

The Data Magician

**Version 1.5
User Manual**



Folland Software Services
Copyright 2000 (Lawrence E. Folland)

PROGRAM DESIGN AND IMPLEMENTATION: Lawrence E. Folland

DOCUMENTATION: Lawrence E. Folland

THE DATA MAGICIAN LOGO: David A. Calvert

ACKNOWLEDGEMENTS:

The following people have also provided help and/or encouragement throughout the development of The Data Magician: Beverley Folland, Lisa Weiss, Maria Phipps, Chris Bruner, Betty Wilson, Karen Brothers, Nancy Fish, Ken Noble, Frank Knor, Susan Pasquariella, Francesco Dell'Orso and others. They are all very much appreciated.

COPYRIGHT NOTICE

Copyright 2000 Lawrence E. Folland

Folland Software Services
36 Thornhill Drive
Guelph, Ontario
Canada N1H 8B6
(519) 836-9682
FAX: (519) 836-3436
Internet: support@folland.com
Web Site: www.folland.com

All rights reserved

This software and accompanying manual are copyrighted under Canadian and International law, and may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without the prior written permission of Folland Software Services.

Printed in Canada

Trademark Acknowledgements

INMAGIC®, **DB/TextWorks™**, **The Biblio Guide™**, and **The Library Guide™** are registered trademarks of Inmagic, Inc.

Pro-Cite™ is a registered trademark of Personal Bibliographic Software Inc.

dBase, **dBase II**, **dBase III**, and **dBase IV** are registered trademarks of Borland International.

Library Master™ is a registered trademark of Balboa Software.

STAR® is a registered trademark of Cuadra Associates, Inc.

BiblioFile, **CAN/OLE**, **DOBIS**, **DIALOG**, **FoxBase**, **GEAC**, **Laser Quest**, **MicroISIS** and **UTLAS** are trademarks or service marks of their respective companies.

TABLE OF CONTENTS

OVERVIEW	1
CONVENTIONS USED IN THIS MANUAL	3
Instructions to Type or Press a key	3
'Press any key'	3
Return/Enter	3
DOS Commands and File Names	3
Sample Screens	3
Sample Data	4
GETTING STARTED	5
Hardware Requirements	5
Package Contents	5
Hard Disk Installation	5
Notes for Windows Users	6
Registering a Demonstration Version	6
Location of data and settings files	6
What to read!	7
INVOKING The Data Magician	9
Change to your "working" directory	9
Starting the program	9
The Copyright Screen	9
The Main Menu	9
Selecting items in the Main Menu	10
Returning to the Main Menu	10
Colour Selection	10
Getting Help at the Main Menu	11
GETTING HELP FROM The Data Magician	13
Sample Help Screen	13
Explanation of the Help Screen	13
Leaving Help	13
Keystrokes in Help	13
Help on Error Messages	14
Function Key Help	15
Getting Help via the Internet	15
GENERAL STEPS IN A DATA CONVERSION	17
Set up	17
Saving Settings	18

Editing Keys	18
The Data Conversion	18
THE SAMPLE FILE - "STAFF"	19
The STAFF file in dBase III	19
The STAFF file in INMAGIC	19
DEFINING THE INPUT FILE SPECIFICATIONS	21
Getting to the Input File Specifications screen	21
Selecting an Input File Type	21
Input file names	22
ASCII delimited files	22
Tagged Files	24
dBase files	29
INMAGIC "dump" files	30
MARC Communication files	31
STAR loadable file	33
Micro CDS/ISIS file	33
Library Master file	34
Fixed Length Fields	35
Sample Input File Specifications for the 'STAFF' data file conversion	36
DEFINING INPUT FIELDS	37
Introduction	37
The Input Fields Screen	37
Adding/Inserting New Fields	38
Number of Input Fields Allowed	38
Reading sample data	38
Adding Processing Codes	38
Testing Processing Codes	39
Resetting the Input File	39
Saving Settings	39
Sample Input Fields screen for the "STAFF" database	40
DEFINING THE OUTPUT FILE SPECIFICATIONS	41
Selecting an Output File Type	41
Output File Names	42
Global Output field processing	42
Global Output *POST* processing	42
ASCII delimited files	43
Tagged file	43
INMAGIC load file	46
MARC Communication files	47
STAR loadable file	48
Micro CDS/ISIS file	49
Library Master file	49
Sample Output File Specifications for the 'STAFF' data file conversion	50

OUTPUT FIELDS	<u>51</u>
Introduction	<u>51</u>
The Output Fields Screen	<u>51</u>
Adding/Inserting New Fields	<u>52</u>
Number of Output Fields Allowed	<u>52</u>
Reading sample data	<u>52</u>
Adding Processing Codes	<u>53</u>
Testing Processing Codes	<u>53</u>
Saving Settings	<u>53</u>
Sample Output Fields screen for the "STAFF" database	<u>54</u>
THE DATA CONVERSION PROCESS	<u>55</u>
Introduction	<u>55</u>
The Conversion Summary Screen	<u>55</u>
Processing Log files	<u>56</u>
Starting the Conversion Process	<u>56</u>
The Conversion Progress Screen	<u>56</u>
File size and completion time estimation	<u>57</u>
Unrecognized Input Field names	<u>57</u>
Viewing the Output File during conversion	<u>57</u>
Switching back to the Summary Screen	<u>57</u>
Process Completion	<u>57</u>
Interrupting the conversion process	<u>57</u>
EDITING KEYS FUNCTION SUMMARY	<u>59</u>
Introduction	<u>59</u>
Move Among Prompts	<u>59</u>
Moving within a Prompt	<u>59</u>
Inserting/Typeover	<u>61</u>
Deletion	<u>60</u>
Movement Among Screens	<u>62</u>
Direct Movement to any screen	<u>62</u>
Help	<u>61</u>
Start a DOS Shell	<u>63</u>
Directory Assistance	<u>62</u>
File Display	<u>62</u>
Save/Load/Print Functions	<u>64</u>
Highlight/Cut/Paste	<u>64</u>
Field Display Controls	<u>65</u>
Read/Process Data	<u>66</u>
Colour Display Control	<u>66</u>
Entering Non-displayable ASCII Codes	<u>65</u>
PROCESSING CODES	<u>67</u>
Introduction	<u>67</u>
General Notes:	<u>67</u>
BRIEF SUMMARY OF PROCESSING CODES	<u>70</u>
Append text string to current data	<u>72</u>

Append comment to processing codes	73
And: True if field contains search text AND condition was True	74
Break out a string of text	75
Break up a Call number	76
Break Left of search string	77
Break Left of character position in data	78
Break Left of input Field name/number	79
Break Left of Word number	80
Break out MARC subfield "\$a"	81
Break Right of search string	83
Break Right of character position in data	84
Break Right of input Field name/number	85
Break Right of Word number	86
Break into Subfields on search text	87
Break out Word number n	88
Condition: True if field contains search text	89
Conditional processing: true	90
Conditional processing: false	91
Conditional processing: reverse (not)	92
Condition: True if Empty field	93
Conditional processing: store False in memory n	94
Condition: True if field contains all Lowercase characters	95
Conditional processing: Recall memory n	96
Conditional processing: Store memory n	97
Conditional processing: store True in memory n	98
Condition: True if field contains all Upper case characters	99
Date Today in form of YYYYMMDD	100
Date conversion: from YYMMDD to Mmm. DD, YYYY	101
Include contents of input Field fieldname/fieldnum	102
Lowercase conversion	105
Number Count number of occurrences of "searchtext"	106
Numbering: Increment counter by value of n	107
Numbering: set Starting value of counter to n	108
Numbering: Use value of counter in current data	109
Or: True if field contains search text OR previous condition was True	110
Or: True if field is Empty OR previous condition was True	111
Proper (capitalize) the 1st word	112
Proper (capitalize) All words	113
Quit Conversion	114
Quit Field	115
Quit Record	116
Substitute "search" text with "replace" text	117
process only Subfields n to m	119
append a SubField marker to current data	120
Substitute Marc subfield separator with "text"	121
look up data via Substitute Table in "filename"	122
Text Begin if not empty	124
Text End if not empty	125

Text <u>R</u> ecall string number n	126
Text <u>S</u> tore string number n	127
<u>U</u> ppercase conversion	128
<u>W</u> ildcard change: * for ! (or another)	129
<u>W</u> ildcard change: ? for ! (or another)	130
Delete: <u>A</u> ll extra spaces	131
Delete: specified punctuation from <u>B</u> eginning of field data	132
Delete: <u>C</u> lear everything in current field	133
Delete: specified punctuation from <u>E</u> nd of field data	134
Delete: blanks on <u>L</u> eft	135
Delete: <u>P</u> redefined punctuation (" .,/-") from <u>E</u> nd of field data	136
Delete: blanks on <u>R</u> ight	137
Delete: excess <u>S</u> ubfield markers	138
Command Line Options (Batch Processing)	139
Introduction	139
Command Options	139
General Notes	142
Command Line Errors	142
Usage examples	143
Miscantations (Error Messages)	145
Introduction	145
Anticipated Errors	145
System Errors	145
Recording and Reporting Problems	146
Appendix A: Standard Error Messages	147
Appendix B: System Error Messages	149
Appendix C: Sample Settings Files	151
Introduction	151
Note on square brackets [] in settings examples	151
Technical aside on how the settings print-outs were made	151
BIP.SET - Conversion of Books In Print to MARC Communications Format	152
CANPROC.SET - Conversion of CANOLE records to Pro-Cite	154
DIALOG.SET - Conversion of DIALOG records to INMAGIC load format	156
DOBISINM.SET - Conversion of DOBIS records to INMAGIC	158
DOBISMRC.SET - Conversion of DOBIS (MARC) records to MARC	161
EBSCO.SET - EBSCO Serials (MARC communications format) to INMAGIC	163
GEACINM.SET - Conversion of GEAC records to INMAGIC	166
GEACPROC.SET - Conversion of GEAC records to Pro-Cite	170
INM2MARC.SET - Conversion of INMAGIC records to MARC	174
LIB2MARC.SET - DB/TextWorks LIBRARY or CATALOG data to MARC	177
LIF2COM.SET - Conversion of MicroLIF MARC records to MARC	180
LM2MARC.SET - Conversion of Library Master records to MARC	185
MARC2LIB.SET - Conversion of MARC records to DB/TextWorks	187
MEDCDINM.SET - Conversion of MedLine CD-ROM records to INMAGIC	190

MEDCDPRO.SET - Conversion of MedLine CD-ROM records to Pro-Cite	192
PROC2MRC.SET - Conversion of Pro-Cite records to MARC	194
STAFF.SET - dBase III to INMAGIC	197
UTLASINM.SET - UTLAS (MARC communications format) to INMAGIC	198
INDEX	201

OVERVIEW

The basic purpose of The Data Magician is to take information from one database system and convert it to a format usable by another database system. While most databases will import or export their data in an assortment of formats to accommodate conversions, few, if any, allow any transformation of that data in the process, and many just do not provide compatible formats.

The Data Magician provides the flexibility to take information from various sources, put it in a standardized format and then write the data back out in an entirely different format. In the process, a variety of transformations may be performed on that data, including: breaking up a database field into component pieces, combining two or more separate fields, converting dates from numeric to text formats, changing from upper to lower case or vice versa, and many others.

These transformations give you the opportunity to make comprehensive changes to existing data that would not have been practical manually, even with a word processor. Such changes may make your data more usable and logical or even more aesthetically pleasing.

Facilities are provided to import information from dBase II, III or IV, INMAGIC "dump" files (including DB/TextWorks), delimited ASCII files (a standard import/export format for many database and spreadsheet systems), almost any kind of "tagged" files (such as those created by on-line text databases), MARC Communication Files (a standardized library format), STAR (a minicomputer-based text management system), Library Master (for bibliographic references), Micro CDS/ISIS and Fixed-length field files. The information may be exported as delimited ASCII files, INMAGIC "dump" files, MARC Communication files, STAR load files, Library Master files, Micro CDS/ISIS load files or a customized "tagged" format.

CONVENTIONS USED IN THIS MANUAL

Instructions to Type or Press a key

Keys that you are expected to press or type will generally appear in bold, like this: **DATAMAGE** and will be set away from other text by two or more spaces.

The word "press" will generally precede a command to press a single, special key like: **F1** (which would be Function Key 1)

The word "type" will generally precede a command to type a series of standard keys like: **TITLE**

If you are instructed to press (for example): **Ctrl-S** or **Alt-F1** this means to press and hold the 'Ctrl' key, or 'Alt' key, then, while still holding that key, press the other key after the '-', eg: S or F1. Then release both keys.

'Press any key'

When told to 'Press any key' to continue, you can usually do just that. Some keys, however, won't work, such as Shift, Ctrl, or Alt. The easiest key (and generally the safest) is just the (big) space bar.

Return/Enter

You will often be instructed to press the 'Return' key. Unfortunately, keyboards use different conventions for the label of that key. It may be called 'Return', 'Enter' or simply a bent arrow pointing left. It is typically a larger grey key above the right shift key. There may be another on the numeric keypad, usually called 'Enter'. Whatever the label, they all do the same thing. Find that key on your keyboard and press it when appropriate. It is required after all commands in 'DOS' (the Disk Operating System) and is used frequently in The Data Magician to select an item, or to end entry of a filename or whatever.

DOS Commands and File Names

DOS commands and file names will be displayed in a bold, mono-spaced font. DOS Commands will be in lower case, eg: **a:install**. File Names will be in all caps, eg: **C:\DATAMAGE\STAFF.SET**

Sample Screens

Throughout the manual, sample screens have been included. Those screens appear as Figures within single line boxes. The text within the sample screen is a small mono-spaced font, eg: `Input File Type`. Reverse video on the screen is represented as "highlighted" text in the manual.

Sample Data

Sample data will also appear highlighted, like this: Thompson, Joshua (mostly in the processing code summaries)

GETTING STARTED

Hardware Requirements

To run The Data Magician you will need the following:

- an IBM PC microcomputer or 100% compatible
- minimum 512k total memory, with at least 350k available when you run the program
- 2 floppy disk drives (limited usefulness), or 1 floppy and 1 hard disk drive
- a printer is required if you wish to print settings files, but is not otherwise required to perform conversions

Package Contents

The following items are included with The Data Magician:

- The Data Magician manual
- 1 floppy disk labelled "Program Disk and Sample Data" on 3½" format.
- the Software Licence which should be read before opening the diskette package
- a Software Registration Form which you must fill out and return if you wish to receive information regarding upgrades.

Hard Disk Installation

In order for you to use The Data Magician, it must be accessible to your system. The installation process will copy files from the floppy disk included onto your hard disk in a directory called "**\DATAMAGE**". It will also create a batch file in the "root" directory, called "**DM.BAT**" which provides a quick way to run The Data Magician.

(An aside for advanced users, others should skip to the next paragraph)

For those who do not like to run installation programs, you may simply copy all the files into a directory of your choosing. "**\DATAMAGE**" is recommended since the software will look there for message files if not in the current directory. Copy the sample files if you wish. You should also run the **README** program for last minute notes.

The following are the steps to installing The Data Magician:

- 1) Put the "Program Disk" in floppy drive A
- 2) Make sure you are at the DOS prompt and the default drive is the hard disk on which you wish to install The Data Magician. This will usually be Drive C:, although it may be Drive D: or others. You may switch to that drive by typing the drive letter, then a full colon, and then press Return, eg: **c :**
- 3) Type: **a:install** and press Return (or Enter)
- 4) The software will now begin making the **\DATAMAGE** directory and copying files.
- 5) When completed you will be prompted to remove your disk. You may now put away the disk, preferably in a safe place so that you can get it back should you ever have difficulties with your hard disk (it happens).
- 6) You are then shown the README file which you may browse by pressing the **PgUp** and **PgDn** keys. This file has notes about any last minute changes, additions,

deletions, or other notes. You may print the whole file (if you have a printer attached, on and "on-line") by pressing the **F2** key. When you are finished looking at the **README** file, press **Esc** to leave.

- 7) The installation process is now complete, and you may start The Data Magician by typing: **DATAMAGE** and pressing Return, at the DOS prompt. See the section entitled "Invoking The Data Magician" for an explanation of the use of the menus, the Help system, and general use of the software.

Notes for Windows Users

The Data Magician is well behaved under both Windows 3.x and Windows 95. The author uses it extensively under Windows 95 and prefers it to the DOS environment. Under Windows 95, it is convenient to run a conversion in the background while working on other tasks in the foreground. Processing speed is comparable under Windows (when dedicated to the task) and DOS, although benchmarks showed that by far the fastest processing could be obtained under Windows 95 rebooted into DOS mode. Next fastest is running under Windows 95 in a full-screen DOS screen (not the graphics window). Press Alt-Enter in the graphics window to get into the full-screen text screen.

An icon file is provided for your convenience, **DM.ICO**, which will have been installed in the **\DATAMAGE** directory.

Registering a Demonstration Version

New with Version 1.5, if you have a demonstration version, you may upgrade that copy to a fully operational copy by ordering a registration code. Once you have that code, you may then download from our Web Site (<http://www.folland.com>) any newer release of Version 1.5. The registration code will be stored in the **\DATAMAGE** directory and will automatically be used to serialize the new version and make it fully operational.

Users who purchase a fully operational copy will automatically have the code installed and it will be available should they choose to download a more recent release.

Location of data and settings files

Some consideration should be given to where your data and settings files will be located. Some of the data files may already exist in various directories, such as **\DBASE** or **\INMAGIC**. Others will be created as you use the system. As you create and modify conversion settings, you will need to save them on disk. These are called "Settings files". You may find it useful if most of them are available in the same directory. Typically, you may want to use the **\DATAMAGE** subdirectory as your working directory, with settings files and many of the input and output files available there. That will save time when specifying filenames, and also reduce the chance of "losing" a file on the (complex) hard disk. Those files that exist already elsewhere can be referred to as necessary, but that information is stored in the settings file, so you won't have trouble finding the file later.

In summary, whatever directory you decide to use as your working directory, if any, should be your "default" directory before you run the program. If you need to change directories while working in The Data Magician, you may type the name of the new directory preceded by an equal sign ("=") when at the Load Settings or Save Settings screens (eg: `=\datamage` changes the default directory to `\DATAMAGE` on the current drive and displays all settings files there).

What to read!

For instructions on starting The Data Magician, read the chapter "Invoking The Data Magician".

If you want to get started right away and/or if you're having trouble getting a conversion to work, at least read the chapter "General Steps in a Data Conversion".

For detailed instructions on defining input and output files, and specifying the fields they will use, read the appropriate chapter(s).

To set up The Data Magician to run automatically from a command line or batch file, read the chapter "Command Line Options".

For reference as needed, read the chapters on the help system, the editor, processing codes, and error messages.

INVOKING The Data Magician

Change to your "working" directory

When you start the program, it is generally a good idea to be in the same directory where your data files (the ones you will be converting) will be found. Although it is not necessary, it will save you typing long "pathnames" to your data files. Assuming your working directory is "**\DATAMAGE**" (if not, make the appropriate changes) you type the following to change to that directory: **cd \datamage** and press Return.

Starting the program

If you are in the \DATAMAGE directory, type: **datamage** and press Return to get the program going.

If you are not in the \DATAMAGE directory and you wish to keep the current directory as the "default" directory, you can type: **\datamage\datamage** and press Return.

Alternatively, whenever you are in the "root" directory (or any directory, if you have a PATH set up to the root directory, as most systems do), you can simply type **DM** and press Return. Note that this changes to the **\DATAMAGE** directory and then runs the program.

The Copyright Screen

Shortly after the system starts a message in a box will appear briefly on the screen announcing The Data Magician, version, the address for Folland Software Services, and the copyright notice. A message below the box states "Initializing, please wait a few seconds ...". This message will remain on your screen for about five seconds before the Main Menu is displayed. You may press any key to terminate the display of the screen before the five seconds has elapsed.

Should you wish to keep the Copyright Screen screen on longer, you may press: **Ctrl-Num Lock** while the message is displayed. The message will now remain on the screen indefinitely until you press any key.

The Main Menu

Once the copyright message has been displayed, the Main Menu will appear, as you see below:

Getting Help at the Main Menu

You may get help at the Main Menu, and almost everywhere else, by pressing: **F1**. This will lead you into a full "context-sensitive" help system. This is described fully in the section entitled "Getting Help from The Data Magician". For now, if you get into the Help system and wish to exit, you may simply press: **Esc** or **Space**.

GETTING HELP FROM The Data Magician

At any time while using the system (except while converting, or at a Yes/No prompt), you may request Help by pressing the **F1** key. This will bring up a "context sensitive" full-screen help system. Context sensitive means that the help provided relates directly to where you are in the system. For example, if you are at the Main Menu when you press F1, you will be given help on the Main Menu. If you are located at the prompt requesting type of Input File, you are shown information about the available Input File formats.

Sample Help Screen

An example of a Help Screen (for the Main Menu) is shown below:

```

Help_Screen_#_100
Prompt:   The Data Magician - Main Menu

Keys:     You may do any of the following:
          a)  Use up and down arrows to highlight the item you wish,
              then press Enter (Return), or
          b)  simply press the letter of your choice.  It will be acted
              upon immediately, or
          c)  press the letter in combination with the 'Alt' key, again,
              the option will be acted upon immediately.

Note:     Any of the Main Menu functions can be activated anywhere in the
          system by pressing Alt and the letter of the menu option here.

Example:  To exit the system you may: a) press the down arrow until you
          highlight "[X] Exit The Data Magician", and then press Return,
          or b) press the letter X, or c) press Alt-X.

Press 'PgDn' now for more information on the various options.

[ESC]Exit5[PgUp]5[PgDn]5[G]eneral5[I]ndex5[Any_Function_Key]5Screen_#:

```

Figure 2: Help Screen for the Main Menu

Explanation of the Help Screen

In the top-left corner of each Help Screen is the Screen Number. You can always get directly to any screen if you know the number of the screen you want (described below). The main body of the screen provides further explanation of the choices you have at the moment in The Data Magician. At the bottom, are the keys you may press for further help, or to leave the help system.

Leaving Help

You may always get out of Help by pressing **Esc** or **space**. If you have not yet entered a screen number, you may also leave by pressing **Return**.

Keystrokes in Help

At the bottom of the Help screen is a line telling you briefly what keys you may press. The actual key you press appears between square brackets (eg: "[I]ndex" means press "I" for the Index). In general, you may either leave, or move to another Help screen. Leaving is described above. The other keys are described below:

PgUp If the body of the help text tells you to press 'PgUp' to move to another screen, you may do so. If not, you will be told that "No Help message is available" if you press that key.

PgDn As with 'PgUp', if the body of the help text tells you to press 'PgDn' to move to another screen, you may do so. If not, you will be told that "No Help message is available" if you press that key.

General Moves to a screen which covers the current topic in more general terms. For example, if you are looking at a Help screen about a specific processing code, [G]eneral would move to a screen giving an overview of processing codes. Note that this option may not always be available. If not, you will be told that "No Help message is available" if you press that key.

Index This option always takes you to the "Help Index" screen, which is a list of Help Topics and the screen numbers on which they can be found. To see more information on any topic, simply type the number of the screen you wish to see (see below).

[Any Function key] This option means that if you press any function key that is used by The Data Magician, you will get help on the use of that key.

Screen # Here you may type the number of any Help Screen you wish to see to access it directly. Use this when the text of the current Help Screen tells you to see a certain Screen Number for more help on a particular topic, or if you happen to know the number of the screen you wish to see. All you need to do is type the digits of the screen number. You do not have to press Return unless the number is shorter than three digits. As you type the number, you will see the digits appear on the bottom line of the screen. If you make a mistake before you press the third digit, you may press **Backspace** to correct the error.

Help on Error Messages

Normally, when you request Help, you will receive information about the current prompt. However, if you receive an Error message (eg: for incorrect processing codes), you may press **F1** to get an explanation of that message. The Help screen will display the message, an explanation of the problem, and suggestions on how to correct the problem.

Function Key Help

Anywhere that you need to use the Function keys (which is just about everywhere but the Main Menu), you may request Help on the Function keys by pressing: **Shift-F1**. This will bring up the Help system, displaying a screen of what Function keys are available. From that point, you proceed to use the Help system as described above. The actual screen is reproduced below:

```

Help_Screen # 200
      Editor Function Keys (Ct=Ctrl, Sh=Shift)
      6=====L=====7          +-----,
Al-F1 5 Display Input Fields *          5          * Other Keys: *
Ct-F1 5 Temporary DOS          * Print Settings 5 Ct-F2  G-----I
Sh-F1 5 Editor Help           * Load Settings 5 Sh-F2 *Field Moves: *
F1 5 Help                     * Save Settings 5 F2  * Return/Enter *
Al-F3 5 Reset Input File      *          5          * Tab/Shift-Tab *
Ct-F3 5 Set Size of Data      * Control Chars On/OFF 5 Ct-F4 * Up/Down Arrows*
Sh-F3 5 Display Data          * Display Codes 5 Sh-F4 *          *
F3 5 Read Next Record        * Process Codes 5 F4  *Screen Moves: *
5          *          5          * PgUp/PgDn *
5          * Browse Output File 5          *          *
Sh-F5 5 Highlight Entry      * Browse Input File 5          *Deleting Text: *
F5 5 Start Highlighting     * Enter Subfield Code 5 F6 * Backspace/Del *
5          *          5          * Esc *
5          * List Directory: Wide 5 Ct-F8 *Insert On/Off: *
Sh-F7 5 Full Screen Edit    * List Directory: Full 5 Sh-F8 * Ins *
F7 5 Cut Highlighted Text   * Paste Text 5 F8 *          *
5          *          5          *Word Moves: *
Ct-F9 5 Colours ON/Off      * Set Number of Fields 5 Ct-F10 * Ctrl-Left/Rt. *
Sh-F9 5 Shimmer ON/Off     * Delete Field 5 Sh-F10 *Begin/End: *
F9 5 Beep ON/Off           * Insert Field 5 F10 * Home/End *
9=====N=====8          .-----
[ESC]Exit5[I]Index5[Any Function key]5Screen #:

```

Figure 3: Function key Help Screen

Getting Help via the Internet

You can send email to: support@folland.com to answer any questions you may have. Involved questions that may require creating settings files may require consulting fees. If that is the case, you will be informed and a quote will be provided, if you wish.

You may access the web site at: <http://www.folland.com> Click on the support button, or the BBS button. The BBS allows you to leave questions that any one may respond to.

There is a Data Magician list server. You may subscribe to this by sending email to: majordomo@folland.com with the body of the email being: subscribe datamage

GENERAL STEPS IN A DATA CONVERSION

To give you an idea of what is involved in the process of setting up a data conversion with The Data Magician, we have listed below the steps required. The chapters following describe each of these steps in more detail.

Set up

1. **Define the input file format.** Depending on the type of input file, this process may be very simple or fairly involved. If the data is in a dBase III or IV file, INMAGIC, DB/TextWorks, STAR, Micro CDS/ISIS or a MARC Communication file, you simply tell The Data Magician the name of the file. For a "tagged" file, however (such as a downloaded search from an on-line database, or a tagged MARC file), you will need to specify what form the field tags will take, how to recognize the start and end of a record, etc. This process provides the flexibility to read a great variety of input formats, but also tends to be more challenging to set up.
2. **Define the input fields.** If the file to be read is a dBase or INMAGIC file, the field names can be determined automatically. For a DB/TextWorks, STAR, Micro CDS/ISIS, tagged or MARC communications file, you have the option of specifying the field names, or have The Data Magician determine them by reading the file and prompting you each time it encounters an unknown field name.

For each field, you may specify the field name plus a longer name or description. You may then enter processing codes to affect the input field. For example, you may wish to convert the field to upper or lower case strip leading or trailing spaces, break the field apart, substitute one text string with another, etc. See the chapter on processing codes (page [70](#)) for the complete list of codes.


In the process of defining processing codes, you may use the actual data to test the effect of the codes entered. Pressing **F3** will read in one record at a time, and **F4** will process the record with the current codes. This interactive process helps to ensure that the codes entered will perform as expected for each record.

You may also specify special "input" fields which are the combination of other fields (or other fields broken apart). For example, on input, you may have a "Name" field with both last and first name separated by a comma. You may wish to create two new input fields, "Last" and "First" where "Last" is everything before the comma and "First" is everything after the comma. Your output file may then draw from those two separate fields, rather than the "Name" field.

3. **Specify the output format.** This will tell The Data Magician how to generate the output file. You may specify "ASCII delimited", an INMAGIC load format, a MARC Communications file, a STAR load file or a Micro CDS/ISIS load file. None of these require any special

instruction, other than the name of the output file.

4. **Define the fields to be used in the output file.** This process is similar to Step 2 - Defining the input fields. Enter the names of the fields to be desired in the output file. In some cases The Data Magician is able to determine the field names for you. Eg: INMAGIC structures. The contents of the output field is defined by the process codes. This typically consists of an input field name, but you may also add text, generate incremental numbers, or use any of the other processing codes.

 **Note:** to get data into an output field, either the F"input field" code (page [102](#)) or the FC (Field Copy) code (page [104](#)) is generally required. This is one of the common problems beginning users of The Data Magician have - not specifying which input field to draw the data from. If you do not, you will not get any output!

Saving Settings

Once you have entered your processing information, or any time during, it is a good idea to save your settings. You can do this at any time (by pressing **F2**), overwriting the last copy of the settings.

Editing Keys

Throughout the creation of these settings, you will be using the editing keys to move around the prompts, making changes, saving and loading settings, etc. The chapter on editing keys will outline how to make efficient use of those functions.

The Data Conversion

After the settings are defined, you proceed to the Conversion Summary screen. Here, you will see what the input and output files are, plus some other information. You may change these, or enter a Y to start the conversion. As the conversion proceeds, you will see a status screen of its progress and an estimate of when the conversion will be completed.

In the future, a similar data conversion is easily accomplished by loading the same conversion settings, making appropriate changes to file names, and performing the conversion again.

THE SAMPLE FILE - "STAFF"

In order to clarify some of the functions of The Data Magician, a sample database file has been included called "STAFF". The original file was built in dBase III and is called "STAFF.DBF" and "STAFF.DBT". The file is a fictitious listing of staff members from an intermediate sized manufacturing firm. For each staff person, it includes employee number, name, position, department, extension, office number, start date, salary, and notes. The "notes" field is a dBase III memo field, which allows you to type a page or two of notes, but is not searchable or printable.

A settings file is included called "STAFF.SET" which will move the dBase III data into an INMAGIC file using the "STAFF.STR" structure file. In the process, the name is split into first name and last name, and the start date is changed from a numeric format to a text format. Once in INMAGIC the notes field will be fully searchable.

The STAFF file in dBase III

Below is the structure of "STAFF" in dBase III:

Structure for database: C:\staff.dbf				
Number of data records: 36				
Date of last update : 10/21/88				
Field	Field Name	Type	Width	Dec
1	EMPNO	Numeric	5	
2	NAME	Character	50	
3	TITLE	Character	50	
4	DEPT	Character	50	
5	EXTENSION	Numeric	4	
6	OFFICE	Character	6	
7	STARTDATE	Date	8	
8	SALARY	Numeric	8	2
9	NOTES	Memo	10	
**	Total	**	192	

Figure 4: The structure of STAFF.DBF in dBase III

The STAFF file in INMAGIC

Below is the structure of "STAFF" in INMAGIC:

```
INMAGIC - DEFINE Data Structure
Enter name of structure: STAFF
Enter description line (optional): Sample STAFF data structure
Enter retrieval key field(s): EMPNUM
Enter order key field(s): LASTNAME FIRSTNAME

Enter LABEL NAME INDEX SORT EMPHASIS for each field
F/1 NO EMPNUM T 1 1
F/2 LN LASTNAME Y 7 1
F/3 FN FIRSTNAME Y 7 1
F/4 POS POSITION Y 7 1
F/5 DEPT DEPARTMENT Y 7 1
F/6 EXT EXTENSION T 1 1
F/7 OFF OFFICE T 7 1
F/8 SD STARTDATE T 4 1
F/9 SAL SALARY T 1 1
F/10 NTS NOTES K 1 1

Enter master password (if passwords are desired):
Enter passwords and field lists
P/1
```

Figure 5: The structure of STAFF.STR in INMAGIC

DEFINING THE INPUT FILE SPECIFICATIONS

The first step in setting up a data conversion is usually to define what type of file you will be reading - the "Input File". Here you tell the system the name of the file and (depending on the file type) you may make special adjustments for the particular source of the file. You may also define "global" processing codes - codes that are performed on every field of the input file.

Getting to the Input File Specifications screen

You can get to the "Input File Specifications" screen from the Main Menu by pressing **A**, or from almost anywhere by pressing **Alt-A**.

Selecting an Input File Type

The first option on the Input File Specifications screen (see [Figure 5](#), page [22](#)) is the type of Input File. There are currently nine choices:

- 1) ASCII delimited files
 - data appears in a continuous line with a delimiter between each field (usually a comma). Text may be surrounded in quotes.
- 2) Tagged files
 - data appears with each field on a separate line (usually) with the field name appearing as a text tag in front of the data.
- 3) dBase files
 - data files produced by Ashton-Tate's dBase II, III, or IV or other database products that can create a .DBF file.
- 4) INMAGIC dump files
 - files written out from Inmagic Inc's INMAGIC or DB/TextWorks, using the WRITE or DUMP commands.
- 5) MARC Communication files
 - the standardized MARC communications format used by many library automation products. Allows you to specify a "Block Size" for blocked or "spanned" MARC files.
- 6) STAR output files
 - files written to disk using the default DUMP output format in STAR from Cuadra Associates, Inc. The file must then be moved from the host computer to a PC.

- 7) Micro CDS/ISIS files
 - files created by Micro CDS/ISIS, in the standard output format (ISO 9)
- 8) Library Master files
 - files created using the Export Utility, a tagged text file
- 9) Fixed Length Fields
 - files where the data is stored as contiguous text with predefined lengths

Depending on which input file type you decide, you will be given a different list of prompts on the Input File Specifications screen. The screen for each file type is described below with a sample screen.

Input file names

Whichever Input File type you use, you will need to give the name of the Input file. You may specify a full DOS "pathname", including drive and directory of the file (eg: **C:\DBASE\STAFF.DBF**). If you do not specify the drive and/or the directory, the current default drive and directory are assumed. For example, if you were in the directory \DATAMAGE on drive C: when you started The Data Magician, and give the filename **CATALOG.ASC**, it would assume you mean **C:\DATAMAGE\CATALOG.ASC**. Note that upper/lower case never matters when giving file names. Please note that The Data Magician will convert file names to UPPER case as you type them.

ASCII delimited files

If you choose Input Type **1** you will be presented with the following screen:

```

                                INPUT FILE SPECIFICATIONS

Input File Type:                1
1) ASCII Delimited file          2) Tagged file
3) dBase II, III, or IV file     4) INMAGIC or DB/TextWorks dump file
5) MARC communications file      6) STAR loadable file
7) Micro CDS/ISIS file           8) Library Master Tagged File
9) Fixed Length Fields

Input file name:

Number of Input Fields:

Character(s) around text fields: "
Character(s) separating fields: ,
Does a RETURN end a field? (Y/N): Y
Does a RETURN end a record? (Y/N): Y

Global Field processing:

[F1] Help5[Shift-F1] Editor Help5[Alt-M] Menu5[Alt-X] Exit
```

Figure 6: Input File - ASCII delimited

The questions specific to ASCII delimited format are:

1) **Number of Input Fields**

The answer you give here determines how many incoming fields are considered to be part of each record. Since the fields come one after the other in a continuous stream, this indicates to the system where each record ends. If you give a number which is higher or lower than the actual number of fields per record, you will find the data "drifting" into different fields as you read each record.

2) **Character(s) surrounding text fields**


It is common for text fields to be enclosed in quotes in ASCII delimited files. This is useful if the text might include commas or whatever symbol separates each field. If you do not say otherwise, the system will assume the double-quote ("), is used.

3) **Character(s) separating fields**

Some symbol or symbols are required to separate the values of each field. Although this is typically a comma (,), the default, it may be something else. If the character separating each field is a special control character (such as a tab), you may enter the three-digit code between angle brackets. For example, if each field was separated by a Tab (ASCII code 9), you may specify **<009>** for this field.

4) **Does a RETURN end a field?** (Usually "Y")

It is common for a Return (also known as a Carriage Return/Line Feed) to end each record, and thus the last field. Sometimes, however, the Return symbol may be part of the data, and the field delimiters are the only markers. Type **N** (for No) if you want Return symbols included as part of the data of a field. The default is **Y**, a return ends a field.

 This prompt (and number 5, below) are frequently misunderstood. The question here is: If a Carriage Return is encountered, should it automatically be considered the end of the field (or record)?

5) **Does a RETURN end a record?** (Usually "Y")

As in question 4, above, you may wish a Return included with the contents of a record and thus not signal the end of a record. To do so, respond **N** to this question. The default answer is **Y**.

Tagged Files

Input File type **2** displays the following set of prompts:

```
                                INPUT FILE SPECIFICATIONS

Input File Type:                2
1) ASCII Delimited file        2) Tagged file
3) dBase II, III, or IV file    4) INMAGIC or DB/TextWorks dump file
5) MARC communications file     6) STAR loadable file
7) Micro CDS/ISIS file         8) Library Master Tagged File
9) Fixed Length Fields

Input file name:

Format of Repeated Fields:
0) None, 1) Numbered, 2) Repeat Tag
3) Special Symbol eg: ;

Continuation Line Character(s):
Beginning of Record Starts With:           or Contains:
End of Record Starts With:                 or Contains:
Field Names Prefixed With:                 or End With:

Global Field processing:

[F1]Help5[Shift-F1] Editor Help5[Alt-M] Menu5[Alt-X] Exit
```

Figure 7: Input File Type - Tagged

A "tagged" file is almost any data file where the data is listed with some consistent field label or "tag" in front of the data. For example, if the data file had the line "DATE 1984", 'DATE' would be the field tag, and '1984' would be the data itself. You need to set up the input file to recognize all the field tags that may occur.

Tagged files require the most involved definition process because of the variety of formats that can be handled. In general, you need to tell The Data Magician how to recognize the start and end of each record, how the field tags will appear, and how long text fields are continued over multiple lines.

The following are the prompts that are specific to the Tagged Input File Type:

1) **Format of Repeated Fields**

Here you are expected to enter a number from 0 to 3. The list of options appear underneath the prompt. In some files, there are fields that may be repeated, such as Subject terms. If they are listed on separate lines, The Data Magician needs to know how to recognize them as being multiple occurrences of the same field. It will then put them all into one input field with a special separator between each occurrence.

The most common way of specifying multiple occurrences is to simply repeat the field tag and the data on separate lines. This is option 2. Some systems put the field tag and number the occurrences (option 1). In this case, The Data Magician

needs to ignore the number. Other systems put the field tag for the first occurrence, and then start each following occurrence on a new line beginning with some special symbol, such as a semi-colon ";".

```
SUBJ/1 Software
SUBJ/2 Data Conversion
```

Option 1

```
SUBJ Software
SUBJ Data Conversion
```

Option 2

```
SUBJ Software
; Data Conversion
```

Option 3

2) **Continuation Line Character(s)**

Long text fields must be continued over several lines. There are a variety of ways the continuation of a field can be signalled. The most common way is to indent subsequent lines with one or more spaces. While the field tags usually occur as the first characters on a line, a continuation line may start on the second or subsequent position. You signal this by typing one or more spaces at this prompt. You may also use tabs or any other symbol that may be used to signal a continuation line. Some systems indent the field tags, and have continuation lines wrap back to the first position. In this case you would note that field names start with a space (see number 5, below), and put nothing in this prompt.

```
NOTES Discusses the use of
 software for data conversion.
```

Continue with one space

```
NOTES Discusses the use of
 software for data conversion.
```

Field tag indented, not text

3) **Beginning of Record Starts With &/or Contains**

This, and question 4, are critical for actually finding each record. If the options for these two prompts are not set correctly, you may find that The Data Magician considers everything belonging to one record or does not recognize any records at all. Also, these are how you get The Data Magician to ignore "garbage" in screen capture files - log on procedures, system prompts, command lines, etc. You need to identify some consistent piece of text that signals the start of each record.

There are actually two prompts here, Starts With and Contains. If you type a piece of text into the Starts With prompt, The Data Magician will look for those characters at the beginning of each line. Alternatively, if you put text in the Contains prompt beside it, The Data Magician will look for that text anywhere in a line. If it finds either of these two it will signal the start of a record, and keep reading data into the same input record until the end of the record is found (see number 4, below). The line that signals the start of record may be part of the data and thus a very good Starts With text is often the first field tag (if it is always there, and always first).

Note that either of these two prompts may include "wildcard" characters (* or ?) to look for a pattern of characters, rather than a particular string of text. For example, if you enter 00?, The Data Magician would look for any field tag starting with "00" as signal to start a record. Please note that you will not be able to look for an actual asterisk (*) or question mark as part of the search text. See the note on Wildcards (page [67](#)) for more details.

When filling in this prompt or any other, if you need to include special non-displayable ASCII codes that cannot be entered from the keyboard like the standard keys can, then you may enter a three-digit ASCII code between angle brackets. For example if the end of a record record is indicated by a Form Feed code (ASCII code 12), then you may enter **<012>** at the "End of Record Starts With" prompt.

4) **End of Record Starts With &/or Contains**

Here you must specify what text signals the end of each record. Like the prompts for signalling the beginning of a record (question 3, above), you may specify that the line ending a record starts with a particular string of text and/or contains that piece of text somewhere in the line.

Possible choices for the end of record signals are any special symbols that always end a record (some DIALOG files use `||`), or a field tag which is always present and always the last line.

Another common end of record marker is simply an empty line (one with just a carriage return, or just spaces). You may have The Data Magician look for an empty line by specifying three asterisks (`***`) in the Starts With prompt.

As with the Start of Record (above), you may use "wildcards" to specify the search string to end the record.

