The Concordance Database Manual

M. Alan Haley

Apress®

The Concordance Database Manual

ISBN-13: 978-1-59059-603-6

ISBN-10: 1-59059-603-X

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Lead Editor: Jim Sumser
Technical Reviewer: Sean King
Editorial Board: Steve Anglin, Ewan Buckingham, Gary Cornell, Jason Gilmore, Jonathan Gennick, Jonathan Hassell, James Huddleston, Chris Mills, Matthew Moodie, Dominic Shakeshaft, Jim Sumser, Keir Thomas, Matt Wade
Project Manager: Sofia Marchant
Copy Edit Manager: Nicole LeClerc
Copy Editor: Susannah Pfalzer
Assistant Production Director: Kari Brooks-Copony
Production Editor: Katie Stence
Compositor: Linda Weidemann, Wolf Creek Press
Proofreader: Elizabeth Berry
Indexer: Valerie Perry
Artist: April Milne
Cover Designer: Kurt Krames
Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail `orders-ny@springer-sbm.com`, or visit `http://www.springeronline.com`.

For information on translations, please contact Apress directly at 2560 Ninth Street, Suite 219, Berkeley, CA 94710. Phone 510-549-5930, fax 510-549-5939, e-mail `info@apress.com`, or visit `http://www.apress.com`.

The source code for this book is available to readers at `http://www.apress.com` in the Source Code section. You will need to answer questions pertaining to this book in order to successfully download the code.

CHAPTER 7

Administrative Functions

Concordance databases are rarely static. You can reasonably expect to receive new sets of data; data is often corrected and requires updating; document records are occasionally deleted; users are regularly editing and annotating records; the number and roles of users change—these and other requirements mean you'll be faced with a series of daily or weekly tasks that are necessary to keep Concordance running efficiently.

Indexing Databases

When dealing with full text, data must be indexed before it can be searched. *Indexing* refers to a process in which Concordance scans appropriate fields and identifies textual units; that is, *words*. When a search is applied to a Concordance database, the search is made of the database's index, not the actual underlying record data itself. This method is what causes Concordance to search data so quickly: the index is a simplified roadmap of where data is stored. The existence and position of a word is stored in the index, which contains a pointer to the record or records that match the search criteria.

The concept is familiar to anyone who has searched for information in a book that contains an index. If you were to search for a given topic by starting at the beginning of the book and reading through to the end, finding the actual topic could take some time. If you refer instead to an index in the back of the book, you can find the topic listed with a corresponding page, and locate the appropriate subject matter quickly. This is similar to how Concordance locates words according to a user's search criteria.

Concordance identifies textual units in the following way: strings of alphanumeric characters that are separated by white space or common punctuation marks are considered to be words. Numbers, single letters, indefinite and definite articles, and prepositions are ignored. Consider the following quote:

> *Lady in thy orizons, be all my sinnes remembred.*

It contains seven words that Concordance regards as meaningful: Lady, in, thy, orizons, all, sinnes, remembred.

The words "be" and "my" are ignored because they're so common in the English lexicon; a search that included them would likely produce many false positives. The words would be a valid hit from a technical perspective, but not useful to the user. For a list of all words that are ignored, you can access and modify the Stopword list from the File ➤ Dictionaries ➤ Stopword list menu.

Punctuation can be important in terms of how Concordance recognizes textual units. Concordance recognizes the characters in Table 7-1, by default, as common punctuation.

Table 7-1. *Characters That Concordance Uses As Punctuation*

Character	Description
-:	Hyphen
/:	Forward slash
.:	Period
,:	Comma
;:	Semicolon
‘:	Single quote apostrophe

You can modify this list from the File ➤ Modify menu (see Figure 7-1). You might wish to exclude the hyphen character to retain searches of hyphenated last names, such as Mary Scott-Smith. By default, Concordance treats this name as three words: Mary, Scott, and Smith. If the hyphen is removed from the punctuation list, Concordance will treat the name as two words, Mary and Scott-Smith.

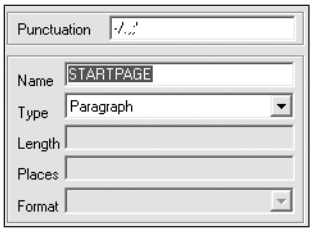

Figure 7-1. *You set punctuation that's used during indexing to identify textual units from the File ➤ Modify menu.*

■ **Note** Modifying the punctuation list requires a full reindex of a database for changes to take effect, and for searches to return accurate results.

Dictionary and Inverted Text Files

Concordance uses two files to facilitate searches, the *dictionary* and *inverted text* files. Both files have the same name of the database being indexed, but have file extensions .DCT and .IVT, respectively.

The dictionary file is a list of every textual unit identified by the index process. Each unique word is stored in the dictionary file only once. In this way, Concordance will build a full vocabulary of every word used in the database, akin to a dictionary.

The inverted text file contains information that identifies the location of each occurrence of a word as it appears in document records that define a Concordance database. This means

that a word listed in the dictionary file will have one or more entries in the inverted text file, depending on how many times that word appears through any record in the entire database.

Working in conjunction, these two files identify the following:

- That a word exists in the database
- Where that word appears in the document records

Indexing vs. Reindexing

To index a database means to rebuild the dictionary and inverted text files completely. To *reindex* a database means to update them. Indexing is always necessary when data is first loaded into an empty database. Reindexing is necessary when indexed fielded data is edited, when data has been deleted, or when more data has been added to a database. Note, too, that actions you take to modify the contents of a database, such as altering a database's synonym or Stopword lists, mandates a reindex before these changes are incorporated into searches.

You're provided with a visual cue from the File ➤ Reindex menu by means of a small check mark that appears next to the word "Reindex." If the check mark is there, the database must be reindexed.

Optimizing Indexing

Indexing and reindexing are resource-intensive procedures. Because of the way that Concordance is configured, you should be aware that the machine that performs the index is the client workstation that calls the procedure, not the server on which database files are stored. (This statement assumes a client/server network installation.) This means that the performance of an index is commensurate with the speed of the calling client workstation's hardware, and how much Random Access Memory (RAM) the client workstation has allocated for indexing—a setting made from within Concordance itself.

You can configure a workstation's RAM from the Tools ➤ Preferences menu. This opens a multi-tabbed Preferences dialog. The tab labeled Indexing contains options that you can use to optimize the procedure (see Figure 7-2).

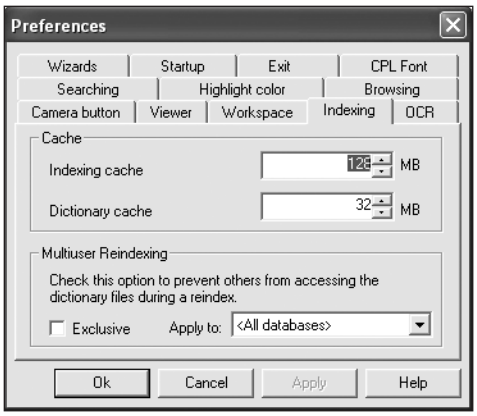

Figure 7-2. *The Indexing tab of the Preferences dialog*

- *Indexing cache*: The value in this field corresponds to the amount of RAM, in megabytes, devoted to indexing by the client workstation. RAM is used for this purpose only while the index procedure is running. When an index has completed, RAM is returned to the workstation and will be used by other programs. In general, the higher this value, the less time it will take an index to complete. However, the value shouldn't equal the total amount of available RAM in the workstation, as the machine's operating system itself requires RAM. How much RAM a workstation's operating system needs varies by operating system. Microsoft recommends that Windows XP be allocated at least 128MB of RAM for it to function properly. This means that if a workstation running Windows XP has 512MB of RAM installed, 128MB should be reserved for XP, leaving 384MB to be divided between the Indexing and Dictionary caches.
- *Dictionary cache*: Unlike RAM used during indexing, Concordance uses RAM allocated for the Dictionary cache continuously, to store data pertaining to various list files, security files, database key files, and—when invoked—by indexing procedures. This RAM isn't released back to the operating system while Concordance is active. Dataflight recommends a setting of 4MB for the Dictionary cache. In the preceding example, where 384MB was calculated to be the amount of RAM that could be split between the Indexing and Dictionary caches, 380MB can be allocated for indexing.
- *Multiuser Reindexing*: Users can continue using a Concordance database while it's reindexing, though this will cause the reindexing itself to proceed more slowly. The effect is small, but if the speed of a reindex is at a premium, you can override this default setting using the Exclusive check box. A database in exclusive mode locks the dictionary file, preventing users from accessing it so that searches cannot be initiated. You can apply the setting to the current database or to all databases using the "Apply to" drop-down box.

