

Making changes to headings

by

Gary L. Strawn

*Authorities Librarian, etc.
Northwestern University Library*

Northwestern University
Evanston, Illinois
2008

The following trademarks, tradenames, registered trademarks, and/or service marks are used in this publication: Visual Basic®, Microsoft Windows™—Microsoft Corporation; Oracle®, Oracle Corporation.

The expression *Vger* is used in this document to represent the name of a well-known client-server integrated library system.

This documentation, and the executable modules it describes, are made available at no cost by Northwestern University to all interested parties. These programs and their documentation may be freely reproduced redistributed. The following restrictions are placed on this free redistribution:

1. There must be no charge of any kind (including ‘shipping and handling’) assessed for these programs and component modules. There must be no charge (including ‘shipping and handling’) assessed for copies of the documentation.
2. The documentation will be distributed as is, without changes of any kind, and especially without removal of marks identifying it as having been produced by Northwestern University.
3. No attempt will be made to remove any identifying marks that may be contained within the executable modules themselves.

Those wishing to distribute these programs under other conditions should contact the following organization for the terms under which this distribution may be allowed:

Technology Transfer Program
Northwestern University
1801 Maple Avenue
Evanston, IL 60208
847/491-3005

Introduction

Background

The need for automated assistance

From time to time, for any number of reasons, it becomes necessary to make the same change to a group of bibliographic records. For example, a cataloger might discover a changed heading in an updated version of an authority record, or might discover an inconsistency among headings in bibliographic records in the local database. In the past, making the same change to a set of bibliographic records¹ has often meant that an operator had to retrieve each record, change the record, and save it back to the database. Naturally, this work was tedious, time-consuming, and prone to error.

The present document describes a pair of programs available from Northwestern University Library that, in conjunction with the *cataloger's toolkit* program (also available from Northwestern), allow libraries using the Vger library system to perform bibliographic heading changes easily, without requiring that an operator visit each record. You use the cataloger's toolkit to formulate correction change requests.² You use the first of the programs described in this document, the *correction review* program, to inspect and adjust correction requests created with the toolkit before those corrections take effect; you use the second program described here, the *correction receiver* program, to make changes to bibliographic records according to the approved correction requests.

Program scope

The current version of the correction receiver program is limited in the types of changes it can make to bibliographic records. Although these limitations will not affect the performance of normal heading changes, it is important that you understand them.

- The correction receiver program can only change *headings*. The term *heading* in this document means *an access point under authority control*. This may be explained further as *that part of a controlled access point that is actually under authority control*. Except when referring explicitly to subject subdivisions (6XX subfields \$v, \$x, \$y or \$z) handled independently of any other text, the term *heading* in this document also means *the part of the access point that begins with subfield \$a*. In the case of a bibliographic record, this includes many subfields of the 100, 440, 650 and 710 fields, but not subfield \$4; and it does not include the bibliographic 245 field except when taken in association with the 1XX field as part of a name/title heading. (The correction receiver program treats the bibliographic 1XX field plus the 240 field, or the 1XX+245 if there is no 240 field, as a name/title heading.)
- The correction receiver program can only change things it can find in your Vger system by searching one of the system indexes—that is, by using instructions in SQL statements to find information that resides in Oracle® tables. All bibliographic headings under authority control (including the name/title heading formed from the 1XX field plus the 240 or 245 field) are exposed by Vger as entries in Oracle tables.

¹ Some institutions may choose to use the programs described in this document to make changes to records by program only when more than a certain number of bibliographic records must be changed, and to continue to make changes through the Vger cataloging client for smaller numbers of records. There is nothing in these programs that enforces such a limit; an institution may use these programs to make a change to a single record.

² Documentation for the cataloger's toolkit is available online at <http://www.library.northwestern.edu/ctkv>. The section of the documentation that describes batch correction requests is at http://www.library.northwestern.edu/ctkv/#requesting_batch_corrections.

- The correction receiver program can perform *one-for-one transformations* on headings. The program can, for example, replace the ‘old’ form of a heading with the ‘new’ form, the most common type of mass change. The program can also create multiple headings from one original heading, can delete subfields from headings, and can delete entire fields. But the program cannot—at the present time—perform other kinds of corrections.

These limitations mean that the correction receiver program can search for the old form of a person’s name or a subject heading, retrieve the relevant bibliographic records, and change headings that contain the old form to the new form; and they mean that the program can search for the old form of a subject subdivision, retrieve the relevant bibliographic records, and change subject headings that contain the subdivision to reflect the new form. (These are of course exactly the things you’ll use the program for most often.) But the program cannot search for records that contain a given piece of text in a bibliographic 500 field and perform some manipulation on that text. Here are two more examples of what the program cannot do:

- These programs cannot change the text in subfield \$h of the 245 field
- These programs cannot change the text in subfield \$e of the 700 field

There is one important exception to the rule about changing only headings under authority control. The correction receiver program can, on option, pass records to a validation module; and the validation module *can* change many other parts of a record. This means that if you configure the correction receiver program to invoke the validation module you can use the correction receiver program to change parts of a record other than headings. See the description of the ‘Supplementary validation rules file’ box on the correction review program’s ‘Files, etc.’ tab: http://www.library.northwestern.edu/ctkv/#files_etc_tab.

There is also one important exception to the rule about changing only records that the program can find via Oracle tables. A correction request can (on option) be accompanied by a file that contains the Vger record numbers of the bibliographic records to be examined in the course of the work. If you can produce such a file using tools outside the correction receiver program,³ the correction receiver program can in fact work on records it would not otherwise be able to find. See the description of the ‘Input file of records to correct’ box on the correction review program’s ‘Files, etc.’ tab on its main panel: http://www.library.northwestern.edu/ctkv/#files_etc_tab.

Although these are severe limitations, the programs are in fact able to perform the great majority of corrections that need to be performed at individual institutions: those involving changes to access points that are under authority control. Libraries will be able to use these programs to clear most items from their backlog of pending changes.

Examples of changes you can make with this set of programs:

Based on these fields (and other information) from an authority record:⁴

```
100 1 $a Kimball, Dexter S. $q (Dexter Simpson), $d 1865-1952
400 1 $w nna $a Kimball, Dexter Simpson, $d 1865-1952
```

These programs can change this bibliographic field:

```
100 1 $a Kimball, Dexter Simpson, $d 1865-1952.
```

to this:

```
100 1 $a Kimball, Dexter S. $q (Dexter Simpson), $d 1865-
1952.
```

³ The cataloger’s toolkit can generate a file containing the Vger record IDs of all of the bibliographic records currently open in the cataloging client and append that file to a correction request.

⁴ These programs do not *require* that an authority record be present in the local system. The examples given in this document are stated in terms of authority records merely for simplicity.

These programs can also change this bibliographic field:

700 1 \$a Kimball, Dexter Simpson, \$d 1865-1952. \$t
Selections. \$f 1997.

to this:

700 1 \$a Kimball, Dexter S. \$q (Dexter Simpson), \$d 1865-
1952. \$t Selections. \$f 1997.

Based on these fields (and other information) from an authority record:

150 \$a Iban (Bornean people)
450 \$w nne \$a Ibans (Bornean people)

These programs can change this bibliographic field:

650 0 \$a Ibans (Bornean people)

to this:

650 0 \$a Iban (Bornean people)

These programs can also change this bibliographic field:

650 0 \$a Ibans (Bornean people) \$z United States \$x
Bibliography.

to this:

650 0 \$a Iban (Bornean people) \$z United States \$x
Bibliography.

Based on these fields (and other information) from an authority record:

180 \$x African American authors
480 \$x \$w nne \$a Afro-American authors

These programs can change this bibliographic field:

650 0 \$a American literature \$x Afro-American authors.

to this:

650 0 \$a American literature \$x African American authors.

These programs can also change this bibliographic field:

650 0 \$a American literature \$x Afro-American authors \$x
Bibliography.

to this:

650 0 \$a American literature \$x African American authors \$x
Bibliography.

Based on these fields (and other information) from two different subject records:

151 \$a Afghanistan \$x Politics and government \$y 1973-1989
451 \$w nnoa \$a Afghanistan \$x Politics and government \$y
1973-

151 \$a Afghanistan \$x Politics and government \$y 1989-2001

These programs can, when requested to do so, change this bibliographic field:

651 0 \$a Afghanistan \$x Politics and government \$y 1973- \$v
Bibliography.

to this pair of fields:

651 0 \$a Afghanistan \$x Politics and government \$y 1973-1989
\$v Bibliography.

651 0 \$a Afghanistan \$x Politics and government \$y 1989-2001
\$v Bibliography.