If you do not specify anything in these prompts, The Data Magician will assume that

the start of one record indicates the end of the previous record.

5) **Field Names Prefixed with &/or End With**

Many systems put special characters on either side of field names. In order to remove these extra characters from the data, put any leading and trailing characters into these two prompts.

For example, DIALOG files often have a hyphen after the field tag (eg: AU-). In that case, you would put a hyphen (-) in the "Ends With" prompt. Others may have symbols on both sides (eg: =TI=). For that, you would put an equal sign (=) in both the "Prefixed" and "Ends With" prompts.

You may specify a particular symbol that a field name **MUST** end with (eg: a colon). As described above, you are able to enter a symbol that **MIGHT** be there which it should strip. You may also, however, specify the symbol that must end a field name by pressing F6 (which puts in the "5" symbol) in the "Field Name Ends With" prompt and then putting the end symbol. That symbol must then be there for the field name to be recognized. For example, the entry: :5: means that a field name must end with a colon (indicated by ":" after the "5"), and that it should be stripped (the ":" before the "5"). This solves the problem when a line has no specific continuation characters on subsequent lines, or when there are several field names with two words (eg: "SALES 1988:" and "SALES 1989:").

Similar to above, you may specify a specific field tag length (eg: 3 characters) when reading a tagged file. This has primarily been useful reading certain MARC files (eg: RLIN, MicroLIF MARC format), where the field tags are right against the data (ie: no spaces). As above, you may enter the field length by pressing F6 (giving the "5" symbol) in the "Field Name Starts With:" prompt. For example to read a file where the first three characters on a line will be the field tag, enter: 53

6) **Input Line PRE-Processing Codes**

Some files can be difficult to process due to blank or "noise" lines in the middle of a record, extraneous characters on the left margin, or a variety of other problems. This prompt lets you specify processing codes that will affect each input line before it is interpreted for start or end of record, or field names, etc. This allows you to strip off extra characters, or clear a whole line if desired.

You get to this prompt by pressing PgDn or Down arrow from the main Input File Parameters screen. You will be in a special screen just for these codes. You may enter up to 20 lines of processing codes. A sample screen is shown below:

```
Input line PRE-process codes:  
  
[F1] Help5[Shift-F1] Editor Help5[Alt-M] Menu5[Alt-X] Exit
```

Figure 8: Input File PRE-Processing Codes

Note: this is really an advanced option and should be used carefully. It may be easy to mess lines up so that the file is unreadable. There is no way to test the effect of these line-by-line pre-processing codes, except by trying to read the file and seeing if you are successful. On the other hand, experienced users who are quite familiar with the processing codes may find this very useful for dealing with "difficult" files.

Example: Say you were trying to read a file where the first 20 characters of each line were spaces and line numbers. The field tags started in column 21, and the data was in column 26 (say). The Data Magician would not be able to read the field tags properly because of the line numbers to the left. So, you could give the PRE-processing codes: **BR 20** (Break Right of character 20) in this prompt. Every line would have the first 20 columns stripped off, and you would be able to read the field names. Watch out, however, if your start and/or end of record symbols are in those first 20 characters. If so, you may want conditional processing to look for them. For example, if in the example above, the end of record marker was a string of three ampersands (ie: &&&) starting in column 1, you could give the following codes to avoid chopping them off: **C"&&&" C~ BR 20** (Which means: Look for "&&&", if so do nothing, otherwise Break Right of character 20).

dBase files

The Data Magician will read database files (those with the extension ".DBF") created by any of the dBase products from Ashton-Tate (including dBase II, III, III+ or IV), or any "work-alike" products, such as Fox-Base.

Input file type **3** displays the following screen:

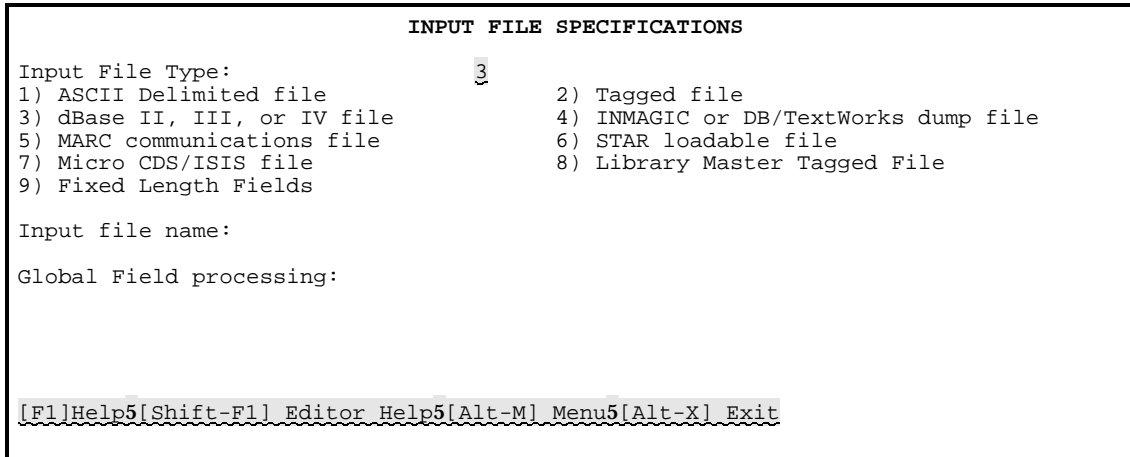


Figure 9: Input File Type - dBase III

There are no special questions for a dBase file aside from the name of the input file. The system will automatically recognize the file format and will read all the field names in the structure of the file.

There is a special indicator at the beginning of a dBase file which The Data Magician checks. This indicator signals whether the file is in dBase II, III, or IV format, and whether there is an associated Memo file.

If the file is in dBase II format, The Data Magician adjusts for this, and then reads the file appropriately. If a memo file should be present (ie: if there are memo fields in the database), it will automatically look for a .DBT file (the dBase Memo file) and they will be included like any other field.. If not found, memos will not be read.

If the indicator is invalid, you will be warned that the specified input file is not a valid dBase file, and the file will not be read. If it is valid, the first record will be read, along with all the field names.

INMAGIC "dump" files

This reads the ASCII file format created by INMAGIC (from Inmagic Inc., Cambridge Mass.) when using the DUMP or WRITE commands (without Using a Report format). Note that this does not read INMAGIC *.DAT* and *.DIC* files. It will also read files created with DB/TextWorks using the Export Tagged format. Data Magician can handle the DB/TextWorks multi-word field names as well as the paragraph markers ("*<*" at the end of a line).

Specifying Input File Type **4** will display the following:

```
                                INPUT FILE SPECIFICATIONS
Input File Type:                4
1) ASCII Delimited file
3) dBase II, III, or IV file
5) MARC communications file
7) Micro CDS/ISIS file
9) Fixed Length Fields
2) Tagged file
4) INMAGIC or DB/TextWorks dump file
6) STAR loadable file
8) Library Master Tagged File

Input file name:
Structure file:
Global Field processing:

[F1]Help5[Shift-F1] Editor Help5[Alt-M] Menu5[Alt-X] Exit
```

Figure 10: Input File Type - INMAGIC "dump" file

Aside from the name of the input file, you may specify the name of the associated *.STR* file (INMAGIC Plus) or *.DBS* file (DB/TextWorks) (ie: the structure of the file from which the dump file was created). If you do specify the structure, The Data Magician will automatically read the field labels and names for you and list them in the Input Fields definition. The field names can also be read from the input file, if the structure file is not available. Press F3 to read a record, and you will be prompted to add each new field name. If the first record does not contain all the names, press F3 to read more records, or manually enter the field names in the Input Fields section.

MARC Communication files

MARC communications files are a common file format used in library cataloguing systems. These files are most commonly sent on disk or tape, not displayed directly on a screen. Some CD-ROM based cataloguing products, such as Laser Quest or Biblio File, can also create MARC records using utility programs that are provided with those products.

Specifying Input File Type **5** will display the following:

```

                                INPUT FILE SPECIFICATIONS
Input File Type:                    5
1) ASCII Delimited file
3) dBase II, III, or IV file
5) MARC communications file
7) Micro CDS/ISIS file
9) Fixed Length Fields
2) Tagged file
4) INMAGIC or DB/TextWorks dump file
6) STAR loadable file
8) Library Master Tagged File

Input file name:
Show MARC subfield delimiters as: C
MARC File Block Size:              0
Global Field processing:

[F1] Help5[Shift-F1] Editor Help5[Alt-M] Menu5[Alt-X] Exit

```

Figure 11: Input File Type - MARC communications file

Because this is a standardized format, you need only specify the input file name. Although a list of field tags are not automatically generated when you specify this format (because there are a large number of possibilities), The Data Magician will prompt you to add any new tags as it encounters them.

MARC Substitution Character

MARC records use a non-displayable character (ASCII Code 31) to designate subfields. Previous versions of The Data Magician converted these to a dollar sign ("\$\$") for display purposes. This, however, can cause a problem if the actual data includes a dollar sign. Therefore, the new default character is a graphic character which is unlikely to occur in normal data. The default character is: **C** (Code 214), and can be generated manually, if needed, by pressing the Alt key with your left hand, and, while still holding the Alt key, typing 2-1-4 on the numeric keypad (not the top row of numbers) with your right hand. Then let go of the Alt key. This character may be changed to another if you wish. Any change made will be saved with the current settings.

Note that the Break MARC subfield processing code (**BM"\$a"**) may still use the dollar sign, or it may use whichever substitution character is specified here.

MARC Block Size (usually "0")

Some MARC files, especially those loaded from tape, are created in fixed block sizes (also known as "spanned" format) usually in increments of 2048 bytes (2048, 4096, 8192, 16384). Each block begins with an extra 5-digit length indicator that specifies how much of the record is in the current block and whether it continues in the next block. The default value, "0", is used for unspanned MARC Communications files.

MARC field tags

Field tags in MARC records are typically three digit codes (eg: 100 for Personal Author, 245 for Title Main Entry). Although there can be codes from 001 to 999, typically only very few of them are actually used, depending on the source of the records. You may either list the field tags that you know are in the data, or you may let The Data Magician prompt you for them. You may wish to check a manual for the system from which the records came to find out the name for the content of each field, and how it is used.

Special MARC field names

Most of the information in a MARC record is contained in the "variable" fields numbered 001 to 999. However, there are a few special pieces of information that come from the "header" portion of a MARC record. Since there are no particular field names for these, but they may be useful, The Data Magician makes up its own name for these, checks if they exist and prompts you to add them if they don't. They are all single character codes. The meaning of the codes would need to be checked in the manual for the system from which they came. You may wish to use the Substitute Table processing code (see page [122](#)) to convert them to more descriptive phrases. Those special fields and their contents are listed below:

STATUS	Record Status (typically "n" for "new")
RECTYPE	Record Type (typically 'a' for monographic materials, although there are many other codes)
BIBLEVEL	Bibliographic Level
ENCODE	Encoding Level
DCF	Descriptive Cataloguing Form

STAR loadable file

Records created by the STAR text database system from Cuadra Associates, Inc. (Los Angeles, CA) can be loaded in under this option. Although they could also be defined using the Tagged File format (Option 2), this simplifies the process.

Records to be converted should be written to disk using the "*DUMP" report that is automatically generated by STAR when a new database is defined. The file must then be transferred to a PC (using a product such as PC-Vision) so that it may be processed by The Data Magician.

Specifying Input File Type **6** will display the following:

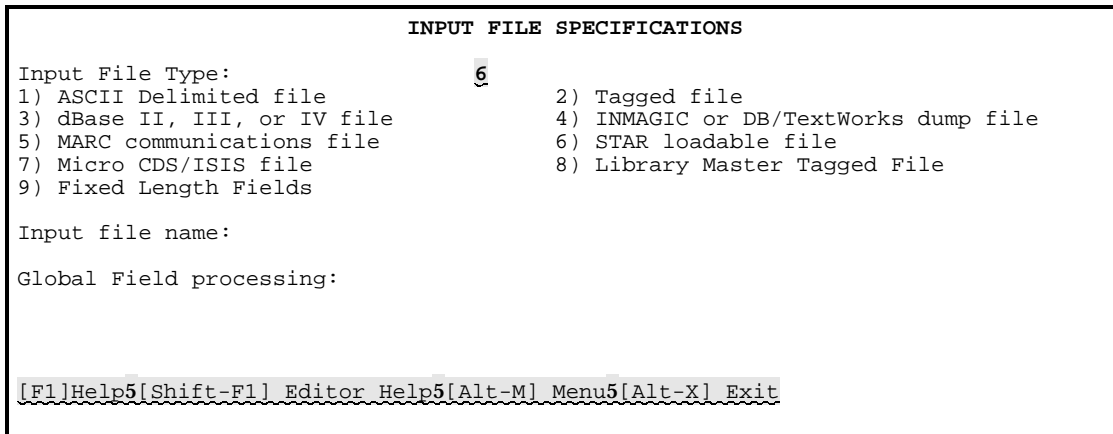


Figure 12: Input File Type - STAR loadable file

Because this is a standardized format, you need only specify the input file name. Although a list of field tags are not automatically generated when you specify this format, The Data Magician will prompt you to add any new tags as it encounters them.

Micro CDS/ISIS file

Records created by MicroISIS or CDS/ISIS (when transferred to a PC) can be read using Option 7. It uses the ISO 9 standard file format, which is similar to a MARC Communication file, with a record header at the beginning (giving record lengths and position of the data in the record following), but is displayable on the screen, using standard ASCII codes and 80 character lines.

Specifying Input File Type **7** will display the following:

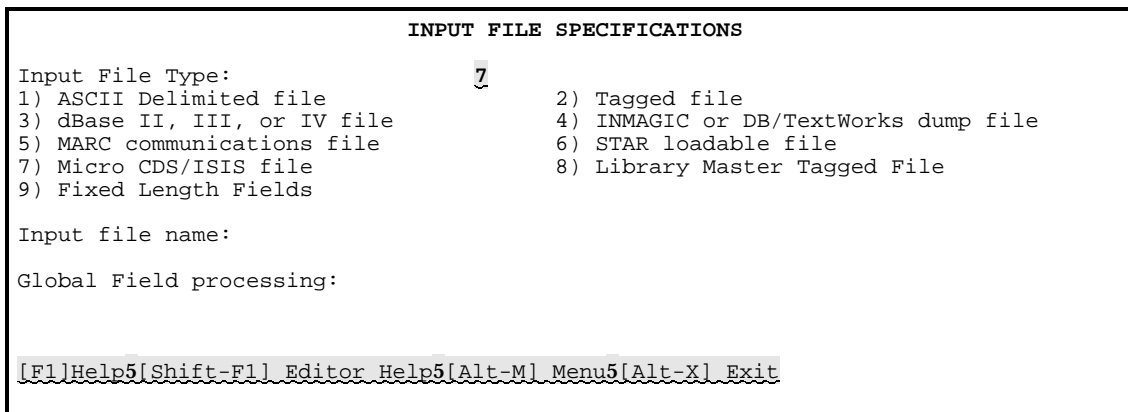


Figure 13: Input File Type - Micro CDS/ISIS File

Again, this is a standardized format, so you need only specify the input file name. Although a list of field tags are not automatically generated when you specify this format (because there are a large number of possibilities), The Data Magician will prompt you to add any new tags as it encounters them.

Library Master file

Records exported from Library Master from Balboa Software (Toronto, ON) can be read using Option 8. Although they could also be defined using the Tagged File format (Option 2), this simplifies the process.

Records to be converted should be created using the Export Utility from within Library Master. Repeat field tag for new paragraphs should be set to "Y" and Include fonts should probably be "N".

Specifying Input File Type **8** will display the following:

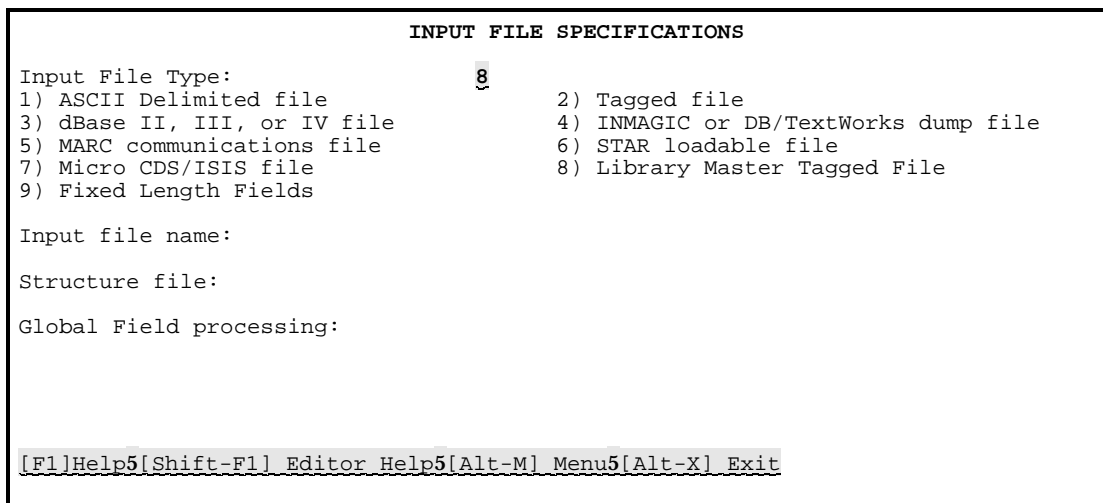


Figure 14: Input File Type - Library Master File

Aside from the name of the input file, you may specify the name of the associated *.STR* file

(ie: the structure of the file from which the exported file was created). If you do specify the structure, The Data Magician will automatically read the field labels for you and list them in the Input Fields definition. If you do not have the structure file available, the field names can be read from the input file. Press F3 to read a record, and you will be prompted to add each new field name. If the first record does not contain all the names, press F3 to read more records, or manually enter the field names in the Input Fields section.

Fixed Length Fields

Records created where the data is in fixed length format (no delimiters between each field) can be read using Input File Type 9. This is a common format for the internal data structure of traditional fixed-length databases. A binary editor (like the Norton Utilities' "DISKEDIT") can be used to look at the data to determine the offset to the data and the size of each field.

Specifying Input File Type **9** will display the following:

```
INPUT FILE SPECIFICATIONS
Input File Type:          9
1) ASCII Delimited file  2) Tagged file
3) dBase II, III, or IV file  4) INMAGIC or DB/TextWorks dump file
5) MARC communications file  6) STAR loadable file
7) Micro CDS/ISIS file      8) Library Master Tagged File
9) Fixed Length Fields

Input file name:
Number of characters to skip:
Global Field processing:

[F1]Help5[Shift-F1] Editor Help5[Alt-M] Menu5[Alt-X] Exit
```

Figure 15: Input File Type - Fixed Length Fields

Aside from the Input File Name, the only option available for this input type is "Number of characters to skip". Some files have a "header" portion that you need to get past to the start of the data. Enter the number of characters or bytes that must be skipped. This may take some experimentation to get the correct offset. You will then need to go to the Input Fields and specify the length of each field. The total of all the fields is the size of each record. If there is any inter-record gap, create a "junk" field to hold that data with a Length equal to the size of that gap.

Sample Input File Specifications for the 'STAFF' data file conversion

Below is how the Input File Specifications look in the conversion of a dBase III file:

```
                                INPUT FILE SPECIFICATIONS

Input File Type:                3
1) ASCII Delimited file
2) Tagged file
3) dBase II, III, or IV file    4) INMAGIC or DB/TextWorks dump file
5) MARC communications file     6) STAR loadable file
7) Micro CDS/ISIS file         8) Library Master Tagged File
9) Fixed Length Fields

Input file name:                STAFF.DBF

Global Field processing:
XL XR ! *** Strip leading/trailing spaces from input data (dBase stores empty s
paces within fixed fields) ***

[F1] Help5[Shift-F1] Editor Help5[Alt-M] Menu5[Alt-X] Exit
```

Figure 16: Input File Specifications - STAFF dBase III file

DEFINING INPUT FIELDS

Introduction

The Input Fields are where the incoming data fields are placed. You must have a list of all incoming field names or tags, except for ASCII delimited files which need only field numbers and Fixed Length Fields which need field lengths. These field names may have been filled in for you if you are reading dBase III files, INMAGIC or DB/TextWorks files or Library Master files. You may type them in yourself if you are reading "tagged" files, MARC communication files, or you may let The Data Magician prompt you to add any unrecognized field tags as it encounters them.

The Input Fields Screen

When you first go to the Input Field screen with a file type that does not know the field names automatically, it will appear as in [Figure 17](#). The highlighting marks the four sections of each Input Field. These are:

- 1) **The Field Label.** This is the name The Data Magician will expect to find in tagged, INMAGIC, MARC, STAR, Micro CDS/ISIS, or Library Master records. ASCII delimited, Fixed Length Fields and dBase files depend on the field position instead.
- 2) **The Field Name/Description** is a longer name or description of that field. Although not recognized when reading the data file, it may be used when you are specifying an Input Field in a processing code.
- 3) **Length.** This is only available for input on files with Fixed Length Fields. It will also display the length of each field for dBase III/IV files.
- 4) **Type.** This is a display-only field for dBase III/IV files.
- 5) **Processing Codes.** The line under the field label and name is where you enter any processing codes that will affect that field. These are applied to the field before the data is drawn into the output fields. Note that the processing codes line will expand if you keep entering codes longer than a line. If you need to enter more than 20 lines of codes, you can press **Shift-F7** (see page [60](#)) to go into an expanded editor. You may suppress the display of the processing codes by pressing **Shift-F4** (see page [64](#)).
- 6) **Data Display.** This is not something you can edit, rather it is where you will see the data for each field as it is read in and processed. By default, you will only see the first line of data, but you may expand the number of rows you see by using the **Ctrl-F3** key (see page [64](#)). You may completely suppress the display of data for a field by pressing **Shift-F3** (see page [64](#)). If the data contains non-displayable codes (eg: ASCII codes 0-31, 127, 255) the display will be messed up. You can press **Ctrl-F4** to have The Data Magician display those codes as **<nnn>** instead, where "nnn" is the ASCII code number (eg: code 9, a tab, would be represented as <009>)

Testing Processing Codes

After you have entered some codes and you wish to experiment with them, you may press **F4** to have The Data Magician process the data via the codes entered. While processing, the message Processing... will appear at the bottom of the screen. When complete, you will again see the screen with the data after processing.

You may repeat the process by reading another record with **F3** and then processing it with **F4**. Continue this way, reading and processing, until you are satisfied that the codes you have created will handle all (or at least most) of the data that can be read.

No data is written to disk while you manually test with **F3** and **F4**, so feel free to experiment.

Resetting the Input File

After you have tried reading a few records and processing them, you may wish to restart at record 1. You may do so by pressing **Alt-F3**. This closes the input file. The next time you press **F3** it starts reading record 1 again. This way, you can keep trying your processing codes on the same records. Also, if you reach the end of the input file, a few presses of **F3** will start back at record 1 again.

Saving Settings

At any time, you may save the current settings (field names, processing codes, etc.). This is important to do frequently in case of power failure or if the computer should have a problem. It is often a good idea to save any settings before you read the first record from the input file, in case there is a disk error while reading the file. You may save the settings file by typing **F2** and then giving a filename (up to 8 letters and/or digits). The extension *.SET* is added automatically if you do not give one. Any existing files with the extension *.SET* will be displayed when saving or loading settings files. If the file already exists, then you will be prompted to replace the old file with the same name. If you do wish to replace the file, you must press **y** or **Y** to confirm the overwriting of the existing file.

Sample Input Fields screen for the "STAFF" database

Below is the first screen of Input Fields for the conversion of a dBase III file to an INMAGIC load file. This screen shot is taken immediately upon entering the Input Fields screen. The first record has been read automatically. Note the extra leading and trailing spaces which are due to reading the fixed field data of dBase III. Using the global processing codes **XL** and **XR** will remove those excess spaces, before they get to the output file.

INPUT FIELDS - Record # 1					
No.	Field Label	Field Name/Description	Length	Type	
1)	EMPNO	()	5	N	
<u>1001</u>					
2)	NAME	()	50	C	
<u>Thompson, Joshua</u>					
3)	TITLE	()	50	C	
<u>President</u>					
4)	DEPT	()	50	C	
<u>Administration</u>					
5)	EXTENSION	()	4	N	
<u>666</u>					
6)	OFFICE	()	6	C	
<u>2-201</u>					
7)	STARTDATE	()	8	D	
<u>19870101</u>					
<u>[F1]Help5[Shift-F1] Editor Help5[Alt-M] Menu5[Alt-X] Exit</u>					

Figure 18: Input Fields - STAFF database conversion

DEFINING THE OUTPUT FILE SPECIFICATIONS

The Data Magician needs to know how the information is to be exported. Here you specify the output format, the output file name and any "global" processing codes to be performed on all fields of the output file.

Selecting an Output File Type

The first option on the Output File Specifications screen (see [Figure 19](#), below) is the type of Output File. The Output File type numbers correspond to the Input File types, although not all Input Types are supported as Output Types.

1) ASCII delimited files

- data is written out with a character (usually a comma) between each field, often with quotes surrounding text, and a Return at the end of each record. This is a common input format for many database systems including: dBase III, Lotus 1-2-3, Pro-Cite, etc. There are options to change what the delimiters between fields and at the end of each record will be.

2) Tagged files

- each field is preceded by a text label. There is typically one field per line, with long fields marked on the next line with a continuation character. Repeating fields can be created by repeating the field tag or starting the line with a special character. The start and/or end of the record may be designated by a special character sequence. There are many options for this format that should be able to accommodate most tagged formats. Note that INMAGIC, STAR and Library Master files could be created with this option, but it is easier to use the predefined formats.

4) INMAGIC or DB/TextWorks load files

- fields are written with a field "tag" followed by a space, then the data. Long lines are wrapped and indented with a single space. Repeating fields are signalled by a semi-colon (";") in place of a field tag. The end of a record is marked by a dollar sign ("\$\$") on a line by itself. This format may be loaded into an INMAGIC or DB/TextWorks database, or other text database systems where you can define the import format, such as Edibase. If the field label includes a space, The Data Magician will put quote marks (') around the label when it creates the output file (eg: Last Name will be written as 'Last Name'). This feature was added for DB/TextWorks.

5) MARC Communication files

- a special "header" section describes the size of the record plus the name and size of all the fields in the record. After the header is all the field data with a field separator between them. This form can be loaded into most library automation systems.

6) STAR loadable file

- fields are written with a field "tag" followed by two spaces, then the data. Long lines are wrapped and indented with a single space. Repeating fields are signalled by repeating the field tag. The end of a record is marked by a blank line. This format may be loaded into a STAR database. When defining the load operation in STAR, the Label Length must be set to zero (0) to allow for variable length field tags.

7) Micro CDS/ISIS file

- a special "header" section describes the size of the record plus the name and size of all the fields in the record. After the header is all the field data with a field separator between them. The record is then written out 80 characters per line, with a Carriage Return/Line Feed (CR/LF) at the end of each line (ASCII codes 13 and 10). The record size does not include the extra space required for the CR/LF codes.

8) Library Master

- each record starts with the record type within curly brackets {}, each field starts on a new line, with the field name in square brackets [], multiple entries and continuation lines wrap back to the left margin, and the record ends with a blank line. The record type is drawn from a special output field that must be named "RECTYPE".

Output File Names

Whichever Output File type you use, you will need to give the name of the Output file. You may specify a full DOS "pathname", including drive and directory of the file (eg: **C:\DBASE\STAFF.ASC**). If you do not specify the drive and/or the directory, the current default drive and directory are assumed. For example, if you were in the directory **\DATAMAGE** on drive C: when you started The Data Magician, and give the filename **STAFF.INM**, it would assume you mean **C:\DATAMAGE\STAFF.INM**. Note that upper/lower case never matters when giving file names. The Data Magician, however, converts any file name to all UPPER case as you type.

Global Output field processing

This prompt allows you to enter processing codes that will be performed before the specified codes for each output field. Note that these codes are repeated for each field. Probably the most useful code to put here is "FC" (see page [104](#)) which allows you to copy the contents of an Input Field to an Output Field with the same name.

Global Output *POST* processing

This prompt allows you to enter processing codes that will be performed after the codes for each output field. These codes are repeated for each output field.

ASCII delimited files

If you choose Output Type **1** you will be presented with the following screen:

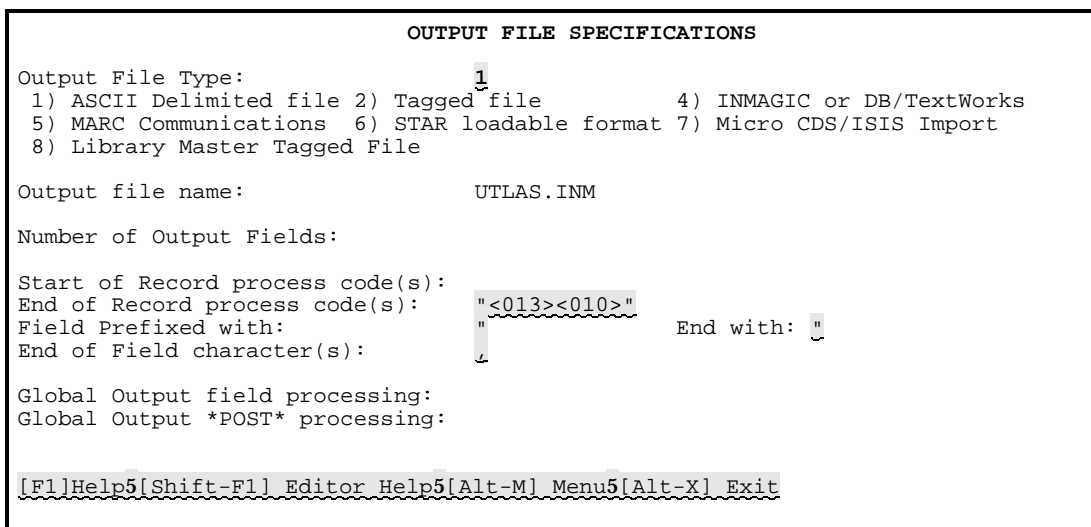


Figure 19: Output File - ASCII delimited

The questions specific to the ASCII delimited format are:

Number of Output Fields

The number of fields you specify here will determine the number of fields actually created for each output record. This is independent of the number of fields displayed in the Output Fields definition. For example, if you have 30 fields displayed in the Output Fields section, and you only specify 25 in this prompt, then only 25 fields will be written - the remaining 5 are ignored.

Start of Record process code(s)

This prompt allows you to specify codes that will include text or the contents of Input Fields that will be written before each record starts. If you need a Carriage Return/Line Feed, use the codes "<013><010>" at the end (include the quotes).

End of Record process code(s)

This prompt will allow you to specify what should mark the end of each record. This defaults to a Carriage Return/Line Feed ("<013><010>") but can be changed. Again, include the quotes since these are processing codes.

Field Prefixed With / End With

This prompt allows you to specify what character(s) (not processing codes) that you want to begin / end each field. These default to a quote symbol (").

End of Field character(s)

This prompt allows you to specify what character(s) (not processing codes) that you want at the end of each field (except the last). This defaults to a comma (,).

Tagged file

If you choose Output Type **2** you will be presented with the following screen:

```

                                OUTPUT FILE SPECIFICATIONS
Output File Type:                2
1) ASCII Delimited file 2) Tagged file          4) INMAGIC or DB/TextWorks
5) MARC Communications 6) STAR loadable format 7) Micro CDS/ISIS Import
8) Library Master Tagged File

Output file name:

Format of Repeated Fields:      Symbol:
(0:None, 1:Numbered, 2:Repeat Tag, 3:Special Symbol)
Continuation Line Character(s):
Start of Record process code(s):
End of Record process code(s):  "<013><010>"
Field tags Prefixed with:      "
Field tag width:                0
Maximum line width:             79
End of Field character(s):      ↵

Global Output field processing:
Global Output *POST* processing:

[F1]Help5[Shift-F1] Editor Help5[Alt-M] Menu5[Alt-X] Exit
```

Figure 20: Output File - Tagged file

This option allows you to custom design a tagged output format. While you could create specifications for any of INMAGIC, STAR or Library Master formats, it is easier to use the specific options provided for those formats.

The questions specific to the Tagged File format are:

Format of Repeated Fields

This option controls how multiple occurrences of the same field will be written out. The four options are:

- 0) The field tag is written for the first occurrence, but not subsequent occurrences:
NOTES This is note 1
This is note 2
- 1) The field tag is followed by a slash and an occurrence number:
NOTES/1 This is note 1
NOTES/2 This is note 2
- 2) The field tag is repeated for each occurrence:
NOTES This is note 1
NOTES This is note 2
- 3) The field tag is written for the first occurrence. Subsequent occurrences use the **Symbol(s)** entered in the next prompt:
NOTES This is note 1
; This is note 2 (assuming the Symbol is ";")

Symbol

This is a companion field for the previous prompt. It is only activated if Option 3 is chosen for the Format of Repeated Fields (above). Simply put the character(s) that you want in place of the field tag for repeated fields (no quotes required).

Continuation Line Character(s)

These characters are placed at the beginning of a line that has been wrapped after exceeding the Maximum line width. Typically, this is a space.

Start of Record process code(s)

Note that you put processing codes in this prompt. You may draw data from an Input Field or perform any other text processing allowed. This can be used when you need a special header before each record that does not come from field data.

End of Record process code(s)

This prompt also accepts processing codes. It used to put any special text you need at the end of a record. This text is put after the text from the last field ("End of field character(s)", see below). If you need a blank line at the end of each record, then you would use the processing codes: "<013><010>". If you needed three asterisks on a separate line, you would use: "<013><010>***<013><010>".

Field tags prefixed with / End With

The character(s) specified here will be put immediately before (or after) the field tag. If you need a space after the field tag (typical), then put a space in the End With prompt.

Field tag width / Left Pad / Right Pad

A value other than 0 specifies that the field tag will be a fixed width. It will be padded on the left or the right with the character specified in either the Left Pad or Right Pad prompts (see below). For example, if you specify a field width of 6 and you have the following field tags: ID, TITLE, AUTHOR, and DESCRIPTION and you enter the indicated padding characters, then the field tags will be created as follows:

Left Pad: . (period)	Right Pad: - (hyphen)
...ID	ID----
.TITLE	TITLE-
AUTHOR	AUTHOR
DESCRI	DESCRI

If both the Left and Right Pad characters are filled in, only the Right Pad character(s) are used.

Any characters specified in the "Field tags prefixed with / End with" (see above) are included within the width. Here is how the field tag would be built: first, both the prefix and End With text are added. Then the entire construction is truncated to the specified width. If there is any room left, right pad characters are appended if specified, otherwise left pad characters are added.

Here's an example if all of these related prompts were filled in and the following field tags were used: ID, STI, DATE, TITLE, AUTHOR, ABSTRACT:

```
Field tags prefixed with:  ==      End With:      --
Field tag width:         8      Left Pad:      .      Right Pad:    +

==ID---+
==STI---+
==DATE--
==TITLE-
==AUTHOR
==ABSTRA
```

Maximum line width

This is the maximum number of characters that can be written on a single line. After this width is reached, The Data Magician will break the text and start on a new line. The next line will be prefixed with the character(s) specified in the "Continuation line character(s)" prompt (above).

Output empty fields?

If the response here is "N" (no) then a tag will not be output if there is no data associated with that output field. If the response is "Y" (yes) then the tag will always be output regardless of field content.

End of Field Characters(s)

These characters are put at the end of each field. Typically, you will want a Carriage Return / Line Feed and thus would need to put: <013><010> These are not processing codes, so you do not put quotes around the text.

INMAGIC load file

This option will create a text file conforming to the INMAGIC or DB/TextWorks load format. This file can be imported into an INMAGIC file using the Add command in the Maintain environment of an INMAGIC database or using the File Import menu item of DB/TextWorks, making sure that the "Inmagic Tagged Format" is selected on the Import Options window.

Output File type **4** displays the following screen:

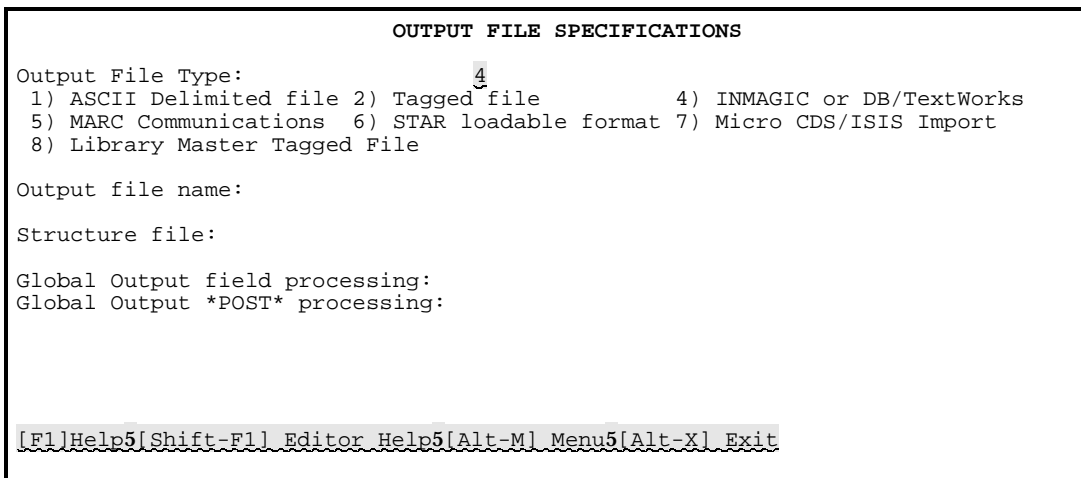


Figure 21: Output File Type - INMAGIC load file

Here you need only specify an INMAGIC or DB/TextWorks structure file (one with an *.STR* extension for INMAGIC or *.DBS* for DB/TextWorks). If you do not provide the extension, The Data Magician will determine what type of structure file it is itself. Providing the structure file name is only to save you from typing in all the required field tags (labels). If you omit the name of the structure of the file into which you will be importing, you will need to enter those field tags yourself (introducing the possibility of miss-spelling one or more tags which will cause INMAGIC to reject the incoming records).

MARC Communication files

Option 5 will create a file in the standard MARC communication format. This format is recognized by most library automation systems. Most of those systems have an optional "MARC loader" program which is required to read the external file of MARC records. Check with the vendor of the system into which you wish to load the records to make sure that it is able to load records in this format.

Output File type **5** displays the following screen:

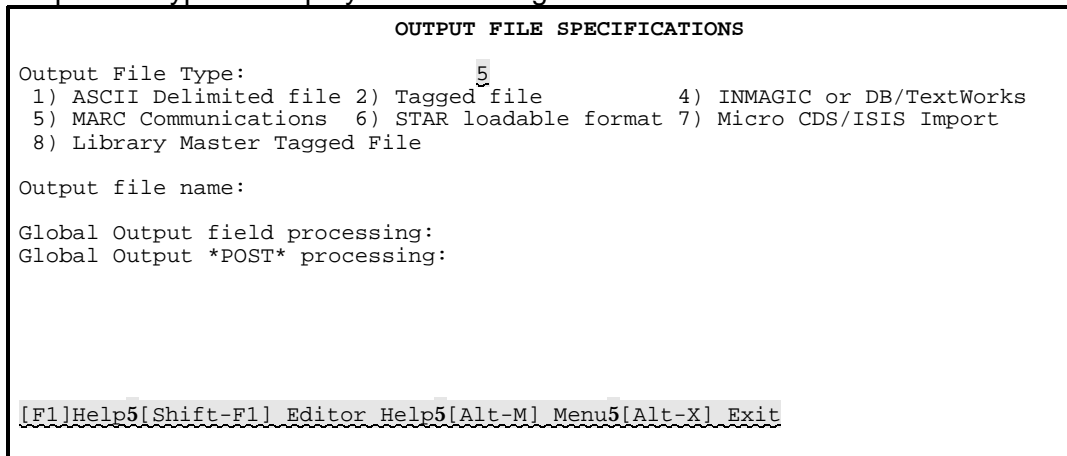


Figure 22: Output File Type - MARC Communications Format

Currently, the only option is to provide the output file name. The Data Magician creates the MARC file in "unspanned" format.

Changing the MARC Subfield character

When writing field data, The Data Magician recognizes "\$" as a subfield delimiter. Since this can cause problems when the data may actually contain dollar signs, you can change this by going to the Input File Parameters, select Input Type 5 (MARC), change the character after "Show MARC subfield delimiters as" to some unused character, such as Alt-214 ("C"), and then switch the Input Type back to what you are actually using. Thereafter, you will need to use the new character to denote MARC subfields, instead of the dollar sign ("\$").

Special MARC field names

If you want to put data into the leader portion of the MARC record, you need to use the following special output field names:

STATUS	Record Status (typically "n" for "new")
RECTYPE	Record Type (typically 'a' for monographic materials, although there are many other codes)
BIBLEVEL	Bibliographic Level
ENCODE	Encoding Level
DCF	Descriptive Cataloguing Form

STAR loadable file

Option 6 will create a file in a format that can be loaded into STAR. The file must then be transferred from the PC to the host system that is running STAR. A package such as PC-Vision may be useful for that purpose. When defining the load operation in STAR, the Label Length must be set to zero (0) to allow for variable length field tags.