Scheduling Indexing Tasks During Times of Nonusage

No damage will be done to a database if an index is triggered during normal work hours when users are actively querying and using Concordance. Administrators and users should be aware, though, that until an index is complete, searches aren't accurate. It's only when the indexing is completely finished that searches can be fully trusted. To this end, you might find that it's best to perform this administrative task at a time when users don't need a database.

Packing Databases and Dictionary Files

Packing is a Concordance-specific term that refers to removing records marked for deletion (when packing the database itself) or updating a database's dictionary files (when packing the dictionary) so that they operate more efficiently. Both are accessible from the File ➤ Pack menu.

Packing a Database

When a Concordance database is packed, those records that have been marked for deletion are removed from the underlying data files, in effect removing document records from the database. Deleting records from a Concordance database is a two-step process: records are

marked for deletion first, then—at a later time—the database is packed. Records aren't completely removed from a Concordance database until the pack has been completed. This separation of steps means that the administrator (or another user with appropriate rights) can audit and verify the records marked for deletion before they're actually removed.

Deleting records is a terminal process, and should be invoked with care. Because document records can contain subjective metadata (document-level tags and annotations), removing a record can also remove the intellectual effort of a user. The deletion cannot be undone.

You can mark records for deletion from the Edit ➤ Delete and Undelete menu. This opens the Delete/Undelete Records dialog (see Figure 7-3). This tool grants the user the ability either to mark records for deletion, or to reset documents that had been previously set for deletion, so that they won't be removed.

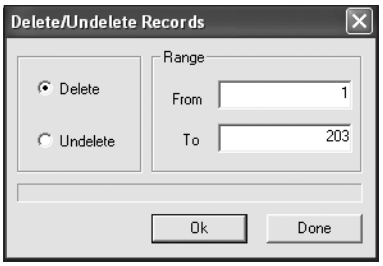

Figure 7-3. *Selecting records for deletion or undeletion*

A user must enter a document range. The document range refers to the ordinal position of the records that are displayed with the last active query issued to the database. If no query has been issued to the database so that all records are visible, the From and To values correspond to the first and last records in the entire database. If the last active query refers to some subset of records in the database, the First and Last values correspond to the first and last records in that query.

If, out of all records in a given database, you wish to mark the third record for deletion, you must set the From and To fields to be the same value: 3.

When a record is marked for deletion, the record will display the letters DEL in the lower right-hand border of the Concordance window. To display only those records that have been marked for deletion, you can click the tagging icon located at the top of the Concordance screen. This opens the Tag/Issue Management dialog (Figure 7-4). Clicking the "Deleted records" button invokes a search of all records in the database that have been marked for deletion, and is the active query when the tool is closed.

You should create a complete backup of the database before records are deleted. This is an additional layer of insurance if records are determined to have been erroneously removed. Because deletions cannot be undone, if it's determined that some or all of the records must be restored, you must either reactivate the backup as the primary database, or move deleted records from the backed-up version of the database to the primary version.

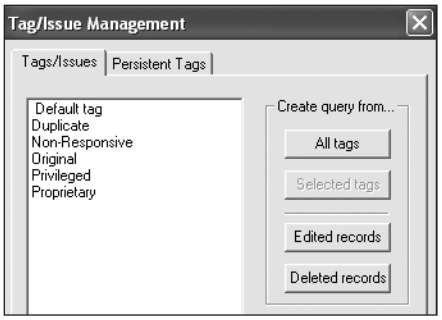

Figure 7-4. *You can use the "Deleted records" button to query for just those records that have been marked for deletion.*

Packing the Dictionary Files

Choosing File ➤ Pack ➤ Database causes the dictionary and inverted text files, essential for speedy search and retrieval, to be compacted. During the life cycle of a database, as it's indexed and reindexed, these files can contain entries that are scattered across a hard drive. Packing these files causes entries to be physically ordered on a drive so that successive entries are contiguous. This reduces the amount of time that the drive head must span the drive to retrieve data. In other words, the hardware itself is able to find data more quickly.

Zapping a Database

Zapping a database means erasing all records. It's the equivalent of marking all records for deletion, and then packing both the database itself and its dictionary file. You trigger zapping from the File ➤ Zap menu. You're prompted to confirm the operation. Zapping a database cannot be undone. When the zap is complete, the database is an empty shell, leaving only field names and data types. Note that any settings applied to Concordance's security model aren't affected.

If no security is applied to a database, the menu item is accessible to all users. Given the potential for catastrophe, users should be blocked from accessing this menu item. You can restrict menu access using the Concordance security model, described later in this chapter in the section "Menu Access."

Before zapping a database, it's best to make a complete backup of all database files.

Deduplicating Records

When a single document is represented by two or more distinct rows in a Concordance database, duplication occurs. Although this concept might seem obvious, you must use some care when identifying duplicates. Because a Concordance record is represented by a series of fields, the question must be asked: what combination of fields that contain equivalent data causes two or more records to be regarded as duplicates? The answer can vary depending on how document records are interpreted.

Selecting Duplication Criteria

We might regard records that contain exactly the same full text to be duplicates. This might be true in terms of our intuition, but what if the SOURCE field of two documents that have the same full text are different, where the SOURCE refers to the individual from whom documents were collected? If it's important to users to know who had the same document in their possession at the time of document collection, records with the same full text, but with different SOURCE values, are not duplicates. The test of duplication depends on the values in both fields.

Another method to identify duplicates might be to locate those records that have the same data across *all* fields in a database. At first glance, this appears to be an absolute method of identifying duplicates, but what if the database contains an accession field, so that each document record has a unique, numerical identifier? If the test for duplication involves the comparison of every field in the database, no records will be identified as duplicates because the autonumber field will have a unique value in each field.

Because of varying interpretations of what criteria should be used to identify duplicates, you should refer the question to end users, so that those who are intimately familiar with the matter represented by the Concordance database can define how duplicates should be identified.

Concordance provides a tool to mark records as *original* or *duplicate* (or other values you specify). You can open it from the Tools ➤ Check for duplicates menu. The Duplicate Detection dialog lists all fields in a database (with data type), so that you may select one or more to be used in the duplicate checking process (see Figure 7-5). Once you highlight the desired fields, the Ok button triggers the procedure. A running total of the number of duplicates identified is displayed under the Duplicate Count label.

■ **Note** When fields that have a PARAGRAPH data type are selected as part of the criteria, only the first 60 characters of those fields are used during the comparison. This means you cannot, in fact, analyze the entire contents of a database record that contains PARAGRAPH fields to determine duplication.