Queues: How the program manages correction requests

This pair of programs implements one or two *queues* that contain messages passed from one program to the next. Each program reads messages from and/or writes messages to one or more of these queues. One queue, called the *pending* queue, contains preliminary requests for changes; a

second queue, called the *approved* queue, contains change requests that have been approved for performance by the correction receiver program.⁵

These ‘queues’ are actually nothing more than folders on some disk drive somewhere. The ‘messages’ in these queues are nothing more than simple text files⁶ that have the general form of configuration files: each file contains several stanzas, each stanza contains one or more lines, and each line consists of a label and a value separated by an equals sign. There is a separate file for each correction request in each queue. The names of the text files can be just about anything; but they must end with the extension ‘.cor’.⁷

Figure 1 presents a general scheme for performing batch corrections. Requests made by some operators (probably via the cataloger’s toolkit) go to the pending queue, and move from that queue into the approved queue only after review. Requests made by other operators go directly into the approved queue.⁸ An institution—most commonly, an institution with only a few operators allowed to make correction requests (or even just a few operators, full stop)—may choose not to implement the pending queue at all, but may allow all operators to write requests directly into the approved queue. (Such institutions will of course not need the correction review program, which exists solely to move requests from the pending queue to the approved queue.) Other institutions—most typically, those with a large number of operators—will probably wish to implement the centralized approval of corrections made possible by the pending and approved queues, so that the effect of each request on the database can be assessed one final time before changes are made.

⁵ This document speaks of the pending and approved queues as if they were single entities. An institution might implement a number of different approved and/or pending queues, depending on the institution’s needs and organizational structure. Indeed, a very large institution may choose to have not only a number of pending and approved queues, but also to employ several instances of the correction receiver program, each instance reading from its own approved queue.

⁶ These files contain only the low-end ASCII characters. MARC21 diacritics and special characters are translated into a plain-text notation when stored in the file, and re-translated when read from the file. This convention for diacritics and special characters—which operators must also employ when modifying correction requests—is described in Appendix A.

⁷ If a correction request is generated by the cataloger’s toolkit, the names of the files consist of the operator’s initials, the date and time on which the request was made, and a second time stamp. This second time stamp shows the amount of time that has elapsed (measured in milliseconds) since the operating system was last rebooted; it helps avoid the possibility of duplicate file names. Here’s a typical name for a correction request file created by the cataloger’s toolkit:

GLS.20010721.133446.1576273411.cor

This file name structure is not required. The correction review and correction receiver programs assume that any file in a queue folder whose name ends with the “.cor” extension is a correction request. For this reason, the folders used for the pending and approved queues must not contain other kinds of files with the ‘.cor’ extension.

⁸ This diagram indicates that programs operating autonomously are also able to write requests into the pending and approved queues. For example, Northwestern University’s authority loader program can notice that a heading has changed and can prepare a suitable correction request; depending in part on the nature of the change and in part on configuration choices, the program can write the request into either a pending or an approved queue.

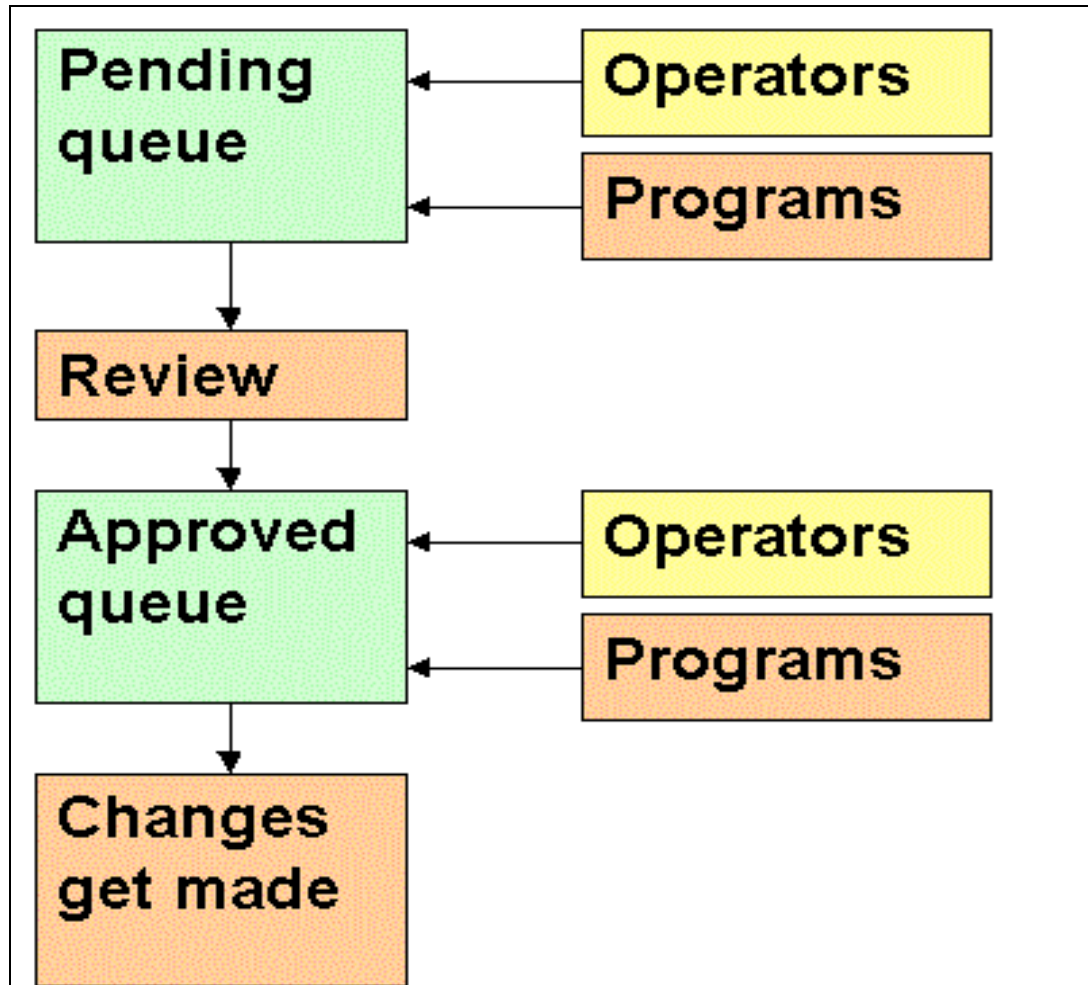


Figure 1. Schematic diagram of the heading correction process

Most operators will use some button on the *cataloger's toolkit* to request a change to records. Several different buttons on the cataloger's toolkit generate correction requests. The *Requesting corrections* section of this document gives just a bit of information about toolkit buttons that create correction requests; the toolkit's documentation goes into far greater detail: http://www.library.northwestern.edu/ctkv/#requesting_batch_connections.

If correction requests go into a pending queue, a qualified operator later uses another program (the *correction review* program, described in this document) to review pending requests; the operator can move requests from the pending queue into the approved queue, can leave them in the pending queue, or can delete them from the pending queue. The operator can modify a request before approving it. All pending requests should eventually either find their way into the approved queue or be deleted. Requests that appear in the approved queue (either because they were written directly into that queue by an authorized operator, or were moved into that queue from the pending queue by an authorized operator) will be performed by a separate program (the *correction receiver* program, described in this document) at the next convenient time. An institution may choose to leave the correction receiver program running all the time, so that it performs each request as the request enters the approved queue; or an institution may choose to accumulate requests in the approved queue and run the correction receiver program at times of low system use.

If an institution implements a pending queue so that requests receive a final review, the pending queue will if possible be a folder on a networked drive to which all appropriate operators have write access. The approved queue should be a folder to which only qualified operators have write access.⁹

Installation

Program availability

The installation programs (ZIP files) described in this document are available from Northwestern University Library's file transfer site:

<http://www.library.northwestern.edu/public/>

Correction review program

If your institution wishes to implement a pending queue for correction requests, you'll need the program that allows an operator to review and dispose of requests. This program is the *correction review* program. Its installation program consists of the following three modules, available within the CorrectionReview.ZIP file in the /public/CorrectionReview¹⁰ folder at Northwestern University Library's file transfer site:

```
CorrectionReview.CAB
setup.exe
Setup.lst
```

Download the ZIP file, expand it into its three component files, and run the program *setup.exe*. The default location for the installed program on the Microsoft Windows™ 'Start' menu is:

```
Programs
  Northwestern University Library
    Correction review
```

Correction receiver program

To make changes to your records based on requests in the approved queue, you'll need the program that acts on correction requests; this is the *correction receiver* program. Its installation program consists of the following three modules, available within the NNNN.NN.NNNN.CorrectionReceiver.ZIP file¹¹ in the /public/CorrectionReceiver folder at Northwestern University Library's program download site:

⁹ If the use of a networked drive is not possible or convenient for either queue, operators may use some other mechanism (diskettes, memory sticks, e-mail messages with attachments) to get correction requests into the pending or approved queues. Such a mechanism is outside the scope of this document.