Output File type **6** displays the following screen:

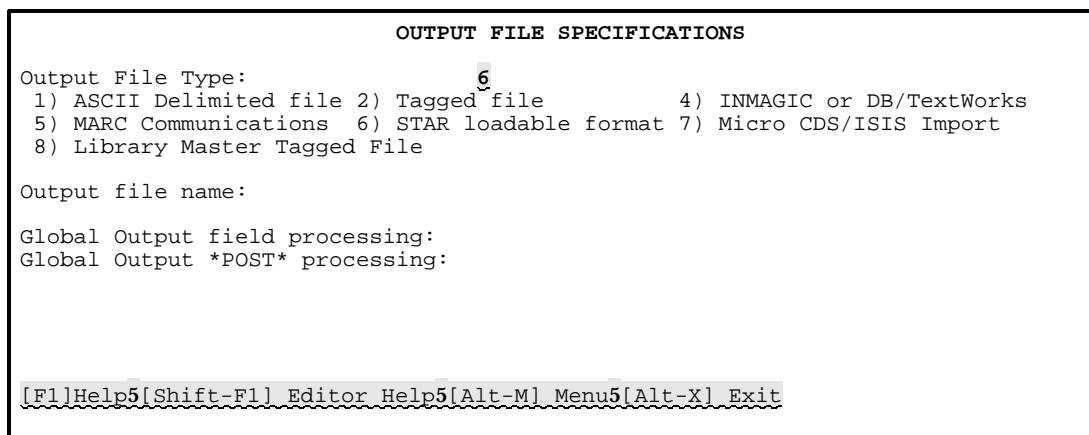


Figure 23: Output File Type - STAR loadable format

Micro CDS/ISIS file

Option 7 will create a file in a format that can be loaded into MicroISIS. MicroISIS field names are always three-digit numbers. The Data Magician will recognize if a number has been entered as a field name with less than three digits, and pad it with leading zeros.

Output File type **7** displays the following screen:

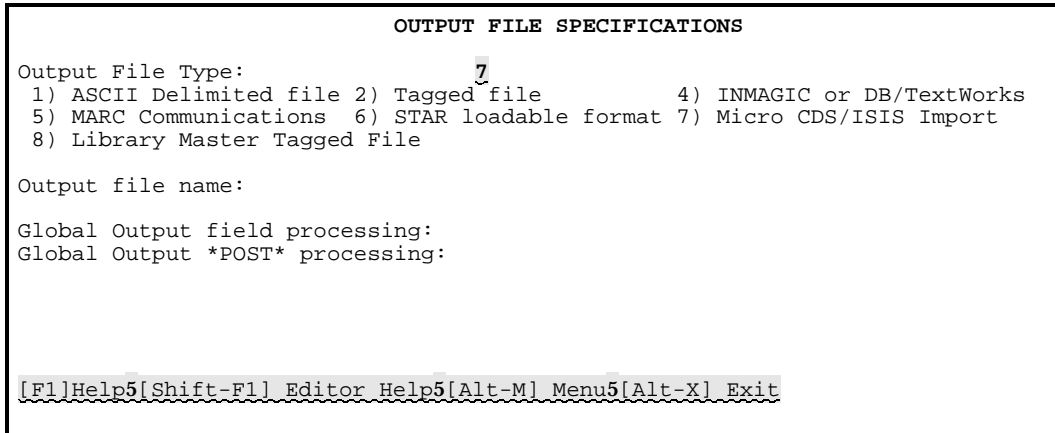


Figure 24: Output File Type - Micro CDS/ISIS Import format

Library Master file

This option will generate an output format that can be read by Library Master from Balboa Software (Toronto, ON). These records are tagged format with the field tag repeated for each occurrence. Although it could also be defined using the Tagged File format (Option 2), this simplifies the process.

Records can be imported into Library Master using the Utilities menu, IMPORT Records.

Specifying Output File Type **8** will display the following:

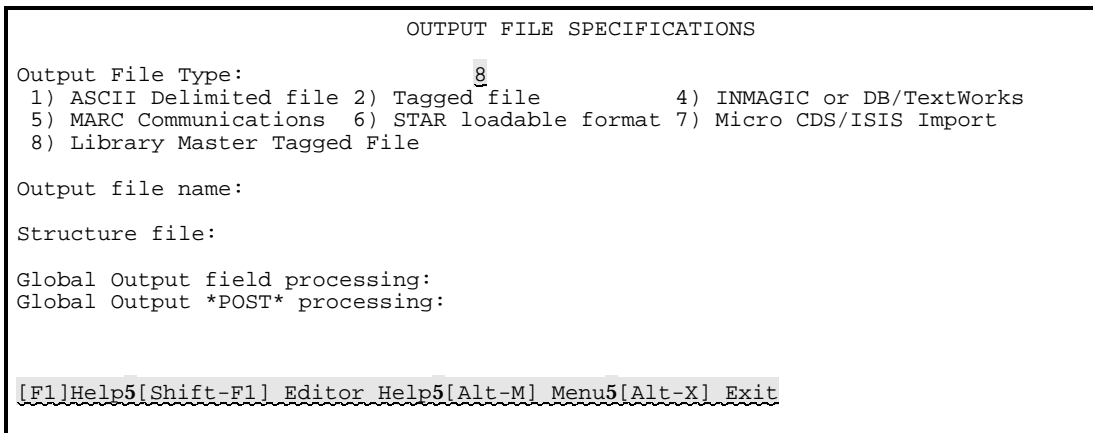


Figure 25: Input File Type - Library Master File

Aside from the name of the output file, you may specify the name of the associated *.STR* file (ie: the structure of the file from which the exported file was created). If you do specify the structure, The Data Magician will automatically read the field labels for you and list them in the Input Fields definition. If you do not have the structure file available, the field names can be manually entered in the Output Fields section.

Sample Output File Specifications for the 'STAFF' data file conversion

Below is how the Output File Specifications look in the creation of an INMAGIC load file from a dBase III file:

```
                                OUTPUT FILE SPECIFICATIONS

Output File Type:                4
1) ASCII Delimited file 2) Tagged file           4) INMAGIC or DB/TextWorks
5) MARC Communications 6) STAR loadable format 7) Micro CDS/ISIS Import
8) Library Master Tagged File

Output file name:                STAFF.INM

Structure file:                  \INMAGIC\STAFF.STR

Global Output field processing:
Global Output *POST* processing:

[F1]Help5[Shift-F1] Editor Help5[Alt-M] Menu5[Alt-X] Exit
```

Figure 26: Output File Specifications - STAFF data file conversion

OUTPUT FIELDS

Introduction

The Output Fields are where the outgoing data fields are placed. You must have a list of all outgoing field names or tags, except for ASCII delimited files which need only field numbers. These field names may have been filled in for you if you are creating INMAGIC or Library Master files. You will need to type them in yourself for other file types (or load an existing settings file and modify it).

The Output Fields Screen

When you first go to the Output Field screen with a file type that does not know the field names automatically, it will appear as in [Figure 27](#). The highlighting marks the four sections of each Output Field. These are:

- 1) **The Field Label.** This is the name The Data Magician will write when creating INMAGIC, MARC, STAR, Micro CDS/ISIS or Library Master records. ASCII delimited files depend on the field number instead.
- 2) **The Field Name/Description** is a longer name or description of that field. Although this is not used by The Data Magician, it may be helpful to remind yourself of what a field is used for.
- 3) **Processing Codes.** The line under the field label and name is where you enter any processing codes that will affect that field. These are applied to the field before the data is written to the output file. Note that the processing codes line will expand if you keep entering codes longer than a line. You may suppress the display of the processing codes by pressing **Shift-F4** (see page [64](#)).
- 4) **Data Display.** This is not something you can edit, rather it is where you will see the data for each field as it is read in and processed. By default, you will only see the first line of data, but you may expand the number of rows you see by using the **Ctrl-F3** key (see page [64](#)). You may completely suppress the display of data for a field by pressing **Shift-F3** (see page [64](#)). If the data contains non-displayable codes (eg: ASCII codes 0-31, 127, 255) the display will be messed up. You can press **Ctrl-F4** (see page [64](#)) to have The Data Magician display those codes as **<nnn>** instead, where "nnn" is the ASCII code number (eg: code 9, a tab, would be represented as <009>)

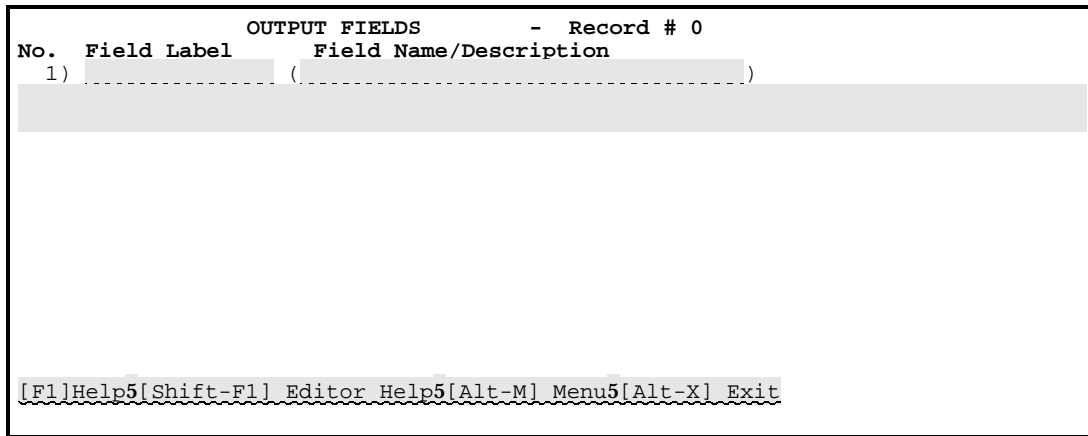


Figure 27: Output Fields (empty)

Adding/Inserting New Fields

You may add new fields by pressing **F10**. If you are in the field Label, the new field will be inserted before the current field. If you are in field Name/Description or Processing Codes, then the new field will be inserted after the current field. You may then fill in any of the prompts.

Alternatively, you may press **Ctrl-F10** to request a number of fields all at once. You then specify how many fields you need (eg: **15**), press **Return**, and that many fields will be available to edit. You may repeat either process to add more at any time. For more information on these and other keys, see the chapter on Editing Keys (page [59](#)).

Number of Output Fields Allowed

You may create up to 100 output fields by default. If you need more than that, you may start the program with a special option which sets that limit higher (or lower). You do this by appending the DOS command with the phrase `"/O=n"` where "n" is the number of fields you need. For example, to boost the number of allowed Output Fields to 200, you would start The Data Magician by entering: `datamage/o=200`

Reading sample data

When you first enter the Output Fields screen, you may not have read any data from the input file (exception: dBase III files automatically read the first record). To read the first and each subsequent record, press the **F3** key. This places the data into the Input Fields. If any field names were unrecognized while reading the file, you will be prompted to append those field names to the end of the current list of fields (if there is room). You will not see any data in the Output Fields until you specify some processing codes (see below) to draw the information from the Input Fields into the Output Fields.

Adding Processing Codes

Once you have some sample data read, you may wish to create some Processing Codes to move the information into the Output Fields and possibly modify the incoming data. As a minimum, you will generally use the **F "fieldname"** code to draw the data from the Input Field "fieldname" (see page [102](#) for more information on this code). You may specify any number of processing codes for each field. See the chapter on Processing Codes (page [67](#)), for more information.

If you put the code **FC** (page [104](#)) in the Global Output field processing (page [42](#)) prompt on the Output File Specifications screen, then all fields of the Output fields that have a field with the same name in the Input Fields will have them automatically moved into the Output Field.

Testing Processing Codes

After you have entered some codes and you wish to experiment with them, you may press **F4** to have The Data Magician process the data via the codes entered. While processing, the message Processing... will appear at the bottom of the screen. When complete, you will again see the screen with the data after processing.

You may repeat the process by reading another record with **F3** and then processing it with **F4**. Continue this way, reading and processing, until you are satisfied that the codes you have created will handle all (or at least most) of the data that can be read.

Note that none of the processing done with the F3/F4 keys is written to disk, so feel free to experiment!

Saving Settings

At any time, you may save the current settings (field names, processing codes, etc.). This is important to do frequently in case of power failure or if the computer should have a problem. It is often a good idea to save any settings before you read the first record from the input file, in case there is a disk error while reading the file. You may save the settings file by pressing **F2** and then giving a filename (up to 8 letters and/or digits). The extension *.SET* is added automatically if you do not give one. Any existing files with the extension *.SET* will be displayed when saving or loading settings files. If the file already exists, then you will be prompted to replace the old file with the same name. If you do wish to replace the file, you must press **y** or **Y** to confirm the overwriting of the existing file.

Sample Output Fields screen for the "STAFF" database

Below is the first screen of Output Fields for the conversion of a dBase III file to an INMAGIC load file.

```
OUTPUT FIELDS          - Record # 1
No.  Field Label      Field Name/Description
  1) NO                (EMPNUM                )
F"EMPNO"
1001
  2) LN                (LASTNAME              )
F"NAME" BL ',,' ! *** Take everything to the left of the comma in the name field
Thompson
  3) FN                (FIRSTNAME             )
F"NAME" BR ',,' XL ! *** Now take everything to the right of the comma ***
Joshua
  4) POS               (POSITION              )
F"TITLE"
President
  5) DEPT              (DEPARTMENT            )
F"DEPT"
Administration
  6) EXT               (EXTENSION              )
F"EXTENSION"
666
  7) OFF               (OFFICE                 )
F"OFFICE"
2-201

[F1]Help5[Shift-F1] Editor Help5[Alt-M] Menu5[Alt-X] Exit
```

Figure 28: Output Fields - STAFF database conversion

THE DATA CONVERSION PROCESS

Introduction

Once you have defined the Input and Output files plus the Input and Output fields, you will want to actually go through an actual conversion process. The Input file will be restarted, so that any experimentation you have done in setting up the processing codes will not affect the output file (ie: any automatic numbering will start back at the original start value, the first records will be processed according to the latest processing codes, etc.)

It is a good idea to save your settings file (**F2** or **Alt-S**) before attempting the complete conversion process. Unforeseen problems (such as disk errors) may cause the computer to "lock up", preventing you from saving later. Also, it is easy to forget to save your work after the conversion process has been completed.

The Conversion Summary Screen

Before a conversion begins, you are shown the Conversion Summary Screen (see [Figure 29](#), below). This reminds you of the Input and Output File names, plus it allows you to specify start and end record numbers for the conversion process. This screen is displayed automatically when you load a previously stored settings file. You may then change the Input or Output file names, or the start and end record numbers, if you do not wish to process the whole file. You may also add a comment or description of these settings.

CONVERSION SUMMARY		Settings: STAFF.SET
Begin Conversion? (Y/N)	N	
Input File Name:	STAFF.DBF	
Output File Name:	STAFF.INM	
Processing Log File:		
Numbering Starts at:		
Starting Record Number:		
Ending Record Number:		
Estimated Number of Records:	36	
Comments/Description:		
Sample Conversion from dBase III to INMAGIC		
<u>[F1]Help5[Shift-F1] Editor Help5[Alt-M] Menu5[Alt-X] Exit</u>		

Figure 29: Conversion Summary Screen (for STAFF database)

File size and completion time estimation

The Data Magician tries to estimate how many records there are in the Input File by calculating the percentage of the Input File read so far, taking the average record size and dividing by the size of the entire file. From the estimated number of records, and the average number of seconds per record for processing, it makes an estimation of when the whole process will be completed. This is updated after each record, so it will always be right (or very close) when it has finished processing. However, it is generally quite close fairly early in the conversion, and gives you a rough estimate of when to come back to check it.

Unrecognized Input Field names

While the file is being processed, any unrecognized field names are displayed in a list on the screen. These are fields that you did not list in the Input Fields definition. You may either choose to ignore these, or go back and add these to the Input Fields, use them in the Output Fields, and then perform the conversion again.

Viewing the Output File during conversion

While the conversion is in progress, you may wish to see the output file being created. To see the output file, either press **V**, the **space bar**, or the **Enter** key. You will then see the output records displayed on the screen as they are written to the disk. It may take a few seconds until the next record appears. Note that the entire record is displayed after processing, so long records may scroll off the screen.

Although The Data Magician will allow you to view any output format, you should be aware that the MARC format contains special "control codes" that prevent them from appearing correctly when printed to the screen so the codes are displayed in numeric form (eg: <030>)

Switching back to the Summary Screen

When viewing the output file, you may switch back at any time to the summary screen by pressing **S**, the **space bar**, or the **Enter** key. Please note that the Summary Screen statistics will not be updated until after the next record is written to disk.

Process Completion

When the conversion process is completed, the completion time is displayed (as well as any unrecognized field names), and you are asked whether you wish to quit. If you respond **Y** the program will end. If not, you will be returned to the Conversion Summary Screen. From there, you may wish to return to the Main Menu (**Alt-M**), or make changes on this screen and perform the conversion again.

Interrupting the conversion process

At any time during the conversion, you may press **Esc** to stop the conversion. You will then be asked if you wish to cancel the conversion. If you do not wish to cancel the conversion, press **N**, or **Enter**. If you do wish to cancel, press **Y**, then proceed as described in the section above.

EDITING KEYS FUNCTION SUMMARY

Introduction

As you create the necessary settings for your conversion, you will require a variety of keys to move and perform special functions. Most of the keys have been designed to work in ways you might be familiar with if you have used a word processor or other standard microcomputer software. Some keys and functions are unique to The Data Magician and knowledge of these keys will greatly enhance your ability to set up conversions effectively.

Move Among Prompts

Return/Enter or Tab	Enters information as is, and moves the cursor to the next prompt
Shift-Tab	Moves the cursor to the previous prompt
Down Arrow	Moves the cursor to next prompt <u>below</u> current prompt. Note that this can be different from Return or Tab, especially in the Input/Output Field specifications which moves the cursor down to the same prompt in the <u>next</u> field.
Up Arrow	Moves the cursor to next prompt <u>above</u> the current prompt. As with the Down Arrow, this can be different from the Shift-Tab.
Ctrl-PgUp	Moves to the first prompt of the group. On the Input or Output File Specifications screens or on the Conversion Summary screen, moves to the first prompt. On the Input or Output Fields screens, it moves to the first field.
Ctrl-PgDn	Moves to the last prompt of the group. On the Input or Output File Specifications screens or on the Conversion Summary screen, moves to the last prompt. On the Input or Output Fields screens, it moves to the last field.

Moving within a Prompt

Right Arrow	Moves the cursor one character to the right. If you move past the right most character, the cursor moves on to the next prompt (like Return or Tab).
Left Arrow	Moves the cursor one character to the left. Like the Right Arrow, a move to the left of the first character will move the cursor back to the previous prompt (like Shift-Tab).

Ctrl-Right-Arrow

Moves the cursor to the first letter of the next "word". If on the last word, the cursor does not move. A "word" is defined as any consecutive sequence of letters and/or digits.

Ctrl-Left-Arrow

Moves the cursor to the first letter of the previous "word", or the current word if not on the first letter. If on the first letter of the first word, the cursor does not move. A "word" is defined as any consecutive sequence of letters and/or digits.

Home Moves the cursor to the first character of the current prompt.

End Moves the cursor to the end of the current prompt.

Inserting/Typeover

Ins Toggles between insert and typeover mode. In typeover mode, characters typed will replace the characters at the cursor. In insert mode, characters typed are inserted before the character at the cursor. Insert mode is signalled by the cursor turning into a half-character sized block. If there is not enough room to enter text, you will get the message "No room to insert. Delete or press INS". You can also press Shift-F7 to go into full-screen mode (see below)

Shift-F7 Switches into full-screen editing mode. You now have the full screen to enter text. If the text is longer than 20 lines, you will see markers at the beginning ("«") or the end ("»") indicating that there is more text in that direction. Press Enter to get out of full-screen mode or any of the movements keys that move out of the editing area to get to the next or previous prompt.

F10 Inserts new field. When defining Input or Output fields, F10 will insert a space for a new field. If the cursor is at the Field Name prompt, the new field is inserted before the current field. If the cursor is at the Description of Process Codes prompt, it is inserted after the current field. If you attempt to insert fields in an ASCII Delimited file, where the order of fields is critical, you will be prompted to verify that that is, in fact, what you wish to do.

Deletion

Backspace Deletes character to the left of the cursor position and moves remaining characters to the left.

Del Deletes character under the cursor and closes the gap.

Ctrl-Home Deletes all characters from left of the cursor to the beginning of the current prompt. The character at the cursor becomes the first character.

Ctrl-End Deletes all characters from the cursor position to the end of the current prompt.

Esc Erases current prompt. If pressed again, or whenever the prompt is empty, will restore the prompt to its original condition before editing.

Shift-F10 Delete current field. When defining Input or output fields, **Shift-F10** will delete the current field and shift all fields up one. You are prompted to confirm the deletion by pressing 'Y'. If you attempt to delete fields in an ASCII Delimited file, where the order of fields is critical, you will be prompted to verify that that is, in fact, what you wish to do.

Movement Among Screens

PgDn displays next screenful of prompts. Progresses from Input File Specifications to Input Fields to Output File Specifications to Output Fields and finally to the Conversion Summary screen. Within Input or Output Field screens, displays next group of fields

PgUp same as PgDn, except it moves backwards. For example, PgUp from the Conversion Summary moves to the first group of Output Fields.

Direct Movement to any screen

Note that all these keys correspond to the same letter choices from the Main Menu, and may also be used at the Main Menu.

Alt-A jumps from anywhere to the first prompt in the Input File Specifications

Alt-B jumps from anywhere to the first prompt in the Output File Specifications

Alt-I jumps from anywhere to the first field in the Input Fields

Alt-O jumps from anywhere to the first field in the Output Fields

Alt-C jumps from anywhere to the Conversion Summary Screen

Alt-M jumps to the Main Menu from anywhere else

Help

F1 context sensitive Help. From anywhere (except a Yes/No prompt) pressing F1 will provide explanation of the prompt and lead into a full help system for more information. Immediately after an error message is displayed (i.e. before any other key is pressed), F1 will display help specific to that error, with suggestions on how to proceed.

Shift-F1 Editor Help. Anywhere, while using the editor, you may press **Shift-F1** to display a screen of all the function keys and what they do. There is additional information about some of the other important editing keys. From there you may also access the full Help system.

Alt-F1 display input fields. Shows a brief summary of the input field names and numbers. This is especially useful while creating your processing codes, to see what fields are available and the exact spelling of field names.

Start a DOS Shell

Ctrl-F1 Exit to DOS. This provides quick access to the operating system you may execute any DOS commands, even run other programs (if there is enough memory space). When finished, type EXIT at the DOS prompt to return back to where you left.

Directory Assistance

Shift-F8 Displays all files in current directory in "full" format - includes filenames, extensions, size, date and time of creation. Pauses between each screenful.

Ctrl-F8 Displays the current directory in "wide" format - includes only filenames and extensions, but more files fit per screenful. Pauses between each screenful.

File Display

Shift-F6 Display Input File. Anytime, once the Input File name has been given, you may press **Shift-F6** to browse the input file. Once in the browse mode, you may use up and down arrows to scroll back and forth, or **PgUp** and **PgDn** to move a screen at a time. Home takes you to the top of the file and End takes you to the bottom. Return to where you left off by pressing **Esc**. Note that non-ASCII files such as dBase III or MARC will not work properly with this function.

Ctrl-F6 Display Output File. Once you have done a conversion, you may browse the Output File by pressing **Ctrl-F6**. In the browse mode, you may use up and down arrows to scroll back and forth, or **PgUp** and **PgDn** to move a screen at a time. **Home** takes you to the top of the file and **End** takes you to the bottom. Return to where you left off by pressing **Esc**.

Note: both the above functions depend on the BROWSE command being available in DOS via the PATH command. The installation procedure for The Data Magician puts that command in the "root" directory of your hard disk, which typically is on the "path". If you have trouble with that, you may try copying the BROWSE command to your working directory (typically \DATAMAGE).

Save/Load/Print Functions

F2
(or **Alt-S**)

Save settings. Saves current settings to disk. You are shown a list of settings files on the default directory. If you have previously given these settings a name, that name appears as the default. Simply pressing Enter will save the settings. Alternately, you may press **Esc** to clear the name and enter a new one.

If a file already exists with the name given, you will be prompted to confirm overwriting the existing file. If you wish to replace the file on the disk with the new file, press **Y**. If you are unsure what that file may be, press **N** and then give a new setting name.

Shift-F2
(or **Alt-L**)

Load settings. Retrieves a previously saved settings file. Note: this will erase your current settings. As with Save settings, you are shown the list of settings files in the current directory. You may enter one of those, or any other. You need not give the extension '.set' as that will be assumed.

If you wish to change to a different directory, type an equal sign followed by the directory path name (eg: to change to \INMAGIC, enter: =\inmagic). You will then be shown any settings files in that directory. That now becomes your default directory.

Ctrl-F2
(or **Alt-P**)

Print settings. After giving you a chance to make sure your printer is ready, it will print the current setting file. Note that square brackets ([,]) are used to enclose text strings which may include spaces.

Highlight/Cut/Paste

F5

Highlight On/Off. When highlighting is on, any text from between the cursor position and the original position when **F5** was pressed will be displayed in reverse (highlighted). Highlighted text may be cut (see **F7**) and moved elsewhere (with Paste - **F8**). All other functions work normally with highlight on, i.e. you can move, insert, delete, etc. Highlighting is cancelled by pressing **F5** again or by leaving that prompt and moving to another.

Shift-F5 Highlight prompt. Turns highlighting on and marks the entire prompt. You may be anywhere on the prompt when you press **Shift-F5**. Essentially, it sets the start highlight position as the first character, and then moves to the end of that prompt. You may then proceed as with **F5** (above).

F7 Cut highlighted text. Removes the text which is currently highlighted and puts it into a temporary cut/paste buffer. This text may be retrieved with the Paste function (**F8** - see below). The text in the cut/paste buffer is not lost until the next time you use **F7**, or until you leave the program.

F8 Paste. Retrieves text from the cut/paste buffer (see Cut - **F7**, above) and inserts it into the current prompt, if there is enough space to accommodate it. If there is not enough room, a message will appear to warn you.

Field Display Controls

Shift-F3 Display Data On/Off. When working in input/output fields, turns the display of the actual data on or off. It may be useful to turn off the display to see more of the fields on the screen. You would want to see the data, however, when reading data from the input file.

Shift-F4 Display Processing Codes On/Off. As with **Shift-F3** (data display On/Off), permits more fields to be displayed on one screen, by suppressing the display of the processing codes line. After processing codes have been created and you are reading experimental records, it may be helpful to see more fields at a time to view the results of your processing codes.

Ctrl-F10 Set number of fields. In either Input or Output Fields, sets the total number of fields that will be displayed. You may work with a maximum of 100 Input and/or Output Fields, unless you requested more (or less) when you started the program.

The contents of fields will not be lost if not displayed, they simply will be ignored, until they are re-displayed. They would not be saved or printed, however.

Ctrl-F3 Set Data Display size. Sets the number of lines of data to display for the current field.
By default, only the first line of data will be displayed. **Ctrl-F3** will allow you to display up to 20 lines of data. If you request more lines than there actually are, you simply see all of the data. You can be in any of the prompts of that field when setting the size.

Ctrl-F4 Display Codes On/Off. If the data contains non-displayable codes (eg: ASCII codes 0-31, 127, 255) the display will be messed up. You can press **Ctrl-F4** to have The Data Magician display those codes as **<nnn>** instead, where "nnn" is the ASCII code number (eg: code 9, a tab, would be represented as <009>)

Read/Process Data

F3 Read Next record. Reads the next record from the input file. Increments the record counter on the top right corner of the screen. Automatically clears all data in the Input and Output fields. When you run out of records, it will reset the input file and start again from the beginning. If, while reading tagged files, field names are encountered that are not in the list of input field names, you will be prompted to add them. No processing of the data is done until you press F4 (see below).

Alt-F3 Reset Input File. Closes the input file, ready to start again. The next time you press **F3**, it will read the first record again.

F4 Process Codes. After the data is read in, you are given a chance to see it in the raw form. Once you have specified some processing codes, you may press **F4** to actually process the data with those codes. After a brief pause (depends on how involved the codes are), you will again see the data in its new form. If you are not satisfied with the result, change any of the processing codes, press **F3** to read the next record, then press F4 to process the new data. Repeat the process for several records to be sure it works for different situations. The order of processing is very important, since some codes will refer to the contents of other fields. Input fields are processed before the output fields. Within that, the fields are processed starting at the first field and continuing to the last field. For each field, the codes specified as Global Processing Codes (on the Input/Output File Specifications screens) are applied first followed by the specific codes for that field. Finally, individual codes are processed left to right.

Colour Display Control

Ctrl-F9 Turns display of colour on or off. This will help if you are using The Data Magician with a monochrome monitor with a colour capable display adapter (CGA, EGA, VGA, SVGA). This works just like the command line switch /M.

Entering Non-displayable ASCII Codes

It may be necessary at times to be able to refer to special ASCII codes that cannot be displayed or entered directly by The Data Magician. This may happen when specifying how

a record begins or ends, or when creating processing codes to eliminate special codes from the data. In any case, The Data Magician will not allow you to type ASCII Codes 0-31, 127, or 255. If you try, it will either "beep" at you or perform the special function that code represents (ie: ASCII Code 8 means "backspace"). To enter such codes, then, you may create the code using a three-digit number enclosed between angle brackets. For example, to enter ASCII Code 8, you would type **<008>**. The Data Magician will show the code this way in all displays, but will still work with the code as ASCII Code 8.

PROCESSING CODES

Introduction

The real power and flexibility of The Data Magician is in the variety of processing options available. The processing codes are how you tell The Data Magician to manipulate the data in the process of moving them from the input format to the output format. The codes are single letter mnemonic commands that are somewhat like a programming language designed especially for data conversion. As you enter each set of codes, The Data Magician checks to make sure they are valid. If not, an error message is displayed indicating the problem. You may press **F1** (Help) for assistance on the message, and may even look up the description of any code.

General Notes:

Upper/Lower Case	When entering alphabetic codes, it does not matter whether they are entered in upper or lower case letters. Case only matters in literal text strings (text within quotes).
Spacing Between Codes	It does not matter how many spaces are between codes, or whether there are any spaces at all. In general, use spaces to improve the clarity of your codes. Spaces only matter within literal text strings.
Comments	You may put a comment anywhere in a processing code line by entering an exclamation mark (!) followed by the comment. <u>Any</u> text after that exclamation mark <u>will be ignored</u> .
Errors Within Codes	When you enter a code and then try to move out of that field, The Data Magician automatically checks the syntax of the codes entered. If it encounters a problem (eg. an unrecognized code or misspelled field name), it will stop, position the cursor on, or just after, the code in question, and display a message describing the problem. If you need clarification, press F1 to get help (if the Help file is available).
Quotes	When entering anything in quotes, you may use: ' (single quote, under the double quote), " (double quote, over the single quote), or ` (reverse apostrophe, under the tilde, ~). Whichever symbol is used to start the quoted text must also be used to end it. This way, you may embed any two of those symbols in a literal string of text.

Wildcards

Whenever you specify "search text" in a code (eg: Break Left/Right of "search text", Search for "search text" & replace with new text), you may include "wildcard" symbols that will match one or many characters. These symbols, * (asterisk) and ? (question mark) were chosen because they are the same as those used by DOS. The * acts as a multiple character wildcard - it may match any number of characters. The ? matches a single character position. For example to find anything with round brackets, you could specify the search text: "(*)". To match a year, you could use: "19??".

You may combine these wildcards, in any combination with literal text, as many times as you wish. For example: "[19??,*]v.*,(*)" might be used to identify a particular sequence of characters with variable text between each piece.

Since the text you may be trying to search for may include actual *'s and ?'s, you may temporarily re-assign those wildcard symbols with the **W*!** and **W?!** codes (see pages [129](#) and [130](#)).

You may also use those wildcards in "replace" text. The Data Magician keeps track of what text was matched with each wildcard. For example, if the raw text was:

[1992,1993] v.17,(12)

and you gave the following processing code:

```
s"[19??,*]v.*,(*)"Year: 19?? and *,*Volume *, Number **
```

The result would be:

Year: 1992 and 1993, Volume 17, Number 12

This happens because each wildcard is matched, in order, with the text it matched, as illustrated in the table below:

Wildcard #	Wildcard	Matching text
1	?	"9"
2	?	"2"
3	*	"1993"
4	*	" "
5	*	"17"
6	*	"12"

Search Strings

When specifying a piece of text as a search string, you may enter the double bar ("`^`", Alt-186, F6) to indicate you want the search string to start at the beginning of text or at the end. If the search string starts with the double bar, then the text must be found at the beginning. If the search text ends at the double bar, then the text must be found at the end. If the search text starts and ends with the double bar, then the whole text must match (although you would probably do this with wildcards). For example, if the text being searched is "The Wind in the Willows", here are some Search and Replace codes and their results:

S"the"A"	"A Wind in A Willows"	(both "the"s matched)
S"5the"A"	"A Wind in the Willows"	(only matches the start)
S"5the5"A"	"The Wind in the Willows"	(didn't match the whole thing)
s"5the*s5"A"	"A"	Did match whole thing

BRIEF SUMMARY OF PROCESSING CODES

<u>Code</u>	<u>Description</u>
"text"	Append text string to current data
! comment	Append comment to processing codes
A "search text"	<u>A</u> nd: True if field contains search text AND condition was True
B "search text"	<u>B</u> reak out search text
B C	<u>B</u> reak up <u>C</u> all Number (or any alphanumeric code)
B L "search text"	<u>B</u> reak <u>L</u> eft of search text
B L characterposition	<u>B</u> reak <u>L</u> eft of character position in data
B L F "fieldname"	<u>B</u> reak <u>L</u> eft of input <u>F</u> ield "fieldname"
B L F n	<u>B</u> reak <u>L</u> eft of input <u>F</u> ield number n
B L W n	<u>B</u> reak <u>L</u> eft of <u>W</u> ord number n
B M '\$a'	<u>B</u> reak out <u>M</u> ARC subfield '\$a'
B R "search text"	<u>B</u> reak <u>R</u> ight of search text
B R characterposition	<u>B</u> reak <u>R</u> ight of character position in data
B R F "fieldname"	<u>B</u> reak <u>R</u> ight of input <u>F</u> ield "fieldname"
B R F n	<u>B</u> reak <u>R</u> ight of input <u>F</u> ield number n
B R W n	<u>B</u> reak <u>R</u> ight of <u>W</u> ord number n
B S "search text"	<u>B</u> reak into <u>S</u> ubfields wherever "search text" is found
B W n	<u>B</u> reak out <u>W</u> ord #n
B X "search text"	<u>B</u> reak e <u>X</u> tract "search text"
C "search text"	<u>C</u> onditional processing: contains search text
C +	<u>C</u> onditional processing: true
C -	<u>C</u> onditional processing: false
C ~	<u>C</u> onditional processing: reverse (not)
C E	<u>C</u> onditional processing: field <u>E</u> mpy
C F n	<u>C</u> onditional processing: store <u>F</u> alse in memory n
C L	<u>C</u> onditional <u>L</u> owercase only
C R n	<u>C</u> onditional processing: <u>R</u> ecall memory n
C S n	<u>C</u> onditional processing: <u>S</u> ore memory n
C T n	<u>C</u> onditional processing: store <u>T</u> rue in memory n
C U	<u>C</u> onditional <u>U</u> ppercase only
D T	<u>D</u> ate <u>T</u> oday in form of YYYYMMDD
D X	<u>D</u> ates: translation of YYYYMMDD to MMM. DD, YYYY
F "fieldname"	include contents of input <u>F</u> ield "field name"
F n	include contents of input <u>F</u> ield number n
F C	<u>F</u> ield <u>C</u> opy (from Input field with same name)
L	case conversion to <u>L</u> ower case
N C "search text"	<u>N</u> umber <u>C</u> ount occurrences of "search text"
N I n	<u>N</u> umbering: <u>I</u> ncrement counter by value of n
N S n	<u>N</u> umbering: set <u>S</u> tarting value of counter to n

N U	<u>N</u> umbering: <u>U</u> se value of counter in data
O "searchtext"	<u>O</u> r: True if field contains search text <u>O</u> R previous cond. was True
O E	<u>O</u> r: True if field is <u>E</u> mpy <u>O</u> R previous condition was True
P 1	case conversion: <u>P</u> roper (capitalize) <u>1</u> st word
P A	case conversion: <u>P</u> roper (capitalize) <u>A</u> ll words
Q C	<u>Q</u> uit <u>C</u> onversion
Q F	<u>Q</u> uit <u>F</u> ield
Q R	<u>Q</u> uit <u>R</u> ecord
S "search"replace"	<u>S</u> ubstitute "search" text with "replace" text
S n-m	process only <u>S</u> ubfields n to m
S F	include a <u>S</u> ub <u>F</u> ield marker in data
S M "text"	<u>S</u> ubstitute <u>M</u> arc subfield separator with "text"
S T "filename"	look up data in <u>S</u> ubstitute <u>T</u> able in "filename"
T B "text"	<u>T</u> ext <u>B</u> egin if field already has data
T B n	<u>T</u> ext <u>B</u> egin with stored string number n (0-9)
T E "text"	<u>T</u> ext <u>E</u> nd if field already has data
T E n	<u>T</u> ext <u>E</u> nd with stored string number n (0-9)
T R n	<u>T</u> ext <u>R</u> ecall stored string number n (0-9)
T S n	<u>T</u> ext <u>S</u> ore current text as string number n (0-9)
U	case conversion: <u>U</u> pper case
W * !	replace multi <u>W</u> ildcard (*) with ! (or any other)
W ? !	replace single <u>W</u> ildcard (?) with ! (or any other)
X A	Delete: <u>A</u> ll extra spaces
X B "punctuation"	Delete: specified punctuation from <u>B</u> eginning of field data
X C	Delete: <u>C</u> lear everything in that field
X E "punctuation"	Delete: specified punctuation from <u>E</u> nd of field data
X L	Delete: blanks on <u>L</u> eft
X P	Delete: <u>P</u> redefined punctuation (" ./-") from <u>E</u> nd of field data
X R	Delete: blanks on <u>R</u> ight
X S	Delete: extra <u>S</u> ubfield markers

"text"

Append text string to current data

Description: Appends the text within quotes to existing data in that field.

- Notes:
- 1) The quotes may be any matched pair of: ", ', or ` . This way, you can include any one of the symbols as a literal string of text.
 - 2) No extra spaces are added unless they are included within the quotes.

Examples:

- 1) Data Before: This is an
Process Codes: "example"
Data After: This is an example
- 2) Before: ABCDE
Codes: '123'
After: ABCDE123
- 3) Input Field "Num": 7792
Output field codes: 'X ' F 'num'
Result: X 7792
- 4) Input Field "Last": West
Input Field "First": Ellen
Output codes: F "last" ", " F "first"
Result: West, Ellen

! comment

Append comment to processing codes

Description: Allows comments to be included within the processing codes. All characters after the exclamation mark (!) are ignored.

Notes: 1) You may use this code to disable codes without actually erasing them (see Example 2)

Examples:

1) Data Before: Smith, John
Process Codes: XL XR !deletes spaces on left and right
Data After: Smith, John

2) Data Before: Smith, John
Process Codes: ! XL XR !deletes spaces on left and right (disabled)
Data After: Smith, John

A "searchtext" And: True if field contains search text AND condition was True

Description: This code will set the conditional processing "flag" to True if the field contains the specified search text and any previous conditions were "True". This way you can have processing codes that are only performed if the field contains a particular string of text as well as some other condition. For instance, you may wish to perform certain processing only if two particular words are there.

- Notes: 1) This code is processed regardless of whether the conditional processing flag is currently True or False.
- 2) To get the reverse, True if the field does not meet all of the conditions, follow the A"searchtext" code with the C~ code (Condition reverse, page [92](#)).
- 3) The test ignores all subfield boundaries. The string of text may occur in any subfield, or even across subfields.
- 4) The search text may include wild card symbols (* and ?). See the General Notes on wildcards (page [67](#)) for more details.
- 5) You may enclose the search string in any pair of the three quote symbols (" , ' , `)
- 6) When searching, case is ignored unless you type any of the search text in UPPER case.
- 7) If there has been no previous Conditional code ("C") in the current field, then this code will act as if it were C"searchtext", since the conditional flag always starts out "True".
- 8) See also the Conditional codes on pages [89](#) to [92](#) and the Or codes on pages [110](#) and [111](#).
- 9) The double bar ("5", F6) can be used to indicate that the search text must be found at the start or end of the field (see "Search Strings", page [68](#))

Examples:

		<u>Comments</u>
1)	Before: M.Sc. University of Toronto Codes: C"Univ" A"Tor" BL"Univ" After: M.Sc.	Want to check for University of Toronto. If so, chop off "Univ..." Would also work for "M.Sc. Univ.Tor."
2)	Before: B.A. University of Washington Codes: C"Univ" A"Tor" BL"Univ" After: B.A. University of Washington	Unaffected, doesn't have "Tor"

B "search text"

Break out a string of text

Description: Search the current data for the string of text within quotes. If found, replace the current data with the search text. Otherwise, does not change the existing data. This is useful when you wish to extract a piece of text if it is there.

- Notes:
- 1) The search text may include wildcard symbols (* and ?) See the General Notes on Wildcards (page [67](#)) for more details.
 - 2) You may enclose the search string in any pair of the three quote symbols (" , ' , or `).
 - 3) When searching, case is ignored unless you type any of the search text in UPPER case.
 - 4) Contrast this code with B X "Search text" (page ?) which will leave the field blank if the search text is not found.
 - 5) The double bar ("5", F6) can be used to indicate that the search text must be found at the start or end of the field (see "Search Strings", page [68](#))

Examples:

		<u>Comments</u>
1)	Data before: <u>Record Type: Book</u> Process Codes: B 'book' Data After: <u>Book</u>	Search string all <u>lower</u> case therefore not case dependant
2)	Before: <u>Record Type: Book</u> Codes: B 'BOOK' After: <u>Record Type: Book</u>	Search string all <u>UPPER</u> case therefore did not match
3)	Before: <u>Vol. 1, No. 2, (1982)</u> Codes: B '(*)' After: <u>(1982)</u>	* is the wildcard

B C

Break up a Call number

Description: This code was specifically designed to insert spaces between the parts of a Library of Congress call number, if there were no spaces. It will also work for other alphanumeric codes where the letters and numbers run together.

