can specify as many TagTest/MergeRule pairs as you need.
- **DefaultMergeRule** is the default merge rule applied to all tags for which a specific **TagTest** is not defined.

Hint: To use the same merge rule for ALL tags, omit the **TagTest/MergeRule** pair and specify only the **DefaultMergeRule** (i.e., **-rMergeString{/DefaultMergeRule}**), which will apply to all tags when not other rules are defined.

The following sections describe each of these components in detail. For examples of merge rules, see the section "[Examples of Duplicate Tests and Merge Options](#)" in this chapter.

3.5.2.1.1 About the MergeString

The **MergeString** is a user-defined string that links a duplicate test (**-x**) to a merge option (**-r**). When Virtua finds duplicate records, it looks at the **MergeString** defined for the duplicate test to determine which merge options should be used.

For example, the following options use the **MergeString** *Process*:

```
-x001{Process} -rProcess{T900-999/E//I}
```

According to this example, when **vload.exe** detects an incoming record with an 001 tag that matches that of an existing record, it will merge the two records according to the rules specified by the **-rProcess** option.

Note: There does not need to be a one-to-one ratio of **-x** options to **-r** options. While you can define specific merge rules for each duplicate test, the **MergeString** lets you link multiple duplicate tests to the same merge rules.

The merge rules defined for a given **MergeString** are used to merge *only* the records that pass a duplicate test that calls that specific **MergeString**. For additional information on the **-x** option, see the section "[Determine the Criteria for a Duplicate Record \(-x\)](#)" and "[About the Format of the Duplicate Test Option](#)" in this chapter.

Note: The **MergeString** cannot contain spaces or brackets.

3.5.2.1.2 About the TagTest

The **TagTest** works in conjunction with the **MergeRule**. The **TagTest** tells the system which tags are to be merged by the specified **MergeRule**. The **TagTest** must use one of the following four formats:

- **T[tag]_f[first indicator]_s[second indicator]**
- **T[tag]_f[first indicator]**
- **T[tag]_s[second indicator]**
- **T[tag]**

Where . . .

- **[tag]** is the tag or range of tags to which you want to apply a given **MergeRule**.
For example . . .
 - ◆ **T100** - Applies the **MergeRule** to the 100 tag.
 - ◆ **T900-999** - Applies the **MergeRule** to the 900-999 tags.

Hint: To apply the **MergeRule** to the Leader, set the **[tag]** value to **LDR**.

- **[first indicator]** is the first indicator or range of first indicators associated with a given **[tag]** value. For example:
 - ◆ **T420_f1** - Applies the **MergeRule** to the 420 tag, where the first indicator is 1 (one).
 - ◆ **T420-480_f2** - Applies the **MergeRule** to the 420-480 tags where the first indicator is 2.
 - ◆ **T520_f1-3** - Applies the **MergeRule** to the 520 tag where the first indicator is any value 1-3.
- **[second indicator]** is the second indicator or range of second indicators associated with a give **[tag]** value. For example . . .
 - ◆ **T500-599_s1** - Applies the **MergeRule** to the 500-599 tags where the second indicator is 1 (one).
 - ◆ **T520_f3_s1-8** - Applies the **MergeRule** to the 520 tag where the first indicator is 3 and the second indicator is any value 1-8.

3.5.2.1.3 About the MergeRule

The **MergeRule** specifies the rule that Virtua will follow if it finds a duplicate. For each **TagTest**, you can specify any of the following rules. Additionally, you can use any of the following rules to define the **DefaultMergeRule**, which will apply to all tags for which a specific **TagTest** is not defined.

- **I** - For tags that satisfy the preceding **TagTest**, keep the tags from the incoming record and discard the tags from the existing record.
- **E** - For tags that satisfy the preceding **TagTest**, keep the tags from the existing record and discard the tags from the incoming record.
- **1IE** - For tags that satisfy the preceding **TagTest**, keep the tags from the incoming record. If a tag in the existing record satisfies the **TagTest** AND if a tag in the incoming record satisfies the **TagTest**, then the tag from the existing record

will be discarded. If no such tag exists in the incoming record, the tag in the existing record will be kept.

- **1EI** - For tags that satisfy the preceding **TagTest**, keep the tags from the existing record. If a tag in the incoming record satisfies the TagTest AND if a tag in the existing record satisfies the TagTest, then the tag from the incoming record will be discarded. If no such tag exists in the existing record, the tag in the incoming record will be kept.
- **bIE** - Keep ALL the tags from *either* record that satisfy the preceding **TagTest**. Such tags from the *incoming* record will be placed in the merged record prior to such tags with identical tag numbers from the existing record.
- **bEI** - Keep ALL the tags from *either* record that satisfy the preceding **TagTest**. Such tags from the *existing* record will be placed in the merged record prior to such tags with identical tag numbers from the incoming record.
- **D** - Discard all tags from either record that satisfy the preceding **TagTest**.
- **W** - For tags that satisfy the preceding **TagTest**, keep only the ones from the record having the *weaker* level of cataloging. If the record being loaded has the same level of cataloging as its duplicate match, the duplicate test will be ignored.

Note: This rule is for use with bibliographic records only.

- **S** - For tags that satisfy the preceding **TagTest**, keep only the ones from the record having the *stronger* level of cataloging. If the record being loaded has the same level of cataloging as its duplicate match, the duplicate test will be ignored.

Note: This rule is for use with bibliographic records only.

- **WI** - If the record being loaded has a different level of cataloging than the existing record, keep the tags from the record with the *weaker* level of cataloging. If the records have the *same* level of cataloging, keep the tags from the *incoming* record.

Note: This rule is for use with bibliographic records only.

- **WE** - If the record being loaded has a different level of cataloging than the existing record, keep the tags from the record with the *weaker* level of cataloging. If the records have the *same* level of cataloging, keep the tags from the *existing* record.

Note: This rule is for use with bibliographic records only.

- **SI** - If the record being loaded has a different level of cataloging than the existing record, keep the tags from the record with the *stronger* level of cataloging. If the records have the *same* level of cataloging, keep the tags from the *incoming* record.

Note: This rule is for use with bibliographic records only.

- **SE** - If the record being loaded has a different level of cataloging than the existing record, keep the tags from the record with the *stronger* level of cataloging. If the records have the *same* level of cataloging, keep tags from the *existing* record.

Note: This rule is for use with bibliographic records only.

3.5.3 Examples of Duplicate Tests and Merge Options

This section provides some examples of duplicate tests and merge options, along with how records will be merged according to the rules defined in the example.

Example 1:

-x001{Drop4xx} -rDrop4xx{T400-499/D//bEI}

According to this example, an incoming record that passes an 001 duplicate test will be merged by discarding all 4xx tags from both existing and incoming records. All other tags will be retained, but the tags in the existing record will be placed later in the merged record.

Example 2:

-x010{Example2} -rExample2{T001-699/1IE//D}

According to this example, an incoming record that passes an 010 duplicate test will be merged by keeping tags 001 - 699 from the incoming record and discarding all other tags.

Example 3:

-x020{OldFixed} -rOldFixed{T001-008/1EI//bIE}

According to this example, an incoming record that passes an 020 duplicate test will be merged by discarding tags 001, 002, 003, 004, 005, 006, 007 and 008 from the incoming record. All other tags will be retained, but the tags in the incoming record will be placed earlier in the merged record.

Example 4:

-x022{Example4} -x035{Example4} -rExample4{T500_s1/D//I}

According to this example, an incoming record that passes a duplicate test for both the 022 and 035 tags will replace the existing record after discarding any 500 tags that have 1 as the second indicator in either record.

Example 5:

-x020{Example5} -rExample5{T420_f2/D/T400-499/I//E}

According to this example, the merged record will contain all of the tags from the existing record except the 4xx tags. The 4xx tags from the existing record will be replaced by the 4xx tags from the incoming record except for the 420 tag with a first indicator 2, which will be discarded from both the incoming and existing records.

Example 6:

-x022{Example6} -rExample6{T520_f3_s1-8/bEI/T520_f3_s/bEI//E}

According to this example, the merged record will contain all of the tags from the existing record, plus any 520 tags from the incoming record that have indicators 3,1; 3,2; 3,3; 3,4; 3,5; 3,6; 3,7; 3,8; or 3,- which will appear later in the record than all such 520 tags from the existing record.

4. Setting Load Options in the Virtua Client

Using the Load Options feature in the Virtua client, you can set load options that will be used when you save or import records via the Virtua client. This chapter describes how you can set load options in the Virtua client. Options that you set on the Load Options window in the client will be applied when you use the following features:

- Save Record to Database function after opening a record in the MARC Editor.
- Import Records tool.
- Import to Purchase Requests tool.

Important: The load options that you set on the Load Options window are persistent. Before using any of the above functions, it is important to check the load option settings so that you do not inadvertently save or import records with inappropriate load values.

This chapter covers the following topics:

- ⇒ [Accessing the Load Options Window](#)
- ⇒ [Working with Load Options on the Load Options Window](#)
- ⇒ [Working with Duplicate Tests and Merge Load Options in the Client](#)

4.1 Accessing the Load Options Window

To display the Load Options window in the client,

1. On the Virtua main menu, click Options.
2. Choose Load Options.

The Load Options window appears (*Figure 4-1*).

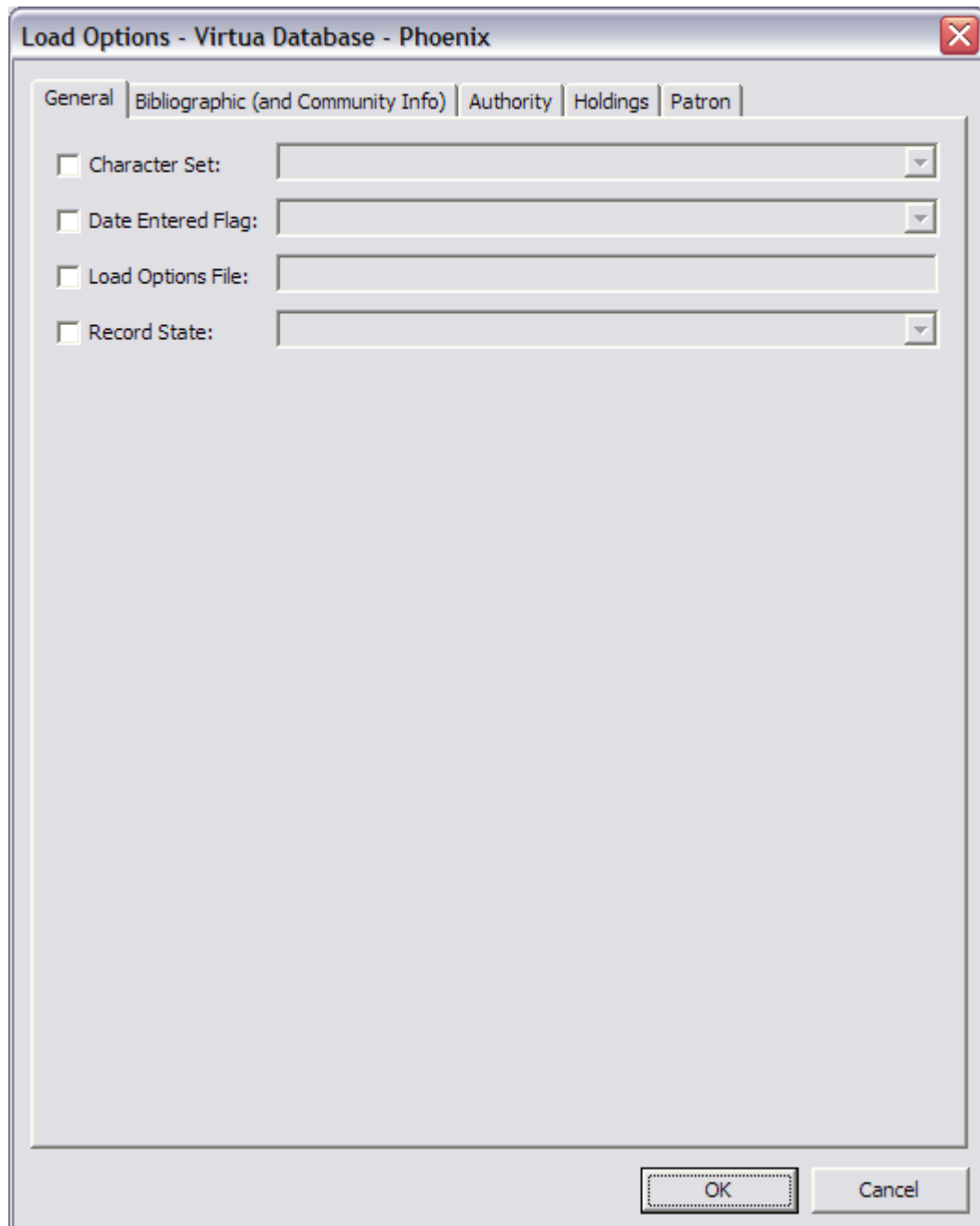


Figure 4-1. Load Options Window

The Load Options window displays five tabs. The General tab contains options that apply to ALL records regardless of record type. The Bibliographic, Authority, Holdings, and Patron tabs contain options specific to those record types.

To choose or set a specific value for an option on the Load Options window,

1. Select the check box for the option.

The corresponding field becomes active.

2. Set the option as desired. The following sections provide detailed instructions for setting specific options.

4.2 Working with Load Options on the Load Options Window

This section describes each load option that appears on the Load Options window (*Figure 4-1*). For information on how the settings on the Load Options window correspond to options available for **vload.exe** and the subfields in the 999 LOADOPT tag, see the appendix "[Available Load Options](#)" in this guide.

Important: Once you specify load options in the client, Virtua will use them for any record that is brought into the MARC Editor and saved to the database.

Keep in mind that some load options have default values that Virtua will use unless you specify otherwise. For options that have default values you want to use, you do NOT need to set options on the Load Options window. For a list of default load options, see the appendix "[Available Load Options](#)" in this user's guide.

4.2.1 Working with the General Tab

The General tab of the Load Options window (*Figure 4-2*) displays load options that apply to ALL types of records.