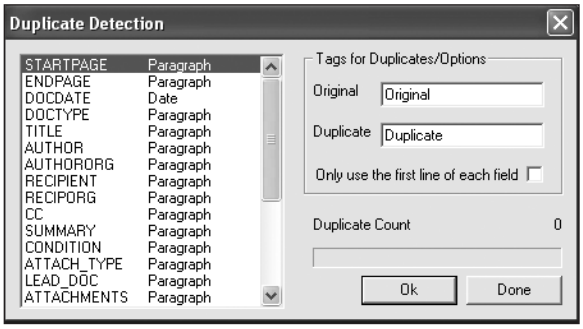

Figure 7-5. *Selecting records to be identified as duplicates*

Original vs. Duplicate Tags

When document records are identified as duplicates, they're given document-level tags that, by default, are labeled original or duplicate. The assignment of these tags is arbitrary, in that the first record used as the record to be compared to is given the original tag, and other records that appear later in the database and that fit the duplicate criteria are given the duplicate tag. You can change the names of these tags from the Duplicate Detection tool, and you should, in fact, change these values for subsequent checks for duplicates, as the labels are reused. For example, if two unrelated checks for duplicates are run against the database, and if the labels of the original and duplicate tags aren't changed, both batches will share these tags, making it difficult to group duplicates together according to criteria.

Security

By default, a new Concordance database doesn't have any security applied. If a database contains data that isn't of a sensitive nature, it might not be necessary to enable security at all. However, in many applications it's necessary to restrict the information that users can view, and more importantly, to prevent unauthorized users from accessing data. However, until you explicitly enable security in a Concordance database, any user that has access to the program and possesses the appropriate network permissions can open a Concordance database and view Concordance.

When enabled, Concordance adds a layer of security over regular network permissions and enforces that security from within Concordance itself. This means that users who have full network rights to read, write, execute, and delete files in the Concordance directory can still do so, but when they open a Concordance database that's secure, they're restricted by Concordance itself from modifying or viewing data from within the program. For example, while users in Concordance might be prevented from deleting an entire database from the File ➤ Zap menu, they can still delete all Concordance files from the database's directory if their network permissions allow them to do so. A thorough treatment of security in Concordance involves both Concordance's own security model and users' network permissions as well.

Concordance's security model is applied to each database instance. To administer security for a specific database, you must open that database first. There's no external tool for administering security, nor is there a method to apply security across several sessions of multiple databases.

Security in Concordance affects a user's ability to read (and search) data from specific fields, or to access menu items. Restrictions are placed on fields within records, not on the records themselves. If a database contains records that a user shouldn't view, you must remove those records from the database entirely or prevent the user from accessing the database.

When security is applied in Concordance, passwords and other security settings are stored in the database's .SEC file. If a database is named DOCREVIEW.DCB, its security file will be named DOCREVIEW.SEC. This file won't exist until security has been enabled. Once security is activated, deleting this file will lock all users out of the database, thereby rendering it unusable.

Managing Security

You manage security from the File ➤ Administration ➤ Security menu, which opens the Security dialog. Because security isn't applied by default on a new Concordance database, any user will be able to access this tool. A login screen is displayed with the user name OWNER. The password is initially blank (see Figure 7-6).

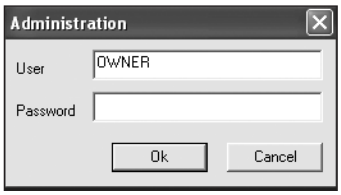

Figure 7-6. *When "Login required" is enabled from the Modify dialog, this login screen will pop up after a user has activated Concordance, but before the program opens.*

When the Security dialog is opened for the first time, there will be two preexisting accounts: *default* and *owner* (see Figure 7-7). Owner is an actual account that should be reserved for the administrator. Default is an account that's used when an individual provides Concordance with a login name that isn't recognized by the database. Removing this account effectively locks out all unauthorized users.

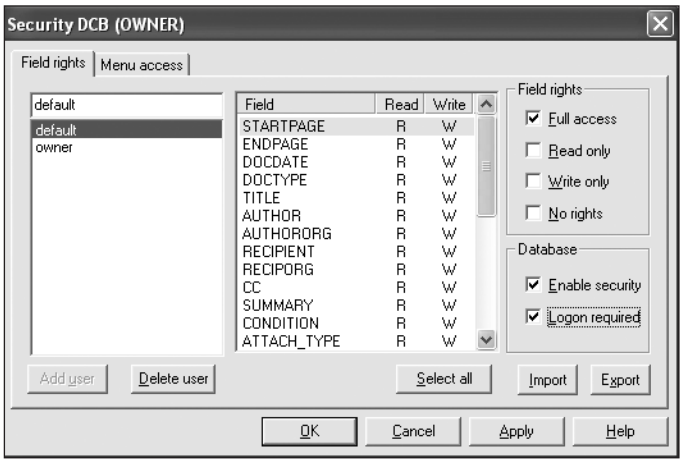

Figure 7-7. *The Security dialog, accessible from the File ➤ Administration ➤ Security menu item*

■ **Note** Although the default and owner accounts are preset by Concordance, they aren't activated until you check the "Login required" check box.

Managing Users and Field-Level Permissions

You can use two settings, individually or in combination, to secure a database. They are to "Enable security" or to require a login ("Login required"). When "Enable security" is selected, field-level permissions are enforced. When "Login required" is selected, users are required to submit a valid login and password before a database will open. The most secure method combines these features: security is enabled and a login is required. If the default account has been deleted, this will grant access only to those users with user names that you've added, and will prevent those users from viewing or writing to fields for which they have no permission.

■ **Note** If both "Login required" and "Enable security" are checked, but no field-level permissions have been set, users will be unable to view any fields in the database.

Managing Users

You add or delete users from the "Field rights" tab of the Security tool (see Figure 7-8). To add a user, click into the text field that appears on the left-hand side of the tab just under the label "Field rights," add the desired user name, and then click the "Add user" button. The user's name is added to the list box. To delete a user, select the name from the list, and then click the "Delete user" button.

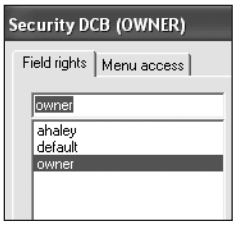

Figure 7-8. *User names that have been added to the database*

When "Login required" isn't checked, Concordance will capture the user's network login name, if there is one. If "Enable security" is checked, Concordance will attempt to match the user's network name with the list of names saved in Concordance's security model. If Concordance finds a match, the user will proceed with the field- and menu-level settings that have been stored in Concordance under his or her name. If Concordance doesn't find a match, it will grant the user permissions defined by the default user. If there's no default user account name, the user cannot use the database.

When an account is initially created, its corresponding password is blank. There's no menu item or button to change or set a password; this feature is built into the login screen if the user enters his or her password in a specific way.

To change an existing password, at the login prompt the user should enter his or her old password, a forward slash (/), and the new password, in this form:

```
oldpassword/newpassword
```

This indicates to Concordance that the user wishes to change his or her password. Concordance responds with a new login/password screen with the title Confirm New User/Password. The user should type the new password, and then click the Ok button. The new password is now set.