¹⁰ Unlike many other programs available at this site, the correction review program does not depend on the Vger version. But because the correction request files shuttled from program to program may change in format from one version to the next, you should always use the latest version of the correction review program, as well as the latest version of the correction receiver program available for your version of Vger.

¹¹ "NNNN.NN.NNNN" represents the Vger version number; most correctly, this is the version number of the BatchCat DLL file that comes with each version of Vger. As explained on the download site's home page, there is a different version of this DLL for each different build of Vger. You must use the version of the correction receiver that exactly matches your version of Vger. For example, if you use Vger version 2006.52 (which comes with BatchCat build 826), you

CorrectionReceiver.CAB
setup.exe
Setup.lst

Download the ZIP file, expand it into its three component files, and run the program *setup.exe*. The default location for this program on the Windows 'Start' menu is:

Programs
Northwestern University Library
Correction receiver

Running the installation program is not all you have to do before you can run these programs. The correction review and correction receiver programs require an ODBC connection to your Vger system. This documentat does not describe the installation of the Oracle ODBC driver, or its configuration.

The correction receiver uses the BatchCat DLL that comes as part of the Vger client installation package. This means that Vger clients for the version of Vger installed at your institution must be installed on the workstation that runs the correction receiver program. However, the clients do not need to be *running* while you run the correction receiver program. The correction receiver program uses one of the DLLs that comes with the client installation, not the clients themselves, to send changed records to Voyager.

What if the installation goes bad?

If you have any difficulty during the installation of either of these programs, see the suggestions in the online documentation for the cataloger's toolkit:

http://www.library.northwestern.edu/ctkv/#installation_gone_wrong

must select 2006.52.826.CorrectionReceiver.ZIP. See the instructions at the download site's initial page for determining which version of the program is right for your version of Vger.

Requesting corrections



An operator may use the ‘big red cross,’ ‘split headings’ or ‘queue review’ button on Northwestern University Library’s cataloger’s toolkit to make a correction request. As described in detail in the documentation for the cataloger’s toolkit, the operator can base a correction request on information in an authority record, can base a correction request on information pulled from a combination of authority and/or bibliographic records, or can create a correction request from scratch. The toolkit writes each correction request approved by an operator into the folder named on the toolkit’s Options panel. This folder may serve as either a pending queue or an approved queue, depending on the manner in which corrections are requested and performed at a given institution. Since each operator has a separate toolkit configuration, different operators can write to different queues.¹²

The toolkit doesn’t actually know, or care, whether it’s writing requests into a pending or an approved queue—it uses whichever folder it’s told to use. The function of the queues is defined by local policy, not the programs. All requests written by the toolkit—regardless of the name of the folder, and regardless of the manner in which the operator composes the correction request—have exactly the same format.

Programmers at other institutions can write their own programs to generate correction requests. As long as a file in the approved queue has the proper structure and has a name that ends with the extension ‘.cor’, the correction receiver program will perform the indicated work.¹³

¹² If different operators sharing one workstation need to write to different queues, then each must use a shortcut to start the program that names individual toolkit INI files. See http://www.library.northwestern.edu/ctkv/#multiple_configurations

¹³ The task of writing a program to generate a correction request is not one for the casual programmer. Northwestern University Library will make available on request a Visual Basic® source code module that contains routines for writing requests to and reading requests from a file in a queue. This is the same code module used by the correction review and correction receiver programs. If this module is included in other programs, locally-written programs will be able to create correction requests in the proper form; but the task of putting valid data into the various buckets is left for the local programmer.

Reviewing correction requests

Preliminaries

If an institution establishes a pending queue for requests generated by some or all of its operators, a qualified operator must use a special program to review the requests and identify the correct disposition for each. This program is the *correction review* program. An operator uses this program to move a correction request from the pending queue into the approved queue. If you don't implement a pending queue, then you don't need this program.

Here's what the program looks like when you start it:¹⁴

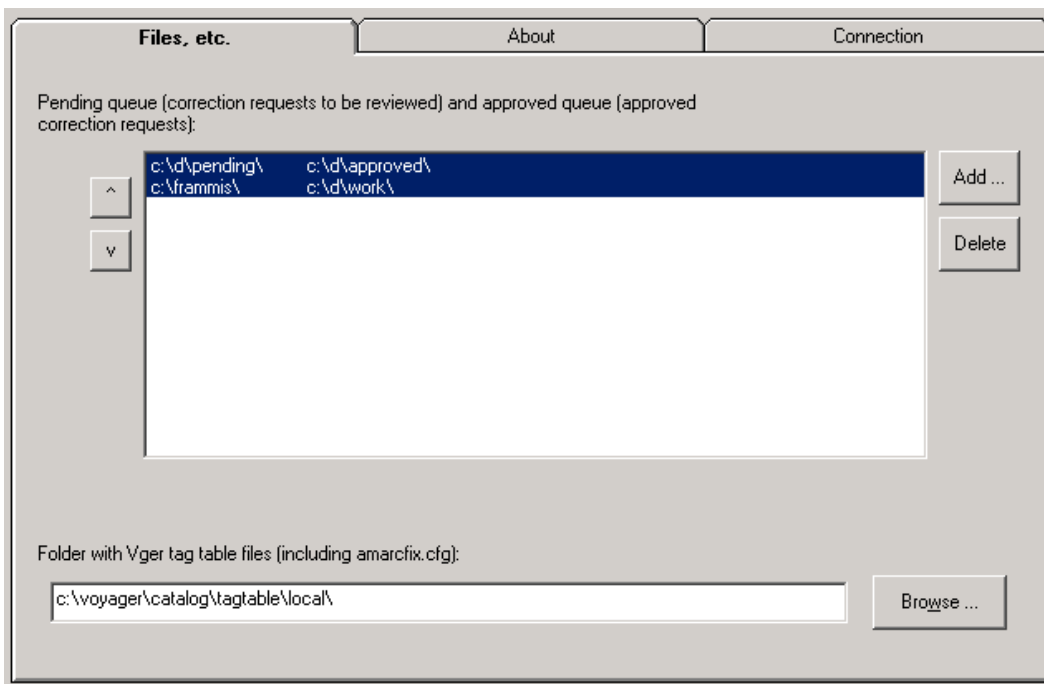
The screenshot shows the 'Correction Review' program interface. The window title is 'Correction Review'. The menu bar includes: Exit, Begin review, Notify, Approve, Delete, Make pending again, Next, Previous, Options. The main area is divided into three tabs: 'Headings', 'Files, etc. (nothing here yet)', and 'Multiples (nothing yet)'. The 'Headings' tab is active and contains two sections: 'Old heading' and 'New heading'. The 'Old heading' section has fields for 'Tag' and 'Ind' (both dropdowns), a large text input field, and buttons for 'Pick up...', 'Add as 4XX', and 'Add as 688'. Below these are 'Series numbering pattern' and 'Indirect subdivision' fields. The 'Subject system' is set to 'a LC subject headings', with checkboxes for 'Ignore \$v/\$x coding differences' and 'Series heading'. The 'New heading' section has similar fields for 'Tag', 'Ind', a large text input field, and buttons for 'Pick up...' and 'New authority'. It also includes 'Series numbering pattern', 'Indirect subdivision', and 'Geographic subdiv. comes after' fields. The 'Subject system' is set to '* All subjects', with checkboxes for 'Geog. subdiv. can follow' and 'Change subject code'. The 'Control information' section at the bottom has radio buttons for 'Normal replacement', 'Zap a subdivision', and 'Zap whole field', and a 'Requester' text field. It also includes several checkboxes: 'Change only whole geog. subdiv.', 'Change if NOT whole geog. subdiv.', 'Romanization change (series title)', 'Change only subjects', 'Change only non-subjects', 'Change only 651, not \$a of X10/X11', 'Move following \$p or \$n to \$v', and 'Preserve \$v/\$x coding during change'.

The correction review program requires an ODBC connection to your Vger system, but does not change records in Vger, so it doesn't use the BatchCat interface into the Vger system. (That's why there isn't a different version of the correction review program for each build of Vger.) This program *only* works with requests in the pending and approved queues. You can only use this program to review requests generated by another program; you cannot use it to create a new request.

¹⁴ The screen layout changes somewhat from version to version. Don't let that throw you. The illustrations in this document may reflect older versions if the difference isn't relevant to the point under discussion.

Configuration

Before you can use the correction review program the first time to inspect correction requests, you must supply a few pieces of information on the program's Options panel. The following illustration shows the Options panel, which you find by clicking the Options choice on the program's menu.



On the Files, etc. tab:

- The big list in the middle consists of pairs of folder names. Each line contains the name of a *pending queue* and an *approved queue*. The program will read files from the pending queues and write them (if the operator so directs) to the corresponding approved queues. You can define as many pairs of queues as you need. You can define more queue pairs than you want to work with in a given session; highlight the queues you're interested in working with during a particular session, and highlight different ones another time. The program will read correction requests from the highlighted queues and ignore the other queues. Perhaps not surprisingly, the pending and approved folders in a given pair cannot have the same name.
- *Folder with Vger tag table files:* the name of the folder that contains the Vger configuration file 'amarcfix.cfg.' This folder doesn't actually have to be the same folder as your Vger tag table folder (the only requirement is that the folder contain the file amarcfix.cfg), but it probably will be best if you identify in this box the copy of that file also used by the Vger cataloging client.