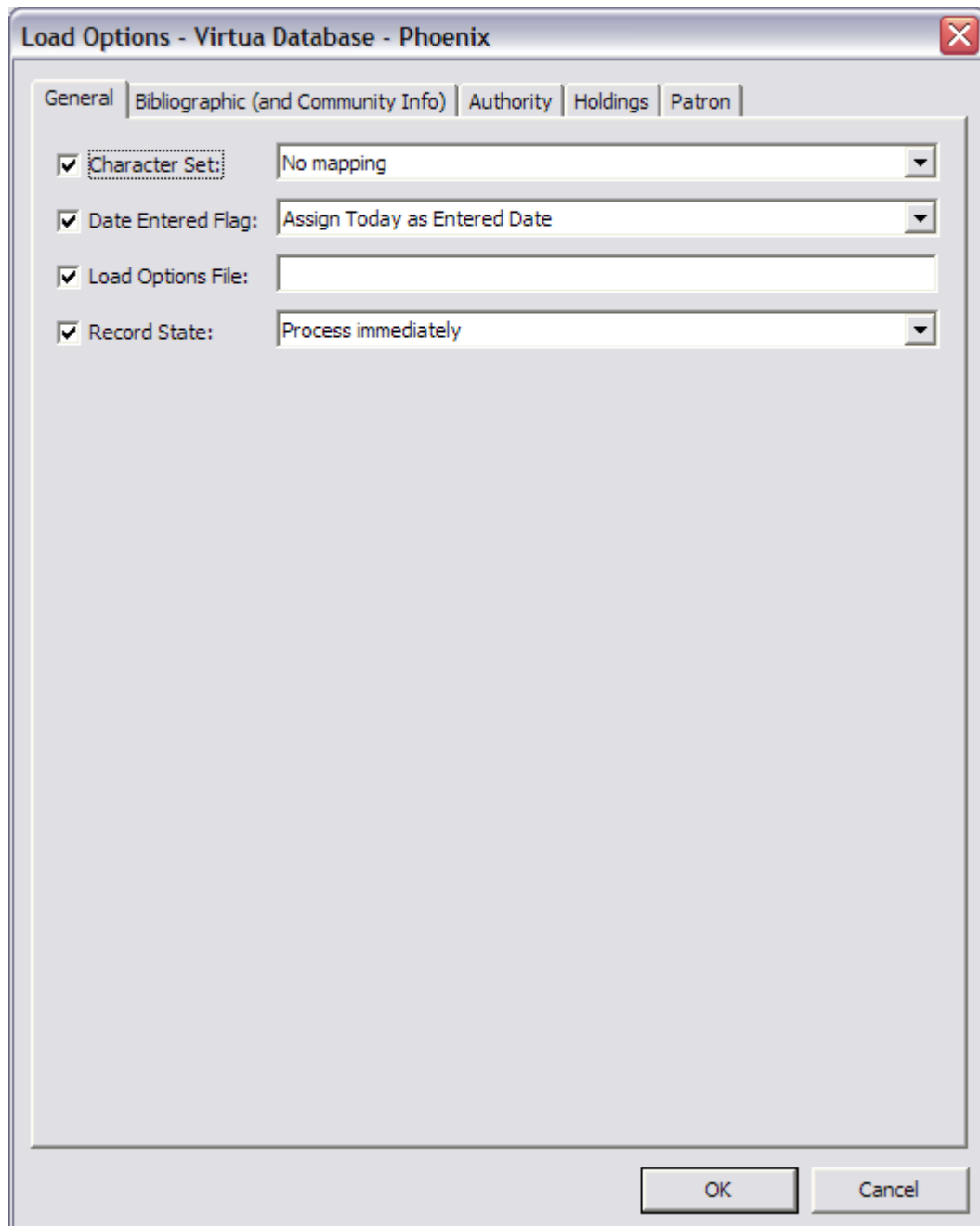


Figure 4-2. Load Options Window - General Tab

Four options are available on the General tab:

- **Character Set** - Lets you specify the character set of the incoming record(s). You can choose from the following character sets:
 - ◆ UTF-8 (no mapping is necessary)

- ◆ MARC 21 (MARC-8)
- ◆ ANSI extended for UCL + ANSI 8 + ANSI Z39.47
- ◆ Europa-3
- ◆ Windows Latin1
- ◆ PC-8
- ◆ Windows ALA
- ◆ Windows Arabic
- ◆ Windows Hebrew
- ◆ Windows Cyrillic
- ◆ Windows Latin 2
- ◆ ISO 6937/2
- ◆ CP 850 (Microsoft Code Page)
- ◆ ISO 6937/2 + Arabic
- ◆ ISO 6937/2 + Greek
- ◆ Big5 (variant used by Tamkang University)
- ◆ ANSI-8 + Hebrew
- ◆ UTF-8 character sets in which some separate diacritics need to be combined.
- ◆ ANSI-8 (Swiss version)
- ◆ GBK (encoding of CJK characters)
- ◆ TIS620 Classic Thai
- ◆ CCCII (Chinese Character Code for Information Interchange)
- ◆ GB 18030 (Chinese National Standard GB 18030-2000)
- ◆ Big5-HKSCS (Hong Kong Special Character Set)

Important: For incoming records, the Virtua *server* will read this setting and map any non-UTF-8 character set to UTF-8, the character set of the Virtua database. When you use the Import Records tool or Import to Purchase Requests tool and specify the character set of incoming records, it is the Virtua *client* that does the mapping. So to avoid “double” mapping, when you use the Import Records tool or Import to Purchase Requests tool, disable the Character Set option in the Load Options window.

- **Date Entered Flag** - Lets you specify which date to use as the Entered Date for the record(s). You can choose either . . .
 - ◆ Assign Today as Entered Date
 - OR-
 - ◆ Keep Record's Entered Date
- **Load Options File** - Lets you type the name of a file, which is saved on the server, that contains preset load options (for details, see the section “[Working with Load Options Files](#)”).

Note: If you specify only the filename of the load options file, Virtua looks for the file in the directory from which you started **psdriver.exe**. If you want Virtua to look for the file in a different directory, specify the full path and filename.

- **Record State** - Lets you choose a state, as defined by your library in the Cataloging Record States parameter, in which you want to save the incoming record(s).

4.2.2 Working with the Bibliographic (and Community Info) Tab

The Bibliographic (and Community Info) tab of the Load Options window (*Figure 4-3*) lets you set load options related to bibliographic records, including duplicate tests, merge rules, and a specific action for non-duplicate records.

The screenshot shows the 'Load Options - Virtua Library' dialog box with the 'Bibliographic (and Community Info)' tab selected. The dialog has a title bar with a close button (X) in the top right corner. Below the title bar are five tabs: 'General', 'Bibliographic (and Community Info)', 'Authority', 'Holdings', and 'Patron'. The 'Bibliographic' tab is active, displaying various configuration options. Each option consists of a checked checkbox followed by a label and a control element (text field or dropdown menu). At the bottom of the dialog are 'OK' and 'Cancel' buttons.

Load Options - Virtua Library

General | **Bibliographic (and Community Info)** | Authority | Holdings | Patron

- ☒ Database Action: Load Item Bib Parent
- ☒ Create Items: Process data in item tags regardless of location
- ☒ Require Virtua 949 Label
- ☒ Shelf Location:
- ☒ Item Class:
- ☒ Duplicate Tests:

Add Edit Delete
- ☒ Merges:

Add Edit Delete
- ☒ Non-Dupl. Action: Do not save non-duplicate incoming records.
- ☒ Split FRBR
- ☒ Load Options File:
- ☒ Validation: Validate and save
- ☒ Move 880 tag data to associated tags
- ☒ Local Options File:

Save To File...

OK Cancel

Figure 4-3. Load Options Window - Bibliographic Tab

You can set the following options on the Bibliographic tab:

- **Database Action** - Lets you tell Virtua which action to take when loading records. You can choose either . . .
 - ◆ Load items and parent records.
 - OR-
 - ◆ Create items only.
- **Create Items/Block Item Creation** - Lets you specify the rules that govern how item records are created from information in the 949 VIRTUAITEM tag of the incoming parent bibliographic records.
- **Require Virtua 949 Label** - Lets you specify whether a Virtua label is required in the 949 tag subfield \$A.
- **Shelf Location** - Lets you specify a shelf location for loaded items.
- **Item Class** - Lets you specify the item class for loaded items.
- **Duplicate Tests** - Lets you specify duplicate tests for incoming records based on particular tags.
- **Merges** - Lets you create rules for merging incoming records with existing records when duplicate records are detected.
- **Non-Duplicate Action** - Lets you specify which action to take against incoming records that are NOT duplicates of records in the database.


Note: For details about Duplicate Tests, Merges, and Non-Duplicate Action, see the section "[Working with Duplicate Tests and Merge Load Options in the Client](#)" in this guide.

- **FRBR Splitting** - Lets you specify whether incoming records are split into three FRBR records: Work, Expression, and Manifestation.
- **Load Options File** - Lets you type the name of a file, which is saved on the server, that contains preset options (for details, see the section "[Working with Load Options Files](#)"). The options defined in the file that you specify here will be used for bibliographic record *only*.

Note: If you specify only the filename of the load options file, Virtua looks for the file in the directory from which you started **psdriver.exe**. If you want Virtua to look for the file in a different directory, specify the full path and filename.

- **Validation** - Lets you specify whether incoming bibliographic records are validated and saved to the database, *only* validated, or not validated at all.
- **Move 880 Tag Data to Associated Tags** - Lets you control whether, when a bibliographic record containing an 880 field is saved, Virtua will move the 880 tag data and use it to replace equivalent data in an associated tag. The original data in the associated tag will be placed in a 921 tag. The 921 tag will have a first indicator 1, which means this tag has former "romanized" data that was associated with an 880 tag, and a subfield 6, which will contain the equivalent tag number (245, 260, etc.).

Note: Moving the original data to a 921 field allows the retention of the original data, but prevents the data from being used for any indexing purposes unless "user-defined" keyword indexing is applied to the 921 field.

- **Local Options File** - Lets you type the name and location or use the  button to browse to a load options file on your workstation or local network (for details on the file contents, see the section "[Working with Load Options Files](#)"). The options defined in the file that you specify here will be used for bibliographic records *only*.
- **Save To File** - Click this button to save the current load options configuration as a file. After you save the file, you can keep it on your workstation to use with the Local Options File setting, or you can upload it to your server so the file is available to all workstations via the Load Options File setting.

4.2.3 Working with the Authority Tab

The Authority tab of the Load Options window (Figure 4-4) lets you set options that will be used for loading authority records, including duplicate tests, merge rules, and a specific action for non-duplicate records.

The screenshot shows a dialog box titled "Load Options - Virtua Library" with a close button (X) in the top right corner. The dialog has five tabs: "General", "Bibliographic (and Community Info)", "Authority" (which is selected), "Holdings", and "Patron".

Under the "Authority" tab, there are several sections:

- Database Action:** A checked checkbox followed by a dropdown menu showing "ESU_Merge".
- Blind References:** A checked checkbox followed by a dropdown menu showing "Load blind references".
- Heading Type:** A checked checkbox followed by a dropdown menu showing "Type is determined by tag and other data in the record."
- Duplicate Tests:** A checked checkbox followed by an empty text area. Below the text area are three buttons: "Add", "Edit", and "Delete".
- Merges:** A checked checkbox followed by an empty text area. Below the text area are three buttons: "Add", "Edit", and "Delete".
- Non-Dupl. Action:** A checked checkbox followed by a dropdown menu showing "Do not save non-duplicate incoming records."
- Load Options File:** A checked checkbox followed by an empty text field.
- Local Options File:** A checked checkbox followed by an empty text field and a folder icon button. Below this is a "Save To File..." button.

At the bottom right of the dialog are "OK" and "Cancel" buttons.

Figure 4-4. Load Options Window - Authority Tab

You can set the following load options for authority records:

- **Database Action** - Lets you tell Virtua which action to take against incoming records. You can choose either . . .
 - ♦ **ESU_Merge** - Merge/If a 1xx/4xx conflict is found, put record in Error state.
-OR-
 - ♦ **ESM_Over** - Save to database/If a conflict is found, override Error state.


Note:

- ESU_Merge lets you change the bibliographic headings in *provisional* authority records without making them permanent authority records.
 - ESM_Over lets you save an authority record to the database after using the MARC Editor to switch, or flip, 1xx and 4xx headings in the record OR to change a 5xx tag to a 4xx tag.
- **Blind References** - Lets you specify whether blind references, i.e., main headings that have no associated bibliographic records, are loaded.
 - **Heading Type** - Lets you specify whether the heading type is determined by tag and other data in the record or whether the type is automatically considered to be *subject*, in which case, a name authority will be loaded as a subject authority. **Hint:** To load records as both name and subject heading types, load the records twice, once with the value for this option set to 0 and once with it set to 1.
 - **Duplicate Tests** - Lets you specify duplicate tests for incoming records based on particular tags.
 - **Merges** - Lets you create rules for merging incoming records with existing records when duplicate records are detected.
 - **Non-Duplicate Action** - Lets you specify which action to take against incoming records that are NOT duplicates of records in the database.

Note: For details about Duplicate Tests, Merges, and Non-Duplicate Action, see the section "[Working with Duplicate Tests and Merge Load Options in the Client](#)" in this guide.

- **Load Options File** - Lets you type the name of a file, which is saved on the server, that contains preset options (for details, see the section "[Working with Load Options Files](#)"). The options defined in the file that you specify here will be used for authority record *only*.

Note: If you specify only the filename of the load options file, Virtua looks for the file in the directory from which you started **psdriver.exe**. If you want Virtua to look for the file in a different directory, specify the full path and filename.

- **Local Options File** - Lets you type the name and location or use the  button to browse to a load options file on your workstation or local network (for details on the file contents, see the section “[Working with Load Options Files](#)”). The options defined in the file that you specify here will be used for authority records *only*.
- **Save To File** - Click this button to save the current load options configuration as a file. After you save the file, you can keep it on your workstation to use with the Local Options File setting, or you can upload it to your server so the file is available to all workstations via the Load Options File setting.