To set a password for the first time, the process is slightly different. When an administrator creates an account, that account's password is initially blank. The first time the user attempts to log in to the Concordance database, he or she should enter the desired password and then click the Ok button. A confirmation login screen appears that prompts the user to type the desired password again. Clicking the Ok button sets the password and opens the database. In future sessions, the user is required to use this same password. If the user wishes to change the password, the preceding method is used, with the following form:

```
oldpassword/newpassword
```

Setting Field-Level Permissions

Field-level permissions are applied uniquely to each user. This means that, each time you create a new user account, if "Enable security" is checked, you must take the time to determine how the user can interact with every field in the database. You must apply this same care when modifying the structure of a database. After creating a new field, you should immediately set permissions for that field, for all users. You must reset existing field-level permissions for a field that's renamed via the File ➤ Modify menu, for each user. Because of the additional overhead involved in modifying the structure of a database for which the security model is mature, you'll want to approach structural changes to a database methodically.

To set permissions for a field, click the desired field, highlighting it, and then select one of the four options—Full access, Read only, Write only, and No rights—described in the following sections. Pressing the Ctrl key on the keyboard allows for multiple selections. The interface also provides a Select All button to highlight all fields.

Four security settings are available for each field. These settings are exclusive, despite the check boxes that appear next to the respective labels (check boxes usually indicate multiple selections are possible). For example, a field cannot be both "Read only" and have "No rights."

Full Access

The user can both read and write to the field. Note that field-level permissions affect administrative functions as well. For example, a user granted the ability to index the contents of a field can't do so successfully unless he or she is granted full access to the field.

A byproduct of the fact that changing the name of a field resets its permissions means that, when the field has been renamed, even the owner account loses full access to it. Changing the name of a field prevents you from modifying a database unless you use the Security tool to grant your account full access to the renamed field.

Read Only

The user is able to search and view data in the field, but not modify it in any way. Even administrative functions such as loading data are affected, so that when a user attempts to load data into a Concordance database, those fields for which he or she has read-only access won't appear in the list of fields available in the database.

The effect of setting a field to read only for an individual user is similar to the effect of setting the field to read only for all users from the Data Entry Attributes dialog, accessible from the Edit ➤ Validation menu: the field is locked, though visible, when a record is opened for editing. However, setting a field to read only from the Data Entry Attributes dialog supersedes individual field rights. A field set to read only from the Data Entry Attributes dialog remains locked to a user even if he or she has full rights to the field from the Security tool.

■ **Note** Whether a user can load data to a field is determined by the field-level permissions set using the Security dialog, not from the Data Entry Attributes dialog.

Write Only

The field is hidden to the user from the Concordance graphical user interface, though the user can interact with the field via administrative functions such as loading or overlaying data. Otherwise, the field isn't available for searching, viewing, or editing.

This setting also prevents a user from modifying a database.

No Rights

The user may not search, edit, or view the field.

Menu Access

You can apply security to menu items in the same way that it's applied to fields, enabling items for some users and disabling them for others. This is particularly important, given that some of the items accessible from Concordance's menus can permanently alter data. To safeguard a database, you should take the time not only to control which fields a user can view, but also which menu items a user can access.

Modifying menu-level access can be accomplished from the "Menu access" tab of the Security tool (see Figure 7-9). Here, all menu items are displayed in an expandable tree. You can navigate through this tree to select the lowest-level menu item, then activate or deactivate it per user as needed. Each node on the tree has a check box. When a check box for a particular node in the tree appears with a gray color, this indicates that the node contains sub elements—other menu items—and that some of them have been disabled.

You should be careful when modifying menu access, as you can render the database unusable. For example, if you remove access to the File menu, you'll lose the ability to modify the database or administer security. As a result, you should ensure that your account has access to *every* menu item.

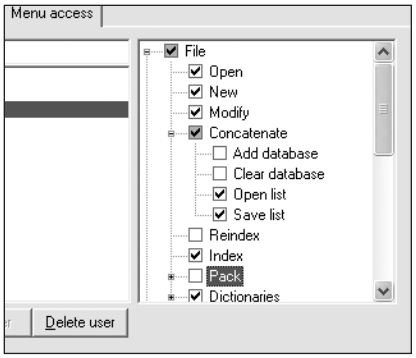

Figure 7-9. *Menu-level access, from the Security tool. Note that items that have a check mark but that also have a gray color indicate that they have child menu items that have been disabled.*

For regular users who should have no access to the administrative features contained in Concordance, you should hide the following menu items:

- *File menu*: New, Modify, Concatenate, Reindex, Index, Pack, Dictionaries, Status, Administration, Zap, Begin Program, Edit Program
- *Edit menu*: Validation
- *Documents menu*: Import, Export, Replication
- *Tool menu*: Convert to V8, List file management, Preferences

However, Concordance offers a series of presets that will aid you in assigning menu item privileges to users according to the role they assume when using the database. Table 7-2 summarizes each role.

Table 7-2. *Preset Roles Provided by the Concordance Security Model*

Role	Disabled Menu Items	Purpose
Supervisor	None	The Supervisor role is one of a super user. The Concordance supervisor is an administrator with complete access to all menu items, and therefore, all administrative tools in the database.
Administrator	*File*: Administration (and all submenus), Added menu items	The Concordance administrator has access to most administrative functions in the database, but cannot set security or modify menu items. This role is appropriate for one who oversees the daily function of a Concordance database.

Continued

Table 7-2. *Continued*

Role	Disabled Menu Items	Purpose
Editor	*File*: New; Modify; Reindex; Index; Pack; Administration (and all submenus); Zap; Begin Program, Edit Program *Edit*: Validation, Delete and undelete *Documents*: Replication (Enable replication, Create a replica, Purge events) *Tools*: Check for duplicates, List file management	This is the most common role for users who both read and write to document records. All menu items relating to administration are disabled, while all menu items relating to the editing of individual document records are enabled.
Researcher	*File*: New; Modify; Reindex; Index; Pack; Administration (and all submenus); Zap; Begin Program, Edit Program *Edit*: Append, Edit, Validation, Global edit, Delete and undelete, Find, Find again, Replace, Send to *Documents*: Import (Delimited text, E-documents, Overlay); Export (Structure) *Tools*: Toolbar (Edit toolbar), Check for duplicates, List file management, Preferences	Users who are researchers have the ability to search and view data. They cannot edit document records, and they have no access to Concordance's administrative features.
No access	*File*: New; Modify; Concatenate (and all submenus); Reindex; Index; Pack (and all submenus); Dictionaries (Database dictionary, Stopword list); Administration (and all submenus); Zap; Page Setup; Print Preview; Print; Begin Program, Edit Program *Edit*: Append, Edit, Validation, Global edit, Delete and undelete, Undo; Cut; Copy; Paste; Find, Find again, Replace *Search*: All submenus disabled *Documents*: Import (Delimited text, E-documents, Overlay); Export (To a delimited text file, Structure); Print documents *Reports*: Report writer *Replication*: All submenus disabled *Tools*: Bell; Empties; Split screen; Toolbar (Main, Browser toolbar, Edit toolbar, Report toolbar, Review toolbar, Table toolbar), Check for duplicates, List file management, Preferences	This role is appropriate for the *default*, or unknown, user. When a user attempts to open a Concordance database where security is enabled, if Concordance doesn't recognize the user's login, it will grant the user default access. The "No access" role allows the user to open other databases, or close Concordance.

Carrying Security Between Databases

When records from a secure database are exported to another Concordance database, the source database's security model doesn't export with the document records. This means that an administrator could export data from a database for which security has been carefully prepared to an unsecured database, thus circumventing Concordance's security model. You should use care when transferring data between databases to ensure that security isn't breached.

Security settings are portable, though, using the Security tool. The Export button allows you to save user names, field-level permissions, and menu item permissions to an external,

comma-delimited file. This file can then be imported into other databases using the Import button. For a company or firm with a large user base, keeping a basic template of user permissions can save you a great deal of time.