The pending and approved folder names (as well as the Vger tag table file folder name) must end with a reverse slash.

On the Connection tab:

- *ODBCdata source name*: The name of the ODBC connection that points to your Vger database. This is the DSN you define for your Vger database in the ODBC Data Source Administrator client.¹⁵
- *Table name prefix*: The name of the database at your Vger installation whose records are to be corrected. A Vger installation contains at least two databases: the production database and a training database. Each of these has a different name. The name of a Vger database typically consists of some text plus the letters 'DB': 'BIGDB.' and 'CORNELLDB.' are typical examples. This name must end with a full stop.
- *Read-only signon* and *Read-only password*: The logon and password the program should use for read-only access to your Vger database. This is a logon and password defined to Oracle by a system administrator, and is unrelated to any Vger logons.

In case you're wondering, the 'About' tab on the Options panel simply gives information about the program. If you have any difficulties with the program or suggestions for changes, it would be best if you include in your message the version number you'll find on this tab when you report the problem or make the suggestion.

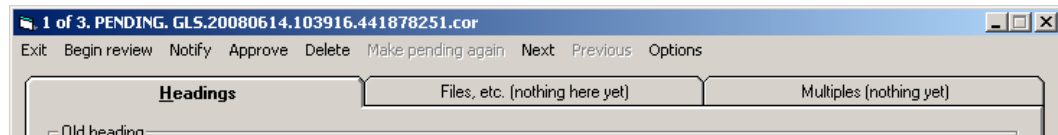
While you're in the middle of a review of correction requests, you can view the Options panel, but you cannot change anything you find there. To change options once you've started a review, either cancel the program, start it up again and immediately change the options; or take the 'Next' choice from the program's menu until you've passed all the way through the pending queue, and then change the options.

Operating the program

The program's menu contains the items you'll use to work with correction requests. You've already seen how to use the 'Options' item to set the program's configuration. Use the *Exit* menu item when you're done using the program.

Use the *Begin review* menu item to begin your review of pending correction requests. When you select this menu item, the program builds an internal list of the correction requests in all of the pending queues highlighted on the options panel. The program then opens the first such file and displays each element of the correction definition in the appropriate work area. You use the program's menu to move through the set of correction requests, and approve or reject the proposed corrections. (These menu choices are described in just a bit.)

The program's caption shows how far along you are in the review, and the status of the currently-displayed request.



In this work, you may need to search your Vger system to see just how matters stand at the moment. The review program doesn't require the cataloging client to be present, but you'll probably find it useful to keep it nearby as you review correction requests.

¹⁵ The correction review program makes an ODBC connection to your Vger database to fill the drop-down lists of subject heading systems.

The correction review program's display areas

The correction review program unpacks each element of a correction request into a separate work area on its main form. The pieces of information that constitute a correction definition are distributed over three tabs. Most of the time, you'll only need to look at the *Headings* tab; occasionally you'll also need to look at the *Files, etc.* tab or the *Multiples* tab. The captions of the *Files, etc.* and *Multiples* tabs will tell you if there's something to see there.

The following picture shows a typical request displayed by the correction review program.¹⁶ The documentation for the cataloger's toolkit describes the contents of the various work areas in great detail: http://www.library.northwestern.edu/ctkv/#connection_request_own.

The screenshot displays the 'Headings' tab of a correction review program. The window title is '1 of 1. PENDING. AuthLoader20080625.155244.5073484.cor'. The interface is organized into three main sections:

- Old heading:** Contains fields for 'Tag' (430), 'Ind' (#0), and a text field with 'laReclams Universal-Bibliothek'. It also includes a 'Series numbering pattern' field, an 'Indirect subdivision' field, and a 'Subject system' dropdown set to 'a LC subject headings'. There are checkboxes for 'Ignore \$v/\$x coding differences' (unchecked) and 'Series heading' (checked). Buttons for 'Pick up...', 'Add as 4XX', and 'Add as 688' are present.
- New heading:** Contains fields for 'Tag' (130), 'Ind' (#0), and a text field with 'laUniversal-Bibliothek (Leipzig, Germany)'. It also includes a 'Series numbering pattern' field, an 'Indirect subdivision' field, and a 'Geographic subdiv. comes after' field. The 'Subject system' dropdown is also 'a LC subject headings'. There are checkboxes for 'Geog. subdiv. can follow' (unchecked) and 'Change subject code' (unchecked). Buttons for 'Pick up...' and 'New authority' are present.
- Control information:** Features radio buttons for 'Normal replacement' (selected), 'Zap a subdivision', and 'Zap whole field'. It includes several checkboxes: 'Change only whole geog. subdiv.' (unchecked), 'Change if NOT whole geog. subdiv.' (unchecked), 'Romanization change (series title)' (unchecked), 'Change only subjects' (unchecked), 'Change only non-subjects' (unchecked), 'Change only 651, not \$a of X10/X11' (unchecked), 'Move following \$p or \$n to \$v' (unchecked), and 'Preserve \$v/\$x coding during change' (unchecked). A 'Requester' field contains the value 'gls'.

When the program shows you a correction definition, you can do a number of things:

- Modify the request by manipulating information in the various work areas
- Inspect the list of persons to whom e-mail notification will be sent after the correction has been performed by selecting the *Notify* menu item. You cannot alter the list of notification recipients; you can only look at the list.
- Transfer the request to the approved queue by selecting the *Approve* menu item. This approved request contains any modifications you made to the original request when it was displayed by the correction review program. The program writes the correction request into the approved queue and deletes the original request from the pending queue. Finally, the program automatically displays the next request in the pending queue (if any) for your inspection.

¹⁶ Layout and features change slightly from one version to the next.

- Skip the request (leaving it in the pending queue) by selecting the *Next* menu item. The program displays the next request in the pending queue (if any) for your inspection; the skipped request remains, unmodified, in the pending queue.
- Move to the previous request by selecting the *Previous* menu item. If you move back to a request you have previously deleted or approved, you can send it back to the pending queue by selecting the *Make pending again* menu item.
- Delete the request from the pending queue by selecting the *Delete* menu item. The program deletes the file from the pending queue, and then displays the next request in the pending queue (if any) for your inspection.

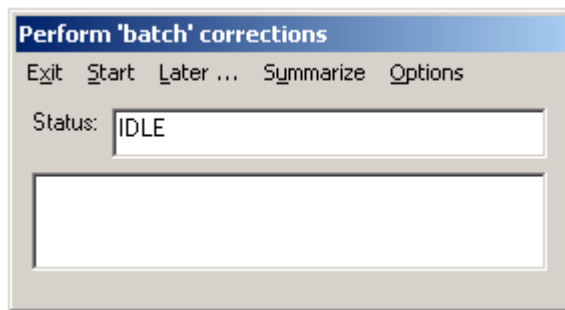
When you arrive at the end of the files to review, you can click the *Begin review* menu item to begin the review again, and re-visit any requests you may have skipped over, and to pick up any requests that have been added to the pending queue since you began your review. (You can do this while a review is in process, or after you've finished a pass through all of the requests.) Until you begin a new review, you have the opportunity to change your mind about things you've marked for deletion or sent to the approved queue. Once you begin the review again, all of your decisions become final. The correction review program will build a new list of just those requests that remain in the pending queue, and present them to you.

Performing corrections

Preliminaries

The final program in the heading correction stream is the one that actually changes records. This program pulls requests one at a time from the approved queue¹⁷ and follows the instructions it finds. This is the *correction receiver* program.

In the normal course of things, you'll just start up the correction receiver program and leave it alone as it does its work; for this reason, the program doesn't have much of a user interface. Here's what the program looks like when you start it:



There's nothing on this panel you can modify directly. The Status box shows the program's current status: *idle*, *searching* for records that contain a heading, actually *performing* a correction, *pausing* between requests, and so on. The background color of the Status box also changes: red if the program is actively modifying records, green if it is waiting for something to do, white if you haven't started the program yet. As the program works its way through requests in the approved queue, it changes the contents and the background color of the Status box. If the program is busy performing a correction, the box below the Status box shows the new form of the heading; otherwise, this box is blank.