- Notes: 1) This procedure takes into account a variety of possibilities of Library of Congress style codes. If you have a similar code you wish to break up, try it a few times to make sure it works for you.
- 2) No extra spaces are added where spaces already exist (see example 1).

Examples:

- 1) Before: QA76.7.F21M77 1988
Codes: BC
After: QA 76.7 .F21 M77 1988
- 2) Before: KF52.P63
Codes: BC
After: KF 52 .P63
- 3) Before: X500
Codes: BC
After: X 500

B L "search text"

Break Left of search string

Description: This code will leave all the text to the left of the search text. The data is unaffected if the search text is not found. This (and Break Right) is particularly useful when trying to break a field into component pieces.

- Notes:
- 1) The search text may include wild card symbols (* and ?). See the General Notes on wildcards (page [67](#)) for more details.
 - 2) You may enclose the search string in any pair of the three quote symbols (" , ' , `)
 - 3) When searching, case is ignored unless you type any of the search text in UPPER case.
 - 4) The double bar ("5", F6) can be used to indicate that the search text must be found at the start or end of the field (see "Search Strings", page [68](#))

Examples:

Comments

- | | | |
|----|--|-------------------------|
| 1) | Before: <u>Smith John</u>
Codes: <u>B L ' '</u>
After: <u>Smith</u> | |
| 2) | Before: <u>Smith John</u>
Codes: <u>B L ' '</u>
After: <u>Smith John</u> | Search string not found |
| 3) | Before: <u>Vol. 1, No. 2, (1982)</u>
Codes: <u>B L '(*)'</u>
After: <u>Vol. 1, No. 2</u> | |

B L characterposition
B L -characterposition

Break Left of character position in data

Description: This code returns all the data to the left of the specified position in the data. If you specify a negative number, the characters are counted from the right side. This is useful when the contents of a field has several pieces of information in a specific order, each piece with a specific length (like old computer punch cards). In MARC records, and in some databases, these are referred to as "fixed fields".

- Notes:
- 1) If the data does not extend to the specified character position, the data is left unchanged.
 - 2) To extract a range of characters from the middle, use a combination of Break Left and Break Right of character position.

Examples:

		<u>Comments</u>
1)	Before: <u>CAN0090..ENGXC</u> Codes: <u>BL4</u> After: <u>CAN</u>	'CAN' is a country code
2)	Before: <u>CAN0090..ENGXC</u> Codes: <u>BL 13 BR 9</u> After: <u>ENG</u>	'ENG' is a language code 2 steps required (or BR9 BL4)
3)	Before: <u>ABC123ZZ</u> Codes: <u>B.L -2</u> After: <u>ABC123</u>	we want to remove the 'ZZ' chop off last 2 characters

B L F "fieldname"

Break Left of input Field name/number

Description: This code is similar to the Break Left/Right of search text, except that instead of specifying the search text in quotes, you tell it to use the contents of an Input Field as the search text.

- Notes: 1) Only an Input Field can be used for this code.
2) If the search string is not found, the data is unaffected.

Examples:

- 1) Input Field "Title": Data Processing
Data Before: Smith, John (1985) Data Processing
Processing Codes: B L F 'Title'
Data After: Smith, John (1985)

- 2) Input Field #5: (May 1985)
Data Before: Science (May 1985)
Processing Codes: BLF5
Data After: Science

B L W wordnumber
B L W -wordnumber

Break Left of Word number

Description: This code will return all text to the left of a specified "word". A "word" is any group of letters and/or digits separated by spaces or punctuation. If the word number is positive, the words are counted from the left side. If the number is negative (preceded by a minus sign), words are counted from the right side.

Notes: 1) If the specified word number is not found, the data is left unchanged.

Examples:

1) Before: Joshua Thompson
Codes: B L W 2
After: Joshua

2) Before: X 62 Main
Codes: B L W -1
After: X 62

Comments

'Main' is a location
Break left of last word

B M "\$abc"

Break out MARC subfield "\$a"

Description: Generally used only in processing MARC communication files, this code helps quickly identify a MARC subfield. If the specified subfield is found, the current data is replaced with the contents of that subfield. If the subfield is not found, the current data is replaced with nothing.

- Notes:
- 1) You may enclose the search string in any pair of the three quote symbols (" , ' , or `).
 - 2) You may be able to use this code for other purposes. The Data Magician looks for the first occurrence of the symbols in quotes. It then looks for the same first character to be repeated (eg: the '\$'). The text, not including the surrounding symbols replace the current text.
 - 3) If the end symbol is not found, but the starting symbols are, then all text following the starting symbols are returned.
 - 4) You may specify multiple subfields to extract (see example 3)
 - 5) If the last character you specify is the subfield marker, that indicates to The Data Magician to include the subfield markers before each subfield (example 4)
 - 6) You may use the asterisk "*" to indicate that you want all subfileds (see example 5)
 - 7) You may use the period (".") as a subfield code to represent the two-character indicator codes before the first subfield (see example 6)
 - 8) Subfields are extracted in the order of the original data unless the first character after the first subfield marker is an exclamation point ("!") (see example 7)
 - 9) If multiple subfields are extracted they are separated, by default, by a space. You may override this with the SM"text" code (see page ?) (see example 8). Nothing is added, however, if you are also extracting the subfield markers (see note 5, above)

Examples:	Comments
1) Data before: <u>\$aToronto\$bMcGraw-Hill\$c1984</u> Process Codes: B M '\$b' Data After: <u>McGraw-Hill</u>	
2) Before: <u>\$aNew York\$c1987</u> Codes: B M '\$b' After:	Subfield doesn't exist therefore nothing is left
3) Data before: <u>\$aToronto\$bMcGraw-Hill\$c1984</u> Process Codes: B M '\$ac' Data After: <u>Toronto 1984</u>	Requested multiple subfields Note that they are separated by a space
4) Data before: <u>\$aToronto\$bMcGraw-Hill\$c1984</u> Process Codes: B M '\$ac\$' Data After: <u>\$aToronto\$c1984</u>	Subfield marker at end, therefore subfield codes are included
5) Data before: <u>\$aToronto\$bMcGraw-Hill\$c1984</u> Process Codes: B M '\$*' Data After: <u>Toronto McGraw-Hill 1984</u>	Specify that all subfields should be extracted

- | | | | |
|----|---|--|---|
| 6) | Data before:
Process Codes: | <u>14CaThe titleCb: subtitle</u>
<u>B M 'C.aC'</u> | Note the subfield marker (Alt-214)
Extract indicator codes (".") and
subfield "a",
plus the subfield markers ("C") |
| | Data After: | <u>14CaThe title</u> | |
| 7) | Data before:
Process Codes: | <u>_aSmith bJohn cE. dMr.</u>
<u>B M '!dbca'</u> | Just happen to use underscore ("_")
Specify the order of subfields and
use "!"
Results in order specified |
| | Data After: | <u>Mr. John E. Smith</u> | |
| 8) | Data before:
Process Codes:
Data After: | <u>CaComputingCxCanada</u>
<u>SM " -- " BM'C*</u>
<u>Computing -- Canada</u> | Separate subfields with " -- " |

B R "search text"

Break Right of search string

Description: This code will return all the text to the right of the search text. The data is unaffected if the search text is not found. This (and Break Left) is particularly useful when trying to break a field into component pieces.

- Notes:
- 1) The search text may include wild card symbols (* and ?). See the General Notes on wildcards (page [67](#)) for more details.
 - 2) You may enclose the search string in any pair of the three quote symbols (" , ' , `)
 - 3) When searching, case is ignored unless you type any of the search text in UPPER case.
 - 4) The double bar ("5", F6) can be used to indicate that the search text must be found at the start or end of the field (see "Search Strings", page [68](#))

Examples:

Comments

- | | | |
|----|------------------------------|-------------------------|
| 1) | Before: Thompson, Joshua | |
| | Codes: B_R ',' | |
| | After: Joshua | |
| 2) | Before: Thompson Joshua | |
| | Codes: B_R ',' | |
| | After: Thompson Joshua | Search string not found |
| 3) | Before: (1982) Vol. 1, No. 2 | |
| | Codes: B_R '(*)' | |
| | After: Vol. 1, No. 2 | |

B R characterposition
B R -characterposition

Break Right of character position in data

Description: This code returns all the data to the right of the specified position in the data. If you specify a negative number, the characters are counted from the right side. This is useful when the contents of a field has several pieces of information in a specific order, each piece with a specific length (like old computer punch cards). In MARC records, and in some databases, these are referred to as "fixed fields".

- Notes:
- 1) If the data does not extend to the specified character position, the data is left unchanged.
 - 2) To extract a range of characters from the middle, use a combination of Break Left and Break Right of character position.

Examples:

		<u>Comments</u>
1)	Before: <u>CAN0090..ENGXC</u> Codes: <u>BR12</u> After: <u>XC</u>	we want to extract 'XC'
2)	Before: <u>CAN0090..ENGXC</u> Codes: <u>BR 9 BL 4</u> After: <u>ENG</u>	'ENG' is a language code 2 steps required (or BL13 BR9)
3)	Before: <u>ABC123ZZ</u> Codes: <u>B R -2</u> After: <u>ZZ</u>	we want to extract the 'ZZ' chop off everything before last 2 characters

B R F "fieldname"

Break Right of input Field name/number

Description: This code is similar to the Break Left/Right of search text, except that instead of specifying the search text in quotes, you tell it to use the contents of an Input Field as the search text.

- Notes:
- 1) Only an Input Field can be used for this code.
 - 2) If the search string is not found, the data is unaffected.
 - 3) You may enclose the field name in any pair of the three quote symbols (" , ' , `)

Examples:

- 1) Input Field "Date": (1985)
Data Before: Smith, John (1985) Data Processing
Processing Codes: B R F 'date'
Data After: Data Processing

- 2) Input Field #7: Science
Data Before: Science (May 1985)
Processing Codes: BRF7
Data After: (May 1985)

B R W wordnumber
B R W -wordnumber

Break Right of Word number

Description: This code will return all text to the right of a specified "word". A "word" is any group of letters and/or digits separated by spaces or punctuation. If the word number is positive, the words are counted from the left side. If the number is negative (preceded by a minus sign), words are counted from the right side.

- Notes: 1) If the specified word number is not found, the data is left unchanged.
2) When breaking right of a word, it starts immediately after the last character of the indicated word (see Example 3)

Examples:		<u>Comments</u>
1)	Before: Joshua Thompson Codes: B R W 1 After: Thompson	
2)	Before: X 62 Main Codes: B R W -2 After: Main	'Main' is a location Break right of 2nd last word
3)	Before: 1987 Salary: 55000.00 Codes: BRW 2 After: : 55000.00	Starts after the 'y' in Salary

B S "search text"

Break into Subfields on search text

Description: This code breaks up data into separate subfields which are then easily manipulated as individual units. For software that recognizes subfields (or "multiple entries"), such as INMAGIC, Pro-Cite, or Revelation, you may need to have these subfields handled in special ways. INMAGIC, for instance, needs these subfields listed each on a separate line, while Pro-Cite needs slashes between each one.
The Data Magician uses an internal format of placing a special double vertical bar ("5", ASCII code 186) between each piece. This is how it will be displayed when reading and converting data manually.

- Notes:
- 1) The search text may contain wild card symbols (* and ?). See the General Notes on wildcards (page [67](#)) for more information.
 - 2) The double bar ("5", F6) can be used to indicate that the search text must be found at the start or end of the field (see "Search Strings", page [68](#))

Examples:

1) Before: computers; calculators; typewriters
Codes: BS "; "
After: computers5calculators5typewriters

2) Before: Dogs Cats Fish Pets
Codes: BS " "
After: Dogs5Cats5Fish5Pets

B W n
B W -n

Break out Word number n

Description: Extracts from the data a single "word" specified by the number n. A "word" is any group of letters and/or digits separated by spaces or punctuation. If the word number is positive, the words are counted from the left side. If the number is negative (preceded by a minus sign), words are counted from the right side.

Notes: 1) If the specified word number is not found, nothing is returned.
2) Only the word itself is extracted, no attached punctuation or spaces.

Example:		Comments
1)	Before: <u>Vol. 3, No. 7 (1974)</u> Codes: <u>BW 2</u> After: <u>3</u>	We want to extract the Volume
2)	Before: <u>Vol. 3, No. 7 (1974)</u> Codes: <u>BW -1</u> After: <u>1974</u>	We want to extract the Date
3)	Before: <u>Vol. 3, No. 7 (1974)</u> Codes: <u>B W 6</u> After:	Doesn't exist

C "searchtext"

Condition: True if field contains search text

Description: This code will set the conditional processing "flag" to True if the field contains the specified search text. This way you can have processing codes that are only performed if the field contains a particular string of text.

- Notes:
- 1) This code is processed regardless of whether the conditional processing flag is True or False.
 - 2) To get the reverse, True if the field does not contain the search text, follow the C"searchtext" code with the C~ code (Condition reverse).
 - 3) The test ignores all subfield boundaries. The string of text may occur in any subfield, or even across subfields.
 - 4) The search text may include wild card symbols (* and ?). See the General Notes on wildcards (page [67](#)) for more details.
 - 5) You may enclose the search string in any pair of the three quote symbols (" , ' , `)
 - 6) When searching, case is ignored unless you type any of the search text in UPPER case.
 - 7) The double bar ("5", F6) can be used to indicate that the search text must be found at the start or end of the field (see "Search Strings", page [68](#))

Examples:

		<u>Comments</u>
1)	Before: S/N 009877 Codes: C"S/N" XC After:	Situation: mixed serial numbers and call numbers, want to clear serial numbers if they exist.
2)	Before: Z 76.5 .F33 1987 Codes: C"S/N" XC After: Z 76.5 .F33 1987	Unaffected, does not have "S/N"
3)	Before: Z 76.5 .F33 1987 Codes: C"S/N" C~ XC After:	Want reverse of above Therefore cleared, no "S/N"
4)	Before: PC Magazine, (1988) Codes: C'(*)' B'(*)' C~ XC After: (1988)	Break out date if found, else clear Note that C~ gives the reverse

C +

Conditional processing: true

Description: This sets the conditional processing "flag" to "true". That is, all codes following this one (until a "false" is found) will be processed. Other conditional processing codes may stop further processing. This is also useful if you wish to turn processing back on after a False result from one of the other conditional codes (eg: CE or C"search text"), see example 3.

- Notes: 1) You may use this to have some codes temporarily disabled with a C- (see below) and then later codes can be executed once a C+ is encountered (see examples 1 and 2). This can be helpful when you know you will reinstate some codes later that you want to temporarily disable now.
- 2) This code is processed regardless of whether the conditional processing flag is True or False.

Examples:		<u>Comments</u>
1)	Before: <u>computers; calculators; typewriters</u> Codes: <u>BS "; " C- U C+ "-xxx"</u> After: <u>computers-xxx5calculators-xxx5typewriters-xxx</u>	Disable Upper Code
2)	Before: <u>computers; calculators; typewriters</u> Codes: <u>BS "; " U "-xxx"</u> After: <u>COMPUTERS-xxx5CALCULATORS-xxx5TYPEWRITERS-xxx</u>	Enable Upper Code
3)	Before: <u>RC</u> Codes: <u>CE "MAIN" C+ "- " NS 1 NU NI 1</u> After: <u>RC-1</u>	Assume this is a location code We want "MAIN" if nothing, then add a number

C -

Conditional processing: false

Description: This sets the conditional processing "flag" to "false". That is, all codes following this one (until a "true" is found) will not be processed. Other conditional processing codes may restart processing.

- Notes: 1) One use of this code could be to have some codes temporarily disabled with a C- and then later codes can be executed once a C+ (see above) is encountered. This can be helpful when you know you will reinstate some codes later that you want to temporarily disable now.
- 2) This code is processed regardless of whether the conditional processing flag is True or False.

Example:

		<u>Comments</u>
1)	Before: computers; calculators; typewriters	
	Codes: BS "; " C- U C+ "-xxx"	Disable Upper Code
	After: computers-xxx5calculators-xxx5typewriters-xxx	
2)	Before: computers; calculators; typewriters	
	Codes: BS "; " U "-xxx"	Enable Upper Code
	After: COMPUTERS-xxx5CALCULATORS-xxx5TYPEWRITERS-xxx	

C ~

Conditional processing: reverse (not)

Description: This code reverses the conditional processing "flag", ie: from "true" to "false", or "false" to "true". When the flag is "false" further processing of codes is suspended, except for other Conditional codes, until the flag is set back to "true". This is especially useful if you want the opposite of CE or C"searchtext", ie: when not empty, or does not contain the search text.

Notes: 1) This code is processed regardless of whether the conditional processing flag is True or False.

Example:

Comments

- | | | | |
|----|---------|------------------------|-------------------------------------|
| 1) | Before: | | |
| | Codes: | <u>CE C~ F"AUTHOR"</u> | Put author here if <u>not</u> empty |
| | After: | | -> nothing happens |
| 2) | Before: | <u>ger</u> | Assume this is a language code |
| | Codes: | <u>c"eng" c~ qr</u> | Don't process (QR) if not english |
| | After: | | -> record is abandoned |

C E

Condition: True if Empy field

Description: This code will set the conditional processing "flag" to True if the field is currently empty. This way you can have processing codes that are only performed if the field is empty.

- Notes:
- 1) This code is processed regardless of whether the conditional processing flag is True or False.
 - 2) To get the reverse, True if the field is not empty, follow the CE code with the C~ code (Condition reverse, page [92](#)).
 - 3) The test ignores all subfield boundaries. If the field has any data (even spaces), then the result is False. It must be completely empty to be True.

Examples:

1) Before:
Codes:
After:

CE "None."
None.

Comments

Sometimes you want to have something there even if the field is empty.

2) Before:
Codes:
After:

Smith, John
CE "None."
Smith, John

3) Before:
Codes:
After:

XX
CE C~ "-" ns 1 nu ni 1
XX-1

Assume this is some code
Add an incrementing number if not empty.

4) Before:
Codes:
After:

CE QR

If this field is empty, we don't want this record.
Therefore, record is abandoned

C F n

Conditional processing: store False in memory n

Description: This code will set the conditional processing "flag" to False in one or all of the conditional processing memory "stores". There are 99 memory stores in all (numbered 1 to 99). A single store may be referenced by giving a single number (eg: CF 5), or you may specify all memory stores with an asterisk (eg: CF *) which will set all stores to False. These memory stores allow you to record some condition and then have later processing act on that condition (using the CR n - Condition Recall code, see page [96](#)). For instance, you may wish to execute some codes if a particular field had an "M" vs. an "S", or if a field had been blank or not.

- Notes:
- 1) This code is only processed if the current conditional processing flag is True.
 - 2) The memory stores are all set to False when processing starts and are not reset or changed during the conversion unless explicitly told by the processing codes. When processing records manually, one-at-a-time, the memory stores are reset whenever the input file is opened or reset (**Alt-F3**).

Examples:

		<u>Comments</u>
1)	Before: <u>S</u> Codes: <u>C"M" C~ CF 10</u> After: <u>S</u>	We want to note if an "M" was there Store False in #10 if NOT there Memory store 10 is now "False"
2)	Before: Codes: <u>CE CF*</u> After:	If this field is empty, we want to set all memory stores to "False" They would all now be False

C L

Condition: True if field contains all Lowercase characters

Description: This code will set the conditional processing "flag" to True if the field contains all Lowercase characters. If any of the text is uppercase, this will give the result of "False"

- Notes:
- 1) This code is processed regardless of whether the conditional processing flag is True or False.
 - 2) The test ignores all subfield boundaries. All characters in the entire field must be lowercase for this to be "True".

Examples:

		<u>Comments</u>
1)	Before: computer software review Codes: CL PA After: Computer Software Review	Journal name, all lowercase If it is all lowercase, capitalize each word
2)	Before: USA today Codes: CL PA After: USA today	Unaffected, some uppercase letters
3)	Before: life5USA today Codes: CL PA After: life5USA today	Two occurrences Unaffected, some uppercase in field

C R n

Conditional processing: Recall memory n

Description: This code will recall the conditional processing status of one of the conditional processing memory "stores". There are 99 memory stores in all (numbered 1 to 99). A memory store may be recalled by giving the memory store number (eg: CR 5). The Condition Recall code allows you to recall some earlier condition set with Condition Store (see page 97), True (page 98), or False (page 94). For instance, you may wish to execute some codes if a particular field had an "M" vs. an "S", or if a field had been blank or not.

- Notes:
- 1) This code is processed regardless of whether the current conditional processing flag is True or False.
 - 2) The memory stores are all set to False when processing starts and are not reset or changed during the conversion unless explicitly told by the processing codes. When processing records manually, one-at-a-time, the memory stores are reset whenever the input file is opened or reset (**Alt-F3**).
 - 3) Remember that the fields are processed in order of field number, input fields then output fields. This may require some special consideration if you wish to process an input field based on the condition of a higher numbered input field. In general, this can be resolved by creating a special input field, which comes after the field to be tested. Copy the information in from the original, earlier field, then perform the conditional processing.

Examples:

		<u>Comments</u>
1)	Before:	Starting with blank output field
	Codes:	<u>CR 10 F"Date" C~ F"Vol"</u>
	After:	<u>1991</u>
		If an "M" was encountered before, get "Date" field, otherwise get "Vol" field
		Say there had been an "M", got "Date" field
2)	Before:	<u>Ontario</u>
	Codes:	<u>CR 3 TE ", Canada" C~ TE " California"</u>
	After:	<u>Ontario, Canada</u>
		If Cond 3 set, then pub. in Canada

C S n

Conditional processing: Store memory n

Description: This code will store the current conditional processing "flag" in one of the conditional processing memory "stores". There are 99 memory stores in all (numbered 1 to 99). A memory store may be referenced by giving the number (eg: CS 5). These memory stores allow you to record some condition and then have later processing act on that condition (using the CR n - Condition Recall code, see page 96). For instance, you may wish to execute some codes if a particular field had an "M" vs. an "S", or if a field had been blank or not.

- Notes:
- 1) This code is processed regardless of whether the current conditional processing flag is True or False.
 - 2) The memory stores are all set to False when processing starts and are not reset or changed during the conversion unless explicitly told by the processing codes. When processing records manually, one-at-a-time, the memory stores are reset whenever the input file is opened or reset (**Alt-F3**). This allows you to record conditions that have EVER occurred during the conversion.
 - 3) Remember that the fields are processed in order of field number, input fields then output fields. This may require some special consideration if you wish to process an input field based on the condition of a higher numbered input field. In general, this can be resolved by creating a special input field, which comes after the field to be tested. Copy the information in from the original, earlier field, then perform the conditional processing.

Examples:

1) Before: S
 Codes: C"M" CS 10
 After: M

Comments

We want to note if an "M" was there or not
 Store True or False in #10
 Memory store 10 is now "False"

2) Before:
 Codes: CE CS 1
 After:

Store in memory store #1 whether this field
 was blank or not
 Memory store 1 is now True

CT n

Conditional processing: store True in memory n

Description: This code will set the conditional processing "flag" to True in one or all of the conditional processing memory "stores". There are 99 memory stores in all (numbered 1 to 99). A single store may be referenced by giving a single number (eg: CT 5), or you may specify all memory stores with an asterisk (eg: CT *) which will set all stores to True. These memory stores allow you to record allow you to record some condition and then have later processing act on that condition (using the CR n - Condition Recall code, see page [96](#)). For instance, you may wish to execute some codes if a particular field had an "M" vs. an "S", or if a field had been blank or not.

- Notes:
- 1) This code is processed regardless of whether the current conditional processing flag is True or False.
 - 2) The memory stores are all set to False when processing starts and are not reset or changed during the conversion unless explicitly told by the processing codes. When processing records manually, one-at-a-time, the memory stores are reset whenever the input file is opened or reset (**Alt-F3**).

Examples:

1) Before: M
Codes: C"M" CT 10
After: M

Comments

We want to note that an "M" was there
Store True in #10 if true
Memory store 10 is now "True" (Note: if there was no M, but #10 had been True, it still would be True)

2) Before:
Codes: CE CT*
After:

If this field is empty, we want to set all memory stores to "True"
They would all now be True

C U

Condition: True if field contains all Upper case characters

Description: This code will set the conditional processing "flag" to True if the field contains all Uppercase characters. If any of the text is lowercase, this will give the result of "False"

- Notes:
- 1) This code is processed regardless of whether the conditional processing flag is True or False.
 - 2) The test ignores all subfield boundaries. All characters in the entire field must be uppercase for this to be "True".

Examples:

		<u>Comments</u>
1)	Before: SMITH Codes: CU L P1 After: <u>Smith</u>	Last name, all upper case If it is all upper case, drop it to lower and capitalize the first word
2)	Before: Van HORNE Codes: CU L P1 After: <u>Van HORNE</u>	Unaffected, some lowercase letters
3)	Before: SMITH5Van HORNE Codes: CU L P1 After: <u>SMITH5Van HORNE</u>	Two occurrences Unaffected, some lowercase in field

D T

Date Today in form of YYYYMMDD

Description: This code will automatically insert today's date in the form of YYYYMMDD (eg: May 4, 1989 would appear as 19890504).

- Notes: 1) Your system clock must be set to today's date for this code to work correctly.
- 2) To create today's date in the form "Mmm. DD, YYYY" (eg: May 4, 1997), then follow the DT code by the DX code (see page [101](#), and example 2)

Examples:		<u>Comments</u>
1)	Before: Codes: DT After: <u>19970522</u>	Assuming today is May 22, 1997
2)	Before: Codes: DT DX After: <u>May 22, 1997</u>	Same assumption

D X

Date conversion: from YYMMDD to Mmm. DD, YYYY

Description: This code provides a simple conversion of date formats from the International standard "YYMMDD" (also used by dBase III), or "YYYYMMDD" to a text format "Mmm. DD, YYYY".

Notes: 1) If the format of the date does not seem to be either "YYMMDD" or "YYYYMMDD", or if the Month or Day are invalid, then the data will remain unchanged.

Examples:

		<u>Comments</u>
1)	Before: 881024 Codes: DX After: Oct. 24, 1988	
2)	Before: 18890115 Codes: DX After: Jan. 15, 1889	
3)	Before: 761603 Codes: DX After: 761603	Invalid Month (16) therefore, left unchanged

- 4) Input Field "Survey": bx17q993a5bf18h922
Codes for field "Loc": f"survey5[5-8]" Just want 5th to 8th characters
Results: g9935h922
- 5) Input Field "245": 10\$aComputers\$b: care and feeding\$wqfg345
Codes for "Title": f'2455\$abcd' Just want subfields a, b, c, d
Results for "Title": Computers : care and feeding
- 6) Output Field "ID": 77212 Note that this is an *Output* field
Codes for "CallNo": "HQ " f'id5/o" CallNo is also an output field, but after
Results for "CallNo": HQ 77212 "ID" (since it will be processed later)
- 7) Input Field "260": \$aQue\$bToronto\$c19965\$aQue\$bNew York\$c1997
Codes for "DATE": F"2605(1)\$c" Just want first date
Results for "DATE": 1996

F C

Field Copy (from Input field with same name)

Description: Looks for an Input field with the same name as the field for this code. Then draws the information from that input field. This is especially useful in the Global Field Processing Codes for the Output file to simply copy all fields with the same name.

- Notes:
- 1) You may use either the "Field Name" or the "Long Name/Description". If there are two Input fields with the same name, it will draw from the field with the lower number.
 - 2) If the field is divided into Subfields, it will include them all unless a range has been specified by the "S n-m" processing code (see page [119](#)) which specifies the first and/or last Subfields to be included in processing.
 - 3) You may enclose the field name in any pair of the three quote symbols (" , ' `)

Examples:

- | | | |
|----|---|-----------------------|
| 1) | Input Field "AU": | <u>Burton, Pierre</u> |
| | Output Field "AU": | |
| | Codes (for "AU"): | <u>FC</u> |
| | "AU" after processing: | <u>Burton, Pierre</u> |
| | | |
| 2) | Input Field "DT": | <u>1987</u> |
| | Output Field "DT": | |
| | <u>Output</u> Global Field Process Codes: | <u>FC</u> |
| | Codes (for "DT"): | |
| | "DT" After: | <u>1987</u> |

L

Lowercase conversion

Description: Takes the current field and converts all letters to lowercase.

Notes: 1) This is often useful in conjunction with the "P 1", or "P A" processing codes (see pages [112](#) and [113](#)) which then capitalize the first letters of the 1st word, or All words, respectively (see Example 2).

Examples:

		<u>Comments</u>
1)	Before: <u>LAW-HISTORY</u> Codes: L After: <u>law-history</u>	
2)	Before: <u>CALDWELL, DAVID A.</u> Codes: l pa After: <u>Caldwell, David A.</u>	lowercase, capitalize each

N C "searchtext"

Number Count number of occurrences of "searchtext"

Description: This code can be used to count the number of occurrences of any string in an entire field. The current field is replaced by that number. This is best used by making a copy of an existing field, use N C code and then use the results elsewhere.

- Notes:
- 1) This code works across the entire field, not separately for each occurrence.
 - 2) The number of occurrences of a field can be counted by using NS "5" (see examples 3-5)
 - 3) The search text may include wild card symbols (* and ?). See the General Notes on wildcards (page [67](#)) for more details.
 - 4) You may enclose the search string in any pair of the three quote symbols (" , ' , `)
 - 5) When searching, case is ignored unless you type any of the search text in UPPER case (see example 2).
 - 6) The double bar ("5", F6) can be used to indicate that the search text must be found at the start or end of the field (see "Search Strings", page [68](#))

Examples:		<u>Comments</u>
1)	Before: HQ 1235REF 1175HQ 22 Codes: NC "HQ" After: 2	Count number of items in "HQ"
2)	Before: HQ 1235REF 1175HQ 22 Codes: NC "hq" After: 2	Search text lower case Still finds both
3)	Before: HQ 1235REF 1175HQ 22 Codes: NC "5" After: 3	Count number of occurrences
4)	Before: Codes: NC "5" After: 0	Blank Count occurrences None
5)	Before: 55 Codes: NC "5" After: 3	Blank, except for occurrence markers Count occurrences There are 3 occurrences (all blank)

N I n

Numbering: Increment counter by value of n

Description: If a numeric counter has been started (see "N S n", page [108](#)), then increment it by the value of 'n'. This number can then be used in a field with the "N U" code (see page [109](#)).
This is especially useful for generating automatically numbered records for systems that require a unique number for each record.

- Notes:
- 1) You may only increment by positive numbers.
 - 2) This code is typically used only in the same set of processing codes with the "N S n" and "N U" codes (see Example 1). Also, it is usually used only in the Output Fields.
 - 3) You may only have one counter running at a time. Once a "N S n" is encountered, all further occurrences of that code are ignored until the conversion process is re-started.
 - 4) Separate "N I n" codes would all increment the same counter. Likewise, separate "N U" codes would draw from the same counter value.

Examples:

		<u>Comments</u>
1)	Before: Codes: <u>NS 1 NU NI 1</u> After: <u>1</u>	Blank, to start Start at 1, use the number, and increment by 1.
2)	Before: Codes: <u>NS 100 NU NI 10</u> After: <u>110</u>	Blank, to start Start at 100, Incr. by 10 after the <u>second</u> record

N S n

Numbering: set Starting value of counter to n

Description: Activates an automatic numeric counter, starting at the value of n. This number can then be drawn into a field with the "N U" code (see page [109](#)). Once this code has been encountered (ie: on the first record), it will no longer have any effect on subsequent records.
This is especially useful for generating automatically numbered records for systems that require a unique number for each record.

- Notes:
- 1) You may only start with a positive number.
 - 2) This code is typically used only in the same set of processing codes with the "N U" and "N I n" codes (see Examples). Also, it is usually used only in the Output Fields.
 - 3) You may only have one counter running at a time. Once a "N S n" is encountered, all further occurrences of that code are ignored until the conversion process is re-started.

Examples:

		<u>Comments</u>
1)	Before: Codes: <u>NS 1 NU NI 1</u> After: <u>1</u>	Blank, to start Start at 1, use the number, and increment by 1.
2)	Before: Codes: <u>NS 100 NU NI 10</u> After: <u>110</u>	Blank, to start Start at 100, Incr. by 10 after the <u>second</u> record

NU

Numbering: Use value of counter in current data

Description: If a numeric counter has been started (see "N S n", page [108](#)), then take the value of that counter and append it to the contents of the current field. This is especially useful for generating automatically numbered records for systems that require a unique number for each record.

- Notes:
- 1) You may only increment by positive numbers.
 - 2) This code is typically used only in the same set of processing codes with the "N S n" and "N I n" codes (see Examples). Also, it is usually used only in the Output Fields.
 - 3) You may only have one counter running at a time. Once a "N S n" is encountered, all further occurrences of that code are ignored until the conversion process is re-started.
 - 4) Separate "N U" codes would all draw from the same counter. Likewise, separate "N I n" codes would all increment the same counter value.

Examples:

		<u>Comments</u>
1)	Before: Codes: <u>NS 1 NU NI 1</u> After: <u>1</u>	Blank, to start Start at 1, use the number, and increment by 1.
2)	Before: Codes: <u>NS 100 NU NI 10</u> After: <u>110</u>	Blank, to start Start at 100, Incr. by 10 after the <u>second</u> record
3)	Before: <u>RC</u> Codes: <u>NS 1200 NU NI 1</u> After: <u>RC 1200</u>	Location: RC Start at 1200 to add to file

O "searchtext" Or: True if field contains search text OR previous condition was True

Description: This code will set the conditional processing "flag" to True if the field contains the specified search text or if the previous condition was "True". This way you can have processing codes that are only performed if the field contains a particular string of text or some other condition. For instance, you may wish to perform certain processing if either of two or more particular strings of text are present.

- Notes:
- 1) This code is processed regardless of whether the conditional processing flag is currently True or False.
 - 2) To get the reverse, True if the field does not meet any condition, follow the O"searchtext" code with the C~ code (Condition reverse, page [92](#)).
 - 3) The test ignores all subfield boundaries. The string of text may occur in any subfield, or even across subfields.
 - 4) The search text may include wild card symbols (* and ?). See the General Notes on wildcards (page [67](#)) for more details.
 - 5) You may enclose the search string in any pair of the three quote symbols (" , ' , `)
 - 6) When searching, case is ignored unless you type any of the search text in UPPER case.
 - 7) If there has been no previous Conditional code ("C") in the current field, then this code will act as if it were C+, since the conditional flag always starts out "True".
 - 8) See also the Conditional codes on pages [89](#) to [92](#) and the And code on page [74](#).
 - 9) The double bar ("5", F6) can be used to indicate that the search text must be found at the start or end of the field (see "Search Strings", page [68](#))

Examples:

		<u>Comments</u>
1)	Before: ISSN: 7123-8810	Given a field whose contents may vary Look for "ISBN:" <u>or</u> "ISSN:". If found, take it. If not, clear the data. It was there, so take it.
	Codes: <u>C"ISBN:" O"ISSN:" BR":" C~ XC</u>	
	After: <u>7123-8810</u>	
2)	Before: Stock #: 79-14467	Different data Look for ISBN/ISSN Not found, so field cleared
	Codes: <u>C"ISBN:" O"ISSN:" BR":" C~ XC</u>	
	After:	

O E

Or: True if field is Empty OR previous condition was True

Description: This code will set the conditional processing "flag" to True if the field is currently empty or if the previous condition was "True". This way you can have processing codes that are performed if the field is empty or contains a particular piece of text.

- Notes:
- 1) This code is processed regardless of whether the conditional processing flag is True or False.
 - 2) To get the reverse, True if the field does not meet any of the previous conditions, follow the OE code with the C~ code (Condition reverse, page [92](#)).
 - 3) The test ignores all subfield boundaries. If the field has any data (even spaces), then the result is False. It must be completely empty to be True.
 - 4) If there has been no previous Conditional code ("C") in the current field, then this code will act as if it were C+, since the conditional flag always starts out "True".
 - 5) See also the Conditional codes on pages [89](#) to [92](#) and the And code on page [74](#).

Examples:

- | | |
|----|--|
| 1) | Before: <u>Smith, John</u>
Codes: C", " OE C~ <u>XC "Corporate"</u>
After: <u>Smith, John</u> |
| 2) | Before: <u>University of Guelph</u>
Codes: C", " OE C~ <u>XC "Corporate"</u>
After: <u>Corporate</u> |

Comments

Want to note 'Corporate' when there's a corporate author, but leave alone if empty or a comma (indicating a personal author)

P 1

Proper (capitalize) the 1st word

Description: Capitalizes the first letter of the first word of the current data.

- Notes:
- 1) If the first letter of the first word is already in upper case, then it is not affected.
 - 2) The first "word" need not be the first characters in the entry. The Data Magician finds the first "word" (using it's standard definition of any group of letters or digits), and then checks to see if the first character of that word is lowercase. If so, it is changed.
 - 3) This process is repeated for every "subfield" within the field (unless limited by the "S n-m" code, see page [119](#)). Therefore, several multi-word phrases (eg: Authors, subject terms, etc.) can be changed if they are first put into separate subfields.
 - 4) This (and the "P A" code, see below) is often used in conjunction with the "L" (convert to lowercase) code to take something in all uppercase and convert it to upper/lower case.
 - 5) Currently, this code only handles the ASCII character set.

Examples:

		<u>Comments</u>
1)	Before: <u>taxation-laws</u> Codes: P1 After: <u>Taxation-laws</u>	
2)	Before: <u>indians5inuit5native people</u> Codes: p1 After: <u>Indians5Inuit5Native people</u>	Multiple subfields
3)	Before: <u>INDIANS5INUIT5NATIVE PEOPLE</u> Codes: L P 1 After: <u>Indians5Inuit5Native people</u>	All Uppercase drop to lowercase first

P A

Proper (capitalize) All words

Description: Capitalizes the first letter of the every word of the current data.

- Notes:
- 1) If the first letter of a word is already in upper case, then it is not affected.
 - 2) This process is repeated separately for every "subfield" within the field (unless limited by the "S n-m" code, see page [119](#)).
 - 3) This (and the "P 1" code, see above) is often used in conjunction with the "L" (convert to lowercase) code to take something in all uppercase and convert it to upper/lower case.
 - 4) Currently, this code only handles the ASCII character set.

Examples:		<u>Comments</u>
1)	Before: <u>taxation-laws</u> Codes: PA After: <u>Taxation-Laws</u>	
2)	Before: <u>indians5inuit5native people</u> Codes: pa After: <u>Indians5Inuit5Native People</u>	Multiple subfields
3)	Before: <u>INDIANS5INUIT5NATIVE PEOPLE</u> Codes: L P A After: <u>Indians5Inuit5Native People</u>	All Uppercase drop to lowercase first

Q C

Quit Conversion

Description: If this code is processed, it will cancel the conversion in progress. You would generally only use this code after one of the conditional processing codes (see pages [89](#) to [92](#)). This may be useful if you are concerned that if a particular field is empty or contains invalid characters, that the whole conversion is invalid.

Notes: 1) When this code is encountered, all further processing on the current record is cancelled, and the output file is closed. If you are in manual mode, you will be asked if you wish to quit. If you are in "auto execute" mode then the program will end.

Examples:

1) Before: 22317
Codes: CE QC
After: 22317

Comments

Assume this is a record number
If there is no record number,
something is assumed wrong with the file
and the conversion would be cancelled.

2) Before: RSN.00189
Codes: c"rsn" c~ qc
After: RSN.00189

Similarly, if "RSN" isn't there,
cancel the conversion.
But it's okay here.

Q F

Quit Field

Description: If this code is processed, it will cease processing codes in the current field. You would generally only use this code after one of the conditional processing codes (see pages 89 to 92). This may be useful if you are doing some complex processing and you do not want any more processing done after a given point if some condition is met (eg: the field is empty, or not).

- Notes:
- 1) When this code is encountered, all further processing on the current field is stopped. The field is left as it was at the point where the Q F code was encountered. It has no effect on other fields.
 - 2) One benefit of this code is to avoid processing time on a field that is empty, or fails some other condition. If there are a lot of other codes following the conditional, they would all be analyzed in case some condition became "true". The Q F code stops The Data Magician from even looking at the remaining codes, thus saving time.

Examples:		<u>Comments</u>
1)	Before: QC76.2.F56 1984 Codes: CE QF C" 19" S" 19" Year: 19" C" 18" S" 18" Year: 18" C+ BC After: QC 76.2 .F56 Year: 1984	A somewhat lengthy set of codes Note that all the processing was done
1)	Before: Codes: CE QF C" 19" S" 19" Year: 19" C" 18" S" 18" Year: 18" C+ BC After:	The same set of codes, but no data The codes after QF would be ignored, saving time

Q R

Quit Record

Description: If this code is processed, it will cancel processing of the record in progress. You would generally only use this code after one of the conditional processing codes (see pages [89](#) to [92](#)). This may be useful if you do not want to include records that contain (or don't contain) a particular code, or if a particular field is empty (or not).

Notes: 1) When this code is encountered, all further processing on the current record is cancelled. Nothing is written to disk for that record.

Examples:

		<u>Comments</u>
1)	Before: 22317 Codes: CE QR After: <u>22317</u>	Assume this is a record number If there is no record number, then this record is invalid and should not be included.
2)	Before: RSN 00189 Codes: c"rsn" c~ qc After: <u>RSN 00189</u>	Similarly, if "RSN" isn't there, cancel the conversion of this record. But it's okay here.
3)	Before: ger Codes: c"eng" c~ qr After:	Assume this is a language code Don't process if not english -> record is abandoned

S "search"replace"**Substitute "search" text with "replace" text**

Description: This code will search through the data in the current field and replace all occurrences of the "search" text with the specified "replacement" text. Especially useful is the ability to include "wildcard" characters (* and ?) in the "search" text.