Adding Custom Menu Items

The File ➤ Administration ➤ Added Menu Items menu opens the Added Menu Items dialog displayed in Figure 7-10. You can use this tool to create new menu items, set their placement on the menu bar, and configure them to trigger customized actions, such as to open another database, to print a report, or to activate a program written in the Concordance Programming Language (CPL). CPL programs are described in greater detail in the section "The Concordance Programming Language," later in this chapter. Top and second-tier–level menu items can be added. Menu items can be configured to appear only in the currently active database, or in all databases, and menu items can be configured only to appear to certain users.

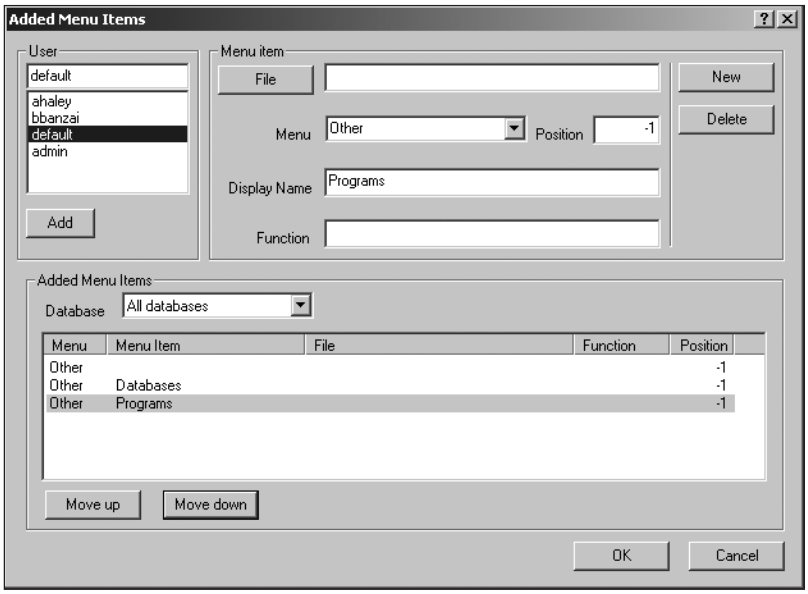

Figure 7-10. *The Added Menu Items dialog*

You must define the database in which the menu item will appear. You determine database scope by the Databases drop-down box under the Added Menu Items label in the middle of the dialog. If the menu item should appear in all databases, you should select the "All databases" option.

You must also select the users to whom the menu item will appear by clicking and highlighting user names in the User list box. Note that the contents of this list box aren't prepopulated with entries from a database's security settings. You must add users by entering their login names in the empty text field under the User label, and by clicking the Add button. You should use some care when entering login names, as they should match the users' actual names as

defined by the Concordance security model (if it is used) or as defined by a network security model. (Recall that if security in a Concordance database is enabled, but a login isn't required, Concordance will capture a user's network login and use that value when determining how security should be applied for the user.)

Note that any menu item associated with a default user will be accessible to all users. To use this feature, you must create the default user with the methods described in the previous paragraph.

You must define access for custom menu items for each user from the Added Menu Items dialog. The other method of configuring menu-level security, from the "Menu access" tab of the Security dialog (accessed from the File ➤ Administration ➤ Security menu) applies only to Concordance's default menu items.

To add a top-level menu item, select the database scope and the user to whom the menu will be visible. Enter the menu name in the Menu drop-down box. If the menu item is to appear at the end of the menu bar, the Position of the menu item should be -1. Otherwise, the number in this field corresponds to the vertical position of the menu item on the menu bar. For example, if the File menu is the first top-level menu item, and your custom menu item is to appear next to it, the value in the Position field should be 2. The Display Name field is left blank. If the top-level menu is to trigger an action, you can associate a file with the item by clicking the File button and selecting a program to run. If the top-level menu item is to take no action and will contain other menu items, this field should be left empty.

To add a submenu item under a parent, select the database scope and the user to whom the menu will be visible. You select the parent menu name from the Menu drop-down box. The value in the Position field is irrelevant. The name of the submenu item is entered in the Display Name field. If the submenu item is to trigger an action, you can associate a file with the item using the method described in the previous paragraph. You can set the position of submenu items using the Move Up and Move Down buttons.

You can enter the path and name of a valid CPL program in the File text field. You use the Function text field when CPL programs are invoked by menu items. Some CPL programs contain *functions* (a section of programming code). If the name of a function in the CPL program referenced in the File text field appears in the Function text field, it's activated by the menu item.

Concatenation

Concordance features a way to conjoin up to 128 databases so that they appear to the user as a single, virtual database. One database is designated as primary (Concordance refers to it as the *main* database); when a user opens it, he or she sees all records in that database, plus all records in other databases that appear in the primary database's *concatenation* file. The concatenation file is an ASCII text file that contains file paths to the .DCB files of other Concordance databases. It has the same name as the .DCB file for the main database. For example, if the main database file is DOCREVIEW.DCB, the concatenation list, if there is one, will be named DOCREVIEW.CAT.

Both indexing and searching are applied on all records in all databases that have been concatenated if those actions are triggered from the main database. The concatenated databases themselves aren't altered in any way, and can be opened and used individually.

When Is It Necessary to Concatenate a Database?

There are times when it may be useful for you to create several databases to represent a single matter. The reasons for this might be for performance or purely administrative.

The theoretical limit of a single Concordance database is 32 million document records, although you'll find that, in practice, users will experience database performance issues before a database becomes that large. For example, if the database contains several PARAGRAPH fields, indexing will take longer. When index time becomes prohibitive, regardless of the number of records in the database, you should consider splitting the database into parts. You can index each database individually, perhaps at separate workstations, and the procedure will complete more quickly. Though searching across a series of concatenated databases can take a little longer, the concatenation itself will otherwise be invisible to the user.

There might also be times when security that is to be applied to a series of related databases isn't uniform: fields for some records should be restricted, while others may not be. Here, you may segregate groups of records that share security characteristics into different databases, apply security to each one individually, and then concatenate them into a single, virtual database. The scope of security is specific to each database, so that the security model applied to the primary database doesn't supersede that of secondary, concatenated databases.

Another reason to concatenate databases is one of simple management: it might be easier for you to create separate databases that are related to the same matter, but that represent slightly different material per database, or that come from a variety of sources. You can color code records in each database so that users may see, at a glance, which records come from which database, and therefore, from which source.

■ **Note** To change the color of records in a Concordance database, use the Font button  from Browse or Table view.

How Concatenation Works

Concatenation works like this: one database is designated as the main database. If other databases have been concatenated, they'll appear in a database concatenation file—a file that has the same name as the primary database, and that has a .CAT extension. Note that there are no restrictions as to where the individual databases are stored across a network. As long as users have the appropriate permissions to read and write to files in all folders referenced in the concatenation file, the virtual database will behave normally. However, concatenated databases spread over a Wide Area Network (WAN) might experience latency issues, and perform more slowly than a series of concatenated databases that reside on the same network server.

■ **Note** For each user, you should consider placing each database in its own dedicated folder.

Databases with different structures can be concatenated. Users can use Concordance's table layout from the primary database to alter the appearance and order of fields of all concatenated databases (see Figure 7-11).