The program contains a menu with a small number of choices:

- 'Exit' cancels the program. If the program is in the middle of a correction when you make this choice, the program puts the current correction definition back into the approved queue before quitting.
- 'Start' tells the program to begin its work of correcting headings immediately.
- 'Later ...' tells the program that you want it to look in the approved queue for correction requests, but not quite yet. The program first shows you a form that allows you to indicate exactly when you'd like the program to start, and when you'd like it to stop. (See below.) Once the program starts, it looks from time to time into the approved queue; when the program finds a request in the queue, it performs the request immediately. The program stops looking in the approved queue when you've told it to stop, even if there are additional requests in the queue.¹⁸
- 'Summarize' tells the program to prepare a report of the work it has completed in previous sessions. When you click this, the program reads its summary report files¹⁹ and creates a text

¹⁷ Whether written directly to the approved queue by qualified operators, or moved from a pending queue to the approved queue with the correction review program.

¹⁸ The program only considers the 'stop' time when it's looking in the approved queue for something to do; it performs all of the work connected for a given correction request, even if that work extends beyond the indicated 'stop' time.

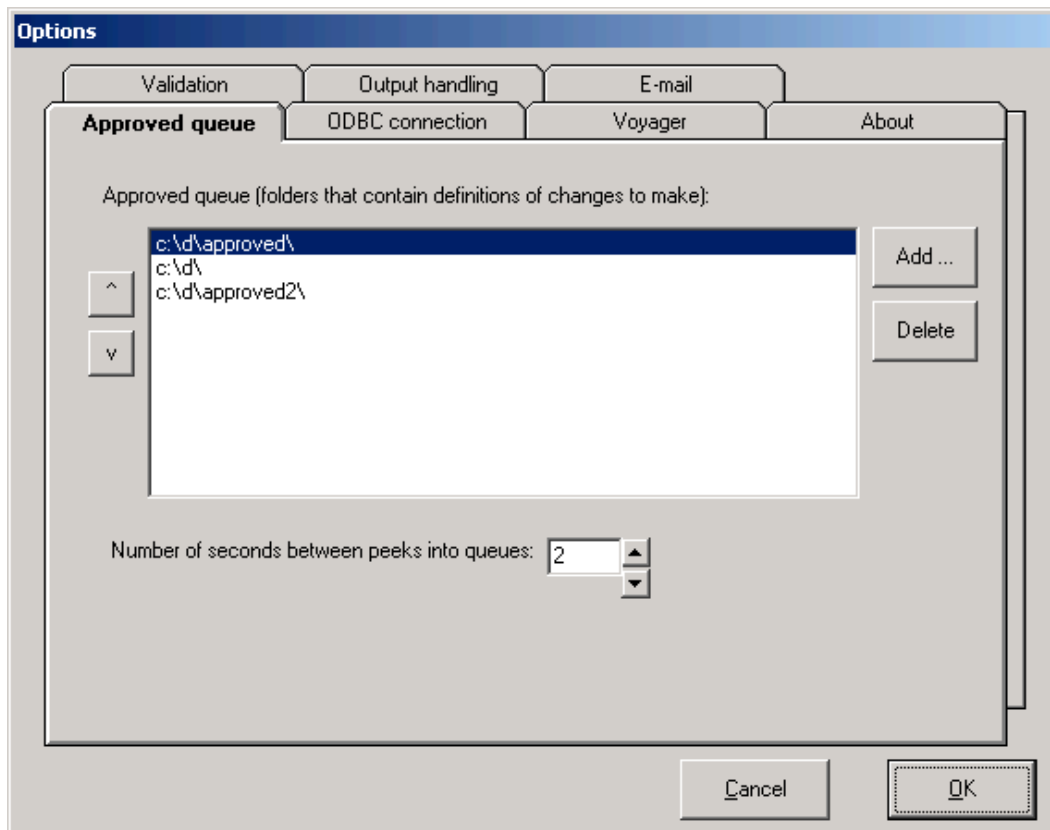
¹⁹ The program only generates statistical summary report files if you ask for them; request them by checking the 'Write summary report' box on the General tab of the Options panel.

file named VgerHdg.FinalSummary.txt. After clicking this menu item, you'll find the report file in the same folder as the summary reports whose contents it distills.

- 'Options' brings up the program's configuration panel.

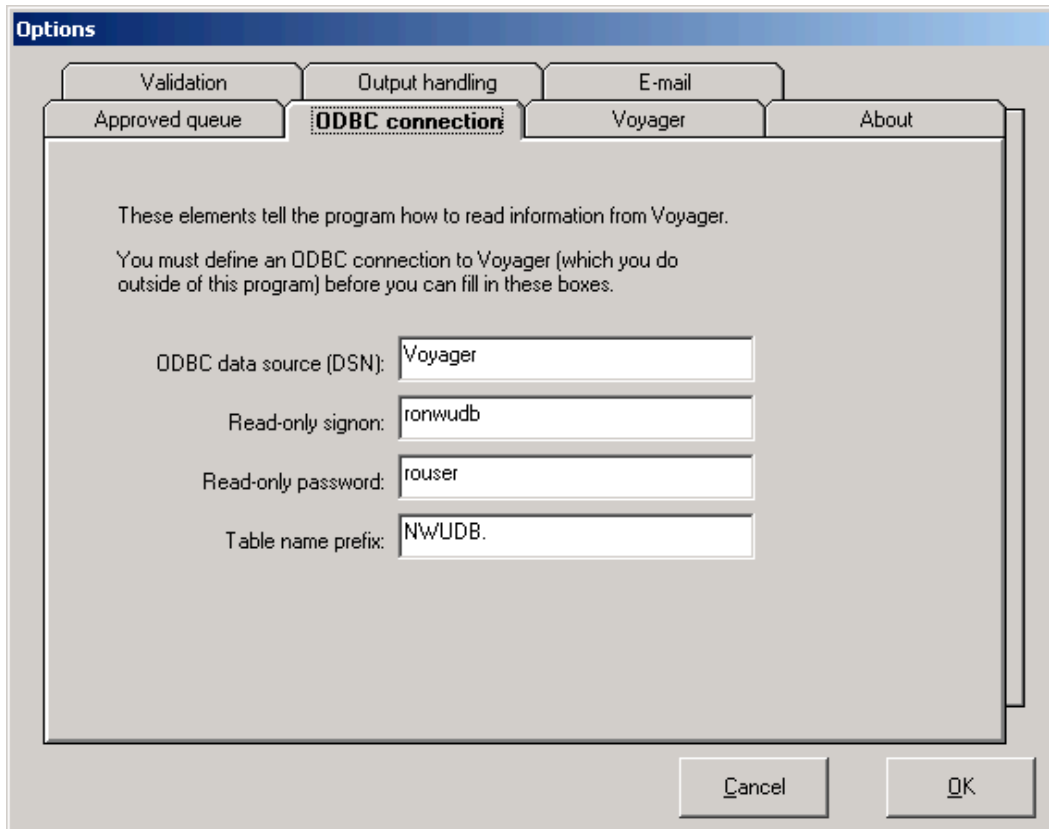
Configuration

Before you can use the correction receiver to perform requests, you must supply several pieces of information on the program's Options panel. You find the Options panel by clicking 'Options' on the program's menu bar. Except as noted, you must supply *all* of the information on *all* of the tabs, or the correction receiver program won't work properly.



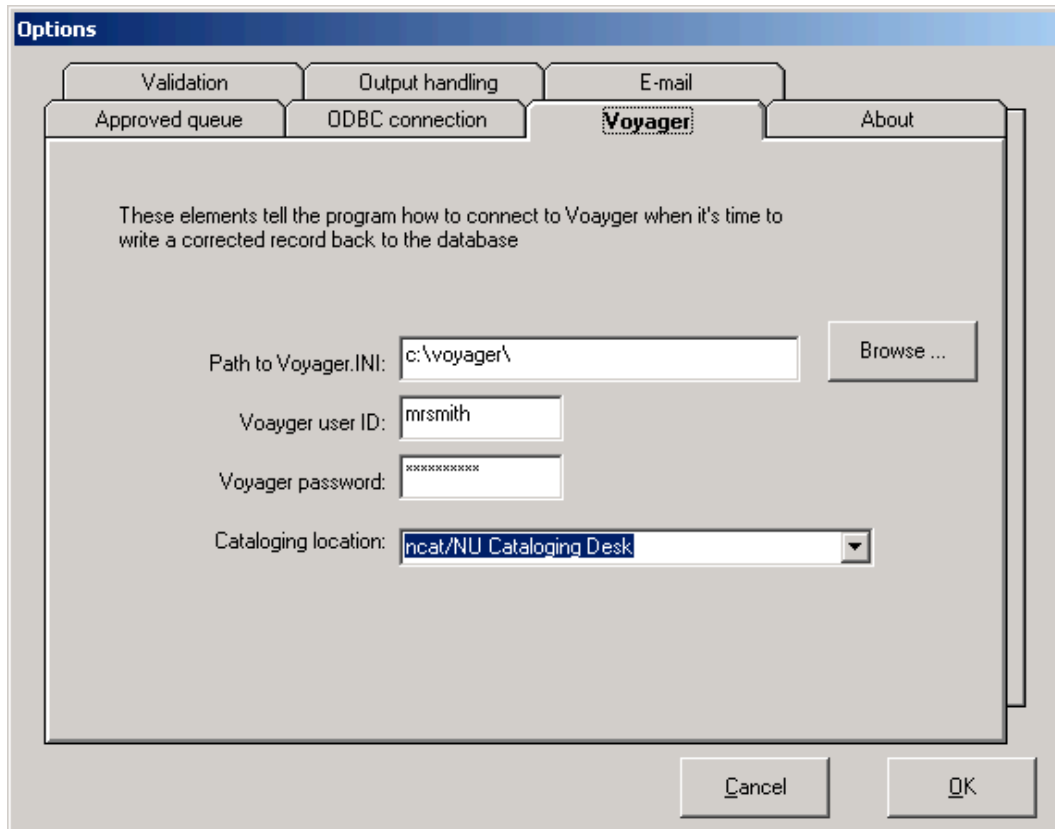
Information on the 'Approved queue' tab identifies the queue of approved correction definitions, and tells the program how to use the queue.