4.2.4 Working with the Holdings Tab

The Holdings tab of the Load Options window (*Figure 4-5*) lets you set options that will be used for loading holdings records.

Load Options - Virtua Library

General | Bibliographic (and Community Info) | Authority | **Holdings** | Patron

☒ Database Action: Load Item Holdings Parent

☒ Create Items: Process data in item tags regardless of location

☒ Virtua 949 Label: A Virtua Label is not needed

☒ Hold Migration Flag: No holdings migration

☒ Item Class:

☒ Load Options File:


☒ Local Options File: Save To File...

OK Cancel

Figure 4-5. Load Options Window - Holdings Tab

You can set the following options on the Holdings tab:

- **Database Action** - Lets you tell Virtua which action to take against a file of parent holdings records:
 - ◆ Load item and/or parent holdings records.
 - ◆ Create item records only; do not process associated parent holdings records.
 - ◆ Load item and/or parent holdings records AND check in issues (as defined in tag 949 subfield \$9).
 - ◆ Create item records and check in issues only; do not process associated parent holdings records.
- **Create Items/Block Item Creation Flag** - Lets you specify the rules by which item records are created from information in the 949 VIRTUAITEM tag of the incoming parent holdings records.
- **Virtua 949 Label** - Lets you specify whether a Virtua label is needed in the 949 tag subfield \$a.
- **Hold Migration Flag** - Lets you specify whether holdings records are being migrated from the classic VTLS system to Virtua.
- **Item Class** - Lets you specify the item class for loaded items.
- **Load Options File** - Lets you type the name of a file, which is saved on the server, that contains preset options (for details, see the section “[Working with Load Options Files](#)”). The options defined in the file that you specify here will be used for holdings records *only*.

Note: If you specify only the filename of the load options file, Virtua looks for the file in the directory from which you started **psdriver.exe**. If you want Virtua to look for the file in a different directory, specify the full path and filename.
- **Local Options File** - Lets you type the name and location or use the  button to browse to a load options file on your workstation or local network (for details on the file contents, see the section “[Working with Load Options Files](#)”). The options defined in the file that you specify here will be used for holdings records *only*.
- **Save To File** - Click this button to save the current load options configuration as a file. After you save the file, you can keep it on your workstation to use with the Local Options File setting, or you can upload it to your server so the file is available to all workstations via the Load Options File setting.

4.2.5 Working with the Patron Tab

The Patron tab on the Load Options window ([Figure 4-6](#)) offers a variety of load options related to patron records, including duplicate tests, merge rules, and a specific action for non-duplicate records.

The screenshot shows a dialog box titled "Load Options - Virtua Library" with a close button (X) in the top right corner. The dialog has five tabs: "General", "Bibliographic (and Community Info)", "Authority", "Holdings", and "Patron". The "Patron" tab is currently selected. The dialog contains several options, each with a checked checkbox:

- Date Pattern:** A dropdown menu showing "YYYYMMDDHH24MI".
- Check Date:** A dropdown menu showing "No date check".
- Expiration Date:** An empty text input field.
- Delete Date:** An empty text input field.
- Duplicate Tests:** A large empty text area with "Add", "Edit", and "Delete" buttons below it.
- Merges:** A large empty text area with "Add", "Edit", and "Delete" buttons below it.
- Non-Dupl. Action:** A dropdown menu showing "Do not save non-duplicate incoming records."
- Load Options File:** An empty text input field.
- Local Options File:** A checked checkbox next to an empty text input field, with a folder icon button to its right.

At the bottom left, there is a "Save To File..." button. At the bottom right, there are "OK" and "Cancel" buttons.

Figure 4-6. Load Options Window - Patron Tab


You can set the following options on the Patron tab:

- **Date Pattern** - Lets you specify the pattern that the system will use to interpret the dates found in the 042 tag subfields \$a, \$b, and \$c of the incoming records. Currently, Virtua supports only ONE date pattern.
- **Check Date** - Lets you specify whether to check the transaction date in the 005 tag of the incoming record and use it as a reason to override the loading of the record. If you enable the date check, the transaction date in the incoming record is checked against the last transaction date in the database record. If the transaction date in the incoming record is older than the date in the database record, then the record is not loaded.
- **Expiration Date** - Lets you specify an expiration date, which Virtua will insert in tag 042 subfield \$b if such a date does not already exist in the incoming record. The date must be in the format specified in the Date Pattern field, i.e., YYYYMMDDHHMM.
- **Delete Date** - Lets you specify a delete date, which Virtua will insert in tag 042 subfield \$c if such a date does not already exist in the incoming record. The date must be in the format specified in the Date Pattern field, i.e., YYYYMMDDHHMM.
- **Duplicate Tests** - Lets you specify duplicate tests for incoming records based on particular tags. For details about special considerations for patron merges, see the section "[A Note about Duplicate Tests for Patron Records](#)" in this chapter.
- **Merges** - Lets you create rules for merging incoming records with existing records when duplicate records are detected.
- **Non-Duplicate Action** - Lets you specify which action to take against incoming records that are NOT duplicates of records in the database.

Note: For details about Duplicate Tests, Merges, and Non-Duplicate Action, see the section "[Working with Duplicate Tests and Merge Load Options in the Client](#)" in this guide.

- **Load Options File** - Lets you specify the name of a file, which is saved on the server, that contains preset options (for details, see the section "[Working with Load Options Files](#)"). The options defined in the file that you specify here will be used for patron records *only*.

Note: If you specify only the filename of the load options file, Virtua looks for the file in the directory from which you started **psdriver.exe**. If you want Virtua to look for the file in a different directory, specify the full path and filename.

- **Local Options File** - Lets you type the name and location or use the  button to browse to a load options file on your workstation or local network (for details on the file contents, see the section “[Working with Load Options Files](#)”). The options defined in the file that you specify here will be used for patron records *only*.
- **Save To File** - Click this button to save the current load options configuration as a file. After you save the file, you can keep it on your workstation to use with the Local Options File setting, or you can upload it to your server so the file is available to all workstations via the Load Options File setting.

4.3 Working with Duplicate Tests and Merge Load Options in the Client

The Bibliographic, Authority, and Patron tabs of the Load Options window offer tools that give you an easy way to specify complex load options that tell Virtua to perform testing for duplicate records, take a specific action if duplicates are found, and, if desired, merge two records according to a specified rule. The tabs also provide a setting that determines how Virtua handles non-duplicate incoming records.

The Duplicate Tests option lets you specify . . .

- Tags for which Virtua will perform duplicate testing.
-AND-
- The action that you want Virtua to take if it finds duplicates.

One action you might want Virtua to take is to merge the two duplicate records. The Merges option works in conjunction with Duplicate Tests to let you specify criteria by which two records will be merged to create a new record in the database. All the tags in the new record come from the two source records:

- The **incoming record**, which comes from an input file or is sent to **psdriver.exe** from the Virtua client.
- The **existing record**, which is the one currently in the database.

The following sections discuss [Duplicate Tests](#) and [Merges](#) options and provide step-by-step instructions for setting each one. Although these load options are available for bibliographic, authority, and patron records, we will focus our discussion on bibliographic records. All aspects of duplicate tests and merges are the same for patron and authority records except for the choice of dominant tags for the duplicate tests.

Important: Keep in mind that once you specify duplicate tests and merge load options, the client will use them for any record that is brought into the MARC Editor and saved to the database.

4.3.1 Specifying One or More Duplicate Tests

To specify one or more duplicate tests for incoming records,

1. On the Bibliographic, Authority, or Patron tab of the Load Options window, click the Duplicate Tests check box.
2. Click the Add button.

The Add Duplicate Test window appears (*Figure 4-7*).

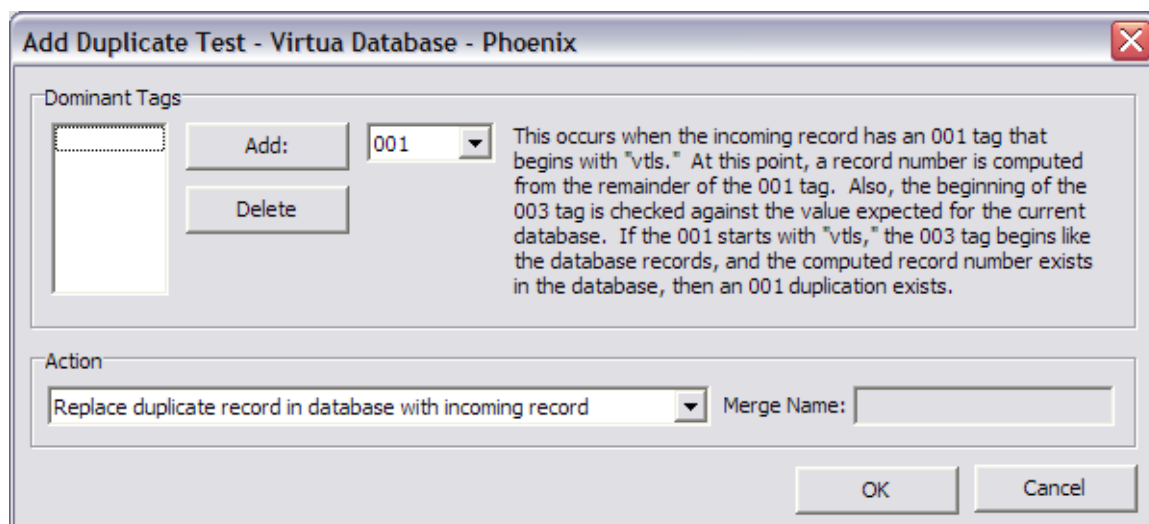


Figure 4-7. Add Duplicate Test Window

The Add Duplicate Test window displays three main sections:

- **Dominant Tags** - Specifies the tags to be used for duplicate testing. When you select a tag from the drop-down list, the display area on the right provides an explanation of the use of each tag in a duplicate test.
- **Action** - Tells Virtua what action to take when duplicates are detected.
- **Merge Name** - Allows you to type a merging string name that will be linked to the duplicate test option. **Note:** This field is active only when the Action is set to *Merge duplicate record...*

3. In the Dominant Tags area, select each tag that you want Virtua to use as criteria to detect duplicate records. You will be able to choose different tags depending on the record type (bibliographic, authority, or patron).
4. Click the Add button to add the tag to the list box. You must add at least ONE tag here.
5. In the Action area, select the action that you want Virtua to take when it detects duplicate records. The choices are to . . .

- Replace the duplicate record in the database with the incoming record.

Note: If a new MARC bibliographic or authority record is added that contains a heading that differs from an existing permanent authority heading in *ending punctuation only*, the new heading will be considered identical to the existing heading, and the new record will replace the existing record in the database.

- Ignore the incoming record and leave the database unchanged.
- Save the incoming record in Error state for later processing.
- Remove the duplicate record in the database and discard the incoming record.
- Merge the duplicate record in the database with the incoming record.
- Keep the record with the weaker cataloging level.*
- Keep the record with the stronger cataloging level.*

* When either of these options are used, you MUST select the Non-Dupl. Action check box on the [Working with the Bibliographic \(and Community Info\) Tab](#) of the Load Options window and configure the corresponding setting to “Do not save non-duplicate incoming records.”

6. If you selected the Action *Merge the duplicate record in the database with the incoming record*, type a Merge Name, which is simply a user-defined linking string. For any other Action, skip this step and proceed to the next step.

Note: If you want duplicate records that meet the criteria for different duplicate tests to be merged in different ways, you can specify multiple Duplicate Tests, specify the Action as "merge" for each one, and assign a different Merge Name to each.

7. Click the OK button.

Your selection appears on the Load Options window ([Figure 4-8](#)).

Load Options - Virtua Database - Phoenix

General | Bibliographic (and Community Info) | Authority | Holdings | Patron

☐ Database Action:

☐ Create Items:

☐ Require Virtua 949 Label

☐ Shelf Location:

☐ Item Class:

☒ Duplicate Tests: 001 - Replace old record

☐ Merges:

☐ Non-Dupl. Action:

☐ Split FRBR

☐ Load Options File:

☐ Validation:

☐ Move 880 tag data to associated tags

Figure 4-8. Load Options Window - Displaying a Duplicate Test

If you specified the Action *Merge the duplicate record in the database with the incoming record*, you must now set the Merges option, as described in the following section.

4.3.2 Specifying the Merges Option

To specify the Merges option for incoming bibliographic records,

1. On the Bibliographic, Authority, or Patron tab of the Load Options window, make sure your Duplicate Tests option includes a Merge Name.
2. Click the Merges check box.
3. Click the Add button.

The Add Merge window appears (*Figure 4-9*).

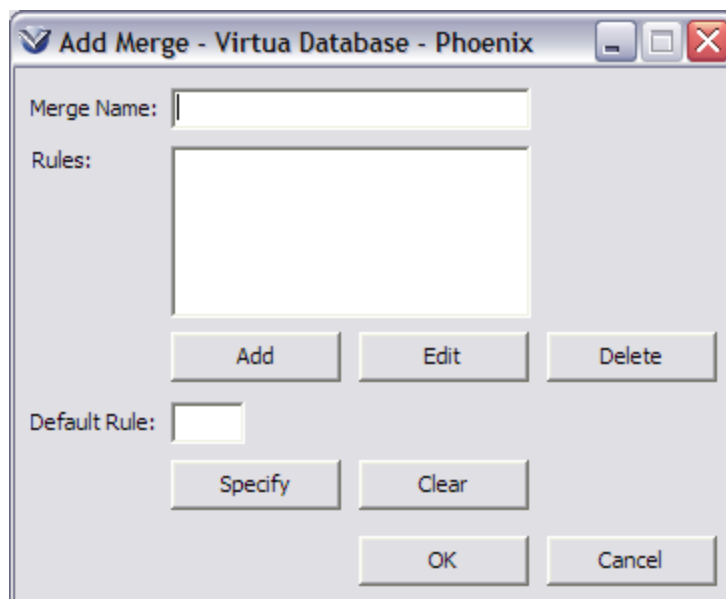


Figure 4-9. Add Merge Window

4. Type the Merge Name that you used in the Duplicate Tests option. **Hint:** If there are multiple names to choose from, choose the Merge Name that you want linked to this particular merge; then repeat the following steps for every other Merge Name.
5. Click the Add button under the Rules list box.