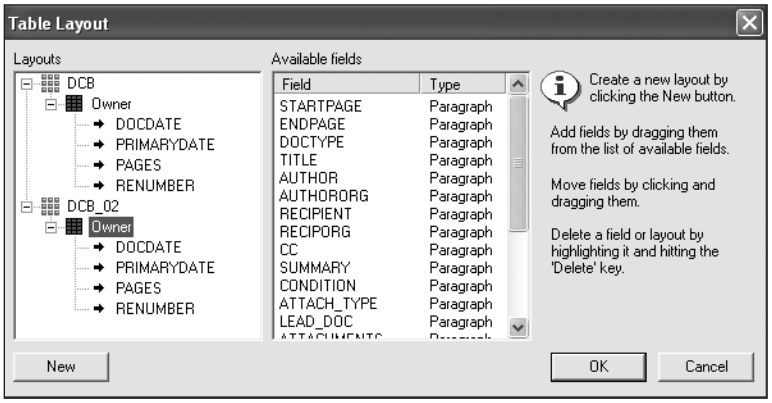

Figure 7-11. *When databases are concatenated, the user can alter the fields that appear in the Table view of Concordance, for each database. In this configuration, the fields displayed for both the DCB.dcb and the DCB_02.dcb databases are the same, though this isn't a requirement. The user could elect to see fields in a different order when records appear from different databases.*

Unless the structure of all databases is exactly the same, it's recommended that you perform data maintenance on each database individually. For example, say the main database contains ten fields while a secondary, concatenated database contains only five. Exporting data from the main database will export *all* records from both databases, though data from the second database won't line up properly with the first. Despite this caveat, indexing from the primary database will trigger across all secondary databases, regardless of each database's structure.

To group databases together, you should open the designated primary database first. Until a concatenation list for a database is created, there is no .CAT file, so when you create a Concordance database for the first time, you won't see it. Only after the list has been created and saved will Concordance create the appropriate file. Once it has been created, you'll find that the .CAT file is an ASCII text file that can be edited from any text editor. However, all modifications to the .CAT file can be accomplished from the Concordance interface.

The following options are available from the File ➤ Concatenate menu, or from the Join button 

that appears at the top of the Concordance screen:

- *Add Database*: This option opens the Concatenated Databases tool (see Figure 7-12). The Concatenated Databases tool lists the main database and all secondary databases that will open with the main database. You may use this tool to both add and delete entries from the .CAT file.

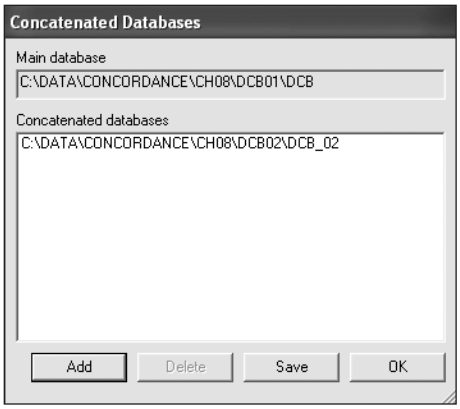

Figure 7-12. *Deleting and adding databases to the main database's concatenation file*

- *Clear Database*: Selecting this option doesn't delete a .CAT file. Rather, it closes all concatenated databases except for the main database. It's a quick way to search just the main database.
- *Open List*: This allows the user to select any concatenation file, which then opens all databases in that list. Any database currently open and active is first closed.
- *Save List*: A concatenation file is created and all currently opened databases are added to it.

The Concordance Programming Language

The Concordance Programming Language (CPL) was created specifically for the purpose of augmenting Concordance's native functionality. CPL programs can perform useful procedures not otherwise available from the tools that can be opened and invoked from within Concordance, or you can use the CPL programs to automate existing Concordance procedures.

During installation, Concordance creates several CPL programs, and places them in a folder named CPL off the main installation directory. These programs have been written and tested by the manufacturer, and can be quite useful to an administrator. Other CPLs are available for download from the CPL Library section on Dataflight's Web site at `http://www.dataflight.com/cpl.library.html`. Programs in the library are grouped according to their use: Administration, Import/Export, Printing, and so on (see Figure 7-13).

Programs written in CPL can have a .CPL extension, and are ASCII text. They can be opened by a text editor, or opened by Concordance from the File ➤ Edit Program menu. When you use this menu item to open a CPL program, you're prompted for the program's location. When you select it, Concordance opens the program in its own simple text editor (see Figure 7-14).

Figure 7-13. *The CPL section of the Dataflight Web site*

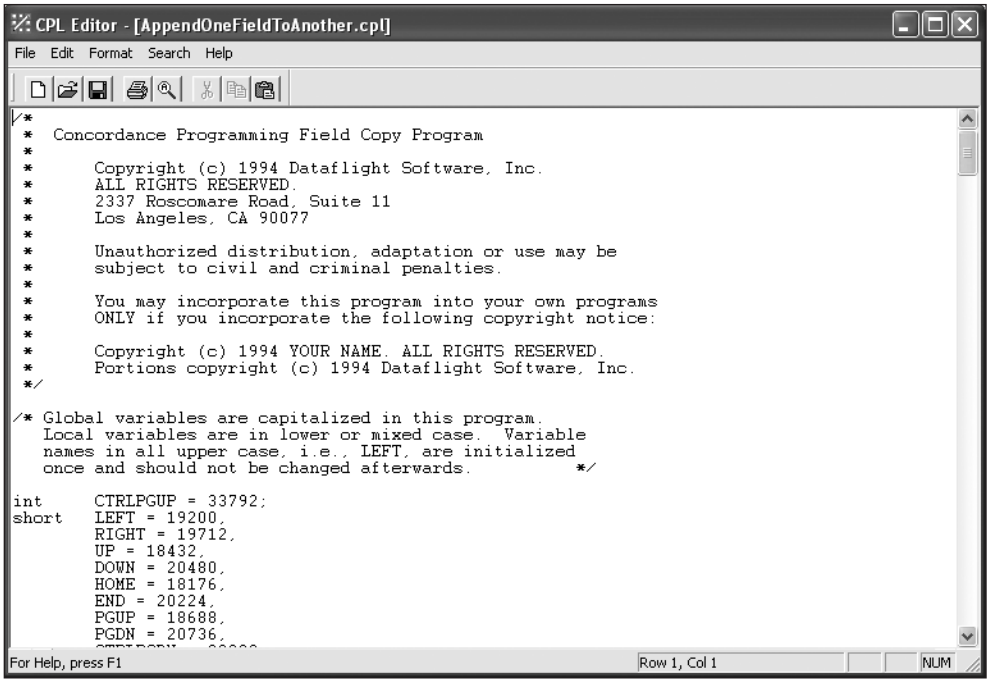

Figure 7-14. *A CPL program opened from the File ➤ Edit Program menu*

CPL programs can also have a .CPT extension. .CPT program files are created the first time Concordance successfully runs a program with a .CPL extension. A .CPT program file is a compiled equivalent to a .CPL file, and runs more efficiently because it has been converted to machine code. Note that if a .CPL file contains a programming error so that it cannot run to completion, Concordance will be unable to compile the file into a .CPT equivalent.

CPL programs can perform a variety of useful procedures. A sampling of CPL programs available at Dataflight's CPL Library include the following:

- *Indxpack.cpl*: Indexes and then packs a database.
- *Userid.cpl*: Displays a user's network login.
- *Send-to-Excel.cpl*: Sends data from the last active query to an Excel file.
- *Calender.cpl*: Displays a calendar. Can be used to augment other CPL programs.