- Notes:
- 1) There must be three (3) quote symbols. The one in the middle ends the "search" text, and begins the "replace" text. You may use any of the three different quote symbols (" , ' , `), thereby allowing you to embed any of them as part of the search and/or replacement text (see Example 3).
 - 2) The "search" text may include the "wildcard" characters (* and ?) to match variations of the search text (see Examples 2 and 5). See the General Notes on Wildcards (page [67](#)) for more information.
 - 3) The replace text may also contain wildcards. Each wildcard corresponds to the matching wildcard in the "search" text. See the note on Wildcards (page [67](#)) for more information. See example 7.
 - 3) If the text you want to "search" for is an '*' or a '?' (ie: the wildcard characters), see the "W * !" or "W ? !" codes (pages [129](#) and [130](#)) to temporarily change the wildcard symbols (see Example 6).
 - 4) If you do not specify any "replace" text (by having nothing between the 2nd and 3rd quote symbols), the effect will be to delete any occurrences of the "search" text (see Example 2).
 - 5) The search & replace will be repeated separately for each "subfield" within the current field, unless limited by the "S n-m" code (see page [119](#)).
 - 6) If the specified "search" text is simply the Subfield marker ("5", ASCII code 186), the subfield markers will be removed and replaced with the "replace" text (see Example 4).
 - 7) The double bar ("5", F6) can be used to indicate that the search text must be found at the start or end of the field (see "Search Strings", page [68](#))

Examples:

		<u>Comments</u>
1)	Before: <u>QA/76.7/.F45/1988</u> Codes: <u>S"/" "</u> After: <u>QA 76.7 .F45 1988</u>	Substitute slashes with spaces
2)	Before: <u>Dogs5<Broader>Animals5<Narrower>Poodle</u> Codes: <u>s'<*>"</u> After: <u>Dogs5Animals5Poodle</u>	Sub. anything between < and > with nothing (ie: delete)
3)	Before: <u>Ben "Butch" McMillan</u> Codes: <u>s'"'"</u> After: <u>Ben 'Butch' McMillan</u>	Sub. " with ' (use reverse apostrophe as quote)
4)	Before: <u>Dogs5Animals5Poodle</u> Codes: <u>s'5/'</u> After: <u>Dogs/Animals/Poodle</u>	Change subfield markers to /

- 5) Before: 123 p.\$bill.\$c24 cm. MARC field (Physical Desc.)
Codes: s'\$?:' Subst. \$ plus a char.
After: 123 p.; ill.; 24 cm. with semicolon (;) space
- 6) Before: *Computers; Word Processing; *Software
Codes: W!* s'!' Change wildcard to !, del *'s
After: Computers; Word Processing; Software
- 7) Before: x12345y67z
Codes: s"x*y??z"a*??bc" The * matches 12345, ?=6 and ?=7
After: a1234567bc Appears in the new order

S n-m**process only Subfields n to m**

Description: This code limits the processing of other codes to the "subfields" specified by 'n' and 'm', for the current field only. This is useful if you do not wish to apply some processing to all subfields, just a subset of them.

- Notes:
- 1) By default, processing is performed on all subfields. Once the "S n-m" code is encountered, processing is limited just to the subfields specified. Note that the next field processed is not affected by this code.
 - 2) The "-m" is optional. If you only specify "n", only that subfield will be processed (see Example 2). If you specify "n-" (no ending subfield number), then all subfields starting with 'n' will be processed.
 - 3) The "S n-m" code has a special effect on the "F 'fieldname'" and "F fieldnum" codes, in that rather than the whole input field being drawn in, only the subfields specified will be included (see Example 3). This feature can be more easily accomplished with the newer F'fieldname5(n-m)' code.
 - 4) The word "subfield" was misunderstood by the author when The Data Magician was created. It was used to refer to a multiple occurrences of a field, as opposed to a part of a field (as it does in MARC).

Examples:

- | | | <u>Comments</u> |
|----|--|--|
| 1) | Before: <u>DOGS5ANIMALS5POODLE</u>
Codes: <u>S.2-3 L P1</u>
After: <u>DOGS5Animals5Poodle</u> | put subfields 2-3 in lowercase and capitalize 1st letter |
| 2) | Before: <u>Z/22.7/.89/198653/4" tape</u>
Codes: <u>s.1 s"/" "</u>
After: <u>Z 22.7 .89 198653/4" tape</u> | Convert slash to space only on first subfield |
| 3) | Input Field "Imprint": <u>Toronto5McGraw-Hill51986</u>
"PUBLISHER" Before: <u>S2 F"IMPRINT"</u>
Codes: <u>S2 F"IMPRINT"</u>
"PUBLISHER" After: <u>McGraw-Hill</u> | We only want to get the publisher name |

S F

append a SubField marker to current data

Description: This codes appends a subfield marker (ASCII code 186, "5") to the current data. This is generally used to separate entries drawn from different input fields.

- Notes:
- 1) Once the data is divided into subfields, they can be manipulated separately using the "S n-m" code (see above). Output Files in the INMAGIC format will list each subfield on a separate line.
 - 2) The word "subfield" was misunderstood by the author when The Data Magician was created. It was used to refer to a multiple occurrences of a field, as opposed to a part of a field (as it does in MARC).

Examples:

- 1) Input Field "ADDR1": 123 Main ST.
Input Field "ADDR2": Suite 205
"STREET" Codes: F'ADDR1' SF F'ADDR2'
"STREET" After: 123 Main ST.5Suite 205

S M "text"

Substitute Marc subfield separator with "text"

Description: This code changes the text that is used to separate MARC subfields when they are extracted using the BM"subfields" code (page [81](#)) or the F"field5\$abc" (see page [102](#)). By default the separator is a space.

Notes: 1) This code only affects the current field and only those codes that occur after this one.

Examples:

		<u>Comments</u>
1)	Before: <u>0CaCanadaCxStatisticsCxYearbooks.</u> Codes: <u>bm"\$*</u> After: <u>Canada Statistics Yearbooks.</u>	Extract all subfields Separated by spaces
2)	Before: <u>0CaCanadaCxStatisticsCxYearbooks.</u> Codes: <u>sm" -- " bm"\$*</u> After: <u>Canada -- Statistics -- Yearbooks.</u>	Change separator to " -- "

S T "filename"

look up data via Substitute Table in "filename"

Description: The Substitute Table code allows you to specify a "look-up" table, which will be found in a file on disk. Each line of the look-up table specifies two items, the "look-up" item, and the item it is to be replaced with.

This is useful in converting standardized codes into text equivalents (eg: "k" to "kit", "b" to "book", etc.)

- Notes:
- 1) The format of the table is specified below. If the table file does not conform to those specifications, unpredictable results may occur.
 - 2) When performing a look-up, it compares the current data in each subfield with each entry in the Substitute Table until it finds a match, or until it reaches the end of the table. If it finds a match, it replaces the data in the subfield with the substitute value from the table.
 - 3) When comparing values, it ignores upper/lower case.
 - 4) You may use as many Substitute Tables as you wish, but see the next note for a discussion on conversion speed.
 - 5) The use of Substitute Tables may dramatically slow down the conversion process. The Data Magician only maintains one table in memory at a time. If you are using more than one table, the system reads the entire table into memory each time it encounters the "S T" code. The size of each table is also a significant factor in conversion speed. If you do need more than one table, and the "look up" values are completely different for each table, you may consider putting them all into a single table for increased speed.
 - 6) The absolute maximum number of entries in a Substitute Table is 32767. Aside from this limit, the practical limit is likely to be much smaller since the table is stored in memory and must share space with all other text being manipulated. The total text space is approximately 64000 characters.
 - 7) The table filename may include drive and path information (eg: "c:\datamage\format.tab"). As with all DOS filenames, it does not matter whether the name is in upper or lower case.

Table Format: The first line of the table must contain the number of entries in the table. If there are actually more entries than that number, the extra will be ignored. If there are less, you will simply waste memory space.

Each line following will contain the look-up and substitution values, separated by a comma (eg: b,book). The two pieces may also be enclosed in double quotes (eg: "NF","Non-Fiction")

Although the table does not have to be sorted, The Data Magician recognizes when a table is sorted and searches the table more efficiently. Note that when putting a table in sorted order, numbers sort before letters, and upper/lower case is ignored. Foreign and graphic characters sort last according to the IBM PC extended ASCII character tables.

Table Example:

7
a,archive
b,book
c,catalog
d,diagram
e,encyclopedia
f,film
g,globe

Table 1
Sample Substitute Table
"FORMAT.TAB"

Examples:

1) Before: B
Codes: S T 'format.tab'
After: book

2) Before: h
Codes: S T 'format.tab'
After: h

Comments

Using Table above
Note: case is ignored

Code not in table above
therefore, left as is

T B "text"

Text Begin if not empty

T B n

Text Begin with string number n

Description: This code checks to see if the current field is empty. If not, it adds the quoted text to the beginning of the current data. If it is empty, it does nothing (see example 2). In the second form, T B n, stored string number 'n' (see T S n, page [127](#)) is added to the beginning of the current data (see example 4).

- Notes:
- 1) The text is added to every non-empty subfield within the current field, unless limited by the "S n-m" code (see page [119](#)). See example 3.
 - 2) The stored strings can be numbered from 0 to 9. They start as blank at the beginning of the conversion (unless the /0 to /9 switches are used to start The Data Magician) and are not cleared between records. This allows you to pass data between records.

Examples:

Comments

- | | | |
|----|---|--|
| 1) | Before: <u>National Law Journal</u>
Codes: <u>TB"\$a"</u>
After: <u>\$aNational Law Journal</u> | Put \$a in front (for MARC records) |
| 2) | Before:
Codes: <u>tb '\$a'</u>
After:
 | Current field is blank
Add '\$a' to beginning of field
Nothing is added to the blank field |
| 3) | Before: <u>computers5programming55C</u>
Codes: <u>s 2- tb ' -- '</u>
After: <u>computers5 -- programming55 -- C</u> | start at 2nd subfield |
| 4) | Stored String 1: <u>HQ</u>
Before: <u>12354565789</u>
Codes: <u>TB 1</u>
After: <u>HQ1235HQ4565HQ789</u> | This may be a location code

Prefix String 1 |

TE "text"
TE n

Text End if not empty
Text End with string number n

Description: This code checks to see if the current field is empty. If not, it adds the quoted text to the end of the current data. If it is empty, it does nothing (see example 2). In the second form, TE n, stored string number 'n' (see T S n, page [127](#)) is added to the end of the current data.

- Notes: 1) The text is added to every non-empty subfield within the current field, unless limited by the "S n-m" code (see page [119](#)).
- 2) The stored strings can be numbered from 0 to 9. They start as blank at the beginning of the conversion (unless the /0 to /9 switches are used to start The Data Magician) and are not cleared between records. This allows you to pass data between records.

Examples:

		<u>Comments</u>
1)	Before: <u>Smith</u> Codes: TE", " After: <u>Smith</u>	going to be appending a first name
2)	Before: Codes: <u>te ' '</u> After: 	Nothing added to a blank field
3)	Stored String 5: <u>xyz</u> Before: <u>12354565789</u> Codes: TE 5 After: <u>123xyz5456xyz5789xyz</u>	Append stored string 5

T R n **Text Recall string number **n****

Description: This code will recall a stored text string and append it to the current field. 'n' can be a number from 0 to 9. The string can be stored by the TS code (see page [127](#)), or using the /0 to /9 command-line switches (see page [142](#)).

- Notes: 1) The text is added to every subfield within the current field (see examples 2 & 3), unless limited by the "S n-m" code (see page [119](#)).
- 2) The stored strings can be numbered from 0 to 9. They start as blank at the beginning of the conversion (unless the /0 to /9 switches are used to start The Data Magician) and are not cleared between records. This allows you to pass data between records.

Examples:			<u>Comments</u>
1)	Stored String 8: Before: Codes: After:	<u>National Law Journal</u> "Journal: " TR8 <u>Journal: National Law Journal</u>	Add literal "Journal: " plus string 8
2)	Before: Codes: After:	 tr8 <u>National Law Journal</u>	Start with a blank field Assume String 8 same as above
3)	Stored string 0: Before: Codes: After:	<u>-HQ</u> <u>1765513901523678</u> tr0 <u>1765-HQ513901-HQ523678-HQ</u>	3 occurrences add string 0 to each

T S n

Text Store string number **n**

Description: This code takes the contents of the current field and stores it into String number "n" (where "n" can be from 0 to 9). The current field is not affected in any way by this code.

- Notes:
- 1) The stored strings can be numbered from 0 to 9. They start as blank at the beginning of the conversion (unless the /0 to /9 switches are used to start The Data Magician, see page [142](#)) and are not cleared between records. This allows you to pass data between records.
 - 2) Storing a blank field will erase the contents of the stored string (see example 2)

Examples:

		<u>Comments</u>
1)	Before: <u>National Law Journal</u> Codes: TS 5 After: <u>National Law Journal</u> Stored String 5: <u>National Law Journal</u>	Store current text in String # 5 Text is unaffected
2)	Before: Codes: <u>TS 5</u> After: Stored String 5:	Current field is blank Field unaffected String 5 is now blank

U

Uppercase conversion

Description: Takes the current field and converts all letters to upper case.

Notes: 1) This is useful for creating consistent use of case in such fields as postal codes or classification codes.

Examples:

- 1) Before: n1h 6j6
Codes: U
After: N1H 6J6

- 2) Before: caldwell, david a.
Codes: u
After: CALDWELL, DAVID A.

- 3) Before: bf 45 .m24 1981
Codes: U
After: BF 45 .M24 1981

W * !

Wildcard change: * for ! (or another)

Description: This code temporarily changes the multiple character wildcard ('*') to some other symbol. This is done to allow the '*' to be included as is in "search" text rather than being treated as a wildcard.

- Notes:
- 1) The allowed wildcard characters are: *, ?, !, @, #, \$, %, ^, &, |, /, \, ~, +, and =.
 - 2) The system will not allow you to set both the multiple character wildcard and the single character wildcard character to the same character.
 - 3) This change remains in effect only to the end of processing codes for the current field. The wildcard then reverts back to the original character.

Examples:

		<u>Comments</u>
1)	Before: *Computers; Word Processing; *Software Codes: W*! s'*! After: Computers; Word Processing; Software	Change wildcard to !, del *'s
2)	Before: Dogs5<Broader>Animals5<Narrower>Poodle Codes: w*! s'<!> After: Dogs5Animals5Poodle	Sub. text between < and > with nothing, using ! wildcard

W ? !

Wildcard change: ? for ! (or another)

Description: This code temporarily changes the single character wildcard ('?') to some other symbol. This is done to allow the '?' to be included as is in "search" text rather than being treated as a wildcard.

- Notes:
- 1) The allowed wildcard characters are: *, ?, !, @, #, \$, %, ^, &, |, /, \, ~, +, and =.
 - 2) The system will not allow you to set both the multiple character wildcard and the single character wildcard character to the same character.
 - 3) This change remains in effect only to the end of processing codes for the current field. The wildcard then reverts back to the original character.

Examples:		<u>Comments</u>
1)	Before: <u>?Computers?; Word Processing; ?Software?</u> Codes: W?! s?" After: <u>Computers; Word Processing; Software</u>	Change wildcard to !, del ?'s
2)	Before: <u>123 p.\$bill.\$c24 cm.</u> Codes: s'\$=:! After: <u>123 p.; ill.; 24 cm.</u>	MARC field (Physical Desc.) Subst. \$ plus a char. with semicolon (;) space

X A

Delete: All extra spaces

Description: Reduces all multiple spaces to a single space. This is helpful when downloading from databases which "format" the text by inserting multiple spaces between words to "right justify" them.

Notes: 1) If there is only a single space, it is left as is.

Examples:

- | | | | |
|----|---------|--|--|
| 1) | Before: | <u>Effects of opioids on smooth muscle</u> | Comments
Multiple spaces
between words |
| | Codes: | <u>X A</u> | |
| | After: | <u>Effects of opioids on smooth muscle</u> | |
| 2) | Before: | <u>Cell membrane potentials5Drug interaction</u> | |
| | Codes: | <u>xa</u> | |
| | After: | <u>Cell membrane potentials5Drug interaction</u> | |

X B "punctuation"

Delete: specified punctuation from Beginning of field data

Description: This code deletes leading punctuation (or any other characters specified) at the beginning of a field. Any of the characters specified, within quotes, will be deleted in any order until a character that is not in that list is encountered. Thus if you wish to delete two dashes at the beginning of a field, you need only specify a single dash. If none of the characters appear at the beginning of the field, this code has no effect.

- Notes:
- 1) This process will be repeated separately for each "subfield" within the current field, unless limited by the "S n-m" code (see page [119](#)).
 - 2) See also the codes **XE** (page [134](#)) and **XP** (page [136](#)) for deleting trailing characters or punctuation.

Examples:

		<u>Comments</u>
1)	Before: -- Aardvarks -- Codes: XB "-" XE "-" After: Aardvarks	Subject (?) Strip dashes and spaces (Note: order doesn't matter)
2)	Before: #23 - 7717 Rodero Drive Codes: XB "0123456789 -#" After: Rodero Drive	Just want street name Strip numbers, dashes, number sign (and spaces!)
3)	Before: - a -5- b -5- c - Codes: xb "-" After: a5b5c	

X C

Delete: Clear everything in current field

Description: This code deletes everything in the current field. This can be useful when used with conditional processing codes to eliminate fields that do not meet some criterion (see example 3).

Notes: 1) This code ignores subfield boundaries, and simply eliminates all the data in the current field (see examples 1 and 2).

Examples:

Comments

1) Before: President
Codes: X C
After:

2) Before: computers5calculators5typewriters
Codes: xc
After:

3) Before: S/N 77145
Codes: c"S/N" C~ xc
After: S/N 77145

We want to erase if not a serial number

X E "punctuation"

Delete: specified punctuation from End of field data

Description: This code deletes trailing punctuation (or any other characters specified) at the end of a field. Any of the characters specified, within quotes, will be deleted in any order until a character that is not in that list is encountered. Thus if you wish to delete two dashes at the end of a field, you need only specify a single dash. If none of the characters appear at the end of the field, this code has no effect.

- Notes:
- 1) This process will be repeated separately for each "subfield" within the current field, unless limited by the "S n-m" code (see page [119](#)).
 - 2) This code is a more generic version of the XP code (see page [136](#)). The XP code could be done using **XE " ./- "** (note the space between the quote and the period). The XP code is simply faster to type.

Examples:

		<u>Comments</u>
1)	Before: <u>Guelph, Ont. :</u> Codes: <u>XE " : "</u> After: <u>Guelph, Ont.</u>	Place of Publication Strip colons and spaces (Note: order doesn't matter)
2)	Before: <u>166744 .C.</u> Codes: <u>XE ".C"</u> After: <u>166744</u>	OCLC-type number Can strip letters, too!
3)	Before: <u>Dogs.5Cats.5Pets -- Household</u> Codes: <u>xe " : , "</u> After: <u>Dogs5Cats5Pets -- Household</u>	

X L

Delete: blanks on Left

Description: This code deletes all leading blanks (ie: on the Left) for each subfield in the current field. This is useful when reading data from a fixed field file (such as dBase III), or after breaking up a field into component pieces.

- Notes:
- 1) This process will be repeated separately for each "subfield" within the current field, unless limited by the "S n-m" code (see page [119](#)).
 - 2) This code (and the "X R" code) is often included as one of the Global Processing Codes when reading a dBase III file, to avoid having to repeat the code for each field.

Examples:

		<u>Comments</u>
1)	Before: .1001 Codes: X L After: 1001	Fixed field number
2)	Before: Thompson Codes: X L After: Thompson	Has a single leading space after previous processing
3)	Before: computers5 calculators5 typewriters Codes: xr After: computers5calculators5typewriters	

X P Delete: Predifined punctuation (" .,-") from End of field data

Description: This code deletes trailing punctuation at the end of a field. Any of the following characters (space, period, comma, slash, dash) will be deleted in any order until a character that is not in that list is encountered. If none of the characters appear at the end of the field, this code has no effect.

- Notes:
- 1) This process will be repeated separately for each "subfield" within the current field, unless limited by the "S n-m" code (see page [119](#)).
 - 2) This code is a more specific version of the XE code (see page [134](#)). The XP code could be done using **XE " .,-"** (note the space between the quote and the period). The XP code is simply faster to type.

Examples:		Comments
1)	Before: <u>Ants on parade /</u> Codes: XP After: <u>Ants on parade</u>	Title entry (MARC 245\$a) Strip trailing punct.
2)	Before: <u>McGraw-Hill,</u> Codes: XP After: <u>McGraw-Hill</u>	Publisher
3)	Before: <u>Dogs.5Cats.5Pets -- Household</u> Codes: xp After: <u>Dogs5Cats5Pets -- Household</u>	

X R

Delete: blanks on Right

Description: This code deletes all trailing blanks (ie: on the Right) for each subfield in the current field. This is useful when reading data from a fixed field file (such as dBase III), or after breaking up a field into component pieces.

- Notes:
- 1) This process will be repeated separately for each "subfield" within the current field, unless limited by the "S n-m" code (see page [119](#)).
 - 2) This code (and the "X L" code) is often included as one of the Global Processing Codes when reading a dBase III file, to avoid having to repeat the code for each field.

Examples:

		<u>Comments</u>
1)	Before: <u>President</u> Codes: <u>X R</u> After: <u>President</u>	Fixed field "POSITION"
2)	Before: <u>Joshua</u> Codes: <u>X R</u> After: <u>Joshua</u>	Has a single trailing space after previous processing
3)	Before: <u>computers 5calculators 5typewriters</u> Codes: <u>xr</u> After: <u>computers5calculators5typewriters</u>	

X S

Delete: excess Subfield markers

Description: This code eliminates redundant subfield markers which may accumulate when adding several fields together or using the Break into Subfields code (page [87](#)). When writing out an INMAGIC or MARC file, The Data Magician automatically ignores empty subfields, however, they may make it more difficult to perform certain functions. You may for example, wish to draw out the contents of the second used subfield.

Notes: 1) The word “subfield” was misunderstood by the author when The Data Magician was created. It was used to refer to a multiple occurrences of a field, as opposed to a part of a field (as it does in MARC).

Examples:

1) Before: computers55calculators5
Codes: XS
After: computers5calculators

Command Line Options (Batch Processing)

Introduction

When you first get started with The Data Magician, you will be using the menus to load previously saved settings files, then modify them to read and create different files with the same settings. If you find yourself doing this frequently, or if you wish to make it easier for someone else to run conversions, you may wish to consider using the command line options.

Right from DOS, you can tell The Data Magician to load a settings file and give the Input and Output file names and execute automatically. When it is done, it will return to the operating system. Thus, you can even create "batch" files in DOS which allow you to give a simple command and have the whole conversion performed without further interaction with you. This is particularly useful if you want to set up the conversion for someone else to perform, who may have little knowledge of DOS (and even less of The Data Magician!). There is a sample batch included called "**CONVBIP.BAT**" which can be used to run the BIP.SET conversion (or modified to run others).

The options are entered following the DATAMAGE command. Each options starts with a forward slash ("/") followed by the option letter. Some options also require additional information (such as a file name), which is given following an equal sign.

Example: to load the settings file "STAFF" and execute automatically, you would give the command: **datamage /s=staff/x**

The options available are given below.

Command Options

/S=filename The Data Magician will load the specified settings file, if it exists. You may specify a full path name, including drive, directory and filename. If no extension is given then ".SET" is assumed.

Example: **/s=c:\datamage\utlasinm**
 will load the settings file "utlasinm.set" from drive C in the \DATAMAGE directory

The conversion will not be run automatically unless you also use the /x option (see below). If you do not include /x, then you will be shown the Conversion Summary Screen after The Data Magician starts.

Note that none of the other options will be processed if you have not specified a settings file on the command line.

/A=filename Use the specified filename as the Input file rather than the one given in the settings file. The filename may include drive, directory, filename and extension.

Example: **/s=utlasinm /a=utlas2.mrc**
will load the settings file "UTLASINM.SET" but expect the Input file to be "UTLAS2.MRC" instead of the file given in the settings.

Note that this option is ignored if the /s option (above) has not been included.

/B=filename Use the specified filename as the Output file rather than the one given in the settings file. The filename may include drive, directory, filename and extension.

Example: **/s=utlasinm /b=utlas2.inm**
will load the settings file "UTLASINM.SET" but will create "UTLAS2.INM" as the Output file instead of the file given in the settings.

Note that this option is ignored if the /s option (above) has not been included.

/F Starting The Data Magician with this option will cause it to add unknown fields to both the Input and Output Fields. This is mostly useful in cases where you are converting a data file but using the same field names for Input and Output. You would likely also need to be using the "FC" processing code (See page [104](#)) to copy the Input data to the Output field without explicitly naming the fields to be copied.

/H or /? Starting The Data Magician with either of these switches will display a Help Page describing all of the switches available and then stop.

/I=n Provide memory space for up to "n" Input fields. By default, The Data Magician allows space for up to 100 Input fields. If you need more than that (or if you are tight on memory space and need less), you may specify the number required. Any settings file saved with more than 100 Input fields will automatically allow those when the file is loaded again at a later time, even if this option is not used. (Note: The Data Magician will actually provide 10 more fields than required, when loading a settings file, to allow for additional fields)

Example: **/i=200**
will allow you to request up to 200 Input fields

This option would typically be used when starting The Data Magician manually since any settings file loaded using the /s option will automatically allow for the number of fields specified by that file.

/L When reading a file with very long lines (ie:greater than 250 characters) running the program with this option will read infinitely long lines. Unfortunately, the conversion is slower, therefore this is only recommended if you experience difficulties reading long fields (like abstracts). So far, this has only been a problem when reading from records downloaded from CD-based products.

/M Starts The Data Magician in Monochrome mode. This is useful if you are using a non-colour monitor with a Colour Graphics display card (eg: CGA, EGA, VGA). The Data Magician detects the presence of such a card and then assumes you have a colour display. If not, starting with the /M option will display everything in black and white (or amber or green).
You may also switch between Colour and Monochrome by pressing **M** for Monochrome/Colour switch at the Main Menu, or by pressing **Ctrl-F9** anywhere else.

/N=startnum Specify a starting number for the automatic number generator processing code. When creating the processing codes, you may use the "NS n" code to specify a starting number. Once you run that conversion, you will presumably need to change that to a new starting number. This option allows you to override the number specified in the processing code and use this number instead.

Example: **/n=527**

will number the first record as 527, despite any processing codes to the contrary

Note that this option is ignored if the /s option (above) has not been included.

/O=n Provide memory space for up to "n" Output fields. By default, The Data Magician allows space for up to 100 Output fields. If you need more than that (or if you are tight on memory space and need less), you may specify the number required. Any settings file saved with more than 100 Output fields will automatically allow those when the file is loaded again at a later time, even if this option is not used.

Example: **/o=200**

will allow you to request up to 200 Output fields

This option would typically be used when starting The Data Magician manually since any settings file loaded using the /s option will automatically allow for the number of fields specified by that file.

/T Tabbed output. If the system you will be loading into requires Tabs between fields (rather than quotes and commas) use Output Format 1 (delimited ASCII) and run the program with this switch. The output file will then be created using Tabs instead of quotes and commas.

/X Automatically execute the conversion once The Data Magician has started. When finished, it will return to the operating system. You are not prompted for anything. Therefore, if the output file already exists, it will be overwritten without confirmation! (so be careful about your filenames).

Example: **/s=geacproc/x**
will load the settings file "GEACPROC.SET" in the default directory and execute automatically, returning to DOS when finished.

If any error condition is encountered, the program stops with a message as to the problem. Examples include: the input file doesn't exist, invalid path or directory name for the output file, etc.

Note that this option is ignored if the /s option (above) has not been included.

/0 to /9 Assigns a text string to Stored String numbers 0 through 9. Put the string in quotes after the equal ("=") sign. This makes a convenient way of passing a fixed string into converted records in a batch file.

Example: **/1="Main Branch"**

If a matching quote is not found, it will take the remainder of the command line as the stored string.

This string can then be recalled any time in the conversion using the TRn code (see page [126](#)) or the TBn or TEn codes (pages ? and ?).

General Notes

- 1) It does not matter whether command options, file names, etc. are entered in UPPER or lower case.
- 2) It also does not matter if a space precedes a slash starting a new option.
- 3) The order of the options is not important.
- 4) The logic of "/a" for specifying the Input file and "/b" for the Output file, is that those are the letters in the Main Menu where you specify those options. The idea being you convert "File A" to "File B".

Command Line Errors

If the command line has invalid option letters, or you give a file name that does not exist, The Data Magician will stop, point (with the ^) to the error and inform you of the problem. You may then correct the problem and re-execute the command. If the command line has been included in a batch file, the batch file will simply continue.

Usage examples

The following are some examples of command lines and what they would mean:

- 1) **datamage /s=canole /a=\smartcom\can1.asc /b=linmagic\can1.inm /n=321 /x**

This would start The Data Magician, load "CANOLE.SET" from the current directory, read "CAN1.ASC" from the "\SMARTCOM" directory, create the file "CAN1.INM" in the "\INMAGIC" directory, record numbering would start at 321, and the whole conversion would be executed automatically.

- 2) **datamage /i=200 /o=150**

This would start The Data Magician and allow space for 200 Input field names and 150 Output field names. You would then see the Main Menu.

- 3) **datamage /m**

This would start The Data Magician in Monochrome mode. It has no effect when used with a Monochrome Display Adaptor, but on a Colour Graphics Adaptor it will force it to display in black and white only.

- 4) **datamage /a=can1.asc /b=can1.inm /x**

This would simply start the program and go to the Main Menu. Since no settings file name was specified with /s, those other options are meaningless.

- 5) **datamage /s=staff /e**

The program would stop and point to the letter "e" and inform you that it is an invalid option letter.

- 6) **datamage /s=marc2lib /1="HQ" /x**

The program would automatically start converting using the marc2lib settings file and have the text "HQ" in Stored Text String #1, which could be used as part of the settings.

Miscantations (Error Messages)

Introduction

There are two types of errors that can arise while working with The Data Magician - "anticipated" errors, and "unanticipated" errors. Anticipated errors will appear on the bottom of the screen with a one-line description of the problem. The majority of the "anticipated" errors deal with invalid processing codes, improper use of editing keys, or disk and file problems. "Unanticipated" errors, also known as "system" errors, are those that The Data Magician has no particular method of handling.

The differences between these, and how to handle them are described in the sections below.

Anticipated Errors

These are errors where The Data Magician recognizes the problem, and provides a one-line description at the bottom line of the screen. Typically, you may correct the problem and continue. If you need clarification of the problem, you may press **F1** to get help on that error message (if the Help file is available).

The one-line error messages are in the file *DATAMAGE.MSG*. If the file is not in the current or *DATAMAGE* directory, then you will simply be presented with the message "Error number ##: Message not available", where ## is replaced by the error message number. In that event, you may refer to Appendix A for a list of those error messages.

When an error message is presented, you will hear a "beep". If you wish to turn that "beep" off for the remainder of that session, press **F9**.

System Errors

There are many situations that can create "system" errors. These may include: insufficient memory, hard disk errors, "bugs" (faulty software design), etc. Whatever the cause, if a problem arises that The Data Magician has no other way to solve the problem, you will be presented with the message:

"System Error #nn has occurred at position: ppppp. Try to continue? "

- where **nn** is the System Error number, and **ppppp** is the program location.

A list of System Errors can be found in Appendix B.

You are now expected to press **Y** or **N**, depending on whether you wish to try to continue. The advantage in trying to continue is that you may get a chance to save a settings file on which you had spent a lot of time and would like to save. The disadvantage is that the error may be such that if tries again the computer may "lock up", and you may need to shut it off and re-start it. It is always unfortunate when this sort of problem arises, and the software tries to prevent these situations, but unforeseen circumstances sometimes generate such severe difficulties.

Recording and Reporting Problems

Whenever possible, it is a good idea to get a printout of the screen when a System Error (or any other unusual situation) occurs. If you have a printer attached this can be done by pressing **Shift-PrtSc**. Otherwise, just write down as much of the information as you can. Note especially the error number, the program location, and, when you are next at the Main Menu, the version and serial number. It is also very helpful to try to record the events that lead up to the error. This information should be available if you call or send email for support.

You may send email to: **support@folland.com** for help. You may also wish to check out the Folland Software Services Web Site at: <http://www.folland.com>. The BBS may be of some help, or there are instructions on subscribing to the The Data Magician list server, if you haven't already.

Appendix A: Standard Error Messages

Standard errors are those that are anticipated by The Data Magician and are handled appropriately. The error message is displayed at the bottom of the screen, accompanied by a "beep" (if not turned off). Pressing **F1** will display a Help message (if available) relating to that error.

<u>Error Number</u>	<u>Error Message</u>
1	Value entered is outside of allowed range
2	No room to insert. Press Shift-F7 for full screen edit
3	Invalid editor key. Press Shift-F1 for Help
4	Cannot CUT, highlighting is Off. Press F5 to Highlight
5	Not a valid dBase II, III or IV file
6	dBase Memo file not found
7	Field name found in input file that was not defined
8	Processing Code Error: (1)st or (A)ll expected after (P)roper
9	Processing Code Error: Field name or number is invalid
10	Processing Code Error: Code not recognized
11	Processing Code Error: Must specify field name, number, or (C)opy
12	(L)eft,(R)ight,(B)egin,(E)nd,(A)ll,(C)lear,(P)unct, or (S)ubfields expected after (X)
13	Processing Code Error: Invalid Search/Replace format, use: S 'old'new'
14	(T)able, sub(F)ield, Number or Search/Replace string expected after (S)
15	(E)mpty, (S)tores, (R)ecall, (T)rue, (F)alse, (L)ower, (U)pper,+,-,~, or quoted text expected
16	No room for requested number of fields. Save, re-run and request more space.
17	A number is expected after (N)umber (S)tart or (N)umber (I)ncr.
18	Processing Code Error: (S)tart, (I)ncr., or (U)se expected after (N)umber
19	Processing Code Error: Date Input or Output format expected in quotes
20	(I)input, (O)utput, (C)onvert, (T)oday, or (X) (special) expected after Date
21	Processing Code Error: a number is expected after (W)ord ...
22	(F)ield, (W)ord, a number, or string expected after (B)reak (L)eft/(R)ight
23	Processing Code Error: quoted string expected after (B)reak into (S)ubfields
24	(C)all, (L)eft, (M)arc, (R)ight, (S)ubfield, (W)ord, e(X)tract or quotes expected
25	Processing Code Error: May not set single and multi-char wildcards the same
26	Processing Code Error: Allowable wildcards: *,?!,@,#,\$,%,&,+,=, ,\,/
27	Processing Code Error: ? or * expected after set (W)ildcard
28	That key is inappropriate at this time.
29	File name specified does not exist.
30	The path/directory given does not exist, please correct.
31	The disk drive is not ready, check drive door.
32	That is not a valid file and/or directory name.
33	Something is wrong with the disk. Unable to access file.
34	Unknown problem reading file or disk.
35	Help File not available

- 36 Not a valid Help File, or file is corrupted
- 37 Help message not available
- 38 Invalid MARC Communications File format
- 39 MARC subfield expected in quotes after (B)reak (M)ARC
- 40 Text in quotes expected after (T)ext (B)egin or (T)ext (E)nd
- 41 (B)egin, (E)nd, (R)ecall, or (S)tore expected after (T)ext
- 42 File name of (S)ubstitute (T)able expected in quotes
- 43 (F)ield, (R)ecord or (C)onversion expected after (Q)uit
- 44 Expecting a specific condition number or * for all
- 45 Conditional Code number specified is greater than the maximum
- 46 Invalid Field Option. Valid sample: F'field5(1-5)[3-9]/o\$abc'
- 47 Marc Subfield Substitution expected after (S)ubstitute (M)arc
- 48 Expecting String Number (0-9) after TS, or TR

Appendix B: System Error Messages

System errors are errors which are not anticipated by The Data Magician and appear with the message:

"System Error #nn has occurred at position: ppppp. Try to continue? "

- where **nn** is the System Error number, and **ppppp** is the program location.

The following is a selected list of possible error codes. There are others which are highly unlikely to occur. Many of these are actually taken care of by The Data Magician, but are listed anyways (just in case!)

<u>Error Number</u>	<u>Error Message</u>
5	Illegal Function call
6	Numeric Overflow
7	Out of Memory
9	Subscript out of range
10	Duplicate Array Definition
11	Division by Zero
14	Out of string space <i>When this error happens, it could be that the record is just too big to be handled by The Data Magician. There is no specific maximum record size, but typically records greater than about 10,000 bytes will cause this problem. Smaller records could also have a problem depending on the processing codes being used (many make copies of fields as they are working, so a single large field could be a problem). Another common problem is that the beginning and end of record definitions may be wrong, resulting in The Data Magician trying to read the entire file into a single record. This is most likely to happen on a Tagged file - check the Start and End of Record characters. Finally, this can happen on records with invalid record sizes, especially MARC and MicroSIS files. If possible, check the data file.</i>
15	String too long
24	Hardware device time-out
25	Hardware device fault
27	Printer out of paper
51	Compiler Internal Error
52	Bad file number
53	File not found
54	Bad file mode
55	File already open
56	Device Input/Output error
61	Disk full
62	Input past end of file

63	Bad record number
64	Bad file name
67	Too many files
68	Device unavailable
70	Disk is write-protected
71	Disk not ready
72	Disk media error
75	Path/File access error
76	Path not found
202	Out of temporary string space
242	String memory corrupted

Appendix C: Sample Settings Files

Introduction

The following pages contain examples of settings files we have created. They are presented alphabetically for easier reference. All of the settings illustrated here are included on the SAMPLE DATA disk, along with the input files. You may run the conversions and play with them. They provide illustrations of possible ways to process various file types. They may be used as starting points for your own conversions, or simply as a guide. Please take careful note about what fields are being put where when using these for your own conversions (especially in MARC conversions), since our needs/ideas may not correspond with yours.

Note on square brackets [] in settings examples

Please note that although the examples on the following pages have square brackets around the codes for each field, they are there only for the printout. They do not appear on the screen if you load the settings file, nor should you enter them if you type in these examples yourself. On the screen, Input and Output File Specifications will appear in reverse video to help identify spaces.

Technical aside on how the settings print-outs were made

The format of the settings files is how they would look if you choose the Print Settings option (**P** from the Main Menu, or **Alt-P** or **Ctrl-F2** from anywhere else). In this case we actually use the print to a file option, by pressing **F** when asked if ready to print. The Data Magician, then prompted for the name of the file to "print" to. This file was then included into this documentation.

BIP.SET - Conversion of Books In Print to MARC Communications Format

This settings file was designed to convert files created by Books In Print to a MARC Communications format, suitable for loading into a library system such as Dynix, Horizon, and others.

There is a batch file included with The Data Magician software called "CONVBIP.SET" which allows you to convert a Books In Print file to MARC very simply. For example, if your input file was: **BIP.TXT**, and you wanted the output file to be: **BIP.MRC**, then you would type:

```
CONVBIP BIP.TXT BIP.MRC
```

Take a look at the batch file to see how it can be applied to other conversions you may want to do.

```
The Data Magician, version 1.5  Release x.x, serial number: 12345
Settings File Name: BIP.SET
```

```
Description:  Convert Books In Print MARC records to standard MARC Communications Format
```

```
Date: 07-18-1997          Time: 23:05:22
```

```
Note:  Square brackets ([...]) around codes are for display purposes only.
      Do NOT type these in when creating settings!
```

INPUT FILE SPECIFICATIONS

```
Input File Name: RUSSIA.MRC
Input File Type: Tagged file
Repeating Field type:      2
Continuation characters: [  ]
Record starts with:       [OCLC:]      and/or contains: []
Record ends with:         [***]        and/or contains: []
Field names begin with:   [53]         and/or end with:  []
Line PRE-Process Codes:
```

```
    [BR1 C"5|" BR4 ! Each line begins with a space - get rid of it.  Every line that
begins with "|" is a MARC field, skip past the line number info.]
```

Global processing codes:

```
    [C"5 " BR 1 S"5??  "??^a" s" b "^b" s" c "^c" s" d "^d" xr c+ ! Clean up indicator
codes and subfield markers (a, b, c & d)]
```

INPUT FIELDS

```
1) OCL          (Leader Line 1)
2) Typ          (Leader Line 2)
3) Rep          (Leader Line 3)
4) Ind          (Leader Line 4)
5) Des          (Leader Line 5)
6) 010         (LCCN)
7) 020         (ISBN)
8) 100         (Personal Author)
9) 245         (Title)
10) 250        (Edition)
11) 260        (Place / Publisher / Date)
12) 300        (Collation)
13) 440        (Series. Often includes 690 info)
14) 44x        (Working field for 440: strip 690)
    [f"440" bl"|???690" ! get everything to the left of a 690]
15) 69x        (690 - Working field, from 440)
    [f"440" bx"|???690*" br"|???690 " ce c~ S"5??  "??^a" s" b "^b" s" c "^c" s" d "^d" xr
tb"5" ! get all the 690 info from the 440 field. Clean up the subfields]
16) 690        (690 with 69x included)
    [f"69x" s"^a"^^ l pa s"^^^a" s" - "^x" s" And " and " s" The " the " s" Of " of "
! Add 69x to end, then change all Upper case to Upper/lower case. Clean up "little" words.]
17) 700        (Additional Authors)
```

Appendix C: BIP.SET - Conversion of Books In Print to MARC Communications Format

OUTPUT FILE SPECIFICATIONS

Output File Name: NEWRUSSIA.MRC
Output File Type: MARC Communications File
Global processing codes: [fc]

OUTPUT FIELDS

1) STATUS ()
[F"OCL" br"Rec stat:" bl"Entrd:" xl xr "n" bl2]
2) RECTYPE ()
[F"Typ" br"e:" bl"Bib lvl:" xl xr " " bl2]
3) BIBLEVEL ()
[F"Typ" br"Bib lvl: " bl2]
4) ENCODE ()
[f"Rep" br"Enc lvl:" bl"Conf pub:" xl xr " " bl2]
5) DCF ()
6) 001 ()
[F"OCL" br"C:" bl"Rec stat:" xl xr]
7) 010 ()
8) 020 ()
9) 100 ()
10) 245 ()
[! Press Ctrl-F4 if the data "cascades" down the screen (because of MARC subfield markers <031>)]
11) 260 ()
12) 300 ()
13) 440 ()
[xc f"44x" ! clear out the 440 field and use 44x instead (no 690s)]
14) 690 (Local Subject Headings)
15) 700 (Author)

CANPROC.SET - Conversion of CANOLE records to Pro-Cite

These settings were created to show how to set up a conversion from an on-line utility such as CAN/OLE to Pro-Cite, a bibliographic reference manager from PBS Inc. The Pro-Cite file must be set up as an ASCII delimited file with 47 output fields. The order of the fields is critical for the file to be imported correctly. After the file is converted via The Data Magician, you must use the IMPORT utility that comes with Pro-Cite to load the records into your Pro-Cite database.