The Add Rule window appears (*Figure 4-10*), displaying four fields, each with its own on-screen description.

Add Rule - Virtua Database - Phoenix

Tag Range: Enter 3 digit tag number or tag number range separated by dash. For example: 100 or 300-400

First Indicator Range: (Optional) - Enter indicator character (including space if desired) or range in the same way. For example 1 or 3-4

Second Indicator Range: (Optional) - Same as above.

Merge Rule: Keep tags from the incoming record that satisfy the preceding range(s) and discard tags from the existing record that satisfy the ranges(s).

OK Cancel

Figure 4-10. Add Rule Window

6. On the Add Rule window, use the appropriate text fields to type the . . .
 - Tag or tag range to which the merge rule will apply.
 - First indicator or first indicator range to which the merge applies.
 - Second indicator or second indicator range to which the merge applies.
7. Select the Merge Rule, which tells Virtua how to conduct the merge for the specified tag.
8. Click the OK button.

The Add Merge window reappears (*Figure 4-11*), displaying the rules that you selected.

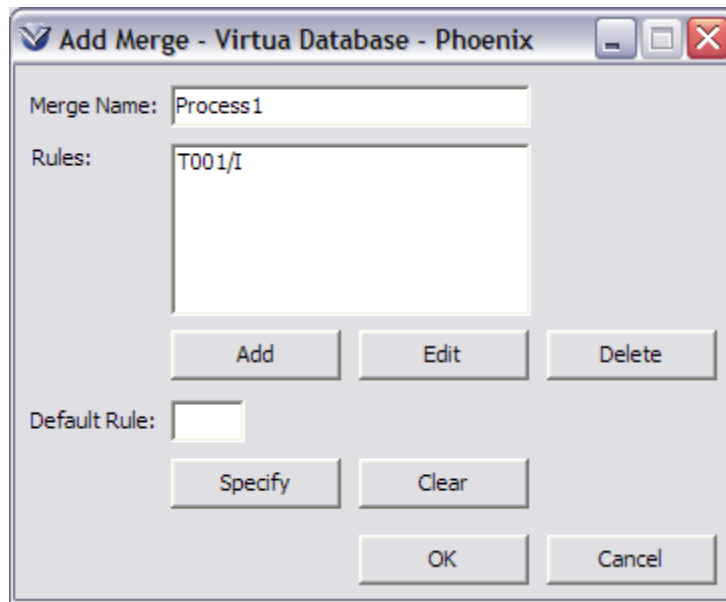


Figure 4-11. Add Merge Window - Displaying a Merge Rule

9. Repeat the process (steps 6-8) for each tag for which you want to specify a merge rule.
10. When you have entered all desired tags, set the Default Rule. To do this, click the Specify button.

The Specify Default Rule window appears (Figure 4-12).

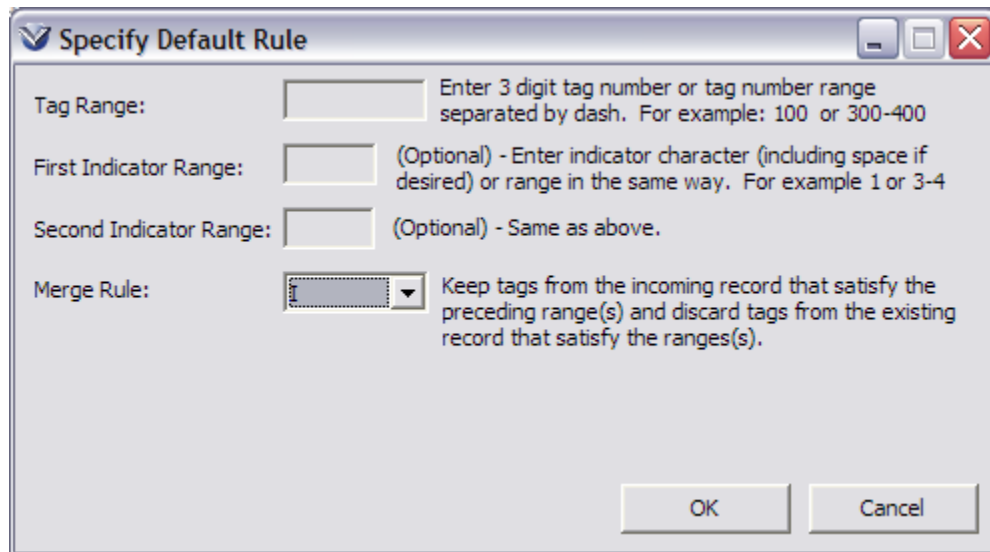


Figure 4-12. Specify Default Rule Window

The Default Rule is the rule that applies to any tag not listed in the Rules list box on the Add Merge window.

11. On the Specify Default Rule window, select the Merge Rule, which is the only field that is enabled.
12. Click the OK button.

Hint: If you want ALL tags in the duplicate records to be merged according to the *same rule*, you do not need to specify any tags for the Rules list box. Simply provide the Merge Name and specify a Default Rule.

4.3.3 Specifying the Non-duplicate Action

The Non-duplicate Action load option tells Virtua how to handle incoming records that are not found to be duplicates according to any duplicate test.

To specify a Non-duplicate Action for incoming bibliographic records,

1. On the Bibliographic, Authority, or Patron tab of the Load Options window, click the Non-duplicate Action check box.
2. Select an action from the drop-down list. The choices are:
 - Do not save non-duplicate incoming records.
 - Save non-duplicate incoming records as new records. (*Default*)
 - Save non-duplicate incoming records as Error state records.

4.3.4 A Note about Duplicate Tests for Patron Records

Virtua's default duplicate tests for patron records specify that the *existing* record should be overwritten when a match is found on the 001, 015, or 016 tag. In the Virtua client, these defaults apply only when you batch load records via the Import Records tool or when you modify an existing record via the MARC Editor. These defaults do NOT apply when you create a NEW record via the MARC Editor.

Rather, when you create a new record in the client, if the 015 or 016 tag of the new record matches that of an existing record, the new record is saved in Error state (i.e., the existing record is not overwritten) by default. The new, Error state record will have a different patron ID (001 tag) than the existing record. If you want Virtua to handle

duplicate NEW records in this way, you do NOT need to set Duplicate Tests options for patron records.

When you *modify* an existing patron record via the MARC Editor in the client, Virtua uses the default duplicate tests for patron records (i.e., the existing record is overwritten when a match is found on the 001, 015, or 016 tag) when you save the record to the database.

For *modified* records to be saved in Error state based on a patron barcode (015 or 016) match, you must set your Duplicate Tests load options in the client as follows:

- **001** - Replace old record.
- **015** - Put new record in error state.
- **016** - Put new record in error state.

Your Load Options must be set *exactly* as they appear above, in the *same* order shown. In this way, if the 015 or 016 tag of an existing record is modified to match that of an existing record, the modified record will be saved in error state. When your Duplicate Test options are set as shown above, Virtua will handle new and modified records the same way.

To summarize . . .

- If you want Virtua to save NEW patron records in Error state based on a patron barcode match, you do not need to set Duplicate Test load options.
- If you want Virtua to save new AND modified patron records in Error state based on a patron barcode match, set your Duplicate Test options as listed above.
- If you want Virtua to handle duplicate patron records in some other way, set your Duplicate Options as appropriate.

5. Batch Loading MARC Format Records in the Virtua Client

After setting your load options on the Load Options window, you can easily load records interactively by using one of several Import Records tools offered by the Virtua client. With these tools, you can import records from various sources such as OCLC Connexion, classic VTLS software, a vendor's website, or any local files containing records saved in 2709 MARC communications format.

Note: Depending on your system, loading records in the Virtua client may take a long time. For this reason, we recommend that you limit your batch loads to no more than 5,000 records. If you need to load more than 5,000 records, we recommend that you create multiple input files, and load them separately.

This chapter covers the following topics:

- ⇒ [Working with the Import Records Tool](#)
- ⇒ [Working with the Import to Purchase Requests Tool](#)
- ⇒ [Working with the Import MARC Orders/Requests Tool](#)

5.1 Working with the Import Records Tool

The Import Records tool in the client offers the following features:

- The ability to load any records saved in 2709 MARC communications format.
- The ability to specify the character set of the incoming file of records so that the Virtua client can convert it to UTF-8 (Unicode), the character set of the Virtua database.

Warning: Before using the Import Records Tool and to avoid any conflicts, disable the character set field on the Load Options window. Because the Virtua *server* reads the character set value in the Load Options window and the Virtua *client* reads the character set value in the Import Records tool, it is important to prevent “double” mapping that would put records in error state.


- Automatic validation and error checking of records upon saving to the database. Additionally, you can configure Virtua to alert you if it detects an error or invalid record.
- The option to validate records with the database *without* saving them to the database.
- The option to open records in the MARC Editor as they are received so you can review and edit them before sending them to the database.
- The ability to work in tandem with the load options that you set via the Load Options function on the Options menu (see the section "[Setting Load Options in the Virtua Client](#)" in this guide).
- Support for performing default duplicate tests on incoming records and if a match is found, for replacing the existing database record.
- Persistent settings.

5.1.1 Accessing the Import Records Tool

To access the Import Records tool,

1. On the Virtua menu bar, click File, choose Records, and then choose Import Records.

-OR-

On the Cataloging toolbar, click Import Records .

The Import Records window appears (*Figure 5-1*).



Figure 5-1. Import Records Window

5.1.2 Aspects of the Import Records Window

The Import Records window has the following aspects:

- **Statistical Counters** - During the import process, display the counts for the following:
 - ◆ **Total Records Found in Source File** - Indicates the total number of records found so far in the source file. The combined total of the other counters equals the value of this counter, which is saved (along with the values of the other counters) and which can be reset to zero only by use of the Reset Button.
 - ◆ **Failed Validity Checks** - Specifies the number of records that have failed validity testing. Virtua makes validity tests if two conditions are met: 1) The selected Destination Option for the imported records is Save Records to Database and 2) The Validate When Saving to Database option is checked on the Cataloging Options window.
 - ◆ **Database Errors** - Indicates how many records that were sent to the database could not be processed due to errors.
 - ◆ **Number of Records Successfully Loaded** - Specifies how many records were imported without validity check failure or database errors.
- **Command Buttons** - Let you perform the following commands:
 - ◆ **Reset** - Resets the import process and associated counters. You can click the button only when the import process is idle. When you click the button, a warning appears, informing you of the reset procedure and allowing you to cancel the reset operation.
 - ◆ **Configure** - Displays the Import Configuration window, where you can set various values such as the character set of the incoming records.
 - ◆ **Start** - Begins the importing process. Virtua saves each record automatically to the current connection, and if there is no connection, the Start button is disabled.
 - ◆ **Stop** - Stops the import process. Stopping the import only keeps Virtua from monitoring the import files. Import files are unaffected. Any records loaded into the MARC Editor are left intact.
 - ◆ **Close** - Closes the dialog box and the import session. If the session is still active, the system prompts you to stop the import process before closing the dialog box.

5.1.3 Configuring the Import Records Tool

Before you can use the Import Records tool, you need to configure it.

To configure the Import Records tool,

1. On the Import Records window ([Figure 5-1](#)), click the Configure button.

The Import Configuration window appears ([Figure 5-2](#)).

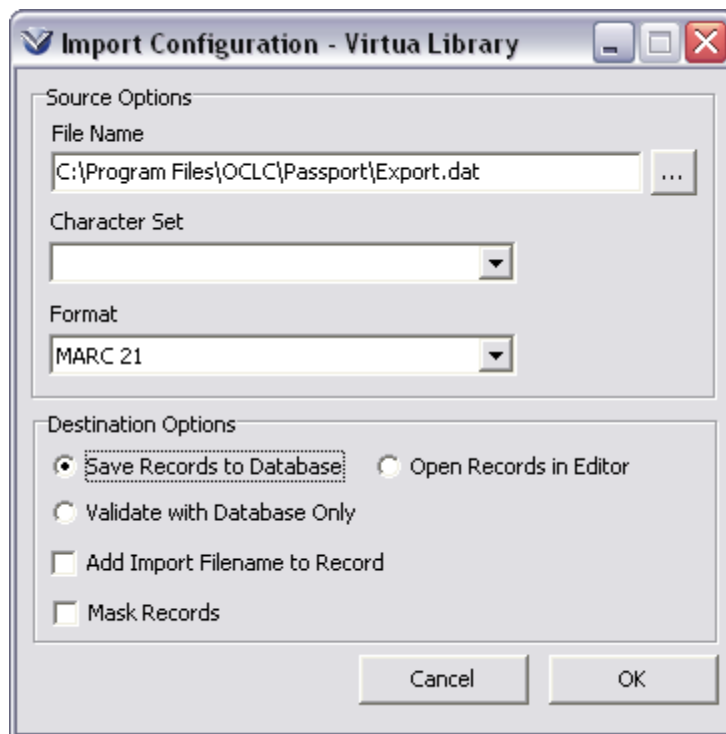


Figure 5-2. Import Configuration Window

2. In the File Name text box, type the name and location of the file containing the 2709 MARC communications format records you are importing. A browse button next to the field lets you find the file via the standard Windows dialog box.

Note: If you download records using OCLC Connexion or any other similar utility, you will find it useful to enable validation via the Cataloging Options so that Virtua not only imports those records but also validates them as you receive them.

3. In the Character Set drop-down list, choose the character set of the input file.

Important:

- The character set you specify *must* match the character set you specified when exporting the records from the utility or vendor. For example, if the file is from OCLC, the character set specified in this field must match the character set specified in the Record Characteristics window of the OCLC Connexion client.
- If you do not know the character set of the records you are importing, you *must* contact the cataloging utility, vendor, or other source supplying the records and verify the *character set* (not the record format).
- The Virtua database supports the UTF-8 (Unicode) character set. Once you specify the character set of the source file, Virtua will automatically convert that character set to UTF-8.
- To avoid “double” mapping by the client and server, disable the character set option in the client’s Load Options window.