The Structure of a CPL Program

A complete discussion of CPL structure exceeds the scope of this book, as it's a full-fledged programming language. However, you should know that CPL programs are divided into functions, where a *function* is a series of programming lines. Functions can perform many tasks, such as executing other functions, performing calculations, activating features otherwise accessible from Concordance's menu items, modifying data in a database, or providing a user with feedback. Sometimes all these actions are executed from within a single function. Programmers define one function within a CPL program to be the *main* or *entry* function. Such a function is the first set of code to be run in a program, and it often does most of the work, calling other functions in the program to perform related tasks.

Recall in the section "Adding Custom Menu Items" that the Added Menu Items dialog has a File text field that you can use to link a CPL program to a menu item. You can use the Function text field in this dialog to call a function from within the CPL program file referenced in the File text field. The function that's called doesn't have to be a CPL's entry function; it may be a function that tests some condition that must be met before the CPL is run.

Executing a CPL Program

You run CPL programs from the File ➤ Begin Program menu. You're prompted to locate the CPL from an Open dialog. When you select and open a .CPL or a .CPT file, the program begins execution. Many CPLs will first prompt you for input parameters needed for the program to run. This is best illustrated using the example in the following section.

Readocr.cpl

Readocr.cpl is created in the CPL folder during the initial installation of Concordance. The purpose of this program is to load full text data into a field that has the PARAGRAPH data type. The program is appropriate when OCR has been extracted for each document record and saved in separate text files, one file per database record. This method of transmitting data is often used when full text extracted from records is potentially voluminous. A delimited load file that contains both fielded and OCR data can be quite large, and difficult to open and

examine using a text editor. A vendor might instead provide a single delimited file that contains fielded data (BEGDOCNO, ENDDOCNO, CUSTODIAN, DOCDATE, and so on), and provide OCR data as separate text files. (Other methods exist of transmitting data—a vendor might provide a Concordance database instead of a delimited text file and also provide OCR in separate text files, with the understanding that you can use these files to reload OCR at some later date if necessary.)

For this program to work, the individual text files must have file names that match some field in the delimited file (assuming that is the structure of the load file). A candidate for this information is the beginning document control number. For example, if the first record in a load file has a BEGDOCNO that is PX000001, its OCR data will be stored in a text file named PX000001.txt. Another requirement is that all OCR text files be in the same folder. No sub-foldering is allowed.

Running the Program

This example uses the COWCO.DCB database that's created during Concordance's initial installation. The beginning document control number in this database is called STARTPAGE. Several text files have been created and stored in a folder named TEXT_TO_LOAD under the Database folder, itself under the Concordance root (see Figure 7-15). These text files have been created specifically for this example, and aren't part of the Concordance installation. The first five records of the COWCO.DCB database have STARTPAGE values of 00010002, 00010003, 00010004, 00010007, and 00010008. The TEXT_TO_LOAD folder contains five text files: 00010002.txt, 00010003.txt, 00010004.txt, 00010007.txt, and 0010008.txt. Note the relationship between file names and STARTPAGE values, a prerequisite for this program to successfully load OCR data. Note also that the last file in the sequence, 0010008.txt, doesn't match the STARTPAGE value 00010008. This error has been introduced on purpose so that the OCR for this record won't load successfully—a failure that's recorded by the CPL in a log file and described later in this section.

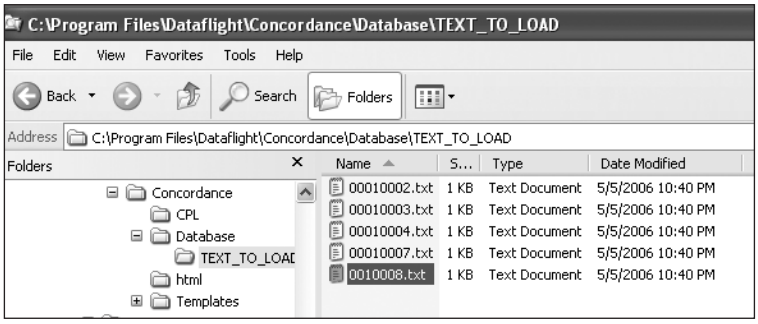

Figure 7-15. *Text files created for this example. Each file name matches a value in the STARTPAGE field of a record in the database, with the exception of 0010008.txt, an error that has been purposely introduced.*

The first step is to select the File ➤ Begin Program menu item. This opens an Open dialog that prompts you to select the CPL to be run (see Figure 7-16).

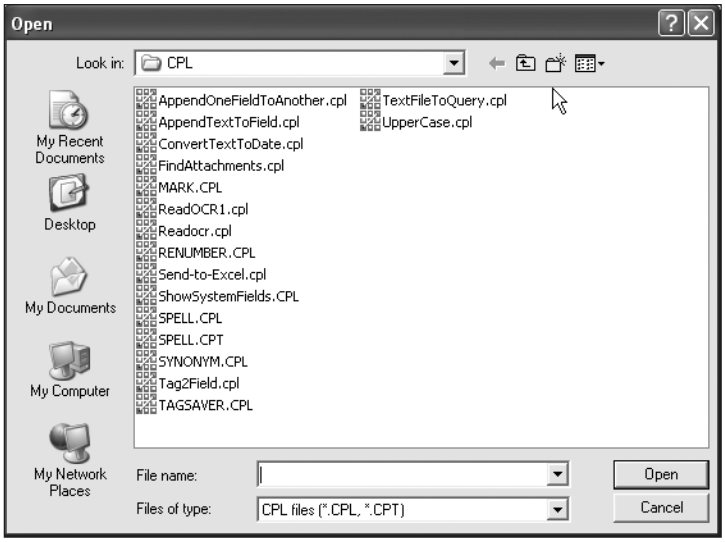

Figure 7-16. *The Open dialog triggered from the File ➤ Begin Program menu. The dialog defaults to the CPL folder under the Concordance folder created during installation.*

Selecting the Readocr.cpl program and clicking the Open button starts the program. Alternatively, you can double-click the program file. The program begins by displaying a message box that describes its purpose (see Figure 7-17).

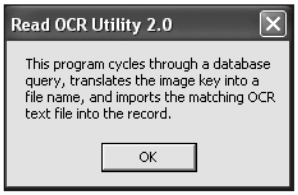

Figure 7-17. *An initial message box created by the Readocr.cpl program describes its purpose.*

When this particular CPL is activated, it provides an interface that prompts you for information needed for the program to run successfully. This interface is created by the CPL itself and is integrated with Concordance; it isn't part of the Concordance software itself (see Figure 7-18).

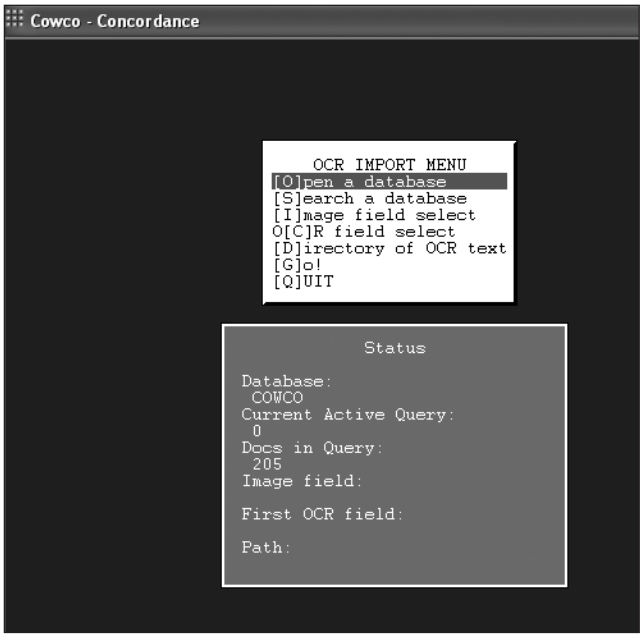

Figure 7-18. *The Readocr.cpl interface. Letters between square brackets are keyboard hotkeys.*

The interface has seven options. The label of each option has a letter between square brackets. This indicates that you can either click the option to activate it, or press the appropriate letter on the computer keyboard.