The sample file used here has only a few fields. Typically, you would request the records displayed in a "full" tagged format to get more information. Simply add the additional tags and specify which Pro-Cite fields will receive them.

The trickiest part of this conversion is recognizing the start and end of a record (bypassing all the search commands, etc.). Since the title field (TI) was always first, we used those characters to recognize the start of a record. Each record ends with a blank line so we used the special sequence "****" to signal the end of a record (see page [26](#) for details).

Note the two extra input fields "JOURNAL" and "DATE". These are not part of the input file, but are temporary fields to try to pull apart the "PU" (publication) field. It tries to recognize a date in a couple of forms. Anything to the left of the date is put into the "JOURNAL" field, while the date is put in "DATE". Further work could be attempted to break down the "JOURNAL" field, but you might have trouble with special cases such as conference proceedings, etc.

The Data Magician, version 1.5 Release x.x, serial number: 123456
Settings File Name: CANPROC.SET

Description: Sample CAN/OLE to Pro-Cite Conversion

Date: 07-18-1997 Time: 01:33:31

Note: Square brackets ([...]) around codes are for display purposes only.
Do NOT type these in when creating settings!

INPUT FILE SPECIFICATIONS

```
Input File Name: CANOLE.ASC
Input File Type: Tagged
Repeating Field type: 2
Continuation characters: [ ]
Record starts with: [TI - ] and/or contains: []
Record ends with: [****] and/or contains: [@]
Field names begin with: [] and/or end with: [ - ]
Line PRE-Process Codes:
[ ]
Global processing codes:
[ ]
```

INPUT FIELDS

```
1) TI (Title)
2) AU (Author)
[BS';' 1 pa xl S"5"// " !*** Break into Subfields, make Upper/lower, strip leadings
spaces, and separate multiple authors with // (for ProCite) ***]
3) PU (Publication Information)
4) JOURNAL (Journal Title only)
[F"PU" bl'(19??)' bl'( 19??)' !*** Journal title is everything to left of the date (in
one of two formats, with or without a space) ***]
5) DATE (Publication Date)
[F"PU" BRF"JOURNAL" b'(19??)' b'( 19??)' !*** Break out date alone ***]
6) LOCATION (Location in Work)
[F"PU" BRF"DATE" !*** Everything else! ***]
```

Appendix C: CANPROC.SET - CAN/OLE records to Pro-Cite

OUTPUT FILE SPECIFICATIONS

Output File Name: CANPROC.ASC
Output File Type: ASCII comma delimited
Number of output fields: 47
Global processing codes: []

OUTPUT FIELDS

1) TYPE (Record Type - Journals Short)
['D']
2) RECNO (Record Number)
[ns 1000 nu ni 10 ! *** Number records by 10's ***]
3) Auth (Author (Analytic))
[f"AU"]
4) AuRo (Author Role (Analytic))
5) Affl (Author Affiliation (Analytic))
6) Titl (Title (Analytic))
[f"TI" XP ! *** Get title, strip trailing punctuation and spaces. This has not been
converted to Upper/lower because of acronymns ***]
7) Medm (Medium Designator)
8) CoPh (Connective Phrase)
9) Auth2 (Author (Monographic))
10) AuRo2 (Author Role (Monographic))
11) Titl2 (Title (Monographic) (Journal))
[F"JOURNAL" XR]
12) Ltto (Letter to)
13) Titl3 (Title)
14) DtMt (Date of Meeting)
15) PlMt (Place of Meeting)
16) Medm2 (Medium Designator 2)
17) Edit (Edition)
18) Auth3 (Author (Subsidiary))
19) AuRo3 (Author Role (Subsidiary))
20) PlPu (Place of Publication)
21) Publ (Publisher Name)
22) Date (Date of Publication)
[F"DATE" xb"(" xe ") " ! *** Get date, strip leading/trailing spaces and brackets
***]
23) Copy ()
24) VoID ()
25) RpID ()
26) IsID ()
27) Loc ()
[F"LOCATION" xl xr]
28) Extn ()
29) PaMe ()
30) Size ()
31) RpRt ()
32) SrEd ()
33) SrRo ()
34) SrTi ()
35) SrVo ()
36) SrIs ()
37) DcTy ()
38) CoPh ()
39) Aval ()
40) StLc ()
41) CODN ()
42) ISSN ()
43) ISBN ()
44) Note (Notes)
45) Abst (Abstract)
46) Call (Call Numbers)
47) Indx (Index)

DIALOG.SET - Conversion of DIALOG records to INMAGIC load format

This is sample conversion of a file downloaded from DIALOG using Format 4, into an INMAGIC format. Many of the features of The Data Magician were built to handle this sample. For instance, the ability to substitute with wildcards was necessary to remove anything in angle brackets (eg: "<Analytic>") - see input fields 3 (TI) and 24 (DE). This file also requires a wildcard change (* to !) to remove the * from the Descriptors (see field 24). You will also note the use of the FC code in the Global Processing Codes for the Output File Specifications. This makes it simpler to create the processing codes, since all fields with the same name will automatically be copied over.

The Descriptor field here is heavily processed. First the wildcard is changed from the * to an ! (w*!). Next any *'s are removed (s"***"). Then anything in angle brackets is removed, using the new wildcard (s"<!>"). Then they are broken into separate entries on a semi-colon (BS"; "). Next, they are made completely lowercase (L) and then the first word is capitalized (P1). Finally, any leading spaces are removed (XL).

The Data Magician, version 1.5 Release x.x, serial number: 123456
Settings File Name: DIALOG.SET

Description: Conversion of a downloaded DIALOG file to an INMAGIC format

Date: 07-18-1997 Time: 02:11:00

Note: Square brackets ([...]) around codes are for display purposes only.
Do NOT type these in when creating settings!

INPUT FILE SPECIFICATIONS

Input File Name: DIALOG.ASC
Input File Type: Tagged
Repeating Field type: 2
Continuation characters: []
Record starts with: [FN] and/or contains: []
Record ends with: [] and/or contains: [||]
Field names begin with: [] and/or end with: [-]
Line PRE-Process Codes:
[]
Global processing codes:
[S"|""]

INPUT FIELDS

1) FN	(File Number)
2) AN	(Accession Number)
3) TI	(Title)
[S"<*>" XL]	
4) AU	(Author)
[BS"; " XL]	
5) JN	(Journal)
6) VO	()
7) CO	()
8) CS	()
9) CT	(Conference Title)
10) CL	(Conference Location)
11) CY	(Conference Year)
12) PU	(Publisher)
13) PY	(Publication Year)
14) PG	(Pages)
15) RN	()
16) NT	()
17) CP	(Country of Publication)
18) GL	()
19) LA	(Language)
20) JA	()

Appendix C: DIALOG.SET - Conversion of DIALOG records to INMAGIC

```
21) AV          ( )
22) DT          ( )
23) AB          (Abstract)
24) DE          (Descriptors)
    [w*! s"" s"<!>" BS"; " L PA XL ! *** Convert multi-wildcard to !, strip all
asterisks, remove anything between angle brackets, break into subfields on a semi-colon and
space, convert to lowercase and capitalize each word, strip leading spaces (whew!) ***]
25) SC          ( )
    [BS"; " w*! s"" ]
26) SF          ( )
```

OUTPUT FILE SPECIFICATIONS

```
Output File Name: DIALOG.INM
Output File Type: INMAGIC load file
Structure File name:
Global processing codes: [FC ! *** Copy fields from the input file to fields with the same
name in the output file ***]
```

OUTPUT FIELDS

```
1) ID          ( )
    [NS 1 NI 1 NU]
2) AU          (AUTHOR)
3) TI          (TITLE)
4) JN          (JOURNAL)
5) PUB         (Publisher)
    [F"PU" BL', ' ]
6) PL          (PLACE)
    [F"PU" BR', ' ]
7) DATE        ( )
    [F"PY" ]
8) DE          (DESC)
9) AB          (ABST)
10) LA         (LANG)
11) NTS        (NOTES)
    [F"NT" ]
```

DOBISINM.SET - Conversion of DOBIS records to INMAGIC

This is a sample conversion of records from DOBIS - The National Library of Canada's cataloguing service, into INMAGIC. Before processing by The Data Magician, however, these records must be pre-processed to convert some of the internal codes to ASCII codes. When you download from DOBIS, the data uses special codes to indicate foreign language characters. For example, "e" is used to represent "è". There is a program available from the National Library called "DOWN.EXE" which will convert the characters in the file to ASCII codes. After that, there are still two codes that need to be converted. They are the subfield delimiters (ASCII Code 31) which need to be converted to dollar signs (\$), and the end of field (ASCII code 30) symbol which needs to be eliminated. This can be done by The Data Magician in the Global Processing Codes. Note how the special ASCII codes are entered as numbers between angle brackets. The end of record (ASCII code 29), must be used to recognize where each record ends.

The Data Magician, version 1.5 Release x.x, serial number: 123456
Settings File Name: DOBISINM.SET

Description: Conversion of DOBIS MARC records to INMAGIC's CATALOG structure

Date: 07-18-1997 Time: 02:20:04

Note: Square brackets ([...]) around codes are for display purposes only.
Do NOT type these in when creating settings!

INPUT FILE SPECIFICATIONS

Input File Name: DOBIS.ASC
Input File Type: Tagged file
Repeating Field type: 2
Continuation characters: []
Record starts with: [LDR] and/or contains: []
Record ends with: [***] and/or contains: []
Field names begin with: [53] and/or end with: []
Line PRE-Process Codes:

[s"<029>" s"<030>" s"<031>"\$! *** Strip end-of-record characters (029),
end-of-field characters (030), and convert subfield delimiters (031) to dollar signs ***]

Global processing codes:
[]

INPUT FIELDS

- | | |
|---------|------------------------------------|
| 1) LDR | (Leader Information) |
| [ce qr] | |
| 2) 001 | (DOBIS Document Number) |
| 3) 007 | (Physical Descriptors) |
| 4) 008 | (Fixed Fields) |
| 5) 009 | (Phys. Desc. for Archive Mat.) |
| 6) 010 | (LC Card Number) |
| 7) 011 | (Alternate LCCN -Utlas convention) |
| 8) 013 | (NLM Catalogue Citation Number) |
| 9) 014 | (DOBIS Document Number) |
| 10) 015 | (National Bibliography Number) |
| 11) 016 | (Nat. Lib. Can. Record Ctrl No.) |
| 12) 017 | (Copyright) |
| 13) 019 | (OCLC Control Number) |
| 14) 020 | (ISBN) |
| 15) 021 | (Related ISBN (UTLAS convention)) |
| 16) 022 | (ISSN) |
| 17) 027 | (Standard Tech. Rept. Number) |
| 18) 030 | (CODEN) |
| 19) 035 | (Local Number (Accession/Barcode)) |
| 20) 036 | (Local System Control Number) |
| 21) 040 | (Cataloguing Source) |

Appendix C: DOBISINM.SET - Conversion of DOBIS records to INMAGIC

22)	041	(Language Code)
23)	042	(Authentication Centre)
24)	043	(Geographic Area Code)
25)	045	(Chronological Code)
26)	046	(Record Source)
27)	050	(LC Call Number)
28)	055	(NLC Call Number (Nat. Lib. Can.))
29)	059	()
30)	060	(NLM Call Number)
31)	082	(Dewey Number)
32)	086	(Gov. Doc. Number)
33)	090	(Local Call Number)
34)	092	(Card Generation Control)
36)	100	(Main Entry - Personal Name)
37)	110	(Main Entry - Corporate Name)
38)	111	(Main Entry - Conference/Meeting)
39)	130	(Main Entry - Uniform Title)
40)	210	(Series Title (abbreviated))
41)	222	(Key Title)
42)	240	(Uniform Title)
43)	243	(Collective Title)
44)	245	(Title Statement)
45)	246	(Variant Title)
46)	250	(Edition)
47)	260	(Imprint (Place/Publisher/Date))
48)	263	(Projected Publication Date)
49)	265	(Source for Acquisition/Subscript.)
50)	300	(Collation)
51)	310	(Frequency)
52)	350	(Price)
53)	362	(Dates & Volumes (Serials))
54)	400	(Series - Personal Name)
55)	410	(Series - Corporate Name)
56)	440	(Series - Title)
57)	490	(Series - Variant Title)
58)	500	(General Note)
59)	503	(Bibliographic History Note)
60)	504	(Bibliography Note)
61)	505	(Contents Note)
62)	510	(Citation Note)
63)	515	()
64)	520	(Abstract/Summary)
65)	525	(Supplement Note (Serials))
66)	535	(Local Use Note)
67)	546	(Language Note (Serials))
68)	550	(Issuing Body Note (Serials))
69)	570	(Editor Note (Serials))
70)	580	(Linking Entry Note)
71)	600	(Subject - Personal Name)
72)	610	(Subject - Corporate Name)
73)	611	(Subject - Conference/Meeting)
74)	630	(Subject - Uniform Title)
75)	650	(Subject - Topical)
76)	651	(Subject - Geographic)
77)	655	(Subject - Genre)
79)	700	(Added Entry - Personal Name)
80)	710	(Added Entry - Corporate Name)
81)	711	(Added Entry - Conference/Meeting)
82)	720	()
83)	730	(Added Entry - Uniform Title)
84)	740	(Added Entry - Title traced diff.)
85)	770	()
88)	780	(Added Entry - Preceding Entry)
89)	785	(Added Entry - Succeeding Entry)
90)	800	(Series Add. Ent. - Personal Name)
91)	810	(Series Add. Ent. - Corp. Name)
92)	811	(Series Add. Ent. - Conf./Meeting)
93)	830	(Series Add. Ent. - Uniform Title)

```
94) 840          (Series Add. Ent. - Title)
95) 850          (DOBIS System holdings)
96) 0##          ( )
97) @@@          (Junk data?)
98) <029>        ( )
```

OUTPUT FILE SPECIFICATIONS

```
Output File Name: DOBIS.INM
Output File Type: INMAGIC or DB/TextWorks load file
Structure File name: \INMAGIC\CATALOG.STR
Global processing codes: []
```

OUTPUT FIELDS

```
1) ID          ( )
   [ns 1 nu ni 1]
2) CL          (CLASS)
   [f"0905$*" CE F"055" CE F"050" C+ s/' ' BC ! *** Get Call number from: Local, NLC,
LC (in that order), then just take subfield a, convert slashes to spaces, and finally insert
spaces at the correct points ***]
3) TI          (TITLE)
   [f"2455$a" xp]
4) STI         (SUBTITLE)
   [f"2455$b" ]
5) AU          (AUTHOR)
   [f"1005$*" xe", " ]
6) CORP        ( )
   [f"1105$*" ]
7) ED          (EDITION)
   [f"2505$a" ]
8) PL          (PLACE)
   [f"2605$a" xe ": " ]
9) PUB         ( )
   [f"2605$b" xe", " ]
10) DT         (DATE)
   [f"2605$c" xe ". " ]
11) PD         (PHYSICAL)
   [f"3005$*" ]
12) SER        (SERIES)
   [f'400' sf f'410' sf f'440' sf f'490' bm"$*" xs]
13) DE         (DESC)
   [f'650' sf f'610' sf f'611' sf f'630' sf f'651' sf f'655' sm" -- " BM '$*' xs]
14) AB         (ABST)
   [f"520" ]
15) NTS        (NOTES)
   [f"500" sf f"503" sf f"504" sf f"505" sf f"510" sf f"525" sf f"535" sf f"546" sf f"550"
sf f"570" sf f"580" BM '$*' XS]
16) AE         (ADDED)
   [f"700" sf f"710" sf f"740" sf f"780" sf f"785" sf f"711" sf f"730" sf f"800" sf f"810"
sf f"811" sf f"830" sf f"840" BM '$*' XS]
17) BN         (ISBN)
   [f"0205$a" xe ": " ]
18) SN         (ISSN)
   [f"0225$a" ]
19) LC         ( )
   [f"0105$a" ]
20) ORD        (ORDER)
21) LOCN       ( )
   [f"0905$a" ]
22) HO         (HOLDINGS)
   [f"3625$a" ]
23) CPS        (COPIES)
24) LBL        (LABEL)
25) PRE        (PRELIM)
   [DT DX ! Get today's date and convert to text format]
26) OK         ( )
27) SEE        ( )
28) SEE_ALSO   ( )
```

DOBISMRC.SET - Conversion of DOBIS (MARC) records to MARC

This is a sample conversion of records from DOBIS - The National Library of Canada's cataloguing service, into an actual MARC Communication file. Once in that format, it should be able to be loaded into most library automation systems. In this sample, the fields are moved one-for-one from the input to the output, retaining all indicator codes and subfield delimiters.

Before processing by The Data Magician, however, these records must be pre-processed to convert some of the internal codes to ASCII codes. When you download from DOBIS, the data uses special codes to indicate foreign language characters. For example, "`e" is used to represent "è". There is a program available from the National Library called "DOWN.EXE" which will convert the characters in the file to ASCII codes. After that, the end of field (ASCII code 30) symbol needs to be eliminated. This can be done by The Data Magician in the Global Processing Codes. Note how the special ASCII codes are entered as numbers between angle brackets. The end of record (ASCII code 29), must be used to recognize where each record ends.

This conversion requires that you start The Data Magician using the /F option which will add field names to both the Input and Output Fields as they are encountered. This way, all MARC tags will be converted. It also makes the conversion settings file very short since you do not need to define all the tags ahead of time.

The Data Magician, version 1.5 Release x.x, serial number: 123456
Settings File Name: DOBISMRC.SET

Description: Conversion of DOBIS downloaded records to MARC Communication format. ***
Note: This conversion must be run with the /F option to add new field tags as encountered ***

Date: 07-18-1997 Time: 15:49:03

Note: Square brackets ([...]) around codes are for display purposes only.
Do NOT type these in when creating settings!

INPUT FILE SPECIFICATIONS

Input File Name: DOBIS.ASC
Input File Type: Tagged file
Repeating Field type: 2
Continuation characters: []
Record starts with: [LDR] and/or contains: []
Record ends with: [<029>] and/or contains: []
Field names begin with: [53] and/or end with: []
Line PRE-Process Codes:
[s"<030>"]
Global processing codes:
[]

INPUT FIELDS

1) LDR (Leader Information)
[ce qr ! *** Skip record if this field empty ***]
2) 001 (DOBIS Document Number)
3) @@@ (This field will be ignored)
4) 0## (This field will be ignored)

OUTPUT FILE SPECIFICATIONS

Output File Name: DOBIS.MRC
Output File Type: MARC Communications File
Global processing codes: [fc ! *** Just copy fields with same name (number) ***]

OUTPUT FIELDS

1) STATUS (Record Status)
[f"LDR5[6]" ! *** Get Record Status from 6th position of LDR***]
2) RECTYPE (Record Type)

```
[f"LDR5[7]" ! *** Get Record Type from 7th position ***]
3) BIBLEVEL      (Bibliographic Level)
[f"LDR5[8]" ! *** Get Bib. Level from 8th position ***]
4) ENCODE       (Encoding Level)
[f"LDR5[9]" ! *** Get Enc. Level from 9th position ***]
5) DCF          (Descriptive Cataloguing Form)
[f"LDR5[10]" ! *** Get DCF from 10th position ***]
6) 001          (DOBIS Document Number)
[! *** All other fields are copied directly from the input file]
```

EBSCO.SET - EBSCO Serials (MARC communications format) to INMAGIC

These settings will allow INMAGIC users to request MARC records from EBSCO or CANEBSCO and convert them to an INMAGIC database using the Biblio Guide SERMAN structure. While many of the fields would make sense for most users, there may be some changes necessary for any particular application. These settings should also work for serials records in MARC Communications format from other vendors, such as FAXON.

Some character translations may need to be added to the Global Processing Codes for the Input File, if you have a lot of French materials.

The Data Magician, version 1.5 Release x.x, serial number: 123456
Settings File Name: EBSCO.SET

Description: Conversion of EBSCO serial records to INMAGIC's SERMAN structure

Date: 07-18-1997 Time: 02:37:08

Note: Square brackets ([...]) around codes are for display purposes only.
Do NOT type these in when creating settings!

INPUT FILE SPECIFICATIONS

Input File Name: EBSCO.MRC

Input File Type: MARC Communications File

Global processing codes:

[ce c~ s"i"e"é" ! *** If current field not empty, convert character(s) from ALA standard to PC format ***]

INPUT FIELDS

1) STATUS	()
2) RECTYPE	()
3) BIBLEVEL	()
4) ENCODE	()
5) DCF	()
6) 001	(RSN)
7) 005	()
8) 008	(Fixed Fields)
9) 010	(LCCN)
10) 012	()
11) 015	()
12) 019	()
13) 022	(ISSN)
14) 030	(CODEN)
15) 032	()
16) 035	(Local Accession Number)
17) 040	(Cataloguing Source)
18) 041	(Language Codes)
19) 049	()
20) 042	()
21) 043	(Geographic Code)
22) 050	(LC Call Number)
23) 051	()
24) 055	()
25) 060	(NLM Call Number)
26) 070	()
27) 072	()
28) 074	()
29) 082	(Dewey)
30) 086	()
31) 088	()
32) 110	()
33) 130	()
34) 210	(Series Title (Abbreviated))
35) 212	()
36) 222	(Key Title)

- 37) 245 (Title Statement)
- 38) 246 (Variant Title)
- 39) 247 ()
- 40) 250 (Edition)
- 41) 260 (Imprint)
- 42) 265 ()
- 43) 300 (Collation)
- 44) 310 (Frequency)
- 45) 321 ()
- 46) 350 (Price)
- 47) 362 (Dates & Volumes)
- 48) 440 (Series - Title)
- 49) 490 (Series - Variant Title)
- 50) 500 (General Note)
- 51) 504 (Bibliography Note)
- 52) 506 ()
- 53) 510 (Citation Note)
- 54) 515 ()
- 55) 520 ()
- 56) 525 ()
- 57) 530 (Source Note)
- 58) 546 (Language Note)
- 59) 550 (Issuing Body Note)
- 60) 555 ()
- 61) 570 (Editor Note)
- 62) 580 (Linking Entry Note)
- 63) 590 (Local Notes)
- 64) 610 (Subject - Corporate Name)

[ce c~ br"\$a" s"\$?" -- " ! *** If current field not empty, take everything after \$a,
and put hyphens between each subfield ***]

- 65) 650 (Subject - Topical)
[ce c~ br"\$a" s"\$?" -- "]
- 66) 651 (Subject - Geographic)
[ce c~ br"\$a" s"\$?" -- "]
- 67) 710 (Added Entry - Corporate Name)
[ce c~ br"\$a" s"\$?" -- "]
- 68) 700 (Added Entry - Personal Name)
- 69) 730 (Added Entry - Uniform Title)
- 70) 752 ()
- 71) 760 ()
- 72) 767 ()
- 73) 770 ()
- 74) 772 ()
- 75) 775 ()
- 76) 776 ()
- 77) 777 ()
- 78) 780 (Added Entry - Preceding Entry)
- 79) 785 (Added Entry - Succeeding Entry)
- 80) 787 ()
- 81) 830 ()
- 82) 850 ()
- 83) 871 ()
- 84) 902 (Supplier Info)
- 85) 936 ()

OUTPUT FILE SPECIFICATIONS

Output File Name: EBSCO.INM
Output File Type: INMAGIC load file
Structure File name: C:\INMAGIC\SERMAN.STR
Global processing codes: []

OUTPUT FIELDS

- 1) ID ()
[NS 1 NU NI 1]
- 2) SUPNO ()
[F"902" BM"\$a"]
- 3) TI (TITLE)

```

    [f"245" bm"$a" xp]
4) STI (SUBTITLE)
    [f"245" bm"$b"]
5) ED (EDITION)
    [f"250" bm"$a"]
6) DE (DESC)
    [f'650' sf f'610' sf f'651' xs]
7) LOCN ( )
8) ORD (ORDER)
9) ODT (ORDERDT)
10) PO ( )
11) ACCT ( )
12) DP (DEPT)
13) PRICE ( )
    [f"350" bm"$a"]
14) COST ( )
15) TP (TYPE)
16) IN (INDEX)
    [f"510" bm'$a' ]
17) FREQ ( )
    [f"310" bm "$a" xe ", "]
18) START ( )
19) EXP (EXPIRES)
20) REN (RENEW)
21) HO (HOLDINGS)
    [f"049" sf f'590']
22) SPEC (SPECIAL)
23) RTN (RETAIN)
24) LOAN ( )
25) SUP (SUPPLIER)
    [f"902" bm'$c' ]
26) SADD (SUPADD)
27) STEL (SUPTEL)
28) SAN ( )
29) NAME ( )
30) NTS (NOTES)
31) Q1 ( )
32) Q2 ( )
33) Q3 ( )
34) Q4 ( )
35) SN (ISSN)
    [f"022" bm'$a' ]
36) AU (AUTHOR)
37) CORP ( )
    [f"110" bm'$a' ]
38) CL (CLASS)
    [f"050" br"$a" s"$?" " bc]
39) PL (PLACE)
    [f"260" bm'$a' xe ", "]
40) PUB ( )
    [f"260" bm'$b' ]
41) DT (DATE)
    [f"362" bm "$a"]
42) SER (SERIES)
43) BIND (BINDERY)
    [ ]
44) PRE (PRELIM)
    [DT DX]
45) OK ( )
46) CPS (COPIES)
47) NUM (NUMBER)

```

GEACINM.SET - Conversion of GEAC records to INMAGIC

These settings were created to show how to set up a conversion of GEAC screen captures to INMAGIC. Once the conversion is performed via The Data Magician, the records must be added to your INMAGIC database using the Add option in the Maintain menu.

Although GEAC records are typically displayed with a brief or full "tagged" display (AU for author, TI for title), the MARC display was chosen for this conversion because of the completeness of the information. The settings could, however, be modified to accommodate the other display formats. All the fields used in the UTLAS conversion were included, even though many of them do not appear in the sample file. Some may occur in any files you may wish to convert. Many of the fields are not actually moved to an output field. If you find fields that are not recognized in the conversion process, simply add them to the input list, and include them somewhere in the output fields. You are likely to find "unusual" fields in special records such as serials, audio-visual materials, government documents, etc.

The Data Magician, version 1.5 Release x.x, serial number: 123456
Settings File Name: GEACINM.SET

Description: Sample CAN/OLE to INMAGIC Conversion

Date: 07-18-1997 Time: 23:16:27

Note: Square brackets ([...]) around codes are for display purposes only.
Do NOT type these in when creating settings!

INPUT FILE SPECIFICATIONS

Input File Name: GEACMARC.ASC
Input File Type: Tagged
Repeating Field type: 2
Continuation characters: []
Record starts with: [008] and/or contains: []
Record ends with: [002] and/or contains: []
Field names begin with: [53] and/or end with: []
Line PRE-Process Codes: []
Global processing codes: []

INPUT FIELDS

- 1) 001 (Reserve Sequence Number)
- 2) 002 (Control Number)
[ce qr ! *** Skip this "record" if there is no control number ***]
- 3) 004 (Amendments)
- 4) 007 (Physical Descriptors)
- 5) 008 (Fixed Fields)
- 6) 009 (Phys. Desc. for Archive Mat.)
- 7) 010 (LC Card Number)
- 8) 011 (Alternate LCCN -Utlas convention)
- 9) 013 (NLM Catalogue Citation Number)
- 10) 015 (National Bibliography Number)
- 11) 016 (Nat. Lib. Can. Record Ctrl No.)
- 12) 017 (Copyright)
- 13) 019 (OCLC Control Number)
- 14) 020 (ISBN)
- 15) 021 (Related ISBN (UTLAS convention))
- 16) 022 (ISSN)
- 17) 027 (Standard Tech. Rept. Number)
- 18) 030 (CODEN)
- 19) 035 (Local Number (Accession/Barcode))
- 20) 036 (Local System Control Number)
- 21) 040 (Cataloguing Source)
- 22) 041 (Language Code)
- 23) 042 (Authentication Centre)

Appendix C: GEACINM.SET - Conversion of GEAC records to INMAGIC

24) 043 (Geographic Area Code)
25) 045 (Chronological Code)
26) 046 (Record Source)
27) 050 (LC Call Number)
28) 055 (NLC Call Number (Nat. Lib. Can.))
29) 060 (NLM Call Number)
30) 082 (Dewey Number)
31) 086 (Gov. Doc. Number)
32) 088 (Document Shelving Number (CODOC))
33) 090 (Local Call Number)
34) 092 (Card Generation Control)
35) 093 (
36) 099 (Alternate Call Number)
37) 100 (Main Entry - Personal Name)
38) 110 (Main Entry - Corporate Name)
[s'\$b' ']
39) 111 (Main Entry - Conference/Meeting)
40) 130 (Main Entry - Uniform Title)
41) 210 (Series Title (abbreviated))
42) 222 (Key Title)
43) 240 (Uniform Title)
44) 243 (Collective Title)
45) 245 (Title Statement)
46) 246 (Variant Title)
47) 250 (Edition)
48) 260 (Imprint (Place/Publisher/Date))
49) 263 (Projected Publication Date)
50) 265 (Source for Acquisition/Subscript.)
51) 300 (Collation)
[s'\$b' ' s'\$c' ']
52) 310 (Frequency)
53) 350 (Price)
54) 362 (Dates & Volumes (Serials))
55) 400 (Series - Personal Name)
56) 410 (Series - Corporate Name)
[s'\$b' ']
57) 440 (Series - Title)
58) 490 (Series - Variant Title)
59) 500 (General Note)
60) 503 (Bibliographic History Note)
61) 504 (Bibliography Note)
62) 505 (Contents Note)
63) 510 (Citation Note)
64) 520 (Abstract/Summary)
65) 525 (Supplement Note (Serials))
66) 535 (Local Use Note)
67) 546 (Language Note (Serials))
68) 550 (Issuing Body Note (Serials))
69) 570 (Editor Note (Serials))
70) 580 (Linking Entry Note)
71) 600 (Subject - Personal Name)
72) 610 (Subject - Corporate Name)
[s'\$b' ']
73) 611 (Subject - Conference/Meeting)
74) 630 (Subject - Uniform Title)
75) 650 (Subject - Topical)
[s'\$b' -- ' s'\$x' -- ' s'\$y' -- ' s'\$z' -- ']
76) 651 (Subject - Geographic)
[s'\$b' -- ' s'\$x' -- ' s'\$y' -- ' s'\$z' -- ']
77) 655 (Subject - Genre)
[s'\$b' -- ' s'\$x' -- ' s'\$y' -- ' s'\$z' -- ']
78) 700 (Added Entry - Personal Name)
79) 710 (Added Entry - Corporate Name)
[s'\$b' ']
80) 711 (Added Entry - Conference/Meeting)
81) 730 (Added Entry - Uniform Title)
82) 740 (Added Entry - Title traced diff.)
83) 780 (Added Entry - Preceding Entry)

```
84) 785          (Added Entry - Succeeding Entry)
85) 800          (Series Add. Ent. - Personal Name)
86) 810          (Series Add. Ent. - Corp. Name)
87) 811          (Series Add. Ent. - Conf./Meeting)
88) 830          (Series Add. Ent. - Uniform Title)
89) 840          (Series Add. Ent. - Title)
90) 966          (Circulation Holdings)
91) BRF          (Junk)
    [! *** The following "fields" come from the search text between (and within) records
***]
92) IND          (Junk)
93) CAT          (Junk)
94) Ent          (Junk)
95) con          (Junk)
96) 074          (Junk)
```

OUTPUT FILE SPECIFICATIONS

```
Output File Name: GEACMARC.INM
Output File Type: INMAGIC load file
Structure File name:
Global processing codes: []
```

OUTPUT FIELDS

```
1) ID          ( )
   [ns 1 nu ni 1]
2) CL          (CLASS)
   [f"090" ce f"099" c+ BM '$a' S"/" " xr ! *** Take field 090, if empty get 099. In
either case just get $a, substitute slashes with spaces, and strip trailing spaces. ***]
3) TI          (TITLE)
   [f"245" BM '$a' XP ! *** Get title, strip trailing punctuation ***]
4) STI         (SUBTITLE)
   [f"245" BM'$b']
5) AU          (AUTHOR)
   [f"100" BM '$a' XE", " ! *** Get Author main entry ($a), and strip trailing commas
***]
6) CORP        ( )
   [f"110" BM '$a' xr]
7) ED          (EDITION)
   [f"250" BM '$a' xe"- " ! *** Get Edition statement, and strip trailing spaces and
hyphens ***]
8) PL          (PLACE)
   [f"260" BM '$a' xe ": " ! *** Get Place of Publication (260$a) and strip trailing
colon ***]
9) PUB         ( )
   [f"260" BM '$b' xe ", "]
10) DT         (DATE)
   [f"260" BM'$c' xr]
11) PD         (PHYSICAL)
   [f"300" BM '$a' xe"- "]
12) SER        (SERIES)
   [f'400' sf f'410' sf f'440' sf f'490' xs bm'$a' ! *** Get series from multiple
sources, delete excess subfield markers (not required, but easier to look at), extract subfield
'a' (only) from each field ***]
13) DE         (DESC)
   [f'650' sf f'610' sf f'611' sf f'630' sf f'651' sf f'655' xs bm'$a' xr ! See Series
notes]
14) AB         (ABST)
   [f"520"]
15) NTS        (NOTES)
   [f"500" sf f"503" sf f"504" sf f"505" sf f"510" sf f"525" sf f"535" sf f"546" sf f"550"
sf f"570" sf f"580" xs bm'$a' xr ! See Series notes]
16) AE         (ADDED)
   [f"700" sf f"710" sf f"740" sf f"780" sf f"785" sf f"711" sf f"730" sf f"800" sf f"810"
sf f"811" sf f"830" sf f"840" xs bm'$a' xr ! See Series notes]
17) BN         (ISBN)
   [f"020" BM'$a' xe ": "]
18) SN         (ISSN)
```

Appendix C: GEACINM.SET - Conversion of GEAC records to INMAGIC

```
[f"022" BM'$a']
19) LC          ()
[f"010" BM'$a']
20) ORD          (ORDER)
21) LOCN         ()
[f"090" BM'$b']
22) HO          (HOLDINGS)
[f"362" BM'$a']
23) CPS          (COPIES)
24) LBL          (LABEL)
25) PRE          (PRELIM)
[dt dx]
26) OK          ()
27) SEE         ()
28) SEE_ALSO   ()
```

GEACPROC.SET - Conversion of GEAC records to Pro-Cite

These settings were created to show how to set up a conversion of GEAC screen captures to Pro-Cite, a bibliographic reference manager from PBS Inc. The Pro-Cite file must be set up as an ASCII delimited file with 47 output fields. The order of the fields is critical for the file to be imported correctly. After the file is converted via The Data Magician, you must use the IMPORT utility that comes with Pro-Cite to load the records into your Pro-Cite database.

Although GEAC records are typically displayed with a brief or full "tagged" display (AU for author, TI for title), the MARC display was chosen for this conversion because of the completeness of the information. The settings could, however, be modified to accommodate the other display formats. All the fields used in the UTLAS conversion were included, even though many of them do not appear in the sample file. Some may occur in any files you may wish to convert. Many of the fields are not actually moved to an output field. If you find fields that are not recognized in the conversion process, simply add them to the input list, and include them somewhere in the output fields. You are likely to find "unusual" fields in special records such as serials, audio-visual materials, government documents, etc.

The Data Magician, version 1.5 Release x.x, serial number: 123456
Settings File Name: GEACPROC.SET

Description: Conversion of on-screen Geac captures, and convert them to a format that can be loaded by Pro-Cite

Date: 07-18-1997 Time: 23:10:22

Note: Square brackets ([...]) around codes are for display purposes only.
Do NOT type these in when creating settings!

INPUT FILE SPECIFICATIONS

Input File Name: GEACMARC.ASC
Input File Type: Tagged
Repeating Field type: 2
Continuation characters: []
Record starts with: [008] and/or contains: []
Record ends with: [002] and/or contains: []
Field names begin with: [] and/or end with: []
Line PRE-Process Codes:
[]
Global processing codes:
[]

INPUT FIELDS

1) 001 (Reserve Sequence Number)
2) 002 (Control Number)
3) 004 (Amendments)
4) 007 (Physical Descriptors)
5) 008 (Fixed Fields)
6) 009 (Phys. Desc. for Archive Mat.)
7) 010 (LC Card Number)
8) 011 (Alternate LCCN -Utlas convention)
9) 013 (NLM Catalogue Citation Number)
10) 015 (National Bibliography Number)
11) 016 (Nat. Lib. Can. Record Ctrl No.)
12) 017 (Copyright)
13) 019 (OCLC Control Number)
14) 020 (ISBN)
15) 021 (Related ISBN (UTLAS convention))
16) 022 (ISSN)
17) 027 (Standard Tech. Rept. Number)
18) 030 (CODEN)

Appendix C: GEACPROC.SET - Conversion of GEAC records to Pro-Cite

19) 035 (Local Number (Accession/Barcode))
20) 036 (Local System Control Number)
21) 040 (Cataloguing Source)
22) 041 (Language Code)
23) 042 (Authentication Centre)
24) 043 (Geographic Area Code)
25) 045 (Chronological Code)
26) 046 (Record Source)
27) 050 (LC Call Number)
28) 055 (NLC Call Number (Nat. Lib. Can.))
29) 060 (NLM Call Number)
30) 082 (Dewey Number)
31) 086 (Gov. Doc. Number)
32) 088 (Document Shelving Number (CODOC))
33) 090 (Local Call Number)
34) 092 (Card Generation Control)
35) 093 (
36) 099 (Alternate Call Number)
37) 100 (Main Entry - Personal Name)
38) 110 (Main Entry - Corporate Name)
[s'\$b' ']
39) 111 (Main Entry - Conference/Meeting)
40) 130 (Main Entry - Uniform Title)
41) 210 (Series Title (abbreviated))
42) 222 (Key Title)
43) 240 (Uniform Title)
44) 243 (Collective Title)
45) 245 (Title Statement)
[s'\$b' ']
46) 246 (Variant Title)
47) 250 (Edition)
48) 260 (Imprint (Place/Publisher/Date))
49) 263 (Projected Publication Date)
50) 265 (Source for Acquisition/Subscript.)
51) 300 (Collation)
[s'\$b' ' s'\$c' ' ! *** Put the whole thing together as one field ***]
52) 310 (Frequency)
53) 350 (Price)
54) 362 (Dates & Volumes (Serials))
55) 400 (Series - Personal Name)
56) 410 (Series - Corporate Name)
[s'\$b' ']
57) 440 (Series - Title)
58) 490 (Series - Variant Title)
59) 500 (General Note)
60) 503 (Bibliographic History Note)
61) 504 (Bibliography Note)
62) 505 (Contents Note)
63) 510 (Citation Note)
64) 520 (Abstract/Summary)
65) 525 (Supplement Note (Serials))
66) 535 (Local Use Note)
67) 546 (Language Note (Serials))
68) 550 (Issuing Body Note (Serials))
69) 570 (Editor Note (Serials))
70) 580 (Linking Entry Note)
71) 600 (Subject - Personal Name)
72) 610 (Subject - Corporate Name)
[s'\$b' ']
73) 611 (Subject - Conference/Meeting)
74) 630 (Subject - Uniform Title)
75) 650 (Subject - Topical)
[s'\$b' -- ' s'\$x' -- ' s'\$y' -- ' s'\$z' -- ' ! *** Substitute subfield delimiters with
dashes ***]
76) 651 (Subject - Geographic)
[s'\$b' -- ' s'\$x' -- ' s'\$y' -- ' s'\$z' -- ']
77) 655 (Subject - Genre)
[s'\$b' -- ' s'\$x' -- ' s'\$y' -- ' s'\$z' -- ']

78) 700 (Added Entry - Personal Name)
79) 710 (Added Entry - Corporate Name)
[s'b' ']
80) 711 (Added Entry - Conference/Meeting)
81) 730 (Added Entry - Uniform Title)
82) 740 (Added Entry - Title traced diff.)
83) 780 (Added Entry - Preceding Entry)
84) 785 (Added Entry - Succeeding Entry)
85) 800 (Series Add. Ent. - Personal Name)
86) 810 (Series Add. Ent. - Corp. Name)
87) 811 (Series Add. Ent. - Conf./Meeting)
88) 830 (Series Add. Ent. - Uniform Title)
89) 840 (Series Add. Ent. - Title)
90) 966 (Circulation Holdings)
91) BRF (Junk)
92) IND (Junk)
93) CAT (Junk)
94) ENTER (Junk)
95) 074 (Junk)

OUTPUT FILE SPECIFICATIONS

Output File Name: GEACPROC.ASC
Output File Type: ASCII comma delimited
Number of output fields: 47
Global processing codes: []

OUTPUT FIELDS

1) TYPE (Record Type - Books Long)
['A']
2) RECNO (Record Number)
[ns 100 nu ni 10]
3) Auth (Author (Analytic))
4) AuRo (Author Role (Analytic))
5) Affl (Author Affiliation (Analytic))
6) Titl (Title (Analytic))
7) Medm (Medium Designator)
8) CoPh (Connective Phrase)
9) Auth2 (Author (Monographic))
[f"100" sf f"700" sf F"110" xs bm'\$a' xe", " s"5"/" ! *** Take Author Main Entry,
Personal Name Added Entry, and Corporate Main Entry (add others if you wish) ***]
10) AuRo2 (Author Role (Monographic))
11) Titl2 (Title (Monographic))
[F"245" BM'\$a' xp]
12) Ltto (Letter to)
13) Titl3 (Title)
14) DtMt (Date of Meeting)
15) PlMt (Place of Meeting)
16) Medm2 (Medium Designator 2)
17) Edit (Edition)
[F"250" BM'\$a' xe"- "
18) Auth3 (Author (Subsidiary))
[F"710" BM'\$a']
19) AuRo3 (Author Role (Subsidiary))
20) PlPu (Place of Publication)
[F"260" BM'\$a' xe": "
21) Publ (Publisher Name)
[F"260" BM'\$b' xe", "
22) Date (Date of Publication)
[f"260" BM'\$c' xr]
23) Copy (Date of Copyright)
24) VoID (Volume ID)
25) RpID (Report ID)
26) IsID (Issue ID)
27) Loc (Location in Work)
28) Extn (Extent of Work)
[F"300" BM'\$a' xe"- "
29) PaMe (Packaging Method)

Appendix C: GEACPROC.SET - Conversion of GEAC records to Pro-Cite

30) Size ()
31) RpRt ()
32) SrEd (Series Editor)
33) SrRo (Series Role)
34) SrTi (Series Title)
 [F"440" bm'\$a' xr]
35) SrVo (Series - Volume)
36) SrIs (Series - Issue)
37) DcTy ()
38) CoPh ()
39) Aval ()
40) StLc ()
41) CODN (CODEN)
42) ISSN ()
 [F"022" BM'\$a']
43) ISBN ()
 [F"020" BM'\$a' xe': ']
44) Note (Notes)
 [F"500" bm'\$a']
45) Abst (Abstract)
 [F"520" bm'\$a']
46) Call (Call Numbers)
 [F"099" BM'\$a' xr]
47) Indx (Index)
 [F'650' SF F'610' SF F'611' SF F'630' SF F'651' SF F'655' xs bm'\$a' xr s'5'/' ! ***
Take various subject fields, in the indicated order, and separate them with slashes ***]

INM2MARC.SET - Conversion of INMAGIC records to MARC

These settings were created to show how MARC records could be created from an INMAGIC database. The INMAGIC structure used here is the CATALOG structure from The Biblio Guide: Using INMAGIC for Libraries. After converting the records with The Data Magician, you would need to use a MARC loader for the system you wanted to load the records into.