- *Approved queue*: the names of the folders that serve as the containers for approved correction requests. You can define as many approved queue folders as you want. The program will only read from the folders whose names are highlighted.
- *Number of seconds between peeks into queues*: After the program finishes processing one request in the approved queue, it waits a number of seconds (indicated here) before it checks the queue for another approved request. Use the up/down arrows to change the amount of time the program pauses between requests. (The maximum wait time is 20 seconds.)



The 'ODBC connection' tab tells the program how to connect to your Vger system for searching and for retrieving records. You must install and configure the ODBC drivers before you can use the correction receiver program to change records.

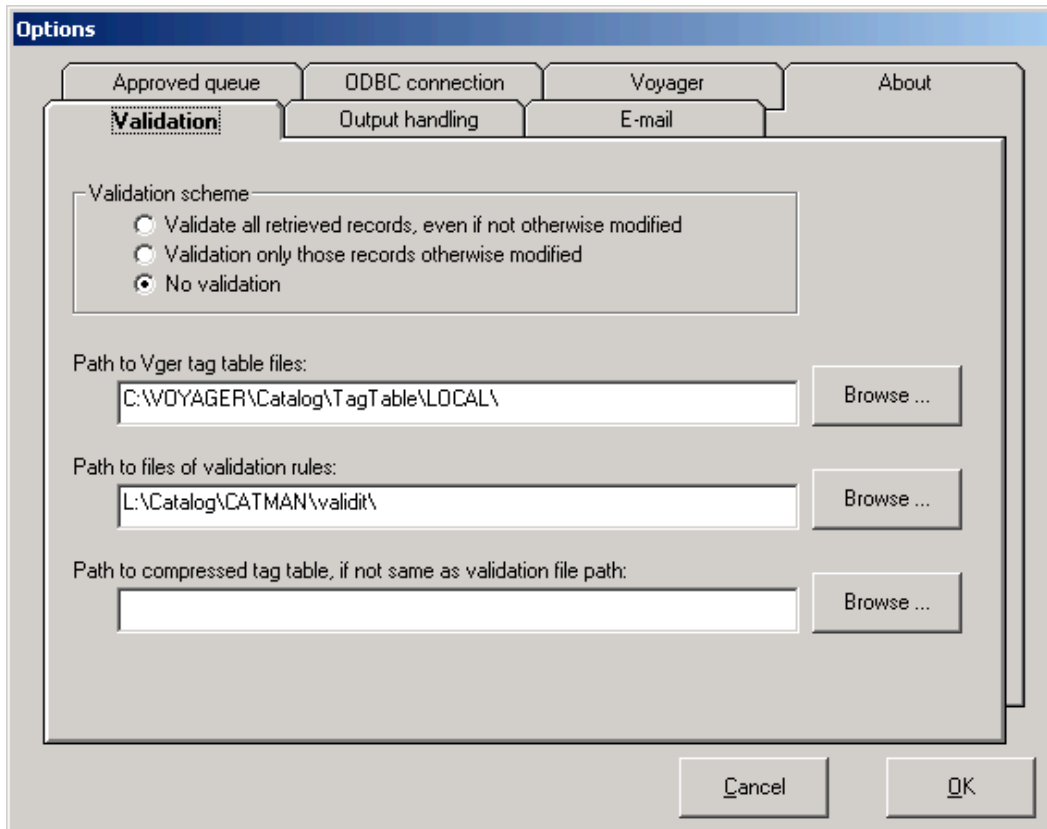
- *ODBC data source (DSN)*: The name of the ODBC connection that points to your Vger database. This is the DSN you define for your Vger database in the ODBC Data Source Administrator client.
- *Read-only signon* and *Read-only password*: The logon and password that the program should use for read-only access to your Vger database. This is a logon and password defined *to Oracle* by a system administrator, and is unrelated to any Vger logons.
- *Table name prefix*: The name of the database at your Vger installation whose records are to be corrected. A Vger installation contains at least two databases: the production database and a training database. Each of these has a different name. The name of a Vger database typically consists of some text plus the letters 'DB': 'BIGDB.' and 'CORNELLDB.' are typical examples. This name must end with a full stop.



The 'Voyager' tab gives the program additional information about your Vger database. You must run the Vger client installation program before you can use the correction receiver program to change records.

- *Path to Voyager.INI*: The folder that contains the 'Voyager.INI' file. This must be the folder that also contains the version of the Vger cataloging client you use work on Vger records.
- *Voyager user ID* and *Voyager password*: The user ID and password, defined in the Vger system administration client, that grants permission to modify bibliographic records. This user's name will appear in the Vger history display for records modified by the program. (You may wish to define a Vger operator ID solely for use by the correction receiver program, so the program's work doesn't get mixed up with a 'real' operator's work.)
- *Cataloging location*: The Vger cataloging 'happening' location from which the program should pretend to be sending records. (It really doesn't matter which location you select. You may wish to create a cataloging 'happening' location for use *only* by the correction receiver program.)²⁰

²⁰ The 'Cataloging location' box contains the *location name* of cataloging 'happening' locations as found in your institution's Vger location table. (The list is limited to those cataloging locations defined for use by the operator named on this tab.) The list in this illustration may look a bit odd because at Northwestern University Library (whose configuration this illustration represents) a location's name consists of the location code, a slash, and an equivalent for the code. (Example: 'nstx/Main Stacks'.) This allows those who assign codes to find the correct one quickly, while also giving sufficient information to other staff, who aren't expected to know all of the codes.



The 'Validation' tab lets you tell the validation tool contained within the program how to go about its work. The validation tool attached to the correction receiver program and the work it performs are described in the toolkit's documentation:
http://www.library.northwestern.edu/ctkv/#suppl_val_rules_file.

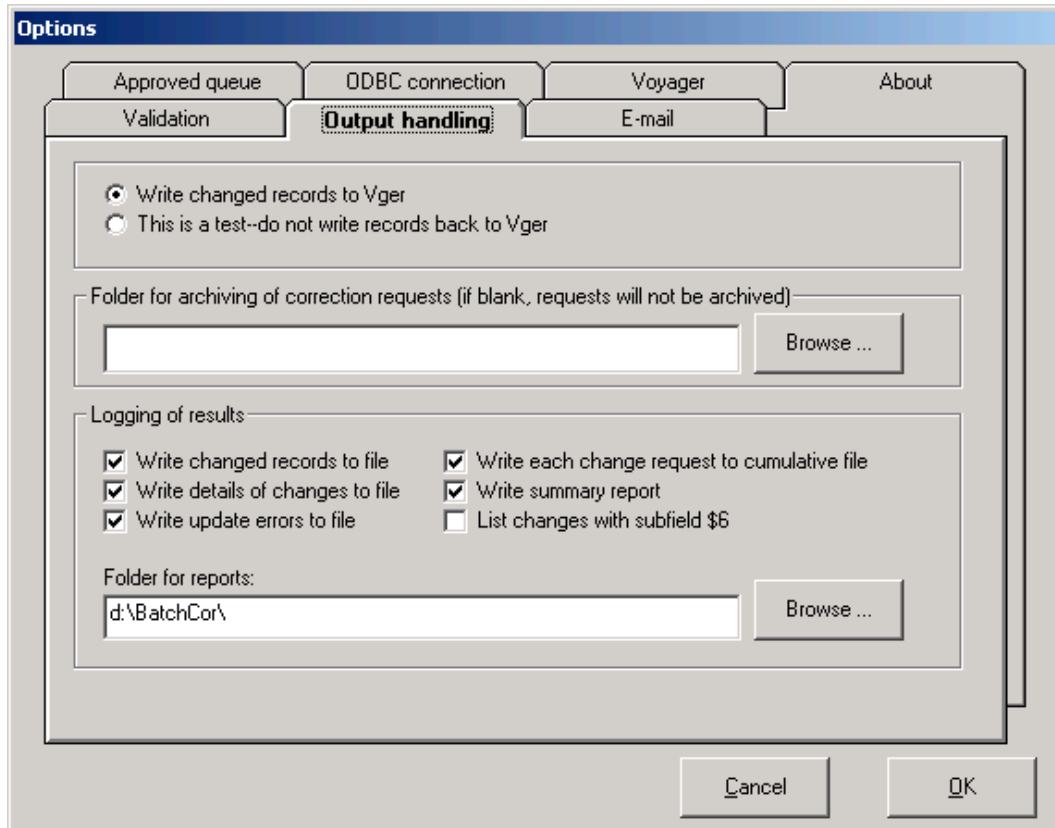
- *Validation scheme:* As mentioned in the description of the correction review program, the correction receiver program makes available the services of a *validation tool* you can use to make changes to records other than changes to their headings. In the normal course of its work, the correction receiver program does one or more searches to identify the set of records likely to require correction, and then looks at each record in more detail; the program usually inspects more records than it ends up correcting. With the appropriate choice in this box, you can ask the correction receiver program to:
 - Send *all* retrieved records to the validation tool. This includes both records to which the program makes a heading change, and those the program retrieved but ended up not changing.
 - Send to the validation tool for additional changes *only* those records to which the correction receiver program made a heading change.
 - Not send *any* of the records it retrieves to the validation tool.