4. If you are an ISSN Centre user, in the Format drop-down list, select ISSN MARC as the format of the records being imported.
5. In the Destination Options section of the Import Configuration dialog box, select ONE of the following radio buttons:
 - **Save Records to Database** - Records will be saved to the Virtua database that is associated with your current server connection.
 - **Open Records In MARC Editor** - The MARC Editor will display a “page” tab for each imported record. In this way, you can select and edit each record as a “local” record before you save it to the database or elsewhere.
 - **Validate with Database Only** – The server will validate the imported records with the database, but the records will not be saved to the database.
6. Select the Add Import Filename to Record check box to add the import filename and record offset to each imported record. The filename will be saved in tag 039 subfield \$w and the offset in tag 039 subfield \$x.

Note:

- The Add Import Filename to Record feature applies to bibliographic records ONLY.
- You can use the filename as a search filter on the Search by State window.

7. Select the Mask Records check box if you want to mask the incoming bibliographic and item records. Masked records will be hidden from users who are not expressly given permission to view them (see the *Virtua Profiler/Global Settings User’s Guide* for information on user permissions).
8. Click the OK button.

Your configuration settings are saved and will be used when you begin the import process. Any errors found during the import process will be displayed in a diagnostic message window along with the invalid record displayed in the MARC Editor.

Note: The diagnostic messages for errors encountered during import will all appear in a single Messages window. To close the Messages window, click the Reset button on the Import Records window.

5.1.4 Running the Import Records Tool

After you have configured the Import Records tool, you can begin loading the records.

To load the records,

- On the Import Records dialog box, click the Start button.

The animation at the top of the window indicates Virtua is waiting for records. If the animation ceases, the import tool is inactive. The Import Records dialog box keeps track of the number of records successfully loaded, records that fail validity checks, and the number of database errors ([Figure 5-3](#)).



Figure 5-3. Import Records Tool in Action

5.1.4.1 Stopping the Import

To stop the load process at any time,

- Click the Stop button.

The import process stops. The state of the import is saved. This means that when you stop the import and even exit Virtua, you can restart the import later beginning where you left off within the source file.

5.1.4.2 Using the Reset Button

If you stop the load process, the Reset button becomes enabled.

Do NOT press the Reset button until all records in the export file have been cleared or deleted.

Pressing the Reset button resets the load process and the counter in the Import Records window. If you reset the load process *before* the export file is cleared or deleted, the records remaining in the file will be loaded a second time (see warning message below).

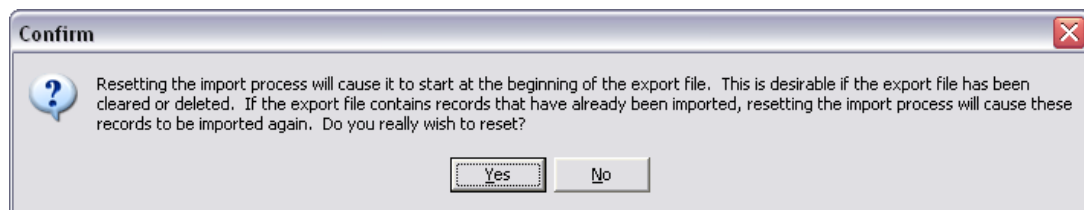


Figure 5-4. Reset Warning - Import Records Function

5.1.5 Availability of Log Files

When you start an import process for a particular source file, the system will create two log files: Summary Log File and Error Log File. The path and name of the files is the same as the path and name of the import source file except for the extension. Note that if log files already exist with this name in the same location, the existing log files will be replaced with new log files.

The log files report the warnings, errors, and statistics of the diagnostics received during the import process up to the time that the import process is stopped.

5.1.6 Summary Log File

The summary log file shows all data and statistics of the diagnostics received during the entire import process. The file extension of the log file will be **.log** for a normal import or **.logv** for a validate-only import (for details about the Validation load option, see the section [“Working with the Bibliographic and Community Info Tab”](#)).

Format of the File:

Import Source File: <source file path and name>

Date/time Import Started: <date/time import started in YYYY/MM/DD 24hh/mm format>

Contents of the File:

- Global Statistics
 - ◆ Number of Records Processed:
 - ◆ Number of Records Successfully Saved to/Validated in DB:
 - ◆ Number of Records Failed by Vericat:
 - ◆ Number of Records Not Saved to/Not Valid according to DB:
 - ◆ Number of Saved/Validated Records without Diagnostics:
 - ◆ Number of not rejected (warning flagged) records:
- Successfully Imported
 - ◆ Bib IDs of successfully imported records
- Error State
 - ◆ Bib IDs of records put into Error state
- Duplicates
 - ◆ Bib IDs of records being imported that had existing duplicates in the database
- Diagnostics Received
 - ◆ Warning messages about headings

5.1.7 Error Log File

The error log file shows the diagnostic information associated with each MARC record to allow easy association of errors with particular records. The file extension of the log file will be **.err** for a normal import or **.errv** for a validate-only import (for details about the Validation load option, see the section [“Working with the Bibliographic and Community Info Tab”](#)).

Format of the File:

Import Source File: <source file path and name>

Date/time Import Started: <date/time import started in YYYY/MM/DD 24hh/mm format>

Contents of the File:

[For each MARC record with diagnostics]:

- ISSN: <ISSN value (022\$a) of the record>
- Record Number: <ordinal number of the record>
- List of all diagnostics, warnings, or errors received for the record in the format:
<Validation ID number> <diagnostic message>
- MARC record in human-readable format <separator line>

5.2 Working with the Import Records to Purchase Request List Tool

The Purchase Request List is a list of titles requested by staff or patrons for addition to the library's collection. You can access the list from the Acquisitions menu on the Virtua menu bar. For details about using the Purchase Request List, see the *Getting Started with Acquisitions Guide*.

Virtua provides a tool called Import Records to Purchase Request List that will import records, automatically create purchase requests using these records, and add those purchase requests to the Purchase Request List. The tool lets you specify a number of criteria for the import process: source file, indications for adding the record to the list, name of the vendor, purchase request category, and whether the records should be masked in the OPAC.

Note: As with the Import Records tool, when you use the Import Records to Purchase Request List option, Virtua will load records into the database using the options specified on the Load Options window.

To import records to the Purchase Request List,

1. On the Virtua menu bar, click File, and then click Records.
2. Choose Import Records to Purchase Request List.

The Import To Purchase Requests window appears (*Figure 5-5*).

Figure 5-5. Import To Purchase Requests Window

3. In the File Name field, type the path and name of the file of records that you want to import. A browse button next to the field lets you find the file via the standard Windows dialog box.
4. In the Character Set drop-down list, choose the character set of the records to be imported.

Important:

- The character set you specify *must* match the character set you specified when exporting the records from the utility or vendor. For example, if the file is from OCLC, the character set specified in this field must match the character set specified in the Record Characteristics window of the OCLC Connexion client.
- If you do not know the character set of the records you are importing, you *must* contact the cataloging utility, vendor, or other source supplying the records and verify the *character set* (not the record format).

- The Virtua database supports the UTF-8 (Unicode) character set. Once you specify the character set of the source file, Virtua will automatically convert that character set to UTF-8.
- To avoid “double” mapping by the client and server, disable the character set option in the client’s Load Options window.

5. Select at least ONE check box indicating the criteria for adding the record to the Purchase Request List . . .
 - **Is New** - Incoming records will be added to the Purchase Request List only if they are new.
 - **Is Discarded** - Incoming records will be added to the Purchase Request List only if they would normally be discarded.
 - **Overwrites Existing Record** - Incoming records will be added to the Purchase Request List only if they overwrite an existing record.
 - **Is Placed in Error State** - Incoming records will be added to the Purchase Request List only if they are placed in Error state.
6. In the Vendor drop-down list, choose a vendor name associated with the purchase requests (*optional*).
7. In the Category drop-down list, choose a purchase request category in which the title will be placed.

Note: Your library can define title-related categories for the Purchase Request List in the Purchase Request Categories parameter on the Acquisitions tab in the Virtua Profiler.

8. Select or deselect the Mask Records check box, if desired. Selecting this check box tells the server to mask the records in the database.

Note: The default selection of the Mask Records check box depends on the setting of the Mask New Records option on the Other Options tab of the Cataloging Options window (see the *Virtua Cataloging User’s Guide*).

9. Click the Import button.

The status of the importation is reflected in the Status area on the window (for a description of the counts, see the section "[Aspects of the Import Records Window](#)"). The records will be added to the database and the titles will be added to the Purchase Request List according to your configuration.

Note: If an imported record goes into Error state, it will still appear in the Purchase Request List, but the title will be accompanied by the message: “Unavailable. Error in the bibliographic record.”

5.3 Working with the Import MARC Orders/Requests Tool

MARC orders and MARC requests are MARC records from a vendor’s website that are embedded with purchase order data. Once Virtua receives these records, the software is programmed to convert the data in the records to Virtua purchase orders:

- MARC orders are MARC records that contain data for items that *have been ordered* through the vendor’s website.
- MARC requests are MARC records that contain order data for items that have been requested through the vendor’s website but *need to be ordered* through Virtua.

Note: A vendor’s website will likely offer several options with regard to MARC orders and/or MARC requests. The vendor may allow you to...

- Download a file of MARC records to your PC.
- Transfer a file of MARC records to the vendor’s FTP server for downloading.
- Transfer a file of MARC records *directly* to your Virtua FTP server.

For details about EDIFACT orders and MARC orders/requests in particular, see the *Getting Started with Acquisitions Guide*.

If you choose to download a file of MARC order/request records from a vendor’s website to your PC, you can manually import the records from your PC into your Virtua database. Virtua provides a tool called Import MARC Orders/Requests for importing the records in batch. The tool lets you specify a number of criteria for the import process: source file, character set of the incoming file, vendor name, type of records, file destination, and whether the records should be masked in the OPAC.

Note: As with the Import Records tool, when you use the Import MARC Orders/Requests option, Virtua will load records into the database using the options specified on the Load Options window.

To import MARC Order/Request records,

1. On the Virtua menu bar, click File, and then click Records.
2. Choose Import MARC Orders/Requests.

The Import MARC Orders/Requests window appears (*Figure 5-6*).

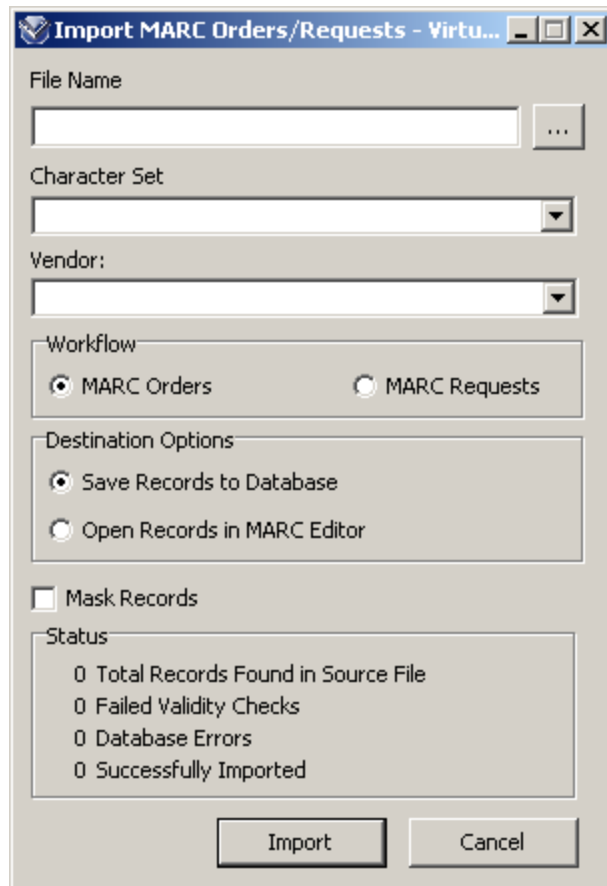


Figure 5-6. Import MARC Orders/Requests Window

3. In the File Name field, type the path and name of the file of records that you want to import. A browse button next to the field lets you find the file via the standard Windows dialog box.
4. In the Character Set field, choose the character set of the records to be imported.

Note:

- Virtua supports the UTF-8 (Unicode) character set. Once you specify the character set as a source option, Virtua will automatically convert that character set to UTF-8.
- A character set value may already be set on the Load Options window. To prevent "double" mapping, make sure that the character set value in *either* the Load Options window or the Import To Purchase Requests window is set to UTF-8.

5. In the Vendor drop-down list, choose the vendor associated with the MARC orders/requests (*optional*).
6. In the Workflow area, select the radio button that specifies the type of MARC records being imported. The setting here will override the Preferred Workflow setting on the EDIFACT tab in the vendor record editor (see the section "EDIFACT Tab" in the *Getting Started with Acquisitions Guide*).

Note:

- If you select MARC Order, Virtua will give the order a status of Ordered when it is created. Virtua assumes that the vendor has already received the order.
- If you select MARC Request, Virtua will give the order a status of Needs Approval or Approved, depending on how the vendor record is set up. Virtua assumes that the order has not been sent to the vendor.

7. In the Destination Options area, select the radio button that specifies whether the record is to be saved to the database (where orders will be created after you click the Import button) or displayed in the MARC Editor (which will open after you click the Import button).
8. Edit the Mask Records check box, if desired. Selecting this check box tells the server to mask the records in the database.

Note: The selection of the Mask Records check box by default will depend on the setting of the Mask New Records option on the Other Options tab of the Cataloging Options window (see the *Virtua Cataloging User's Guide*).

9. Click the Import button.

The status of the importation is reflected in the Status area on the window (for a description of the counts, see the section "[Aspects of the Import Records Window](#)").

Note: If an imported record goes into Error state, a generic error message will be accessible on the MARC Orders/Requests tab of the EDIFACT Tracker (see the *Getting Started with Acquisitions Guide* for details).