The MARC file is only as good as the data going into it. We have include some choices for the MARC indicator codes, although you may wish to adjust some of them. Note especially that we have tried to analyze leading articles in the Title field to use the correct indicator code. Subjects, series and added entries have been put into the local use or undefined categories since they may be a mixed group. You can easily change what MARC fields should be used by changing the number. For example: we put Descriptors into 690 - Subjects, Undefined. If your system will ignore those, you may wish them put in as 650 - Topical Subjects.

The Data Magician, version 1.5 Release x.x, serial number: 123456
Settings File Name: INM2MARC.SET

Description: Conversion of INMAGIC records in the CATALOG Structure of The BIBLIO Guide to MARC Communications Format
Date: 07-18-1997 Time: 23:31:17

Note: Square brackets ([...]) around codes are for display purposes only.
Do NOT type these in when creating settings!

INPUT FILE SPECIFICATIONS

Input File Name: CATSAMP.ASC
Input File Type: INMAGIC load file
Structure File name: CATALOG.STR
Global processing codes: []

INPUT FIELDS

- 1) ID ()
- 2) CL (CLASS)
[tb'0 \$a' s" ."\$b." ! *** Add indicator code and separate call number into subfields
***]
- 3) TI (TITLE)
[c"The " o"Les " tb'14\$a' c"An " o"Le " o"La " tb"13\$a" c"A " o"L'" tb"12\$a" c"\$a"
c"\$a" c~ tb"10\$a" ! *** Create appropriate filing indicator codes ***]
- 4) STI (SUBTITLE)
[tb'\$b' s"5" " ! *** Add subfield delimiter so that 245 can be asssembled later ***]
- 5) AU (AUTHOR)
[tb'10\$a']
- 6) CORP ()
[tb' \$a']
- 7) ED (EDITION)
[tb' \$a']
- 8) PL (PLACE)
[tb'0 \$a']
- 9) PUB ()
[tb'\$b']
- 10) DT (DATE)
[tb'\$c']
- 11) PD (PHYSICAL)
[tb' \$a']
- 12) SER (SERIES)
[tb' \$a']
- 13) DE (DESC)
[tb' 0\$a']
- 14) AB (ABST)
[tb' \$a']
- 15) NTS (NOTES)

Appendix C: INM2MARC.SET - Conversion of INMAGIC records to MARC Comm. format

16)	AE	(ADDED)
	[tb' \$a']	
17)	BN	(ISBN)
	[tb' \$a']	
18)	SN	(ISSN)
	[tb' \$a']	
19)	LC	()
	[tb' \$a']	
20)	ORD	(ORDER)
	[tb ' \$a']	
21)	LOCN	()
	[tb '\$b']	
22)	HO	(HOLDINGS)
23)	CPS	(COPIES)
24)	LBL	(LABEL)
25)	PRE	(PRELIM)
26)	OK	()
27)	SEE	()
28)	SEE_ALSO	()

OUTPUT FILE SPECIFICATIONS

Output File Name: CATSAMP.MRC
Output File Type: MARC Communications File
Global processing codes: []

OUTPUT FIELDS

1)	STATUS	()
	["n" ! *** Create fixed Leader codes. These codes can be refined with conditional Processing Codes. ***]	
2)	RECTYPE	()
	["a"]	
3)	BIBLEVEL	()
	["m"]	
4)	ENCODE	()
5)	DCF	()
6)	001	()
	[F"ID"]	
7)	010	(LC Card Number)
	[f"lc"]	
8)	020	(ISBN)
	[f"bn"]	
9)	022	(ISSN)
	[f"sn"]	
10)	035	(Local Number (Order Number))
	[f"ord"]	
11)	090	()
	[f"cl" f"locn"]	
12)	100	()
	[s1 f"au"]	
13)	110	()
	[s1 f"corp"]	
14)	245	()
	[s1 f"ti" s1- f"sti"]	
15)	250	()
	[f"ed"]	
16)	260	()
	[f"pl" f"pub" f"dt"]	
17)	300	()
	[f"pd"]	
18)	499	(Series Statement - Undefined)
	[f"ser"]	
19)	520	(Abstract)
	[f"ab"]	
20)	590	(Local Note)
	[f"nts"]	
21)	690	(Subject - Local Use)
	[f"de"]	

22) 700 (Added Entry - Personal Name)
[s2- f"au"]
23) 740 (Added Entry - Title traced diff.)
[s2- f"ti"]
24) 799 (Added Entry - Undefined)
[f"ae"]

LIB2MARC.SET - DB/TextWorks LIBRARY or CATALOG data to MARC

These settings were created to show how MARC records could be created from a DB/TextWorks library database. The data may come from a database based on either the LIBRARY or CATALOG data structures from The Library Guide. After converting the records with The Data Magician, you would need to use a MARC loader for the system you wanted to load the records into.

The MARC file is only as good as the data going into it. We have include some choices for the MARC indicator codes, although you may wish to adjust some of them. Note especially that we have tried to analyze leading articles in the Title field to use the correct indicator code. Subjects, series and added entries have been put into the local use or undefined categories since they may be a mixed group. You can easily change what MARC fields should be used by changing the number. For example: we put Descriptors into 690 - Subjects, Undefined. If your system will ignore those, you may wish them put in as 650 - Topical Subjects.

The Data Magician, version 1.5 Release x.x, serial number: 123456
Settings File Name: LIB2MARC.SET

Description: Conversion of DB/TextWorks records in the LIBRARY or CATALOG Structure of The Library Guide to MARC Communications Format

Date: 08-09-1997 Time: 00:35:32

Note: Square brackets ([...]) around codes are for display purposes only.
Do NOT type these in when creating settings!

INPUT FILE SPECIFICATIONS

Input File Name: LIBRARY.ADD
Input File Type: INMAGIC or DB/TextWorks load file
Structure File name: CATALOG.STR
Global processing codes:
[]

INPUT FIELDS

- 1) ID ()
- 2) Class ()
[tb'0 \$a' s" ."\$b." ! *** Add indicator code and separate call number into subfields
***]
- 3) Title ()
[c'The " o"Les " tb'14\$a' c"An " o"Le " o"La " tb'13\$a" c"A " o"L'" tb'12\$a" c"\$a"
c"\$a" c~ tb'10\$a" ! *** Create appropriate filing indicator codes ***]
- 4) Subtitle ()
[tb'\$b' s"5" " ! *** Add subfield delimiter so that 245 can be assembled later ***]
- 5) Author ()
[tb'10\$a']
- 6) Corporate Author ()
[tb' \$a']
- 7) Responsibility ()
[tb'\$c / "]
- 8) Edition ()
[tb' \$a']
- 9) Place ()
[tb'0 \$a']
- 10) Publisher ()
[tb'\$b']
- 11) Pub Date ()
[tb'\$c']
- 12) Physical Description ()
[tb' \$a']
- 13) Record Type ()
- 14) Series ()
[tb' \$a']
- 15) Descriptors ()
[tb' 0\$a']

- 16) Abstract ()
- [*tb' \$a'*]
- 17) Notes ()
- [*tb' \$a'*]
- 18) ISBN ()
- [*tb' \$a'*]
- 19) List Price ()
- 20) ISSN ()
- [*tb' \$a'*]
- 21) LC Card ()
- [*tb' \$a'*]
- 22) Order Number ()
- [*tb ' \$a'*]
- 23) Copy Management ()
- [*tb '\$b'*]
- 24) Holdings ()
- 25) Label Information ()
- 26) When Catalogued ()
- 27) When Approved ()

OUTPUT FILE SPECIFICATIONS

Output File Name: LIBRARY.MRC
Output File Type: MARC Communications File
Global processing codes: []

OUTPUT FIELDS

- 1) STATUS ()
- [*"n" ! *** Create fixed Leader codes. These codes can be made more refined with conditional Processing Codes. ****]
- 2) RECTYPE ()
- [*"a"*]
- 3) BIBLEVEL ()
- [*"m"*]
- 4) ENCODE ()
- 5) DCF ()
- 6) 001 ()
- [*F"ID"*]
- 7) 010 (LC Card Number)
- [*f"LC Card"*]
- 8) 020 (ISBN)
- [*f"ISBN"*]
- 9) 022 (ISSN)
- [*f"ISSN"*]
- 10) 035 (Local Number (Order Number))
- [*f"Order Number"*]
- 11) 090 ()
- [*f"Class" f"Copy Management"*]
- 12) 100 ()
- [*f"Author5(1)"*]
- 13) 110 ()
- [*f"Corporate Author5(1)"*]
- 14) 245 ()
- [*f"Title5(1)" f"Subtitle" f"Responsibility" !Take the first title field, and all of "sti"*]
- 15) 250 ()
- [*f"Edition"*]
- 16) 260 ()
- [*f"Place" f"Publisher" f"Pub Date"*]
- 17) 300 ()
- [*f"Physical Description"*]
- 18) 499 (Series Statement - Undefined)
- [*f"Series"*]
- 19) 520 (Abstract)
- [*f"Abstract"*]
- 20) 590 (Local Note)
- [*f"Notes"*]
- 21) 690 (Subject - Local Use)

Appendix C: INM2MARC.SET - Conversion of INMAGIC records to MARC Comm. format

```
[f"Descriptors]
22) 700 (Added Entry - Personal Name)
    [f"Author5(2-)"]
23) 740 (Added Entry - Title traced diff.)
    [f"Title5(2-)"]
```

LIF2COM.SET - Conversion of MicroLIF MARC records to MARC

These settings were created to convert MARC records that were in the MicroLIF format to the MARC Communications format. MicroLIF format is provided by some book vendors, rather than MARC Communications. MicroLIF is generally easier to design, plus it is easier to read directly than MARC Communications format. Field tags are down the left hand side (thus we use the Tagged Format as the Input Type), with the data to the right. Look at the sample file (with **Shift-F6**) when you have this conversion loaded to see the format.

In this case, library staff needed to load some records from Maclean-Hunter in MicroLIF format, but their library system only handled MARC Communications format. These settings converts between the two.

Most of the fields are converted "as is", with the exception of the Subject fields (6xx). Since these came in all upper, they needed to be converted to upper/lower. The subfield codes pose a problem, however. See those fields for comments on how this was done.

If you run this conversion with the /F option, any unspecified MARC tags will automatically be added to both the Input and Output fields and will get converted.

The Data Magician, version 1.5 Release x.x, serial number: 123456
Settings File Name: LIF2COM.SET

Description: Conversion of MicroLIF MARC records to MARC Communication format
Date: 07-18-1997 Time: 22:33:20
Note: Square brackets ([...]) around codes are for display purposes only.
Do NOT type these in when creating settings!

INPUT FILE SPECIFICATIONS

```
Input File Name: MICROLIF.MRC
Input File Type: Tagged
Repeating Field type: 2
Continuation characters: [ ]
Record starts with: [LDR] and/or contains: []
Record ends with: [] and/or contains: [^`]
Field names begin with: [53] and/or end with: []
Line PRE-Process Codes: []
Global processing codes:
[s"^`" s"^" s"_"$" s"}"é" s"~"è" ! *** Strip End of Record/End of Field symbols,
substitute foreign language characters ***]
```

INPUT FIELDS

- 1) LDR (Leader Information)
[ce qr ! *** If empty, skip record ***]
- 2) 001 (DOBIS Document Number)
- 3) 007 (Physical Descriptors)
- 4) 008 (Fixed Fields)
- 5) 009 (Phys. Desc. for Archive Mat.)
- 6) 010 (LC Card Number)
- 7) 011 (Alternate LCCN -Utlas convention)
- 8) 013 (NLM Catalogue Citation Number)
- 9) 014 (DOBIS Document Number)
- 10) 015 (National Bibliography Number)
- 11) 016 (Nat. Lib. Can. Record Ctrl No.)
- 12) 017 (Copyright)
- 13) 019 (OCLC Control Number)
- 14) 020 (ISBN)
- 15) 021 (Related ISBN (UTLAS convention))
- 16) 022 (ISSN)
- 17) 023 (Film Number)

Appendix C: LIF2COM.SET - Conversion of MicroLIF MARC records to MARC Comm. format

18) 027 (Standard Tech. Rept. Number)
19) 030 (CODEN)
20) 035 (Local Number (Accession/Barcode))
21) 036 (Local System Control Number)
22) 040 (Cataloguing Source)
23) 041 (Language Code)
24) 042 (Authentication Centre)
25) 043 (Geographic Area Code)
26) 045 (Chronological Code)
27) 046 (Record Source)
28) 050 (LC Call Number)
29) 055 (NLC Call Number (Nat. Lib. Can.))
30) 059 (
31) 060 (NLM Call Number)
32) 082 (Dewey Number)
33) 086 (Gov. Doc. Number)
34) 090 (Local Call Number)
35) 092 (Card Generation Control)
36) 093 (
37) 100 (Main Entry - Personal Name)
38) 110 (Main Entry - Corporate Name)
39) 111 (Main Entry - Conference/Meeting)
40) 130 (Main Entry - Uniform Title)
41) 210 (Series Title (abbreviated))
42) 222 (Key Title)
43) 240 (Uniform Title)
44) 243 (Collective Title)
45) 245 (Title Statement)
46) 246 (Variant Title)
47) 250 (Edition)
48) 260 (Imprint (Place/Publisher/Date))
49) 263 (Projected Publication Date)
50) 265 (Source for Acquisition/Subscript.)
51) 300 (Collation)
52) 310 (Frequency)
53) 350 (Price)
54) 362 (Dates & Volumes (Serials))
55) 400 (Series - Personal Name)
56) 410 (Series - Corporate Name)
57) 411 (Series - Meetings)
58) 440 (Series - Title)
59) 490 (Series - Variant Title)
60) 500 (General Note)
61) 501 (Notes)
62) 503 (Bibliographic History Note)
63) 504 (Bibliography Note)
64) 505 (Contents Note)
65) 508 (Credit Note)
66) 510 (Citation Note)
67) 511 (Performers)
68) 515 (
69) 520 (Abstract/Summary)
70) 525 (Supplement Note (Serials))
71) 535 (Local Use Note)
72) 546 (Language Note (Serials))
73) 550 (Issuing Body Note (Serials))
74) 570 (Editor Note (Serials))
75) 580 (Linking Entry Note)
76) 600 (Subject - Personal Name)

[ce c~ s"\$a"\$1 " s"\$x"\$2 " s"\$y"\$3 " s"\$z"\$4 " l pa s"\$1 "\$a" s"\$2 "\$b" s"\$3 "\$y" s"\$4 "\$z" s" And " and " s" Of " of " s" A " a " s" The " the " s" An " an " ! *** We wanted to capitalize each word (instead of all upper), but the subfield codes would have been capitalized, so we converted subfield codes to numbers (plus a space), did the case conversion, then put the codes back. Next, we converted a number of small words to lower case. The "ce c~" at the beginning means do the processing only if the field is not blank.]

77) 610 (Subject - Corporate Name)

[ce c~ s"\$a"\$1 " s"\$x"\$2 " s"\$y"\$3 " s"\$z"\$4 " l pa s"\$1 "\$a" s"\$2 "\$b" s"\$3 "\$y" s"\$4 "\$z" s" And " and " s" Of " of " s" A " a " s" The " the " s" An " an "]

```

78) 611 (Subject - Conference/Meeting)
[ce c~ s"$a"$1 " s"$x"$2 " s"$y"$3 " s"$z"$4 " l pa s"$1 "$a" s"$2 "$b" s"$3 "$y" s"$4
"$z" s" And " and " s" Of " of " s" A " a " s" The " the " s" An " an "]
79) 630 (Subject - Uniform Title)
[ce c~ s"$a"$1 " s"$x"$2 " s"$y"$3 " s"$z"$4 " l pa s"$1 "$a" s"$2 "$b" s"$3 "$y" s"$4
"$z" s" And " and " s" Of " of " s" A " a " s" The " the " s" An " an "]
80) 650 (Subject - Topical)
[ce c~ s"$a"$1 " s"$x"$2 " s"$y"$3 " s"$z"$4 " l pa s"$1 "$a" s"$2 "$b" s"$3 "$y" s"$4
"$z" s" And " and " s" Of " of " s" A " a " s" The " the " s" An " an "]
81) 651 (Subject - Geographic)
[ce c~ s"$a"$1 " s"$x"$2 " s"$y"$3 " s"$z"$4 " l pa s"$1 "$a" s"$2 "$b" s"$3 "$y" s"$4
"$z" s" And " and " s" Of " of " s" A " a " s" The " the " s" An " an "]
82) 655 (Subject - Genre)
[ce c~ s"$a"$1 " s"$x"$2 " s"$y"$3 " s"$z"$4 " l pa s"$1 "$a" s"$2 "$b" s"$3 "$y" s"$4
"$z" s" And " and " s" Of " of " s" A " a " s" The " the " s" An " an "]
83) 682 ( )
[ce c~ s"$a"$1 " s"$x"$2 " s"$y"$3 " s"$z"$4 " l pa s"$1 "$a" s"$2 "$b" s"$3 "$y" s"$4
"$z" s" And " and " s" Of " of " s" A " a " s" The " the " s" An " an "]
84) 697 (French Subject Headings)
85) 700 (Added Entry - Personal Name)
86) 710 (Added Entry - Corporate Name)
87) 711 (Added Entry - Conference/Meeting)
88) 720 ( )
89) 730 (Added Entry - Uniform Title)
90) 740 (Added Entry - Title traced diff.)
91) 770 ( )
92) 775 ( )
93) 776 ( )
94) 780 (Added Entry - Preceding Entry)
95) 785 (Added Entry - Succeeding Entry)
96) 800 (Series Add. Ent. - Personal Name)
97) 810 (Series Add. Ent. - Corp. Name)
98) 811 (Series Add. Ent. - Conf./Meeting)
99) 830 (Series Add. Ent. - Uniform Title)
100) 840 (Series Add. Ent. - Title)
101) 949 (Holdings Info)

```

OUTPUT FILE SPECIFICATIONS

```

Output File Name: COM.MRC
Output File Type: MARC Communications File
Global processing codes: [fc ! *** Copy all field tags with the same name ***]

```

OUTPUT FIELDS

```

1) STATUS (Record Status)
["n" ! *** Force a default of "n" to the Status field. Note: in some MicroLIF files,
the data for these first five fields is contained in the LDR input field. ***]
2) RECTYPE (Record Type)
["a"]
3) BIBLEVEL (Bibliographic Level)
["m"]
4) ENCODE (Encoding Level)
[" "]
5) DCF (Descriptive Cataloguing Form)
[" "]
6) 001 (DOBIS Document Number)
7) 007 (Physical Descriptors)
8) 008 (Fixed Fields)
9) 009 (Phys. Desc. for Archive Mat.)
10) 010 (LC Card Number)
11) 011 (Alternate LCCN -Utlas convention)
12) 013 (NLM Catalogue Citation Number)
13) 014 (DOBIS Document Number)
14) 015 (National Bibliography Number)
15) 016 (Nat. Lib. Can. Record Ctrl No.)
16) 017 (Copyright)
17) 019 (OCLC Control Number)
18) 020 (ISBN)

```

Appendix C: LIF2COM.SET - Conversion of MicroLIF MARC records to MARC Comm. format

19)	021	(Related ISBN (UTLAS convention))
20)	022	(ISSN)
21)	023	(Film Number)
22)	027	(Standard Tech. Rept. Number)
23)	030	(CODEN)
24)	035	(Local Number (Accession/Barcode))
25)	036	(Local System Control Number)
26)	040	(Cataloguing Source)
27)	041	(Language Code)
28)	042	(Authentication Centre)
29)	043	(Geographic Area Code)
30)	045	(Chronological Code)
31)	046	(Record Source)
32)	050	(LC Call Number)
33)	055	(NLC Call Number (Nat. Lib. Can.))
34)	059	()
35)	060	(NLM Call Number)
36)	082	(Dewey Number)
37)	086	(Gov. Doc. Number)
38)	090	(Local Call Number)
39)	092	(Card Generation Control)
40)	093	()
41)	100	(Main Entry - Personal Name)
42)	110	(Main Entry - Corporate Name)
43)	111	(Main Entry - Conference/Meeting)
44)	130	(Main Entry - Uniform Title)
45)	210	(Series Title (abbreviated))
46)	222	(Key Title)
47)	240	(Uniform Title)
48)	243	(Collective Title)
49)	245	(Title Statement)
50)	246	(Variant Title)
51)	250	(Edition)
52)	260	(Imprint (Place/Publisher/Date))
53)	263	(Projected Publication Date)
54)	265	(Source for Acquisition/Subscript.)
55)	300	(Collation)
56)	310	(Frequency)
57)	350	(Price)
58)	362	(Dates & Volumes (Serials))
59)	400	(Series - Personal Name)
60)	410	(Series - Corporate Name)
61)	411	(Series - Meetings)
62)	440	(Series - Title)
63)	490	(Series - Variant Title)
64)	500	(General Note)
65)	501	(Notes)
66)	503	(Bibliographic History Note)
67)	504	(Bibliography Note)
68)	505	(Contents Note)
69)	508	(Credit Note)
70)	510	(Citation Note)
71)	511	(Performers)
72)	515	()
73)	520	(Abstract/Summary)
74)	525	(Supplement Note (Serials))
75)	535	(Local Use Note)
76)	546	(Language Note (Serials))
77)	550	(Issuing Body Note (Serials))
78)	570	(Editor Note (Serials))
79)	580	(Linking Entry Note)
80)	600	(Subject - Personal Name)
81)	610	(Subject - Corporate Name)
82)	611	(Subject - Conference/Meeting)
83)	630	(Subject - Uniform Title)
84)	650	(Subject - Topical)
85)	651	(Subject - Geographic)
86)	655	(Subject - Genre)

87) 682 ()
88) 697 (French Subject Heading)
89) 700 (Added Entry - Personal Name)
90) 710 (Added Entry - Corporate Name)
91) 711 (Added Entry - Conference/Meeting)
92) 720 ()
93) 730 (Added Entry - Uniform Title)
94) 740 (Added Entry - Title traced diff.)
95) 770 ()
96) 775 ()
97) 776 ()
98) 780 (Added Entry - Preceding Entry)
99) 785 (Added Entry - Succeeding Entry)
100) 800 (Series Add. Ent. - Personal Name)
101) 810 (Series Add. Ent. - Corp. Name)
102) 811 (Series Add. Ent. - Conf./Meeting)
103) 830 (Series Add. Ent. - Uniform Title)
104) 840 (Series Add. Ent. - Title)
105) 949 (Holdings Info)

LM2MARC.SET - Conversion of Library Master records to MARC

This is a sample of converting the sample Library Master database "SAMPLE1" to MARC Communications format. As is, this will convert all records to a MARC format capturing titles, authors, and several other fields. Not all fields are converted, however, and perhaps not all records should be converted (eg: Articles). This could be handled by select only Books to export in Library Master or using some Conditional Processing codes with The Data Magician to only process Books.

The Data Magician, version 1.5 Release x.x, serial number: 123456
Settings File Name: LM2MARC.SET

Description: Library Master SAMPLE1 database to MARC Communications format

Date: 08-08-1997 Time: 23:50:29

Note: Square brackets ([...]) around codes are for display purposes only.
Do NOT type these in when creating settings!

INPUT FILE SPECIFICATIONS

Input File Name: SAMPLE1.TXT
Input File Type: Library Master Tagged file
Structure File name: SAMPLE1.STR
Global processing codes:

[c~ qf c~ s'è'e' s'è'Be' s'è'Me' s'á'a' s'à'Ba' s'á'Ma' s'ç'≡c'
s'ç'≡c' s'ñ'En' c+ ! *** Convert some of the PC character codes to the ALA and CAN/MARC Character
set. See Help Screen 214 to extrapolate for other PC codes ***]

INPUT FIELDS

1) RECTYPE ()
2) AUTHORS ()
[TB"10\$a"]
3) TITLE ()
[c"The " o"Les " tb'14\$a' c"An " o"Le " o"La " tb"13\$a" c"A " o"L'" tb"12\$a" c"\$a"
c"\$a" c~ tb"10\$a" ! *** Create appropriate filing indicator codes ***]
4) PUB LOCATION ()
[tb "\$a"]
5) PUBLISHER ()
[tb "\$b"]
6) DATE ()
[tb "\$c"]
7) EDITION ()
[tb " \$a"]
8) CALL NUMBER ()
[tb "\$a"]
9) ABBREVIATION ()
10) LOCATION ()
[tb"\$b"]
11) PERIODICAL ()
12) VOLUME ()
13) PAGES ()
14) TRANSLATORS ()
[tb" \$a" te", trans.]
15) ORIG TITLE ()
[c"The " o"Les " tb'14\$a' c"An " o"Le " o"La " tb"13\$a" c"A " o"L'" tb"12\$a" c"\$a"
c"\$a" c~ tb"10\$a" ! *** Create appropriate filing indicator codes ***]
16) SERIES TITLE ()
[c"The " o"Les " tb'14\$a' c"An " o"Le " o"La " tb"13\$a" c"A " o"L'" tb"12\$a" c"\$a"
c"\$a" c~ tb"10\$a" ! *** Create appropriate filing indicator codes ***]
17) EDITORS ()
[tb "10\$a" TE", ed.]
18) USE ()
19) BOOK TITLE ()
[c"The " o"Les " tb'14\$a' c"An " o"Le " o"La " tb"13\$a" c"A " o"L'" tb"12\$a" c"\$a"
c"\$a" c~ tb"10\$a" ! *** Create appropriate filing indicator codes ***]
20) TYPE ()
21) NUM VOLUMES ()

22) ITEM NUMBER ()
23) DOC SERV/NUM ()
24) DESCRIPTION ()
 [tb " \$a"]
25) SUBJECTS ()
 [tb " 0\$a" s"--"\$x"]
26) ISSUE ()
27) AUTHOR ROLE ()
28) ORIG PUB LOC ()
29) ORIG PUBLISH ()
30) ORIG DATE ()
31) ORIG PERIODI ()
32) ORIG VOLUME ()
33) ORIG PAGES ()
34) ORIG BOOK ()
35) RECORD TITLE ()
36) BOOK AUTHORS ()
37) PERFORMER ()
38) PERFORM ROLE ()
39) SECTION ()
40) TRANS TITLE ()
41) SHORT CITE ()
42) MEETING ()
43) MEETING LOC ()
44) ASSIGNEE ()
45) APPL DAT/NUM ()
46) EQUIV PATENT ()

OUTPUT FILE SPECIFICATIONS

Output File Name: SAMPLE1.MRC
Output File Type: MARC Communications File
Global processing codes: []

OUTPUT FIELDS

1) STATUS ()
 ["n"]
2) RECTYPE ()
 ["a"]
3) BIBLEVEL ()
 ["m"]
4) ENCODE ()
 [" "]
5) DCF ()
 [" "]
6) 090 (Call Number)
 [f"CALL NUMBER" f"LOCATION" TB" "]
7) 100 (Author)
 [F"AUTHORS5(1)"]
8) 245 (Title)
 [F"TITLE"]
9) 250 (Edition)
 [F"EDITION"]
10) 260 (Publication)
 [f"PUB LOCATION" F"PUBLISHER" F"DATE" TB"0 "]
11) 440 (Series Title)
 [F"SERIES TITLE"]
12) 500 (Notes)
13) 520 (Abstract)
 [F"DESCRIPTION"]
14) 650 (Subjects)
 [F"SUBJECTS"]
15) 700 (Added Authors)
 [f"AUTHORS5(2-)" SF F"EDITORS" SF F"TRANSLATORS" XS]
16) 740 (Added Title entries)
 [f"ORIG TITLE" SF F"BOOK TITLE" SF F"TRANS TITLE"]

MARC2LIB.SET - Conversion of MARC records to DB/TextWorks

These settings will convert a MARC Communications format file into DB/TextWorks Library Guide LIBRARY or CATALOG structures.

Note that when you import records created with these settings that have "extended" characters (with accents), that it is creating them using the MS-DOS character set. To load them correctly, you must set the User Preferences (in the File Menu) so that the File Format will Read/write extended MS-DOS characters.

These settings will work for either the LIBRARY or CATALOG structure as it uses the common fields between them.

The Data Magician, version 1.5 Release x.x, serial number: 123456
Settings File Name: MARC2LIB.SET

Description: Conversion of MARC records to INMAGIC's DB/TextWorks LIBRARY or CATALOG structure
Note that you must set the User Preferences (File Menu) so that File Format will Read/write extended characters in MS-DOS format, if you have extended chars.

Date: 08-08-1997 Time: 23:58:16

Note: Square brackets ([...]) around codes are for display purposes only.
Do NOT type these in when creating settings!

INPUT FILE SPECIFICATIONS

Input File Name: UTLAS.MRC
Input File Type: MARC Communications file
Global processing codes:

[ce qf c~ s'e'è' s'Be'è' s'Be'è' s'Me'è' s'Mb'ö' s'Mü'ü' s'Sa'â' s'a'á' s'Ba'à' s'Ba'â' s'Ma'â' s'≡C'ç' s'≡c'ç' s'En'ñ' s'<'f' c+ ! *** Convert some of the ALA and CAN/MARC Character set to PC character codes. See Help Screen 214 to extrapolate for other PC codes ***]

INPUT FIELDS

- | | |
|-------------|------------------------------------|
| 1) STATUS | (Record Status) |
| 2) RECTYPE | (Record Type) |
| 3) BIBLEVEL | (Bibliographic) |
| 4) ENCODE | (Encoding Level) |
| 5) DCF | (Descriptive Cataloguing Form) |
| 6) 001 | (Reserve Sequence Number) |
| 7) 004 | (Amendments) |
| 8) 007 | (Physical Descriptors) |
| 9) 008 | (Fixed Fields) |
| 10) 009 | (Phys. Desc. for Archive Mat.) |
| 11) 010 | (LC Card Number) |
| 12) 011 | (Alternate LCCN -Utlas convention) |
| 13) 013 | (NLM Catalogue Citation Number) |
| 14) 015 | (National Bibliography Number) |
| 15) 016 | (Nat. Lib. Can. Record Ctrl No.) |
| 16) 017 | (Copyright) |
| 17) 019 | (OCLC Control Number) |
| 18) 020 | (ISBN) |
| 19) 021 | (Related ISBN (UTLAS convention)) |
| 20) 022 | (ISSN) |
| 21) 027 | (Standard Tech. Rept. Number) |
| 22) 030 | (CODEN) |
| 23) 035 | (Local Number (Accession/Barcode)) |
| 24) 036 | (Local System Control Number) |
| 25) 040 | (Cataloguing Source) |
| 26) 041 | (Language Code) |
| 27) 042 | (Authentication Centre) |
| 28) 043 | (Geographic Area Code) |
| 29) 045 | (Chronological Code) |
| 30) 046 | (Record Source) |
| 31) 050 | (LC Call Number) |

32) 055 (NLC Call Number (Nat. Lib. Can.))
33) 060 (NLM Call Number)
34) 082 (Dewey Number)
35) 086 (Gov. Doc. Number)
36) 090 (Local Call Number)
37) 092 (Card Generation Control)
38) 093 (
39) 100 (Main Entry - Personal Name)
40) 110 (Main Entry - Corporate Name)
41) 111 (Main Entry - Conference/Meeting)
42) 130 (Main Entry - Uniform Title)
43) 210 (Series Title (abbreviated))
44) 222 (Key Title)
45) 240 (Uniform Title)
46) 243 (Collective Title)
47) 245 (Title Statement)
48) 246 (Variant Title)
49) 250 (Edition)
50) 260 (Imprint (Place/Publisher/Date))
51) 263 (Projected Publication Date)
52) 265 (Source for Acquisition/Subscript.)
53) 300 (Collation)
54) 310 (Frequency)
55) 350 (Price)
56) 362 (Dates & Volumes (Serials))
57) 400 (Series - Personal Name)
58) 410 (Series - Corporate Name)
59) 440 (Series - Title)
60) 490 (Series - Variant Title)
61) 500 (General Note)
62) 503 (Bibliographic History Note)
63) 504 (Bibliography Note)
64) 505 (Contents Note)
65) 510 (Citation Note)
66) 520 (Abstract/Summary)
67) 525 (Supplement Note (Serials))
68) 535 (Local Use Note)
69) 546 (Language Note (Serials))
70) 550 (Issuing Body Note (Serials))
71) 570 (Editor Note (Serials))
72) 580 (Linking Entry Note)
73) 600 (Subject - Personal Name)
74) 610 (Subject - Corporate Name)
75) 611 (Subject - Conference/Meeting)
76) 630 (Subject - Uniform Title)
77) 650 (Subject - Topical)
78) 651 (Subject - Geographic)
79) 655 (Subject - Genre)
80) 700 (Added Entry - Personal Name)
81) 710 (Added Entry - Corporate Name)
82) 711 (Added Entry - Conference/Meeting)
83) 730 (Added Entry - Uniform Title)
84) 740 (Added Entry - Title traced diff.)
85) 780 (Added Entry - Preceding Entry)
86) 785 (Added Entry - Succeeding Entry)
87) 800 (Series Add. Ent. - Personal Name)
88) 810 (Series Add. Ent. - Corp. Name)
89) 811 (Series Add. Ent. - Conf./Meeting)
90) 830 (Series Add. Ent. - Uniform Title)
91) 840 (Series Add. Ent. - Title)

OUTPUT FILE SPECIFICATIONS

Output File Name: LIBRARY.ADD
Output File Type: INMAGIC or DB/TextWorks load file
Structure File name: LIBRARY.STR
Global processing codes: []

Appendix C: MARC2LIB.SET - Conversion of MARC recrds to DB/TextWorks

OUTPUT FIELDS

```
1) ID          ()
   [F"001" xl bw1 !You may just want this to be blank and let DB/TextWorks create the
ID key]
2) Class      ()
   [f"0905(1)$a" s'/' ' ! *** Take subfield "a" of the local call number, and convert
slashes to spaces ***]
3) Label Information  ()
4) Copy Management  ()
5) Title         ()
   [f"2455$a" XP xe": " ! *** Take subfield "a" of the title and strip trailing
punctuation ***]
6) Subtitle     ()
   [f"2455$b" xe"/ "]
7) Author       ()
   [f"1005$c" sf f"7005$c" XE", " xs]
8) Corporate Author  ()
   [f"1105$c" sf f"7105$c" sf f"1115$c" sf f"7115$c" xs]
9) Responsibility  ()
   [f"2455$c" xp]
10) Edition     ()
   [f"2505$a" XE "- "]
11) Source      ()
12) Place       ()
   [f"2605$a" XE ",: "]
13) Publisher   ()
   [f"2605$b" XE ", "]
14) Pub Date    ()
   [f"2605$c" XE ". "]
15) Physical Description (PHYSICAL)
   [f"3005$c" XE "- "]
16) Record Type  ()
   ["book"]
17) Series      ()
   [f'400' sf f'410' sf f'440' sf f'490' sf f'210' sf f"800" sf f"800" sf f"810" sf f"811"
sf f"830" sf f"840" xs bm'$*' xp ! *** Get all the series fields (separated by subfield marks),
clear out any excess marks (xs), extract all MARC subfields, then strip trailing punctuation ***]
18) Descriptors  ()
   [f'650' sf f'610' sf f'611' sf f'630' sf f'651' sf f'655' xs sm" -- " bm'$*' !separate
subfields with " -- "]
19) Abstract     ()
   [f"5205$c"]
20) Notes        ()
   [f"500" sf f"503" sf f"504" sf f"505" sf f"510" sf f"525" sf f"535" sf f"546" sf f"550"
sf f"570" sf f"580" xs bm'$*"]
21) ISBN        ()
   [f"0205$a" xe": "]
22) ISSN        ()
   [f"0225$a"]
23) Order Number  ()
24) List Price   ()
   [f"0205$c"]
25) Holdings      (Serial Holdings)
   [f"3625$c"]
26) LC Card      ()
   [f"0105$a" xl]
```

MEDCDINM.SET - Conversion of MedLine CD-ROM records to INMAGIC

This sample demonstrates the conversion of records downloaded from a MedLine CD-ROM, into an INMAGIC format. Once converted, you can use the Add option in Maintain to load these into INMAGIC.

Of special note in this conversion is the fact the lines coming from the download file are longer than most. An entire abstract is printed on a single line! To accommodate this, you will need to run The Data Magician with the command line option /L (eg: DATAMAGE/L), which tells it to read exceptionally long lines. Note that this process is a little slower than normal, so you wouldn't want to do this for other conversions. For more information on command line options, see page [139](#).

The Data Magician, version 1.5 Release x.x, serial number: 123456
Settings File Name: MEDCDINM.SET

Description: Sample Medline CD to INMAGIC Conversion

Date: 07-18-1997 Time: 23:41:10

Note: Square brackets ([...]) around codes are for display purposes only.
Do NOT type these in when creating settings!

INPUT FILE SPECIFICATIONS

Input File Name: MEDLINE.ASC
Input File Type: Tagged
Repeating Field type: 2
Continuation characters: []
Record starts with: [<*>] and/or contains: []
Record ends with: [***] and/or contains: []
Field names begin with: [] and/or end with: []
Line PRE-Process Codes:
[]
Global processing codes:
[]

INPUT FIELDS

1) UNIQUE ()
[br "Identifier "]
2) AUTHOR ()
[xl xr]
3) AUTHORS ()
[xl xr bs" " ! *** Strip spaces off left and right, break into subfields on 2 spaces
***]
4) TITLE ()
[xl]
5) JOURNAL ()
[xl]
6) ABSTRACT ()
7) date (templ)
[f"journal" br"[*]" br "," br "," br "," br "," xl ! *** Extract the date from the
Journal field. Break right of a comma four times, just to be sure ***]

OUTPUT FILE SPECIFICATIONS

Output File Name: MEDCDINM.ASC
Output File Type: INMAGIC load file
Structure File name:
Global processing codes: []

OUTPUT FIELDS

1) ID ()
["MED "ns 10 nu ni 1 ! *** Add 'MED' for easy identification of the records loaded
from an outside source ***]
2) CL (CLASS)

Appendix C: MEDCDINM.SET - Conversion of MedLine CD-ROM records to INMAGIC

```
3) TI (TITLE)
   [f"title"]
4) STI (SUBTITLE)
5) AU (AUTHOR)
   [f"author" sf f"authors" xs]
6) CORP ( )
7) ED (EDITION)
8) PL (PLACE)
9) PUB ( )
   [F"JOURNAL" BLF"date" xe ", " ! *** Get publication source information, strip off date
information ***]
10) DT (DATE)
   [f"date"]
11) PD (PHYSICAL)
12) SER (SERIES)
13) DE (DESC)
14) AB (ABST)
   [f"abstract" xl]
15) NTS (NOTES)
16) AE (ADDED)
17) BN (ISBN)
18) SN (ISSN)
19) LC ( )
20) ORD (ORDER)
21) LOCN ( )
22) HO (HOLDINGS)
23) CPS (COPIES)
24) LBL (LABEL)
25) PRE (PRELIM)
   [dt dx]
26) OK ( )
27) SEE ( )
28) SEE_ALSO ( )
```

MEDCDPRO.SET - Conversion of MedLine CD-ROM records to Pro-Cite

This sample demonstrates the conversion of records downloaded from a MedLine CD-ROM, into a Pro-Cite format. Once converted, you can use the Pro-Cite utility program called "IMPORT" to load these into Pro-Cite.