If you select either the first or the second radio button in this box, you must fill in the first two 'Path' boxes on the tab—and the third 'Path' box, if the compressed tag table is not in the same folder as the files of validation rules. If you select the 'No validation' button in this box, you can leave the 'Path' boxes blank if you want—the program will ignore whatever is in them.

- *Path to Vger tag table files:* The folder that contains your Vger tag table files. This is the folder that contains such files as 'amarcfix.cfg' and 'hmarc8xx.cfg', not the folder that contains 'lang.cfg'.
- *Path to files of validation rules:* The folder that contains the basic set of validation rules defined for the validation tool. (In addition to this basic set, each correction definition may

refer to one file of supplementary rules. The supplementary rule file is defined within each correction request.)

- *Path to compressed tag table*: If the ‘compressed’ tag table used by the validation tool is in the same folder as the validation rules files (this is the usual setup), you can leave this box blank; if the compressed tag table is in some other folder, that folder’s name belongs here.



On the ‘Output handling’ tab, you tell the correction receiver program how to dispose of the records it has modified, and what kind of reports you would like to see.

- *Write changed records to Vger* and *This is a test—do not write records back to Voyager*: If you check the ‘Write changed records to Vger’ radio button, the program will make changes to records and then send the changed records back to your Vger database; if you click the ‘This is a test’ radio button, the program will make changes to records in memory only, prepare its reports, and then throw the changes away. Check the ‘This is a test’ radio button if you wish to test a correction request before you perform it; check the ‘Write changed records to Vger’ radio button when you want the program to change your records.²¹ If you correct

²¹ Here’s what actually happens if the ‘This is a test’ radio button is checked. When you click ‘Start ...’ from the program’s menu, the program retrieves from Windows the name of the standard folder for temporary files, deletes all ‘*.cor’ files that may be present in that folder and copies all of the correction requests from all of the currently-highlighted approved queues into that folder. (This means that you can’t also use the Windows temporary folder as your approved queue.) The program then works on the files in the temporary directory, as if the temporary directory were the sole approved queue; the program deletes each file in the temporary directory as it handles each correction request. At the end of this work, the temporary directory contains no correction requests at all, and the original approved queue is untouched. This method of proceeding allows you to determine the effect of one or more correction requests, without disturbing the requests in the approved queue: if everything in the log files appears satisfactory,

records as a test, the program quits as soon as it has tested every correction request in the approved queue; it doesn't continue to look into this queue for additional requests.

- *Folder for archiving correction requests:* If you wish the correction receiver to preserve your correction requests for future reference, give in this box the name of the folder the program should use. If you give a folder name in this box, the correction receiver will move into this folder each correction request after it is complete. (If you give here the same folder as the approved queue folder, the program will rename correction requests with the extension "archive" and leave them in the same folder.) If you do not wish the correction receiver to preserve your correction requests in an archive file, leave this box empty.
- *Logging of results:* These choices allow you to define the kind of log files (if any) you wish the program to maintain. You can ask the program to write all changed records (in MARC format) to a file;²² to create a detailed text file listing each change made to each record;²³ to create a statistical summary of its work;²⁴ and to write a description of each change to a cumulative text file.²⁵ You can ask to be notified if any changed field contains subfield \$6 (linking the changed field to an 880 field with vernacular representation of the same data).²⁶ If you want the program to create any of these log files, the program will put them into the folder named in the 'Folder for reports' box.

you can immediately run the program again, this time with the 'Write changed records to Vger' button selected.

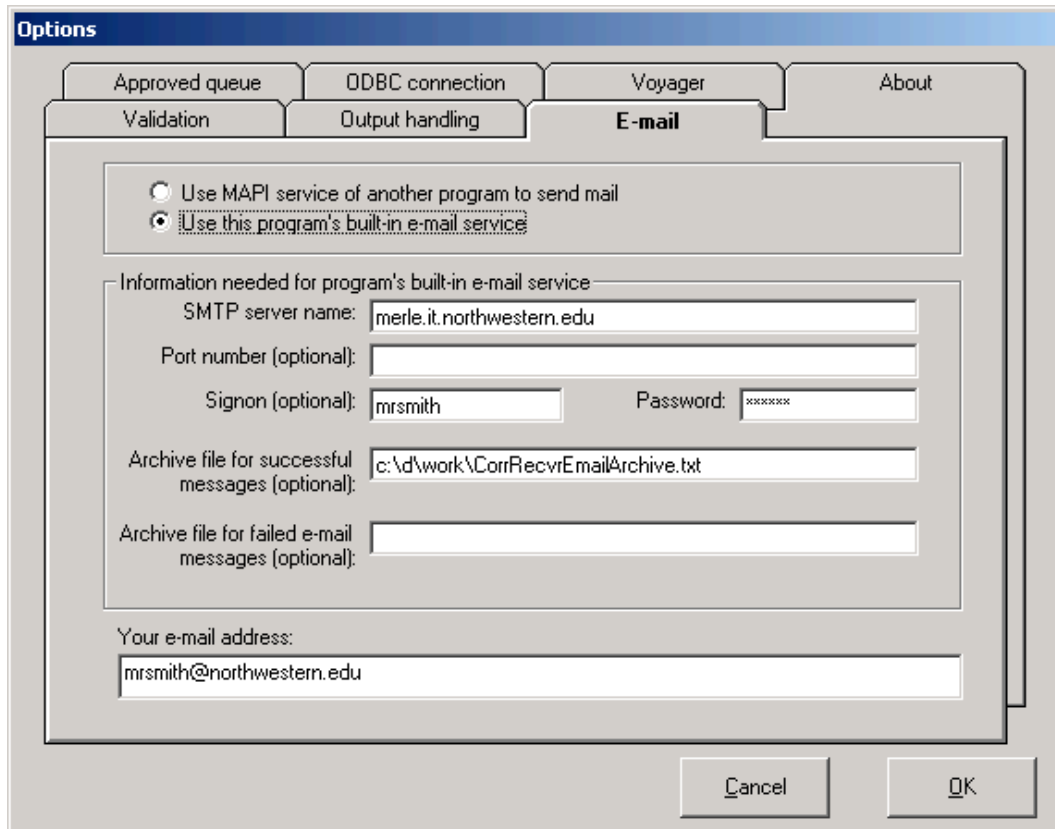
²² These files have names in this form: 'VgerHdg.<date>.<time>.Changed.mrc'. The date and time used in these files is the time the program started up, not the time the program performed a particular correction.

²³ This file is handy when you're previewing the effect a request will have on a record (i.e., when the 'This is a test' radio button is checked), but will have less value when you ask the program to write changed records back to Vger. These log files have names in this form: 'VgerHdg.<date>.<time>.Details.txt'.

²⁴ It is these statistical reports that the program draws on when you select Summarize from the program's menu bar. If you do not ask the program to create these files, it cannot later generate a summary for you. These files have names in this form: 'VgerHdg.<date>.<time>.Summary.txt'.

²⁵ This file can be of great use for the person who should come along afterwards and make sure that everything that should have been changed did in fact get changed. (Errors in tagging or subfield code assignment can cause the program to fail to make a correction you would otherwise expect.) This file is called ChangeRequests.txt; it is a cumulative file, and grows until you delete or rename it.

²⁶ The correction receiver makes no change to 880 fields. If you are interested in keeping 880 fields parallel with the rest of the record, use this list to guide additional changes to bibliographic records.



On the 'E-mail' tab you give the program information it needs to send e-mail notifications of corrections performed. You must tell the program how to send e-mail messages.