6. Appendix A - Available Load Options

This appendix presents all record-loading options in a table format. The column headings in the table are described below.

- **vload.exe** - The command-line option that you use to call the load option when running **vload.exe**. The option is hyperlinked to a more detailed description elsewhere in the guide, if available.
- **Client** - The default label that is used to identify the option in the Virtua client. For options that are NOT available in the client, this column contains the text *N/A*. The option is hyperlinked to a more detailed description elsewhere in the guide, if available.
- **999 LOADOPT Subfield** - The subfield value that is used to identify the load option in the 999 LOADOPT tag. For options that are NOT available in the LOADOPT tag, this column contains the text *N/A*. For details on the 999 LOADOPT tag, see the section "[About the 999 LOADOPT Tag](#)" in this guide.
- **Valid Values** - The values that you can specify for the option.
 - ◆ Values in **bold** represent values that you specify on the command line when using **vload.exe**.
 - ◆ The text provides a description of each value. For load options that are available in the client, the text is similar to that which appears in the client by default.
 - ◆ The LOADOPT tag will contain the value used by **vload.exe**.
- **Default Value** - The default value for the load option. If there is no default value for a given option, this column contains the text *N/A*. For details on default load options, see the section "[About Default Load Option Values](#)" in this guide.
- **Record Types** - The types of records with which you can use each load option.

vload.exe	Client	999 LOADOPT Subfield	Valid Values	Default Value	Record Types
-c	Database Action	\$c	<ul style="list-style-type: none"> • 0 - No Action • 1 - Insert/Replace • 2 - Insert/Replace • 3 - Delete • 5 - Merge incoming record with existing record. (If a 1xx/4xx conflict is found, the record is put in Error state.) • 6 - Save incoming record to the database. (If a 1xx/4xx conflict is found, the system “overrides” it so that the record does NOT get put in Error state.) • 10 - Load items and/or parent bib records. • 11 - Load items and/or parent holdings records. • 12 - Create items only. Do not process associated parent bib records. • 13 - Load items and/or parent holdings records AND check in serials issues. • 16 - Load incoming records and create items as specified in the 949 VIRTUAITEM tag, even if matching encoding levels are encountered. 	2 (N/A for the Client)	<ul style="list-style-type: none"> • 0, 2, and 3 are valid for ALL records types (VLOAD ONLY but may be specified in the Client using the “Load Options File” option.) • 1 is valid for <i>holdings</i> records, especially <i>new</i> holdings records. • 5 and 6 are valid only for <i>authority</i> records, where 0 is the default. • 10, 12, and 16 are valid only for <i>bibliographic</i> records, where 0 is the default. • 11, 13, 22 and 23 are valid only for <i>holdings</i> records, where 0 is the

vload.exe	Client	999 LOADOPT Subfield	Valid Values	Default Value	Record Types
			<ul style="list-style-type: none"> • 22 - Create items only. Do not process associated parent holdings records • 23 - Create items AND check in serials issues only. Do not process associated parent holdings records. 		default.
-@	Load Options File	\$@	The name of a load options file (e.g., loadopts.txt). If there is a conflict, the load options specified in the file override command-line load options.	N/A	ALL
-9	Use 999	N/A	<ul style="list-style-type: none"> • 0 - Do not use the options specified in the 999 LOADOPT tag. • 1 - Use the options specified in the 999 LOADOPT tag. 	0	ALL
-b	Record State	\$b	<ul style="list-style-type: none"> • 0 - Process Immediately -OR- • The numeric ID of any record state defined in the Virtua Profiler. 	0	ALL
-d	Character Set	\$d	<ul style="list-style-type: none"> • 2 - UTF-8 (no mapping) • 10 -MARC-8 • 11 - ANSI 8 + ANSI Z39.47 • 12 – Europa-3 • 13 - Windows Latin1 • 14 - PC-8 	2	ALL

vload.exe	Client	999 LOADOPT Subfield	Valid Values	Default Value	Record Types
			<ul style="list-style-type: none"> • 15 - Windows ALA • 16 - Windows Arabic • 17 - Windows Hebrew • 18 - Windows Cyrillic • 19 - Windows Latin 2 • 20 - ISO 6937/2 • 21 - Microsoft CP 850 • 22 - ISO 6937/2 + Arabic • 23 - ISO 6937/2 + Greek • 24 - Big5 (Tamkang) • 25 - ANSI-8 + Hebrew • 26 - UTF-8 character sets in which some separate diacritics need to be combined. • 28 - ANSI-8 (Swiss version) • 30 - GBK (encoding of CJK characters) • 31 - TIS620 Classic Thai • 32 - ISO 5426 (International Serials Data System interchange character set) • 33 - CCCII • 34 - GB 18030 • 35 - Big5-HKSCS 		

vload.exe	Client	999 LOADOPT Subfield	Valid Values	Default Value	Record Types
-1	Date Entered Flag	\$l	<ul style="list-style-type: none"> • 0 - Assign today's date as the Entered Date • 1 - Keep the "Entered Date" already in the incoming record, where it is stored in the first six positions of the 008 tag. 	0	ALL
-O	N/A	N/A	<ul style="list-style-type: none"> • 0 - Do not use the Record Status to determine deletion of the existing record • 1 - Delete the existing record if the Record Status of the incoming record contains "d" in the leader position 05 (delete status flag). • 2 - Deletes the existing record if the Record Status of the incoming record contains "d" in the leader position 05, or discards the incoming record if its Record Status Flag is set to "x" in the leader position 05. Note that this option value applies ONLY to authority records. 	1	ALL

vload.exe	Client	999 LOADOPT Subfield	Valid Values	Default Value	Record Types
-X	N/A	N/A	Lets you delete all records of a specified record ID type. The record IDs must be contained in a file that is also specified at the command line. Record type IDs to delete: <ul style="list-style-type: none"> • 101 - bibliographic • 102 - authority • 104 - holdings • 105 - patron 	N/A	ALL
-E	N/A	N/A	<ul style="list-style-type: none"> • 0 - No action. • 1 - Existing authority records causing a 4xx tag conflict with records being loaded are automatically merged with the incoming records. 	0	Authority
-G	Blind References	\$G	<ul style="list-style-type: none"> • 0 - Load blind references. • 1 - Do not load blind references. 	0	Authority
-H	Heading Type	\$H	<ul style="list-style-type: none"> • 0 - Type is determined by tag and other data in the record. • 1 - Type is determined to be "subject" regardless of other data. 	0	Authority

vload.exe	Client	999 LOADOPT Subfield	Valid Values	Default Value	Record Types
-N	N/A	N/A	<ul style="list-style-type: none"> • 0 - Deletion of the record is allowed. • 1 - Deletion of the record is blocked. <p>Note: Used with -c3 option (VLOAD ONLY) and intended for use during batch loading only.</p>	0	Authority
-R	N/A	N/A	<ul style="list-style-type: none"> • 0 - Creation of tag 010 requires a DLC 003 tag. • 1 - Creation of tag 010 is not affected by the contents of 003 tag. 	0	Authority
-t	N/A	\$t	<ul style="list-style-type: none"> • 0 - Local tags in existing records are not preserved. • 1 - Local tags in existing records are preserved. 	0	Authority

vload.exe	Client	999 LOADOPT Subfield	Valid Values	Default Value	Record Types
-x[tag] {n}	Duplicate Tests	\$x	<p>Valid values for n are . . .</p> <ul style="list-style-type: none"> • 0 - Do nothing with new record. • 1 - Replace database record with new record. • 2 - Save new record in Error state. • 3 - Discard new record and delete database record. • 4 - Keep database record and add the library symbol specified with -U. • 5 - Keep the new record if it has a less thorough level of cataloging; otherwise, discard. • 6 - Keep the new record if it has a more thorough level of cataloging; otherwise, discard. • merge_name - Merge the incoming record with the duplicate counterpart. The value for merge_name should match the string that you specify for the -r option. <p>Note: -xMRG is a special test to support merging of existing bibliographic records and is ignored unless an -m value is also provided.</p>	<p>Authority: -x001{1} -x010{1} -x1xx{1}</p> <p>Bib: -x001{1} -x035{1} -xMRG{1}</p> <p>Patron: -x001{1} -x015{1} -x016{1} -x035{1}</p>	Authority, Bibliographic, and Patron

vload.exe	Client	999 LOADOPT Subfield	Valid Values	Default Value	Record Types
-s	Non-Dupl Action	\$s	<ul style="list-style-type: none"> • 0 - Do nothing. • 1 - Save to database as new record. • 2 - Save to database in Error state. 	1	Authority, Bibliographic, and Patron
-3	N/A	\$3	<ul style="list-style-type: none"> • 0 - Do not add special 035. • 1 - Add special 035 (TMUA option) • 2 - Add special 035 (UCL option) • 3 - Add special 035 (NLW option) • 4 - Add special 035 (BRO option) 	0	Bibliographic
-C	N/A	\$C	<ul style="list-style-type: none"> • 0 - Do not compare the 005 tag dates. • 1 - Compare the 005 tag dates to determine whether to load the record; only records with newer dates than existing records are loaded. 	0	Bibliographic
-U	N/A	N/A	Any valid library symbol as defined in the Virtua Profiler.	N/A	Bibliographic
-u	N/A	N/A	<ul style="list-style-type: none"> • 0 - Do not use special FRBR load. • 1 - Use special FRBR load. 	0	Bibliographic

vload.exe	Client	999 LOADOPT Subfield	Valid Values	Default Value	Record Types
-f	FRBR Splitting	\$f	<ul style="list-style-type: none"> • 0 - Do not split the bib record. • 1 - Split the bib record into work, expression, and manifestation 	0	Bibliographic
-h	N/A	N/A	<ul style="list-style-type: none"> • 0 - 949 tags are not constructed from 950 and 998 or 852 tags. • 1 - 949 tags are constructed from the 950 and 998 tags. • 2 - 949 tags are constructed from the 852 tag 	0	Bibliographic
-v	Validation	\$v	<ul style="list-style-type: none"> • 0 = Validate and save • 1 = Validate only • 2 = No validation before save 	0	Bibliographic
-w	N/A	N/A	<p>A filename</p> <p>Note: This option provides a filename for the source of loaded records. The filename is stored in \$w of tag 039. (If the record goes into Error state, its position in the file will be stored in \$x of tag 039.)</p>	No filename is provided, so \$w and \$x of the 039 tag are not created.	Bibliographic
-V	N/A	N/A	<ul style="list-style-type: none"> • 0 = Disable MARC order processing • >0 = Value must match an Innovative database vendor ID 	0	Bibliographic

vload.exe	Client	999 LOADOPT Subfield	Valid Values	Default Value	Record Types
-W	N/A	N/A	<ul style="list-style-type: none"> • 1 = Process MARC order information as an order • 2 = Process MARC order information as a request 	1	Bibliographic
-m	N/A	\$m	<p>The name of a file that contains two tab-delimited Innovative IDs per line. The ID on the left corresponds to the incoming record to be absorbed. The ID on the right corresponds to the existing record that will absorb or be merged with the incoming record.</p> <p>Note: To merge rather than absorb a record, you can include an – xMRG {} option to specify merge rules in the same manner used for merging new records with existing records, as described for -x[tag]{n}.</p>	N/A	Bibliographic
-8	Move 880 Tag Data to Associated Tags	N/A	<ul style="list-style-type: none"> • 0 - Do NOT move 880 tag data to associated tags. • 1 - Move 880 tag data to associated tags. 	0	Bibliographic
-Y	N/A	N/A	<ul style="list-style-type: none"> • 0 - Incoming records are in MARC21 format • 1 - Incoming records are in UNIMARC format. 	0	Bibliographic

vload.exe	Client	999 LOADOPT Subfield	Valid Values	Default Value	Record Types
-k	Block Item Creation	\$k	<ul style="list-style-type: none"> • 0 - Do not block item creation. Create items from information in the 949 tag of the parent bib record. • 1 - Block item creation if items already exist with a location that matches the main part of the location code in the 949 \$D being processed. If items exist, but only for <i>another</i> main location, create one item (the first item) for the main location. • 2 - Block item creation if items already exist with a location that matches the entire location code in the 949 \$D being processed. If items exist, but only for <i>another</i> location, create one item (the first item) for the main location. 	0	Bibliographic, Holdings
-r	Merging String	\$r	The text that will be used to merge records when the -x option is set to MergeString .	N/A	Bibliographic, Authority, and Patron
-B	Shelf Location	N/A	<ul style="list-style-type: none"> • When using vload.exe, any valid Virtua location code. • When using the client, any location name. 	N/A	Bibliographic, Holdings
-D	Item Class	\$D	Any valid Virtua item class code.	N/A	Bibliographic, Holdings

vload.exe	Client	999 LOADOPT Subfield	Valid Values	Default Value	Record Types
-L	Virtua 949 Label	\$L	<ul style="list-style-type: none"> • 0 - A Virtua Label is not needed. • 1 - A Virtua Label is needed. 	1	Bibliographic, Holdings
-p	N/A	N/A	A text string that contains translation pairs such that any existing item subfields with a code that matches the first character in the string will be converted into a subfield with a code using the second character in the string. Likewise for a subfield match with the third character being converted into the fourth character in the string, and so on.	N/A	Bibliographic, Holdings
-q	N/A	N/A	A text string that contains translation pairs such that any existing item call number subfields with a code that matches the first character in the string will be converted into a subfield with a code using the second character in the string. Likewise for a subfield match with the third character being converted into the fourth character in the string and so on.	N/A	Bibliographic, Holdings
-z	N/A	N/A	<ul style="list-style-type: none"> • 0 - Do not create items from 949 tags that contain subfield \$z. • 1 - Item record creation is not affected by the existence of subfield \$z in the 949 tags. 	0	Bibliographic, Holdings

vload.exe	Client	999 LOADOPT Subfield	Valid Values	Default Value	Record Types
-I	Check Date	\$I	<ul style="list-style-type: none"> • 0 - Do not check the transaction date. • 2 - Check the last transaction date of the incoming record against the last transaction date in the database record. Do not load if the incoming record date is the earlier date. 	0	Patron
-J	Expiration Date	\$J	A date specified in the default format indicated by the Date Pattern Flag (-o).	N/A	Patron
-K	Delete Date	\$K	A date specified in the default format indicated by the Date Pattern Flag (-o).	N/A	Patron
-o	Date Pattern	\$o	<ul style="list-style-type: none"> • 0 - YYYYMMDDHH24MI • 1 - MMDDYYYYHH24MI 	0	Patron
-P	N/A	\$p	<ul style="list-style-type: none"> • 0 - Do not add default password. • 1 - Add default password. 	0	Patron
-T	N/A	N/A	<ul style="list-style-type: none"> • 0 - Preserve the dates in tag 042 of the incoming record. • 1 - Recalculate the dates in tag 042 of the incoming record according to parameter settings in the Virtua Profiler. 	0	Patron
-i, -a			See the <i>Consortium User's Guide</i> .		Patron

7. Appendix B - Specifications for the 949 VIRTUAITEM Tag

This appendix provides a list of the subfields that you can use in the 949 VIRTUAITEM tag in a holdings or bibliographic record. For each subfield, there is a description, information about the value you can specify, and the default value, if any, along with the rules for using each subfield. Use the following key to decipher the codes in the Rules column.