- *[O]pen a database*: Selecting this option activates an Open dialog. You can select a different Concordance database than the one that's active. If no database is selected, the program assumes the currently active database will be used for the loading procedure. The selected database name is displayed in a status window (see Figure 7-19).

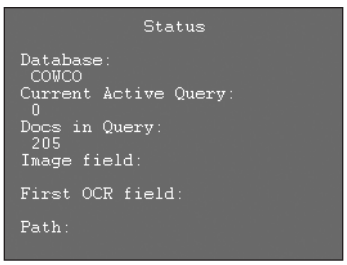

Figure 7-19. *The CPL status window. Note the Database entry—in this example, it's the COWCO.DCB database.*

- *[S]earch a database*: Selecting this option opens a search tool that you can use to select records that will be affected by the loading procedure (see Figure 7-20). When the search is completed, clicking the Done button closes the tool and returns focus to the CPL interface. If you have already queried the database to select appropriate records to be loaded, no search is required. The status window displays the numerical alias of the last active query and the number of records in that query.

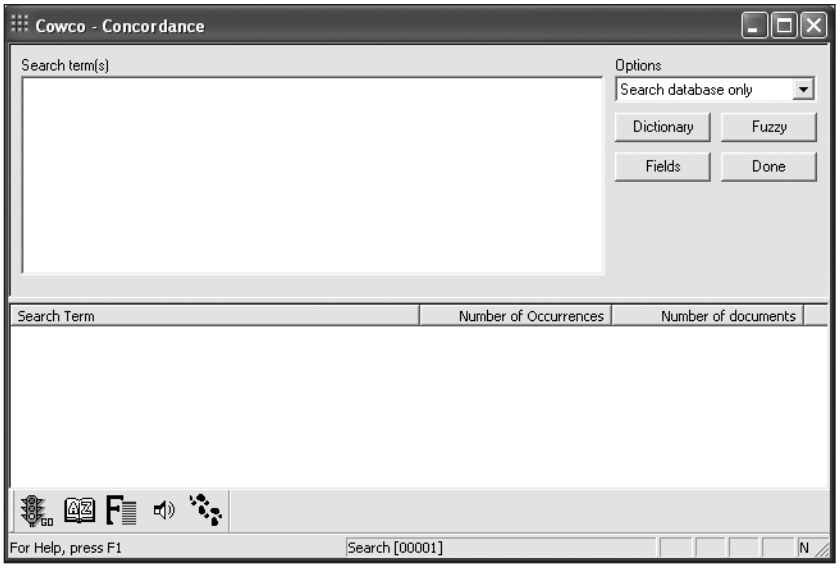

Figure 7-20. *The search tool opened by the "[S]earch a database" option*

- *[I]mage field select*: Selecting this option presents all fields in the database in a list (and each field's data type). You can click the field, or use the up and down arrows on the keyboard to scroll through the list until the desired field is displayed. To select a specific field, click the field name, or scroll to it and press the Enter key. For example, a database may have a field named STARTPAGE that contains the beginning document control number for document records—values that should match OCR text file names. This information is also displayed in the status window.
- *O[C]R field select*: Selecting this option presents all fields in the database in a list (and each field's data type). You select the field to which OCR should be loaded by clicking on the required field or by using the up and down arrows on the keyboard to highlight it.
- *[D]irectory of OCR text*: Selecting this option opens an Open dialog. You should use this tool to navigate to the folder that contains OCR text files. Recall that all files must be in a single folder. Once the appropriate folder is displayed, you must select one of the text files—by double-clicking it or by highlighting it and clicking the Open button—to set the file path. All text files will be loaded, but at least one text file must be selected in this way for the CPL to understand the file path containing OCR text files.

As with the preceding options, your selection is displayed in a status window (see Figure 7-21).

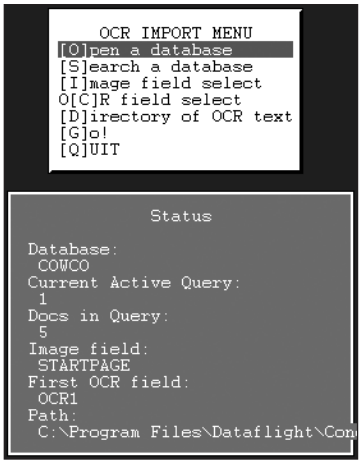

Figure 7-21. *The status window displays the file path selected using the "[D]irectory of OCR text" option.*

- *[G]o!*: Selecting this option prompts you to select a file path for a log file. The log file augments feedback provided by the status window, and is a separate ASCII text file that creates one row per processed file, regardless of success. You can use the log file to locate any OCR text files that failed to load during the procedure. You enter the name of the log file (of your choosing) and then click the Open button. This triggers the actual load.

  When the load procedure is activated, the CPL takes over and processes each database record and each associated OCR text file. The CPL displays a running tally of the number of database records processed. As the CPL attempts to load each database record, the results of the procedure are recorded in the log file. If the CPL encounters a value in the field selected from the option "[I]mage field select" that doesn't have a corresponding text file of the same name, an error message is created as a separate row in the log file. The log file created by this particular load is displayed in Figure 7-22; recall that the file 0010008.txt was purposely misnamed to generate an error message.

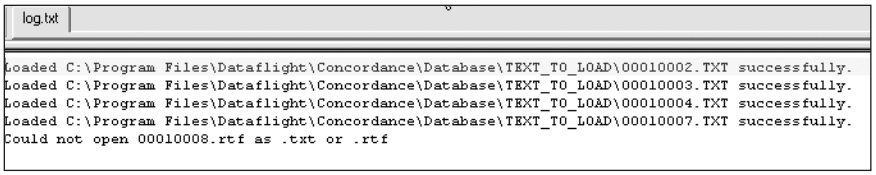

Figure 7-22. *The log file created by the Readocr.cpl in this example. The last line of this log file records the fact that the program was unable to locate a file matching the STARTPAGE value 00010008.txt.*

- *[Q]UIT*: Selecting this option closes the CPL and returns focus back to the Active Workspace screen in Concordance.

Interacting With Other CPL Programs

The interface created by Readocr.cpl is unique to that program. Other CPL programs that perform different procedures will prompt you for different information and will use different interfaces. You might wish to create a sample database, and activate various CPLs, to test how they work and to examine their results. As with any procedure that involves bulk processing and modification of data, you're strongly advised to make a complete backup of a database before a CPL is invoked.

Summary

Because Concordance databases usually involve a dynamic set of underlying data and a dynamic set of users, you'll find that you must interact with databases on a frequent basis to keep them operating at peak efficiency. In this chapter, some basic administrative functions were described: indexing and packing databases, deduplicating records, applying security and concatenating databases, creating custom menu items, and running CPL programs. Although the descriptions in this chapter will give you some insight as to how a Concordance database should be administered, there's no better way to learn than by doing. I recommend that you create test databases and practice the various administrative techniques described in this chapter.

At this point, you should have a good grasp of how databases are created, designed, and deployed. Chapters thus far haven't delved too deeply into how you and end users actually *use* a database, though. The next chapter addresses just that. Among other things, it covers how to view, sort, and edit records, and how to group them using document-level tags. The chapter also introduces the important concept of document *annotations*—subjective comments created by end users to qualify the data presented to them by Concordance.