- If you use an e-mail client that is configured to allow other programs to ask it to send messages via the MAPI protocol, select the 'Use MAPI service' radio button; the program will not display the 'Information needed for program's built-in e-mail service' frame.
- In all other cases, select the 'Use this program's built-in e-mail services' radio button. In the 'Information needed for program's built-in e-mail service' frame, fill in at least the server address. Fill in the port number, signon, and password as necessary. If you wish the program to archive successful or failed e-mail messages, give file names in the corresponding boxes.
- *Your e-mail address*: The complete e-mail address of the account under which the program should send e-mail notifications about completed batch corrections.

The Options panel also contains an 'About' tab, which gives the version number of the program and of its components. In case of problems with the program, you should include this information in your message.

Testing a set of batch corrections

When you are just getting comfortable with the program, or at any other time you aren't sure whether your correction request is formulated as narrowly as necessary or wish to preview the results of a correction for any other reason, you can ask the correction receiver to perform its work as a *test*. When running as a test, the program will go through all of the motions of identifying records to correct, making changes to them and preparing log files; but the program will not actually write the changed records back to Vger. You should carefully inspect the log files to make

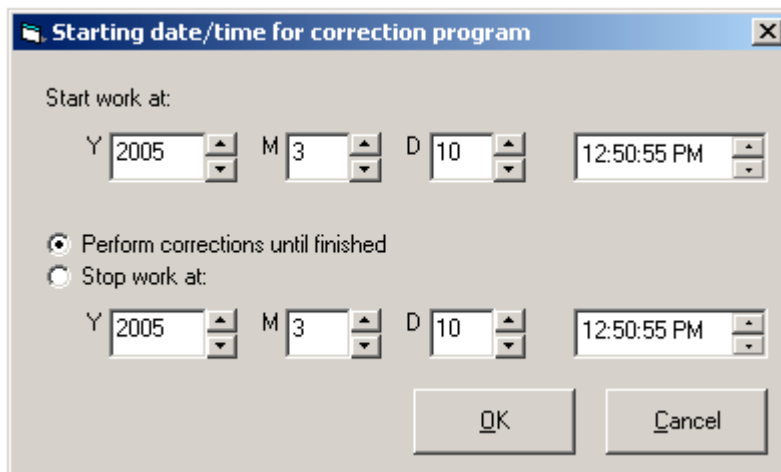
sure that all of the records you expect to be changed (and no others) are being changed, and that the changes being made are in fact the changes you expect.

To test a set of batch corrections, start the correction receiver program and immediately open the Options panel. On the 'Output handling' tab, click the 'This is a test' radio button. Click the 'OK' button to approve the change, and then select 'Start' from the program's menu. The program will process all of the requests in the approved queue, and cancel itself when it is done. At the end of the test, all of the original correction requests are still in the approved queue. Inspect the program's log files before using the program to make any changes.

Using the program to change headings

To use the program to perform corrections (i.e., those corrections whose definitions reside in the approved queue), you should inspect all of the values on the Options panel²⁷ and then select 'Start' or 'Later ...' from the program's menu bar.

If you select 'Later ...', the program shows you its starting date/time dialog:



The starting date/time dialog allows you to set the time at which the program should start looking for corrections to perform, and to set the time it should stop working on corrections. The dialog is primed with information that tells the program to start looking in the approved queue immediately and to perform all corrections it finds there until it's told to stop, but you can change these settings to suit your needs. You can use the arrows on the various boxes²⁸ to change the date and time at which the program will start to look into the approved queue for work to do, and to set the date and time at which the program will stop its work and cancel itself.

When you click the 'OK' button on the starting date/time dialog, you're telling the program you're ready to have it start performing corrections. If you've told the program to start immediately, it will do so; if you've told the program to start at a later time, it will patiently wait until that time arrives, and then begin its work. (If you want the program to start immediately and run until it's finished, there's no need to use this dialog at all.)

²⁷ Once you've got the program set up the way you want, you don't actually have to re-visit the Options panel each time you run the program, except when performing a test.

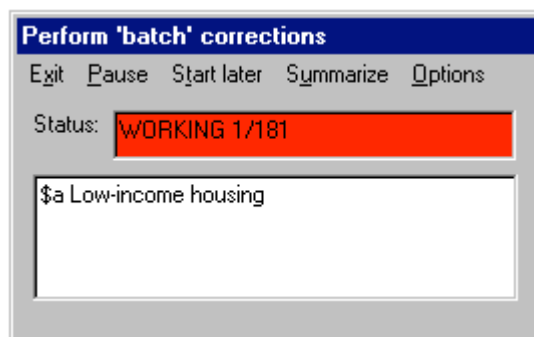
²⁸ For the boxes that contain hours, minutes and seconds: click on one element in the time (hour, minutes, seconds, AM/PM) and then use the arrows to change that one element.

Note for ILC SO participants: The correction receiver program will automatically configure itself for you so that it begins working at 10:00 p.m. and finishes at 2:00 a.m. You do not need to use this dialog to set starting and stopping times. When you're ready to perform corrections, simply start the program by Selecting 'Start' from the menu; the program will wait until 10:00 p.m. to begin.

When the program begins to work, it displays the text *ACTIVE* in its status box, and waits a few seconds. This brief wait period gives you time to cancel the program, should you change your mind about performing corrections. When the proper time has elapsed, the program peeks into the approved queue for a correction request.

If the program doesn't find a request in the queue, it simply waits another interval, and looks again. If the program finds a request in the approved queue, it unpacks the request and deletes the file from the approved queue.²⁹ The program searches Vger for records that contain the heading of interest (the status box reads *SEARCHING*), and modifies those records as instructed (the status box reads *WORKING*).

As it modifies each record,³⁰ the program sends the corrected record back to your Vger database and/or writes information to log files. The box below the Status box shows the new form of the heading the program is working on. Here's an example of what the program looks like while it's performing a correction:



When the program has finished its work on one request, the program writes a summary of its activity up to this point.³¹ If at the time the original request was formulated the operator asked for e-mail notification to be sent to one or more parties, the program will send an e-mail notification describing the correction and its outcome.³² (Appendix A shows a typical e-mail message.) The

²⁹ If you've told the program to archive correction requests, it moves the correction request to the archive folder instead of deleting it.

³⁰ In addition to the text *WORKING*, the Status box shows the number of records retrieved so far from Vger (i.e. the number of the current record), and the total number of records the program is going to retrieve. This allows you to gauge the program's progress on a request. This display is expressed in terms of records *retrieved*, not records *corrected*. The correction receiver program often retrieves records it subsequently discards as irrelevant—but the program sends back to Vger for update only those records it has actually modified. The program's statistical summary file tells you how many records were changed.

³¹ Because the name of this report file includes the date and time at which the current program session begin, this file will overwrite any previous file created by the program during the same session. This is a *cumulative* file, and summarizes the program's activity during a session. The program writes this file after performing each correction request, so that in the case of a catastrophe as little information as possible will be lost. In the normal course of things, you'll only see, or be interested in, the final version of this file.

³² If you use MAPI and if there is any difficulty during the sending of this e-mail message (such as: MAPI client not configured properly), it's possible that the correction receiver program will

program waits another interval (as indicated on the Options panel) and peeks again into the approved queue for another request. If it finds one, it performs it immediately; if it does not find one, it looks again in a little while, and waits until a request appears.

After you've used the program to perform all the requests you wish it to perform (typically: when the approved queue is empty), select Exit from the program's menu to cancel the program. The program writes the final version of its summary report (if you've requested that it prepare such a report), and then cancels itself. You should use the cumulative list of change requests (assuming you've asked the program to prepare this list) to review the records changed by the program, and, more importantly, to find any headings *not* changed by the program.

At suitable intervals, use the program to prepare a summary of its activity (assuming you've asked the program to write its statistical detail files), and then delete the individual statistics files.

freeze while it waits for a response. (This problem, if it occurs at all, occurs after one correction is complete and before the next one is begun; so there's no harm done to the Vger database.)

Appendix A: e-mail notification

At the time a correction request is generated, the operator may request that an e-mail notification be sent to one or more operators when the correction receiver has completed its work.

The e-mail message's subject line shows the 'old' heading (or at least the first part of it). The body of the message gives:

- the original request file name
- the name of the archive version of the request (if applicable)
- the old and new headings
- timings for the begin and end of the correction work
- counts of the number of records inspected, and the number of records changed

Here is a typical example of an e-mail notification sent by the correction receiver program:

```
Subject line: Correction complete for: Poor Housing

Correction request file:
g:\catalog\catman\GLS.20050309.112507.780514953.cor
Request file archived as:
g:\catalog\corrarch\GLS.20050309.112507.780514953.cor

From: 450:  :|a Poor | xHousing
To: 150:  :|a Low-income housing

Work began at 20050309 11:28:17
Work ended at 20050309 11:28:45

Records retrieved: 31
Records changed: 31
```