Rules Key

- **NR** - Not repeatable
- **R** - Repeatable
- **R1** - Repeatable once
- **M** - Mandatory
- **A** - May be mandatory
- **O** - Optional

Subfield	Rules	Description	Value	Default Value if Subfield Not Defined
\$A	NR - A	Tag label	<p>The label that identifies the tag as containing item creation information. It should contain the text <i>VIRTUAITEM</i> in uppercase letters.</p> <p>Note: To support deprecated 949 tag formats, the system also allows for the tag data to consist of the text <i>VTL\$ITEM</i> or <i>VTL\$ITEM</i> followed by a space. Additionally, Virtua allows a 949 tag with either no \$A or a \$A whose data starts with numeric characters to be processed as a VIRTUAITEM 949 tag.</p>	N/A
\$B	NR - A	Virtua parent control number information	<p>The system ID of the record to which the item is to be linked. This value <i>must</i> contain the Virtua parent ID Type (101 for an item linked directly to a bibliographic record or 104 for an item linked to a holdings record) AND the Virtua parent ID. For example . . .</p> <ul style="list-style-type: none"> • \$B101 12345 (for an item linked to a bibliographic record) • \$B104 9876543 (for an item linked to a holdings record) <p>Note: This field is mandatory only if the parent ID cannot be determined from other information in the record.</p>	N/A - If this subfield is not defined, the control number is taken from the associated parent record.
\$C	NR - A	Virtua bibliographic control number information	<p>The Bib-ID of the bibliographic record to which the item is to be linked, either directly OR through a holdings record. This value <i>must</i> contain the Virtua Bib ID Type (101) and the Virtua Bib-ID. (see the example for subfield \$B above).</p> <p>Note: This field is mandatory only if the Bib-ID cannot be determined from other information in the record. Subfield \$C can be used only if subfield \$B is also used. If subfield \$B contains a Virtua Bib-ID and subfield \$C is also used, the two values must match.</p>	N/A - If this subfield is not defined, the Bib-ID is taken from the parent bibliographic record.

Subfield	Rules	Description	Value	Default Value if Subfield Not Defined
\$D	NR - O	Owning location	The location code for the permanent or owning location of the item. This subfield can contain only digits. If you are loading holdings records and this subfield value is not available and the linking \$8 is valid, Virtua will try to get the location from the 852 tag with the same linkage subfield 8. Note: The value you use is assumed to be a valid location code that is stored in the Location Names parameter. The system will not check this.	10000
\$F	NR - O	Copy number	A number that indicates a particular copy when multiple copies exist.	0
\$G	NR - O	Temporary shelf location	The code for the temporary or shelf location of the item. This subfield can contain only digits.	10000
\$H	NR - O	At Location Until date	The date that the item is to be moved from the temporary shelf location (\$H) and sent back to the permanent location (\$D). The date must be in the form YYYYMMDDHHMM, where HHMM is optional.	No date if \$G is not defined or is the same as \$D. Otherwise, the load date.
\$K	NR - A	Loan period	The number of <i>days</i> the item can circulate when overriding the loan period in the matrix. This subfield can contain only digits, with a maximum of 9999. Note: This subfield is mandatory if the third position of subfield \$t (override circulation matrix) is set to Y (yes).	0
\$L	NR - O	Circulation count	The total number of times that the item has circulated. This subfield can contain only digits, with a maximum of 999999999. This count does not include the reserve, in-house, and usage counts.	0
\$M	NR - O	Temporary circulation count	The total number of times that the item has circulated since some fixed date, typically the date specified in subfield \$N. This subfield can contain only digits, with a maximum of 9999.	0

Subfield	Rules	Description	Value	Default Value if Subfield Not Defined
\$N	NR - O	Temporary circulation count date	The date that the temporary circulation count in subfield \$M was reset. This subfield must contain a valid date in the form YYYYMMDDHHMM, where HHMM is optional. Note: This subfield can be included in the tag only if subfield \$M is also included.	Load date
\$O	NR - O	Reserve count	The total number of times that an item has circulated while on reserve. This subfield can contain only digits, with a maximum of 9999.	0
\$P	NR - O	Temporary reserve count	The total number of times that the item has circulated while on reserve since some fixed date, typically the date specified in subfield \$Q. This subfield can contain only digits, with a maximum of 9999.	0
\$Q	NR - O	Temporary reserve count date	The date that the temporary reserve count was reset. This subfield must contain a valid date in the form YYYYMMDDHHMM, where HHMM optional. Note: This subfield can be included in the tag only if subfield \$P is also included.	Load date
\$R	NR - O	In-house circulation count	The total number of times that the item has circulated as an in-house check-out. This subfield can contain only digits, with a maximum of 9999.	0
\$S	NR - O	Temporary in-house circulation count	The total number of times that the item has circulated as an in-house check-out since some fixed date, typically the date specified in subfield \$T. This subfield can contain only digits, with a maximum of 9999.	0
\$T	NR - O	Temporary in-house circulation count date	The date that the temporary in-house check-out count was reset. This subfield must contain a valid date in the form YYYYMMDDHHMM, where HHMM optional. Note: Subfield \$T can be included in the tag only if subfield \$S is also included.	Load date

Subfield	Rules	Description	Value	Default Value if Subfield Not Defined
\$V	NR - O	Item price	The price of the item expressed in the smallest unit of the currency for the database's Home currency. For example, if the Home currency US Dollars, you would enter 1000 (one thousand pennies), NOT \$10.00 (ten dollars). Enter digits only. Do not include currency symbols or separators. The value in this subfield cannot exceed 999999999.	0
\$X	NR - M	Item class	The item class code of the item class assigned to the item. This subfield can contain only digits, with a maximum of 9999. Note: This item class code must be defined in the Item Classes parameter in the Profiler.	0
\$Y	NR - O	Reserve item class	The item class code that will be used when the item is on reserve. This subfield can contain only digits, with a maximum of 9999. Note: This item class code must be defined in the Item Classes parameter in the Profiler.	None
\$0	NR - O	Number of pieces	The number of pieces that constitute this item. This subfield can contain only digits, with a maximum of 9999.	1
\$1	NR - O	Creation date	The date that the item record was created. This subfield must contain a valid date in the form YYYYMMDDHHMM, where HHMM is optional.	Load date
\$2	NR - O	Last use date	The date that the item was last checked in. This subfield must contain a valid date in the form YYYYMMDDHHMM, where HHMM is optional.	None
\$3	NR - O	Last inventory date	The date that the item was last included in the Inventory Control report. This subfield must contain a valid date in the form YYYYMMDDHHMM, where HHMM is optional.	None

Subfield	Rules	Description	Value	Default Value if Subfield Not Defined
\$4	NR - O	Operator ID	The Virtua operator ID of the person who last modified the item record. It is a system-assigned number associated with the Virtua user. This subfield can contain only digits, with a maximum of 999999999.	0
\$6	NR - O	Item barcode	The item barcode associated with the item. This subfield can contain a maximum of 50 characters.	None
\$8	NR - A	852 linking sequence number	<p>The same sequence number as in subfield \$8 of the holdings record 852 tag associated with the item. This subfield is optional if . . .</p> <ul style="list-style-type: none"> • The holdings record contains zero or one 852 tag(s) <p>-OR-</p> <ul style="list-style-type: none"> • No information from an 852 tag is needed to create the item record <p>When this subfield is specified, subfield \$b of the 852 tag will be used when subfield \$D is not specified in the 949 VIRTUAITEM tag. Similarly, the copy number in subfield \$t of the holdings 852 tag will be used when subfield \$F is not specified.</p> <p>Note: Subfield \$8 is applicable only to items attached to holdings records.</p>	None
\$9	NR - O	Units	Free-text that specifies the units information for the item or enumeration/chronology for the serials issue in a comma-delimited list of values. The length is restricted to 64 characters. Note: As the units_sort_form string is also restricted to 64 characters, you need to be aware of the following: Depending on the data in the units field, the units information <i>could</i> result in a units_sort_form string that is longer than 64 characters, causing the following error message:"<DIAGNOSTIC CODE 5059> Input string value is too large to fit into the buffer."	0

Subfield	Rules	Description	Value	Default Value if Subfield Not Defined
\$a	NR - O	First item-level call number	The first item-level call number associated with the item. The length is restricted to 60 characters.	0
\$b	NR - O	Second item-level call number	The second item-level call number associated with the item. The length is restricted to 60 characters.	0
\$c	NR - O	Circulate pieces	Indicates whether the item circulates with multiple pieces/parts (the number of pieces specified by subfield \$0). Zero (0) means it does NOT circulate with multiple pieces/parts; one (1) means it does.	0
\$d	NR - O	EPN Number	An optional item identification number that can be searched in the Virtua client via a control number search. The length is restricted to 24 characters.	None
\$e	NR - O	Accession Number	An optional item identification number that can be searched in the Virtua client via a control number search. The length is restricted to 35 characters.	None
\$f	NR - O	Collection Code	An optional identification number that associates the item with a collection, which can be searched via the Browse search. The length is restricted to 20 characters.	None
\$o	NR - O	Staff note	Text that will be stored in the item record as a "staff" note. The length is restricted to 2000 characters.	None
\$p	NR - O	Public note	Text that will be stored in the item record as a "public" note. The length is restricted to 2000 characters.	None
\$q	NR - O	Check-in note	A note, or circulation alert, that will appear when the item is checked in. The length is restricted to 2000 characters.	None
\$r	NR - O	Check-out note	A note, or circulation alert, that will appear when the item is checked out. The length is restricted to 2000 characters.	None

Subfield	Rules	Description	Value	Default Value if Subfield Not Defined
\$s	R - O	Status	<p>Status data containing the following parts, which are colon delimited:</p> <ul style="list-style-type: none"> • Virtua status code (<i>required</i>) • Date the status was attached to the item (<i>optional</i>) <p>The <i>status code</i> can contain only digits, with a maximum of 9999. If the <i>date</i> is included, it must contain a valid date in the form YYYYMMDDHHMM, where HHMM is optional. For example, \$s2400 or \$s2400:19970901</p>	None
\$t	NR - O	Additional Information	<p>A five-character string of Y's and N's which specify settings for various flags (e.g., YNYNN). Positions 1-5 represent the following flags:</p> <ul style="list-style-type: none"> • 1 - Request flag - Y to use parameters when placing requests and trapping or N if requests are not allowed. • 2 - (<i>flag associated with non-implemented feature</i>) Always set to N. • 3 - Override circulation matrix flag - Y to use the item's local loan period setting (\$K) or N to use the Circulation matrix. • 4 - (<i>flag associated with non-implemented feature</i>) Always set to N. • 5 - (<i>flag associated with non-implemented feature</i>) Always set to N. 	YNYNN, if a valid loan period is specified with \$K. Otherwise, YNNNN.
\$y	NR - O	URL	The URL that allows access to an Internet version of the item.	None
\$z	R - O	Date/time stamp	<p>The date/time will be added when an item is created.</p> <p>Note: If the tag already contains a \$z, no item record will be created <i>unless</i> the z load option is set to allow reloading of items from 949 tags that contain a \$z. Multiple \$z's are created by reloading items.</p>	Load date/time

8. Appendix C - Changes in this Guide

8.1 Changes for Version 15.0

No changes were made.

Index

- GB 18030 · 115

?

-? option · 31

@

-@ load option · 25

0

008 tag values, defaults for patron records · 11
 010 tag used as duplicate test for authority records · 62
 010 tag used as duplicate test for bib records · 62
 015 tag used as duplicate test for patron records · 62
 016 tag used as duplicate test for patron records · 63
 020 tag used as duplicate test for bib records · 62
 022 tag used as duplicate test for bib records · 62
 024 tag used as duplicate test for bib records · 62
 028 tag used as duplicate test for bib records · 62
 035 tag used as duplicate test for authority records · 62
 035 tag used as duplicate test for bib records · 62
 035 tag used as duplicate test for patron records · 63
 035 tag, always generate automatically · 59
 039 tag used to store import filename and offset · 101

1

1xx tag used as duplicate test for authority records · 62

2

222 tag used as duplicate test for bib records · 62
 2709 MARC communications format · 97, 100

3

-3 load option · 37

4

4xx tag, prevent conflicts · 59

8

-8 load option · 51
 852 tag, provides location and copy number · 8
 880 tag data, moving · 51, 79

9

-9 load option · 28
 -9 use 999 load option · 12
 949 VIRTUAITEM tag · 7, 8, 26
 EPN Number and · 8
 role in loading item records · 8
 sample format · 128
 stores item information in parent records · 8
 949 VIRTUAITEMCALLNUMBER tag · 8
 999 LOADOPT tag · 5, 6, 7, 12, 73, 113
 ignoring · 12
 mapping load options · 7
 priority of · 4, 12
 role of · 6
 specifications for · 7
 999 VTLSSORT tag · 53

A

-a load option · 126
 --AAP-only-indexing load option · 55
 Action entries on vload log · 19
 Add Duplicate Test window (fig) · 89
 Add Import Filename to Record check box · 101
 Add Merge window (fig) · 92, 94
 Add Rule window (fig) · 93
 AddDiagnosticText.exe · 20
 running · 20
 used to translate diagnostic code · 20
 AddSpecial035.exe · 38
 --always-auto-generate-035-tag · 59
 ampersands in duplicate tests · 64
 ANSI Z39.47 · 29, 75, 115
 ANSI-8 (Swiss) · 29, 75, 115
 ANSI-8 + Hebrew · 29, 75, 115
 authority records
 duplicate tests for · 62
 load options specific to · 79
 prevent conflicts · 59

Authority tab, Load Options window (fig) · 79
 automated check-in of serials issues · 27, 84

B

-b load option · 28
 -B load option · 43
 Barcode Type · 9
 Barcode Type parameter, setting · 9
 bibliographic records, load options specific to · 77
 Bibliographic tab, Load Options window (fig) · 77
 bibliographic_location table · 57
 Big5 · 29, 75, 115
 Big5-HKSCS · 29, 75, 115
 blind reference flag load option · 81
 block item creation flag load option · 7, 78, 84

C

-c load option · 26
 -C load option · 27
 cataloging level, matching on · 63
 Cataloging parameters · 9
 CCCII · 29, 75, 115
 changes in this guide · 136
 Character Set drop-down list · 100
 character set load option
 default · 10
 diagnostic error messages · 10
 mapping to UTF-8 · 10
 character sets
 client translation of · 97
 client-side conversion · 97
 potential conflict in settings · 97
 server-side conversion · 75
 specifying for incoming records · 29, 74, 101, 107
 character-mapping errors · 17
 check date flag load option · 87
 Circulation parameters · 9
 client
 advantages/disadvantages in using to load records · 2
 load options and the 999 LOADOPT tag · 7
 setting load options in · 71
 comparing record dates when loading · 27
 comparing vload.exe and the client · 2
 Configure button · 99, 100
 CP 850 · 29, 75
 create items flag load option · 78, 84
 --create-sort-tag load option · 53
 creating
 duplicate test/merge load options file · 13
 general load options files · 12
 creating item records · *See* loading item records

D

-d load option · 29
 -D load option · 43
 database action load option · 27, *See* dbAction load option
 database errors · 99
 date entered flag load option · 75
 date information, retaining original · 59
 date of records
 comparing when loading · 27
 date pattern code load option · 87
 dbAction load option · 78, 81, 84
 Default (Merge) Rule
 explanation of · 95
 using in the client · 95
 default character set · 10
 default duplicate tests · 98
 default load options · *See also* load options
 use of · 10
 delete date load option · 87
 detailed processing summary · 17, 19
 diagnostic messages
 vload.exe, getting a report of · 23
 diagnostics
 client batch load · 103
 vload.exe · 16
 dominant tags for duplicate test · 89
 duplicate barcodes, detection of · 8
 duplicate records, similar headings and · 63, 90
 duplicate test load option · 60, 78, 81, 96
 behavior of · 60
 generating a file for · 13
 multiple options allowed · 61
 using in the client · 88, 89
 duplicate tests
 available for all records · 61
 available for authority records · 62
 available for patron records · 62
 defaults · 98
 escaping the ampersand · 64
 examples of · 63
 specifying Virtua's action · 63
 use of the ampersand · 64
 DuplicateOptions.exe · 13

E

-E load option · 51
 EPN Number in item records · 8
 error codes, translating · 19
 error log file
 client batch load · 104
 vload.exe · 20
 error messages, vload.exe and · 23
 error processing
 client batch load · 103

vload.exe · 16
 Error state records, retrieving · 17
 ESM_Over · 81
 ESU_Merge · 81
 Europa-3 · 29, 75, 115
 expiration date load option · 87
 Export window - Language Editor (fig) · 22

F

-f load option · 36
 failed record deletion file · 52
 failed validity checks · 99
 --failed-delete-record-file load option · 52
 File Name text box · 100
 files that contain load options · 11
 duplicate tests/merges · 13
 general · 12
 format of imported records, selecting · 101
 formats
 merging string option · 65
 Tag Test · 66
 FRBR splitting flag load option · 78

G

-G load option · 44
 GB 18030 · 29, 75
 GBK · 29, 75, 115
 general processing summary · 17
 General tab load options · 72
Getting Started with Acquisitions Guide · 106, 109, 111
 Global parameters · 9
 graphical user interface · 2

H

-h load option · 39
 -H load option · 44
 heading type flag load option · 81
 --help option · 31
 hold migration flag load option · 84
 holdings records, load options specific to · 83
 Holdings tab, Load Options window (fig) · 83
 Hong Kong Public Library · 37

I

-i load option · 126
 -I load option · 47
 Import button · 108, 111
 Import Configuration dialog box (fig) · 100

import filename, use of · 101
 Import MARC Orders/Requests tool · 109
 Import MARC Orders/Requests window (fig) · 110
 Import Records icon · 98
 Import Records to Purchase Request List tool · 106
 Import Records tool · 97, 98
 accessing · 98
 character set value · 97
 configuring · 100
 diagnostic messages and · 102
 features of · 97
 generation of log files · 103
 running · 102
 Import Records window (fig) · 98
 Import to Purchase Requests tool
 when records go into Error state · 108
 Import To Purchase Requests window · 106
 Import to Purchase Requests window (fig) · 107
 importing MARC records
 Destination Options · 100
 resetting the import tool · 103
 running the Import Records tool · 102
 Source Options · 100
 stopping the import process · 103
 indexing
 disabling for holdings when running vload.exe · 58
 disabling for patrons when running vload.exe · 57
 disabling when running vload.exe · 56
 indexing load options
 AAPs only · 55
 disabling all indexing · 56
 disabling FRBR subject indexing · 56
 disabling holdings indexing · 58
 disabling location indexing · 57
 disabling patron indexing · 57
 subject headings only · 56
 user-defined headings only · 55
 --indexing-disabled load option · 56
 --input-file load option · 53
 International Serials Data System interchange
 character set · 29
 Is Discarded option · 108
 Is New option · 108
 Is Placed in Error State option · 108
 ISO 5426 · 29, 115
 ISO 6937/2 · 29, 75, 115
 ISO 6937/2 + Arabic · 29, 75, 115
 ISO 6937/2 + Greek · 29, 75, 115
 ISSN version of MARC 21 · *See* MARC 21 (ISSN version)
 item class code load option · 7, 78
 Item Class Definitions · 9
 Item Class Definitions parameter, setting · 9
 item records · *See also* loading item records
 load blocked if parent records assigned special state · 7
 loading · 7
 loading multiple · 7
 setting parameters for loading · 9

Item-level Call Number Flag · 9
Item-level Call Number Flag parameter, setting · 9

J

-J load option · 47

K

-k load option · 40
-K load option · 48
Keyword Indexing Configuration Editor · 2
keyword indexing, performed by load programs · 2

L

-l load option · 35
-L load option · 42
language file, obtaining for AddDiagnosticText.exe · 21
language table - Language Editor (fig) · 23
Leader position 17, matching on · 63
Library Names parameter, setting · 9
load options
 Authority tab · 81
 Bibliographic tab · 77
 default settings · 5
 default values for · 10
 for duplicate tests · 60, 88
 for merging records · 60, 88
 for vload.exe · 24
 General tab · 74
 Holdings tab · 83
 layers of · 4
 organized by record type · 113
 Patron tab · 86
 persistent nature of in the client · 71, 73, 89
 priority of · 4, 5, 11
 scope of · 6
 setting different types of · 73
 setting in the Virtua client · 71
 stored in the 999 LOADOPT tag · 5, 6
 supplied at the command line · 5
load options file option · 76, 78, 82, 84, 87
load options files
 creating for duplicate tests/merges · 13
 creating for general options · 12
 using for duplicate tests/merges · 13
 working with · 11
Load Options function · 6
Load Options function and the Import Records tool · 98
Load Options window
 character set value · 97
 displaying · 71

 working with · 73
Load Options window (fig) · 72
 Authority tab · 80
 Bibliographic tab · 77
 Holdings tab · 83
 Patron tab · 86
loading item records · 7
 multiple · 7
 role of 949 VIRTUAITEM tag · 8
 setting parameters before · 9
loading records in the client
 Destination Options · 100
 resetting the import tool · 103
 running the Import Records tool · 102
 Source Options · 100
 stopping the loading process · 103
LOADOPT tag · *See* 999 LOADOPT tag
LoadXMLFormat.sh · 54
Local Options File option
 Authority tab · 82
 Bibliographic tab · 79
 Holdings tab · 84
 patron tab · 88
locations, disabling indexing of · 57
log entries
 Action codes · 19
log files
 client batch load · 103
 specifying location of on server · 17
 vload.exe · 16

M

-m load option · 34
mapping character sets · 10
MARC 21 (ISSN version) · 101
MARC format, specifying for incoming records · 52
MARC orders/requests
 importing · 109
 options offered by vendor's website · 109
MARC-8 · 29, 75, 115
MARCXML files, loading · 53
Mask New Records option · 108, 111
Mask Records check box · 108, 111
Mask Records check box on Import Configuration window · 101
masking bibliographic records · 58, 101
--mask-records option · 58
merge database action for provisional authorities · 81
merge load option · 60, 81, 87
 about · 65
 examples of use · 69
 format of · 65
 generating a file for · 13
 multiple options allowed · 61
 using in the client · 88, 92
merge rule · 65

merge rule, types of · 67
 merges load option · 78
 MergeString · 66
 merging duplicate records
 general principles for · 65
 merging duplicate records, options for · 64
 Microsoft CP 850 · 115
 MODS records, loading · 54
 MODS2MARCXMLStyleSheet.xml · 54
 MODSSchema.xsd · 54
 move 880 tag data load option · 79
 MoveStateRecords.exe, command-line options for · 24

N

-N load option · 46
 National Library of Wales · 37
 --NoFRBRSubjectIndexing load option · 56
 --no-holdings-indexing load option · 58
 --no-location-filter load option · 57
 non-duplicate action load option · 78, 81, 87
 specifying in the client · 95
 --no-patron-indexing load option · 57
 number of records successfully loaded · 99

O

-o load option · 46
 -O load option · 30
 OCLC · 9
 OCLC Connexion · 97, 100
 Open Records In MARC Editor destination option · 101
 override database action for authority records · 81
 Overwrites Existing Record option · 108

P

-p load option · 40
 -P load option · 48
 patron 008 tag values, creation of · 11
 patron records
 automatic creation of 008 tag values · 11
 duplicate tests for · 62
 patron records, load options specific to · 86
 Patron tab, Load Options window (fig) · 86
 PC-8 · 29, 75, 115
 persistent settings and the Import Records tool · 98
 Preferred Workflow settings · 111
 --prevent-authority-4xx-conflicts · 59
 priority of load options · 4, 5
 Process Immediately · 28
 processing records · *See* loading records
 processing summaries · 16
 Purchase Request Categories parameter · 108

purchase request category · 108
 Purchase Request List, importing to · 106

Q

-q load option · 41

R

-R load option · 45
 -r merge load option · 33, 63, 65
 record state load option · 76
 Record Update Failed diagnostic message · 21
 Reset button · 99
 Reset button, Import Records tool · 103

S

-s load option · 33, 63
 Save Records to Database destination option · 101
 Save To File button · 79, 82, 84, 88
 scope of load options · 6
 Search by State window · 101
 serials issues, creating items for · 27, 84
 shelf location code load option · 7, 78
 shelving location, defining · 43
 Source Options · 100
 special 035, adding · 37
 Start button · 99
 -Start load option · 30
 statistics log file
 vload.exe · 19
 Stop button · 99
 storing import filename and offset · 101
 subfield specifications
 949 VIRTUAITEM tag · 128
 subfields in the 949 VIRTUAITEM tag
 reprocessing · 9
 --SubjectHeading-only-indexing load option · 56
 summary log file
 client batch load · 104
System Management Reference Guide · 30

T

-t load option · 45
 -T load option · 49
 tag 999 LOADOPT · *See* 999 LOADOPT tag
 tag test · 65
 TagTest · 66, 67
 Thai Union Catalog · 37
 TIS620 Classic Thai · 29, 75, 115
 Title Selection List · *See* Purchase Request List

total records found in source file · 99

U

-u load option · 49
 -U load option · 36
 Unicode · 10, 29, 97, 101, 108, 111
 UNIMARC records, loading · 52
 Université catholique de Louvain · 37
 UNIX operating system · 1
 use 999 load option · 12
 --use-incoming-039-tag · 59
 --UserHeading-only-indexing load option · 55
 UTF-8 · 115
 UTF-8 character set · 10, 97, 101, 108, 111
 converting to · 101, 108, 111
 UTF-8 encoding · 29

V

-v load option · 39
 -V load option · 50
 validate only load option
 vload · 19, 20
 Validate with Database Only import destination option
 · 101
 validation and the Import Records tool · 98
 validation load option · 79
 Validation load option
 error log · 104
 summary log · 104
 Virtua 949 tag label flag load option · 78, 84
Virtua Cataloging User's Guide · 23
 Virtua ILS – Integrated Library System · 1
Virtua Installation User's Guide · 3
Virtua Profiler Getting Started Primer · 9
Virtua Profiler/Circulation User's Guide · 49
 vload.exe
 advantages/disadvantages in using · 2

creating error message files · 19
 detailed processing summary · 19
 help · 31
 load options and the 999 LOADOPT tag · 7
 processing summaries · 16
 VTLS_TEMP · 17

W

-w load option · 39
 -W load option · 50
 Windows ALA · 29, 75, 115
 Windows Arabic · 29, 75, 115
 Windows Cyrillic · 29, 75, 115
 Windows Hebrew · 29, 75, 115
 Windows Latin 2 · 29, 75, 115
 Windows Latin1 · 29, 75, 115

X

-x load option · 32
 -X load option · 35
 XML file
 loading · 53, 54
 --xml-element-name load option · 53
 --xml-format load option · 54

Y

-Y load option · 52

Z

-z load option · 42