Of special note in this conversion is the fact the lines coming from the download file are longer than most. An entire abstract is printed on a single line! To accommodate this, you will need to run The Data Magician with the command line option /L (eg: DATAMAGE/L), which tells it to read exceptionally long lines. Note that this process is a little slower than normal, so you wouldn't want to do this for other conversions. For more information on command line options, see page [139](#).

The Data Magician, version 1.5 Release x.x, serial number: 123456
Settings File Name: MEDCDPRO.SET

Description: Sample Medline CD to Pro-Cite Conversion

Date: 07-18-1997 Time: 22:33:02

Note: Square brackets ([...]) around codes are for display purposes only.
Do NOT type these in when creating settings!

INPUT FILE SPECIFICATIONS

Input File Name: MEDLINE.ASC

Input File Type: Tagged

Repeating Field type: 2

Continuation characters: []

Record starts with: [<*>] and/or contains: []

Record ends with: [***] and/or contains: []

Field names begin with: [] and/or end with: []

Line PRE-Process Codes:

[]

Global processing codes:

[s`"```` ! *** Convert all " symbols to '. This avoids any possible problems when loading into ProCite, since the data is normally surrounded by the double quote symbol ("). ***]

INPUT FIELDS

1) UNIQUE ()

[br "Identifier " ! *** Part of the field label gets into the data, so we remove it

***]

2) AUTHOR (Used for Single author)

[xl xe ". "]

3) TITLE ()

[xl]

4) JOURNAL ()

[xl]

5) AUTHORS (Used for Multiple authors)

[xl xr bs " " xe ". " s":o"ö" s":a"ä" ! *** Clear spaces on left and right, break into multiple entries on two spaces, strip trailing periods and spaces, convert Scandinavian characters. ***]

6) ABSTRACT ()

7) date (templ)

[f"journal" br "[*]" br ", " br ", " br ", " br ", " br ", " xl ! *** Take the Journal field, take everything after something in square brackets, keep breaking off to the right of commas. What's left should be the date of publication! ***]

OUTPUT FILE SPECIFICATIONS

Output File Name: MEDCDPRO.ASC

Output File Type: ASCII comma delimited

Number of output fields: 47

Global processing codes: []

Appendix C: MEDCDPRO.SET - Conversion of MedLine CD-ROM records to Pro-Cite

OUTPUT FIELDS

1) TYPE (Record Type - Journals Short)
['D']

2) RECNO (Record Number)
[ns 10 nu ni 10]

3) Auth (Author (Analytic))
[f"author" sf f"authors" xs s"5"/" ! *** Take single and multiple author fields and
turn multiple entries into the double slash required by ProCite. ***]

4) AuRo (Author Role (Analytic))

5) Affl (Author Affiliation (Analytic))

6) Titl (Title (Analytic))
[f"title" xp ! *** Strip trailing punctuation from title ***]

7) Medm (Medium Designator)

8) CoPh (Connective Phrase)

9) Auth2 (Author (Monographic))

10) AuRo2 (Author Role (Monographic))

11) Titl2 (Title (Monographic) (Journal))
[F"JOURNAL" bl "["*]" XP ! *** Take everything to left of the square brackets as the
Journal Title. Strip trailing punctuation ***]

12) Ltto (Letter to)

13) Titl3 (Title)

14) DtMt (Date of Meeting)

15) PlMt (Place of Meeting)

16) Medm2 (Medium Designator 2)

17) Edit (Edition)

18) Auth3 (Author (Subsidiary))

19) AuRo3 (Author Role (Subsidiary))

20) PlPu (Place of Publication)

21) Publ (Publisher Name)

22) Date (Date of Publication)
[f"date" xe"."]

23) Copy ()

24) VoID (Volume Number)
[f"journal" br "["*]" bl "("*)" xl xr ! *** The volume number should be to the right of
what's in square brackets and to the left of the round brackets ***]

25) RpID ()

26) IsID (Issue Number)
[f"journal" br "["*]" b "("*)" xb' (' xe') ' ! *** Get everything to the right of the
square brackets, get whatever is in round brackets, strip leading/trailing spaces and round
brackets ***]

27) Loc ()
[f"journal" br "["*]" br ":" blf"date" xe", " ! *** Take everything to the right of the
square brackets, then past the colon, leave whatever is to the left of what is in the date input
field, then strip trailing commas and spaces ***]

28) Extn ()

29) PaMe ()

30) Size ()

31) RpRt ()

32) SrEd ()

33) SrRo ()

34) SrTi ()

35) SrVo ()

36) SrIs ()

37) DcTy ()

38) CoPh ()

39) Aval ()

40) StLc ()

41) CODN ()

42) ISSN ()

43) ISBN ()

44) Note (Notes)

45) Abst (Abstract)
[f"abstract" xl]

46) Call (Call Numbers)

47) Indx (Index)

PROC2MRC.SET - Conversion of Pro-Cite records to MARC

These settings provide the basis for a conversion of Pro-Cite records to MARC Communications format. The sample here rejects any records that are not Record Type A, B, or J (Books Long, Short, and Manuscripts). These settings could be extended to other workform types (such as Reports, Trade Catalogs and Sound Recordings) with some work. You may wish to make those separate conversions, based on this one, or build in conditional processing based on the type of record.

If you wish to create your own file to convert to MARC, you may use the Export program that comes with Pro-Cite. Either Export from the complete database, and have The Data Magician skip unwanted record types, or perform a search of the records you want, merge them into a new database and Export from the new one. When using Export, do NOT change any of the customization options, or the file will likely not load into The Data Magician.

The Data Magician, version 1.5 Release x.x, serial number: 123456
Settings File Name: PROC2MRC.SET

Description: Conversion of ProCite records (Books - Long and Books - Short) to MARC Communications Format
Date: 07-18-1997 Time: 21:55:35
Note: Square brackets ([...]) around codes are for display purposes only.
Do NOT type these in when creating settings!

INPUT FILE SPECIFICATIONS

Input File Name: PROCITE2.ASC
Input File Type: ASCII comma delimited
Number of input fields: 47
Character around text fields: "
Character between fields: ,
Does a RETURN end a field? Y
Does a RETURN end a record? Y
Global processing codes: []

INPUT FIELDS

1) TYPE (Record Type - Books Long)
[C"A" o"B" o"J" c~ qr ! *** Reject records that are not type A, B, or J ***]
2) RECNO (Record Number)
3) Auth (Author (Analytic))
[s///
4) AuRo (Author Role (Analytic))
5) Affl (Author Affiliation (Analytic))
6) Titl (Title (Analytic))
[c"The " o"Les " tb'14\$a' c"An " o"Le " o"La " tb"13\$a" c"A " o"L'" tb"12\$a" c"\$a"
c"\$a" c~ tb"10\$a" ! *** Create appropriate filing indicator codes ***]
7) Medm (Medium Designator)
8) CoPh (Connective Phrase)
9) Auth2 (Author (Monographic))
[s///
10) AuRo2 (Author Role (Monographic))
11) Titl2 (Title (Monographic))
[c"The " o"Les " tb'14\$a' c"An " o"Le " o"La " tb"13\$a" c"A " o"L'" tb"12\$a" c"\$a"
c"\$a" c~ tb"10\$a" ! *** Create appropriate filing indicator codes ***]
12) Ltto (Letter to)
13) Titl3 (Title)
14) DtMt (Date of Meeting)
15) PlMt (Place of Meeting)
16) Medm2 (Medium Designator 2)
17) Edit (Edition)
[tb" \$a"]

Appendix C: MEDCDPRO.SET - Conversion of Pro-Cite records to MARC Comm. format

18) Auth3 (Author (Subsidiary))
[tb"10\$a"]
19) AuRo3 (Author Role (Subsidiary))
20) PlPu (Place of Publication)
[tb"0 \$a"]
21) Publ (Publisher Name)
[tb"\$b"]
22) Date (Date of Publication)
[tb"\$c"]
23) Copy (Date of Copyright)
24) VoID (Volume ID)
25) RpID (Report ID)
26) IsID (Issue ID)
27) Loc (Location in Work)
28) Extn (Extent of Work)
[tb" \$a"]
29) PaMe (Packaging Method)
[tb" "]
30) Size ()
[tb"\$c"]
31) RpRt ()
32) SrEd (Series Editor)
[tb"10\$a"]
33) SrRo (Series Role)
34) SrTi (Series Title)
[c"The " o"Les " tb'14\$a' c"An " o"Le " o"La " tb"13\$a" c"A " o"L'" tb"12\$a" c"\$a"
c"\$a" c~ tb"10\$a" ! *** Create appropriate filing indicator codes ***]
35) SrVo (Series - Volume)
36) SrIs (Series - Issue)
37) DcTy ()
38) CoPh (Connective Phrase)
39) Aval ()
40) StLc ()
41) CODN (CODEN)
[tb" \$a"]
42) ISSN ()
[tb" \$a"]
43) ISBN ()
[tb" \$a"]
44) Note (Notes)
[tb" \$a"]
45) Abst (Abstract)
[tb" \$a"]
46) Call (Call Numbers)
[tb" \$a"]
47) Indx (Index)
[s"/"5" tb" 0\$a"]

OUTPUT FILE SPECIFICATIONS

Output File Name: PROCITE.MRC
Output File Type: MARC Communications File
Global processing codes: []

OUTPUT FIELDS

1) STATUS ()
["n" ! *** Create fixed Leader codes. These codes can be made more refined with
conditional Processing Codes. ***]
2) RECTYPE ()
["a"]
3) BIBLEVEL ()
["m"]
4) ENCODE ()
[" "]
5) DCF ()
[" "]
6) 001 ()
[F"RECNO"]
7) 020 (ISBN)

[F"ISBN"]
8) 022 (ISSN)
[F"ISSN"]
9) 090 (Local Call Number)
[F"CALL"]
10) 100 (Main Entry - Personal Name)
[S1 F"AUTH2"]
11) 245 (Title)
[f"Titl2"]
12) 250 (Edition)
[F"Edit"]
13) 260 (Imprint)
[F"PlPu" F"Publ" F"Date"]
14) 300 (Collation)
[F"Extn" F"PaMe"]
15) 440 (Series Title)
[F"SrTi"]
16) 520 (Abstract)
[F"Abst"]
17) 500 (General Note)
[F"Note"]
18) 690 (Subject - Topical)
[F"Indx"]
19) 700 (Added Entry - Personal Name)
[s2- F"Auth2" sf s1- F"Auth" xs]
20) 740 (Added Entry - Title traced diff.)
[F"Titl"]

STAFF.SET - dBase III to INMAGIC

These settings were created as a fictitious example of how to move a dBase file to a corresponding structure in INMAGIC. Note that there was a "memo" field in the dBase file (NOTES) which is moved to INMAGIC where it can be keyword searched. Also, note that the NAME field is being broken into two fields - FIRST and LAST, by breaking it apart at the comma.

The Data Magician, version 1.5 Release x.x, serial number: 123456
Settings File Name: STAFF.SET

Description: Sample Conversion from dBase III to INMAGIC

Date: 07-18-1997 Time: 21:58:32

Note: Square brackets ([...]) around codes are for display purposes only.
Do NOT type these in when creating settings!

INPUT FILE SPECIFICATIONS

Input File Name: STAFF.DBF
Input File Type: dBase III
Global processing codes:
[XL XR ! *** Strip leading/trailing spaces from input data (dBase stores empty spaces within fixed fields) ***]

INPUT FIELDS

- 1) EMPNO ()
- 2) NAME ()
- 3) TITLE ()
- 4) DEPT ()
- 5) EXTENSION ()
- 6) OFFICE ()
- 7) STARTDATE ()
- 8) SALARY ()
- 9) NOTES (A free-text Memo field)

OUTPUT FILE SPECIFICATIONS

Output File Name: STAFF.INM
Output File Type: INMAGIC load file
Structure File name: \INMAGIC\STAFF.STR
Global processing codes: []

OUTPUT FIELDS

- 1) NO (EMPNUM)
[F"EMPNO"]
- 2) LN (LASTNAME)
[F"NAME" BL ', ' ! *** Take everything to the left of the comma in the name field ***]
- 3) FN (FIRSTNAME)
[F"NAME" BR ', ' XL ! *** Now take everything to the right of the comma ***]
- 4) POS (POSITION)
[F"TITLE"]
- 5) DEPT (DEPARTMENT)
[F"DEPT"]
- 6) EXT (EXTENSION)
[F"EXTENSION"]
- 7) OFF (OFFICE)
[F"OFFICE"]
- 8) SD (STARTDATE)
[F"STARTDATE" DX ! *** Take the numeric Start Date (eg: 19890522) and convert to a text format (eg: May 22, 1989) ***]
- 9) SAL (SALARY)
[F"SALARY"]
- 10) NTS (NOTES)
[F"NOTES"]

UTLASINM.SET - UTLAS (MARC communications format) to INMAGIC

These settings were created for a client to move their holdings on UTLAS to an INMAGIC database using the Biblio Guide CATALOG structure. While many of the fields would make sense for most users, there may be some changes necessary for any particular application.

These same settings could be used for almost any MARC communications file from a variety of vendors, including: BiblioFile (when converted to "CFILE" format), Laser Quest, OCLC, RLIN, and others. Be sure that the file you are converting is MARC Communications format. There are a wide variety of formats for display MARC records.

These settings could also be used to convert to STAR by simply changing the Output File type from 4 to 7, as well as adjusting any field names that differ.

Some character translations may need to be added to the Global Processing Codes for the Input File, if you have a lot of French materials.

The Data Magician, version 1.5 Release x.x, serial number: 123456
Settings File Name: UTLASINM.SET

Description: Conversion of UTLAS MARC records to INMAGIC format (BIBLIO's CATALOG structure)

Date: 08-09-1997 Time: 00:13:09

Note: Square brackets ([...]) around codes are for display purposes only.
Do NOT type these in when creating settings!

INPUT FILE SPECIFICATIONS

Input File Name: UTLAS.MRC
Input File Type: MARC Communications file
Global processing codes:

[ce qf c~ s' e' é' s' ße' è' s' Be' ê' s' Me' ë' s' ' a' á' s' ßa' à' s' Ba' â' s' Ma' ä' s' =C' Ç'
s' =c' ç' s' En' ñ' ! *** Convert some of the ALA and CAN/MARC Character set to PC character codes.
See Help Screen 214 to extrapolate for other PC codes ***]

INPUT FIELDS

- | | |
|-------------|------------------------------------|
| 1) STATUS | (Record Status) |
| 2) RECTYPE | (Record Type) |
| 3) BIBLEVEL | (Bibliographic) |
| 4) ENCODE | (Encoding Level) |
| 5) DCF | (Descriptive Cataloguing Form) |
| 6) 001 | (Reserve Sequence Number) |
| 7) 004 | (Amendments) |
| 8) 007 | (Physical Descriptors) |
| 9) 008 | (Fixed Fields) |
| 10) 009 | (Phys. Desc. for Archive Mat.) |
| 11) 010 | (LC Card Number) |
| 12) 011 | (Alternate LCCN -Utlas convention) |
| 13) 013 | (NLM Catalogue Citation Number) |
| 14) 015 | (National Bibliography Number) |
| 15) 016 | (Nat. Lib. Can. Record Ctrl No.) |
| 16) 017 | (Copyright) |
| 17) 019 | (OCLC Control Number) |
| 18) 020 | (ISBN) |
| 19) 021 | (Related ISBN (UTLAS convention)) |
| 20) 022 | (ISSN) |
| 21) 027 | (Standard Tech. Rept. Number) |
| 22) 030 | (CODEN) |
| 23) 035 | (Local Number (Accession/Barcode)) |
| 24) 036 | (Local System Control Number) |
| 25) 040 | (Cataloguing Source) |

Appendix C: UTLASINM.SET - UTLAS (MARC communications format) to INMAGIC

26) 041 (Language Code)
27) 042 (Authentication Centre)
28) 043 (Geographic Area Code)
29) 045 (Chronological Code)
30) 046 (Record Source)
31) 050 (LC Call Number)
32) 055 (NLC Call Number (Nat. Lib. Can.))
33) 060 (NLM Call Number)
34) 082 (Dewey Number)
35) 086 (Gov. Doc. Number)
36) 090 (Local Call Number)
37) 092 (Card Generation Control)
38) 093 (
39) 100 (Main Entry - Personal Name)
40) 110 (Main Entry - Corporate Name)
41) 111 (Main Entry - Conference/Meeting)
42) 130 (Main Entry - Uniform Title)
43) 210 (Series Title (abbreviated))
44) 222 (Key Title)
45) 240 (Uniform Title)
46) 243 (Collective Title)
47) 245 (Title Statement)
48) 246 (Variant Title)
49) 250 (Edition)
50) 260 (Imprint (Place/Publisher/Date))
51) 263 (Projected Publication Date)
52) 265 (Source for Acquisition/Subscript.)
53) 300 (Collation)
54) 310 (Frequency)
55) 350 (Price)
56) 362 (Dates & Volumes (Serials))
57) 400 (Series - Personal Name)
58) 410 (Series - Corporate Name)
59) 440 (Series - Title)
60) 490 (Series - Variant Title)
61) 500 (General Note)
62) 503 (Bibliographic History Note)
63) 504 (Bibliography Note)
64) 505 (Contents Note)
65) 510 (Citation Note)
66) 520 (Abstract/Summary)
67) 525 (Supplement Note (Serials))
68) 535 (Local Use Note)
69) 546 (Language Note (Serials))
70) 550 (Issuing Body Note (Serials))
71) 570 (Editor Note (Serials))
72) 580 (Linking Entry Note)
73) 600 (Subject - Personal Name)
74) 610 (Subject - Corporate Name)
75) 611 (Subject - Conference/Meeting)
76) 630 (Subject - Uniform Title)
77) 650 (Subject - Topical)
78) 651 (Subject - Geographic)
79) 655 (Subject - Genre)
80) 700 (Added Entry - Personal Name)
81) 710 (Added Entry - Corporate Name)
82) 711 (Added Entry - Conference/Meeting)
83) 730 (Added Entry - Uniform Title)
84) 740 (Added Entry - Title traced diff.)
85) 780 (Added Entry - Preceding Entry)
86) 785 (Added Entry - Succeeding Entry)
87) 800 (Series Add. Ent. - Personal Name)
88) 810 (Series Add. Ent. - Corp. Name)
89) 811 (Series Add. Ent. - Conf./Meeting)
90) 830 (Series Add. Ent. - Uniform Title)
91) 840 (Series Add. Ent. - Title)

OUTPUT FILE SPECIFICATIONS

The Data Magician, version 1.5

Output File Name: UTLAS.INM
Output File Type: INMAGIC or DB/TextWorks load file
Structure File name: CATALOG.STR
Global processing codes: []

OUTPUT FIELDS

- 1) ID ()
[ns 1 nu ni 1]
- 2) CL (CLASS)
[f"0905\$a" s/' ' ! *** Take subfield "a" of the local call number, and convert slashes to spaces ***]
- 3) TI (TITLE)
[f"2455\$a" XP ! *** Take subfield "a" of the title and strip trailing punctuation ***]
- 4) STI (SUBTITLE)
[f"2455\$b"]
- 5) AU (AUTHOR)
[f"1005\$a" XE", "]
- 6) CORP ()
[f"1105\$a"]
- 7) ED (EDITION)
[f"2505\$a" XE "- "]
- 8) PL (PLACE)
[f"2605\$a" XE ": "]
- 9) PUB ()
[f"2605\$b" XE ", "]
- 10) DT (DATE)
[f"2605\$c" XE ". "]
- 11) PD (PHYSICAL)
[f"3005\$a" XE "- "]
- 12) SER (SERIES)
[f'400' sf f'410' sf f'440' sf f'490' xs bm'\$*' xp ! *** Get all the series fields (separated by subfield marks), clear out any excess marks (xs), extract all MARC subfields, then strip trailing punctuation ***]
- 13) DE (DESC)
[f'650' sf f'610' sf f'611' sf f'630' sf f'651' sf f'655' xs sm" -- " bm'\$*']
- 14) AB (ABST)
[f"5205\$a"]
- 15) NTS (NOTES)
[f"500" sf f"503" sf f"504" sf f"505" sf f"510" sf f"525" sf f"535" sf f"546" sf f"550" sf f"570" sf f"580" xs bm'\$*']
- 16) AE (ADDED)
[f"700" sf f"710" sf f"740" sf f"780" sf f"785" sf f"711" sf f"730" sf f"800" sf f"810" sf f"811" sf f"830" sf f"840" xs bm'\$*']
- 17) BN (ISBN)
[f"0205\$a"]
- 18) SN (ISSN)
[f"0225\$a"]
- 19) LC ()
[f"0105\$a" x1]
- 20) ORD (ORDER)
- 21) LOCN ()
[f"0905\$b"]
- 22) HO (HOLDINGS)
[f"3625\$a"]
- 23) CPS (COPIES)
- 24) LBL (LABEL)
- 25) PRE (PRELIM)
[DT DX ! *** Get today's date (in numeric form), then convert to text form ***]
- 26) OK ()
- 27) SEE ()
- 28) SEE_ALSO ()

INDEX

- .DBF
 - dBase format files [21](#)
- !
 - Comments in Processing Codes [67](#)
- \$
 - Changing MARC subfield character [48](#)
- ***
 - Blank line marks end of record [26](#)
 - Example [154](#)
- /?
 - Command Line Help [140](#)
- /O [142](#)
- /A=filename [140](#)
- /B=filename [140](#)
- /F
 - Automatically add fields [140](#)
 - DOBISMRC.SET [161](#)
 - LIF2COM.SET [180](#)
- /H
 - Command Line Help [140](#)
- /I=n [38](#), [140](#)
- /L [140](#)
 - Example [190](#), [192](#)
- /M [10](#), [65](#), [141](#)
- /N=startnum [141](#)
- /O=n [141](#)
- /S=filename [139](#)
- /T [141](#)
- /X [142](#)
- Adding text
 - If field not empty [124-127](#)
- Alt-A [21](#), [61](#)
- Alt-B [61](#)
- Alt-C [10](#), [61](#)
- Alt-F1 [62](#)
- Alt-F3 [39](#), [65](#), [94](#), [96-98](#)
- Alt-I [61](#)
- Alt-L [63](#)
- Alt-M [10](#), [57](#)
- Alt-O [61](#)
- Alt-P [63](#), [151](#)
- Alt-S [55](#), [63](#)
- ASCII codes
 - Entering non-displayable codes [65](#)
 - Non-displayable [37](#), [51](#), [65](#)
- ASCII delimited files [21](#)
 - Field names [37](#)
 - Input file [22](#)
 - Output file [41](#), [43](#)
 - Output format [17](#)
- Automatic execution mode [114](#), [142](#)
- Automatic number generation [107-109](#)
 - Specifying starting number [141](#)
- Average seconds per record [56](#)
- Backspace [60](#)
- Batch files [139](#)
- BBS
 - Data Magician Web Site [15](#)
- Beep
 - Turning off [145](#)
- Beginning of Record Starts With &/or Contains [25](#)
- BIBLEVEL
 - Special MARC field [32](#), [48](#)
- Biblio File
 - Importing from [31](#)
- Biblio Guide [163](#), [198](#)
- BiblioFile
 - to INMAGIC [198](#)
- Bibliographic Level
 - Special MARC field [32](#), [48](#)
- Binary editor [35](#)
- Books In Print
 - Conversion to MARC [152](#)
- Break MARC subfield
 - specifying the delimiter [31](#)
- CAN/OLE
 - to Pro-Cite [154](#)
- Cancel conversion [114](#)
- Cancel processing
 - of current record [116](#)
- CANEBSCO
 - to INMAGIC [163](#)
- Capitalization
 - Example [156](#)

- Case changes
 - All words [113](#)
 - First word only [112](#)
 - Lowercase [105](#)
 - Uppercase [128](#)
- CD-ROM MARC cataloguing products
 - Importing from [31](#)
- CDS/ISIS [1](#), [22](#)
 - Output format [17](#)
- CGA [141](#)
- Character separating fields [23](#)
- Character surrounding text fields [23](#)
- Character translations
 - Example [198](#)
- Colour
 - turning off (/M) [141](#)
 - turning off/on (Ctrl-F9) [65](#)
- Colours
 - turning off [10](#)
- Command line options [139](#)
- Comments
 - within Processing Codes (!) [67](#)
- Completion time estimation [57](#)
- Conditional processing
 - And contains text [74](#)
 - False (off) [91](#)
 - Memory stores - set to True [94](#), [96-98](#)
 - Or contains text [110](#)
 - Or field is empty [111](#)
 - Reverse [92](#)
 - True (On) [90](#)
 - True if field contains "text" [74](#), [89](#), [110](#)
 - True if field empty [93](#), [111](#)
 - Uppercase [95](#), [99](#)
- Contains [26](#)
- Continuation Line Character(s) [25](#)
 - Tagged file options [45](#)
- Control characters
 - Entering codes as <nnn> [23](#)
- CONVBIP.BAT
 - Sample Batch file [139](#)
- Conversion process
 - Completion [57](#)
 - Interrupting [57](#)
 - Starting [56](#)
- Conversion Progress Screen [56](#)
- Conversion Summary screen [18](#), [55](#)
 - accessing [10](#)
- Count
 - Number of occurrences [106](#)
- Creating Processing Codes [38](#)
- Ctrl-End [61](#)
- Ctrl-F1 [62](#)
- Ctrl-F10 [38](#), [52](#), [64](#)
- Ctrl-F2 [63](#), [151](#)
- Ctrl-F3 [37](#), [51](#), [64](#)
- Ctrl-F4 [37](#), [51](#), [65](#)
- Ctrl-F6 [62](#)
- Ctrl-F8 [62](#)
- Ctrl-F9 [10](#), [65](#), [141](#)
- Ctrl-Home [60](#)
- Ctrl-Left-Arrow [60](#)
- Ctrl-Num Lock [9](#)
- Ctrl-Right-Arrow [60](#)
- Cuadra Associates, Inc.
 - STAR text database system [33](#)
- Current time [56](#)
- Cursor moves
 - Character to the left [59](#)
 - Character to the right [59](#)
 - Conversion Summary Screen [61](#)
 - End of the current prompt [60](#)
 - First character of the current prompt [60](#)
 - Input Fields [61](#)
 - Input File Parameters [61](#)
 - Next Prompt [59](#)
 - Output Fields [61](#)
 - Output File Parameters [61](#)
 - Previous prompt [59](#)
 - Prompt above current prompt [59](#)
 - Prompt below current prompt [59](#)
 - Screen down [61](#)
 - Screen up [61](#)
 - While highlighting [63](#)
 - Word left [60](#)
 - Word right [60](#)
- Cut highlighted text [64](#)
- Cut/paste buffer [64](#)
- Data
 - Display On/Off [64](#)
 - Reading [38](#)
 - Setting display size [64](#)
- Data conversion
 - general steps [17](#)
- Data Display [37](#)
 - Expanding [37](#), [51](#)

- Input Fields [37](#)
- Output fields [51](#)
- Suppressing [37](#), [51](#)
- Data files
 - location [6](#)
- Dates
 - Converting numeric to text [101](#)
 - Generating today's date [100](#)
- DB/TextWorks [1](#)
 - Conversion to MARC [177](#)
 - Converting from MARC [187](#)
 - Importing extended MS-DOS characters [187](#)
 - importing into [41](#), [46](#)
 - input format [17](#)
 - load format [46](#)
 - multi-word field tags [30](#)
 - reading dump files [30](#)
 - Structure file [47](#)
 - Tagged format [21](#)
- DBase II [29](#)
- DBase III [17](#), [21](#)
 - Browsing [62](#)
 - field names [17](#)
 - Filling in field names [37](#)
 - Input file [29](#)
 - Memo fields [19](#)
 - Memo file [29](#)
 - output fields [18](#)
 - Reading first record [38](#)
 - Sample file [19](#)
 - Sample Input Fields [40](#)
 - to INMAGIC [197](#)
- dBase IV [1](#)
 - Automatic recognition [29](#)
- DBF
 - dBase data file [29](#)
 - dBase format files [21](#)
- DBS
 - DB/TextWorks structure file [47](#)
- DBT
 - dBase Memo file [29](#)
- DCF
 - Special MARC field [32](#), [48](#)
- Default directory [7](#)
- Del [60](#)
- Delete
 - All multiple spaces [131](#)
 - Character to left [60](#)
 - Character under cursor [60](#)
 - Characters left of cursor [60](#)
 - Characters to right of cursor [61](#)
 - Current field [61](#)
 - Current prompt [61](#)
 - Entire field [133](#)
 - Leading blanks [135](#)
 - Trailing blanks [137](#)
- Descriptive Cataloguing Form
 - Special MARC field [32](#), [48](#)
- DIALOG [26](#)
 - Field names [27](#)
 - to INMAGIC [156](#)
- Directory
 - changing [7](#), [63](#)
 - default [7](#), [9](#)
 - Input File [22](#)
 - root [9](#)
 - working [9](#)
- Directory display
 - Full format (Shift-F8) [62](#)
 - Wide format (Ctrl-F8) [62](#)
- DISKEDIT [35](#)
- DOBIS
 - to INMAGIC [158](#), [161](#)
 - to MARC Communication format [161](#)
- DOS
 - Temporary exit to [62](#)
- Down Arrow [59](#)
- DOWN.EXE
 - Conversion program for DOBIS [158](#), [161](#)
- Downloading new releases [6](#)
- EBSCO
 - to INMAGIC [163](#)
- Edibase
 - importing into [41](#)
- Editing keys
 - Summary [59](#)
- EGA [141](#)
- Email
 - Getting help via [15](#)
- Empty line
 - Marking end of record [26](#)
- ENCODE
 - Special MARC field [32](#), [48](#)
- Encoding Level
 - Special MARC field [32](#), [48](#)
- End [60](#), [63](#)
- End of Field character(s) [43](#)

- End of Field Characters(s)
 - Tagged file option [46](#)
- End of Record process code(s) [43](#)
 - Tagged file option [45](#)
- End of Record Starts With &/or Contains [26](#)
- End With
 - Tagged file option [45](#)
- Error messages
 - Getting Help [61](#)
 - help [14](#)
 - Standard [147](#)
 - System [149](#)
- Errors
 - Anticipated [145](#)
 - in command line [142](#)
 - System [145](#)
 - Unanticipated [145](#)
- Esc [13](#), [57](#), [61](#), [63](#)
- Estimated number of records [56](#)
- Estimated time of completion [56](#)
- Exit to DOS [62](#)
- Export
 - Pro-Cite utility [194](#)
- Extended MS-DOS characters
 - Converting in DB/TextWorks [187](#)
- F1 [13](#), [14](#), [61](#), [67](#), [145](#), [147](#)
- F10 [38](#), [52](#), [60](#)
- F2 [18](#), [39](#), [53](#), [55](#), [64](#)
- F3 [17](#), [38](#), [39](#), [52](#), [53](#), [65](#)
- F4 [17](#), [39](#), [53](#), [65](#)
- F5 [63](#), [64](#)
- F6 [27](#)
 - in search strings [69](#)
- F7 [63](#), [64](#)
- F8 [63](#), [64](#)
- F9 [145](#)
- FAXON
 - to INMAGIC [163](#)
- Field Label [37](#)
 - Input Fields [37](#)
 - Output fields [51](#)
- Field Name/Description [37](#)
 - Input Fields [37](#)
 - Output fields [51](#)
- Field names
 - Must end with symbol [27](#)
 - Prefixed with &/or End With [27](#)
 - Prompting to add [38](#)
 - Specific length [27](#)
 - unknown [17](#)
 - Unrecognized [38](#), [52](#), [57](#)
 - with embedded spaces [41](#)
- Field tag width [45](#)
 - Tagged file option [45](#)
- Field tags [24](#)
 - Prompting to add [31](#), [33](#), [34](#), [37](#)
- Field tags prefixed with
 - Tagged file option [45](#)
- Fields
 - Inserting [38](#), [60](#)
 - Number allowed - Input [38](#)
 - Reading long lines [140](#)
 - Setting limit number [38](#)
 - Specifying number of [38](#)
- File size estimation [57](#)
- Filename [53](#)
- Fixed Length Fields
 - Input file [35](#)
 - input format [22](#)
- Format of Repeated Fields [24](#)
 - Tagged file option [44](#)
- Fox-Base
 - Importing from [29](#)
- Full-screen editing mode [60](#)
- Function keys
 - Help [14](#), [62](#)
 - Summary [59](#)
- GEAC
 - to INMAGIC [166](#)
 - to Pro-Cite [170](#)
- Global Output field processing [42](#)
- Global processing codes [21](#), [65](#)
- Global search and replace [117](#)
- Hardware Requirements [5](#)
- Help [61](#)
 - accessing [13](#)
 - Editing keys [62](#)
 - Error messages [61](#), [145](#)
 - General [14](#)
 - Index [14](#)
 - leaving [13](#)
 - Main Menu [11](#)
 - Screen number [14](#)
- Highlighting
 - Cutting text [64](#)
 - Entire prompt [64](#)

- On/Off [63](#)
- Paste cut text [64](#)
- Home [60](#), [62](#)
- IMPORT
 - Pro-Cite utility [192](#)
- Incremental numbers [18](#)
- INMAGIC [17](#)
 - Dump files [21](#), [30](#)
 - field names [17](#)
 - Filling in field names [37](#)
 - from BiblioFile [198](#)
 - from dBase III [197](#)
 - from DOBIS [158](#), [161](#)
 - from EBSCO, CANEBSCO, or FAXON [163](#)
 - from GEAC [166](#)
 - from MedLine CD-ROM [190](#)
 - from OCLC [198](#)
 - from RLIN [198](#)
 - importing into [41](#)
 - Load files [41](#), [46](#)
 - Multiple entries [87](#)
 - output fields [18](#)
 - Output file [41](#), [46](#)
 - output format [17](#)
 - Structure file [47](#)
 - Structure files, reading [30](#), [34](#), [50](#)
 - to MARC Communications Format [156](#),
[158](#), [174](#), [177](#), [185](#), [187](#), [190](#), [192](#)
- Inmagic Tagged Format
 - DB/TextWorks import option [46](#)
- Input fields
 - Allowing more space for [140](#)
 - definition [17](#), [37](#)
 - Help display [62](#)
 - Number of [23](#)
 - Sample - dBase III [40](#)
 - Setting number of [64](#)
- Input Fields Screen [37](#)
- Input File
 - Browsing [62](#)
 - Field tags [24](#)
 - format definition [17](#)
 - Location [6](#)
 - Names [22](#)
 - Pre-Processing Codes [27](#)
 - Read next record [65](#)
 - Resetting [39](#), [65](#)
 - Specifying name from command line [140](#)
- Input File Parameters
 - Definition [21](#)
 - Getting to [21](#)
 - Prompts [22](#)
 - Sample - dBase III [36](#)
- Input File Type
 - Selecting [21](#)
- Input Line PRE-Processing Codes [27](#)
- Ins [60](#)
- Insert mode [60](#)
- Insert new field [60](#)
- Installation [5](#)
- ISO 9
 - Micro CDS/ISIS output format [22](#)
- Laser Quest
 - Importing from [31](#)
 - to INMAGIC [198](#)
- Left Arrow [59](#)
- Left Pad
 - Tagged file option [45](#)
- Length
 - Input Fields [37](#)
- Library Guide
 - Converting to MARC [177](#)
- Library Master
 - Conversion to MARC [185](#)
 - importing into [41](#)
 - importing records [49](#)
 - Input file [34](#), [49](#)
 - input format [22](#)
 - output format [42](#)
- List server
 - Subscribing to [15](#)
- load format [46](#)
- Loading settings [63](#)
- Long lines
 - Example reading [190](#), [192](#)
 - Reading from Input File [140](#)
- Look-up table
 - see Substitute Table [122](#)
- Lowercase
 - Example [156](#)
- Maclean-Hunter
 - loading records from [180](#)
- Main Menu [9](#)
 - help [11](#)
 - returning to [10](#)
 - selecting items [10](#)
- MARC Block Size [31](#)
- MARC Communication files [17](#), [21](#)

- Block Size [21](#)
- Browsing [62](#)
- Extracting subfields [81](#)
- Field names [17](#), [37](#)
- Fixed fields [78](#), [84](#)
- Input file [31](#)
- Output file [41](#), [47](#)
- Output format [17](#)
- Problems viewing output [57](#)
 - spanned [21](#)
- MARC field names
 - Special [32](#)
- MARC field tags [32](#)
- MARC Subfield character
 - changing for output files [48](#)
- MARC Substitution Character [31](#)
- Maximum line width
 - Tagged file option [46](#)
- MedLine CD-ROM
 - to INMAGIC [190](#)
 - to Pro-Cite [192](#)
- Memo file [29](#)
- Memory
 - requirements [5](#)
- Memory stores
 - Reset [94](#), [96-98](#)
- Message files [5](#)
- Micro CDS/ISIS
 - Creating an output file [42](#)
 - Format of output file [42](#)
 - input format [17](#)
 - Output file [49](#)
 - see also CDS/ISIS [1](#)
- MicroLIF MARC format
 - converting to MARC Communications [180](#)
 - Reading field tags [27](#)
- Monochrome mode [141](#)
 - Command line switch /M [10](#)
- Multiple entries [87](#)
- National Library of Canada
 - DOBIS [158](#), [161](#)
- Non-displayable ASCII codes
 - Entering codes as <nnn> [26](#)
 - Explanation and creating [65](#)
 - in MARC records [31](#)
- Non-displayable codes
 - making displayable with Ctrl-F4 [37](#), [51](#), [64](#)
- Norton Utilities [35](#)
- Number of characters to skip [35](#)
- Number of Input Fields [23](#)
- Number of occurrences
 - Counting [106](#)
- Number of Output Fields [43](#)
- Numbers
 - Generating incremental [107-109](#)
- OCLC
 - to INMAGIC [198](#)
- Order of processing [65](#)
- Out of string space
 - What to do [149](#)
- Output empty fields?
 - Tagged file option [46](#)
- Output fields
 - Adding new fields [52](#)
 - Allowing more space for [141](#)
 - Defining [51](#)
 - Getting data into [18](#)
 - Number allowed [52](#)
 - Number of [43](#)
 - Processing codes [53](#)
 - Reading sample data [52](#)
 - Sample [54](#)
 - Setting number of [52](#), [64](#)
- Output File
 - Browsing [62](#)
 - Defining [41](#)
 - Location [6](#)
 - Naming [42](#)
 - Specifying name from command line [140](#)
 - Type [41](#)
 - Viewing during conversion [57](#)
- Output File Parameters
 - Sample [50](#)
- Output format
 - definition [17](#)
- Paste [64](#)
- PC-Vision
 - Transferring from a PC to STAR [48](#)
 - Transferring records from STAR to PC [33](#)
- Percentage complete [56](#)
- PgDn [14](#), [61](#), [62](#)
- PgUp [14](#), [61](#), [62](#)
- Power failure [39](#)
- Pre-Processing Codes [27](#)
- Printer
 - requirements [5](#)

- Printing settings [63](#)
- Pro-Cite
 - from CAN/OLE [154](#)
 - from GEAC [170](#)
 - from MedLine CD-ROM [192](#)
 - Multiple entries [87](#)
 - to MARC Communications format [194](#)
 - using the Export utility [194](#)
- Problems
 - Recording/Reporting [146](#)
- Processing Codes
 - Comments in (!) [67](#)
 - Creating [38](#)
 - definition [17](#)
 - Display On/Off [64](#)
 - Errors [67](#)
 - general [17](#)
 - help [14](#), [67](#)
 - Input Fields [37](#)
 - Input fields screen [37](#)
 - Length [37](#)
 - Order of processing [65](#)
 - Output fields [51](#), [53](#)
 - Quotes [67](#)
 - Search Strings [68](#)
 - Spaces [67](#)
 - Summary of [70](#)
 - Suppressing display of [37](#)
 - Test process [65](#)
 - Testing [39](#), [53](#)
 - Upper/Lower Case [67](#)
 - Wildcards [67](#)
- Processing log [56](#)
- Processing start time [56](#)
- Prompting to add
 - Field tags [31](#), [33](#), [34](#)
- Quit processing [114](#)
- Quitting the program [57](#)
- Quotes [67](#)
- Read Next record [65](#)
- README file [5](#)
- README program [5](#)
- Record
 - end of [17](#), [24](#), [26](#)
 - End of (Blank line) [26](#)
 - read next [65](#)
 - start of [17](#), [24](#), [25](#)
- Record numbers
 - Ending [55](#)
 - Starting [55](#)
- Record Status
 - Special MARC field [32](#), [48](#)
- Record type
 - special field for Library Master [42](#)
 - Special MARC field [32](#), [48](#)
- RECTYPE
 - Special MARC field [32](#), [48](#)
- Registering a Demonstration Version [6](#)
- Registration code
 - Updating a new release [6](#)
- Reset Input File [39](#), [65](#)
- RETURN end a field? [23](#)
- RETURN end a record? [23](#)
- Revelation
 - Multiple entries [87](#)
- Right Arrow [59](#)
- Right Pad
 - Tagged file option [45](#)
- RLIN
 - Reading field tags [27](#)
 - to INMAGIC [198](#)
- Sample files [5](#)
- Saving Settings [39](#), [53](#), [63](#)
- Screen Number [13](#)
- Search and Replace [117](#)
- Search Strings [68](#)
- Settings
 - Comment/Description [55](#)
 - Extension [39](#)
 - Loading [63](#)
 - Loading from command line [139](#)
 - Location of files [6](#)
 - Naming [39](#), [53](#)
 - Overwriting [53](#), [63](#)
 - Printing [63](#)
 - Replacing old version [39](#)
 - saving [18](#), [39](#), [53](#), [55](#), [63](#)
- Shift-F1 [15](#), [62](#)
- Shift-F10 [61](#)
- Shift-F2 [64](#)
- Shift-F3 [37](#), [51](#), [64](#)
- Shift-F4 [37](#), [51](#), [64](#)
- Shift-F5 [64](#)
- Shift-F6 [62](#), [180](#)
- Shift-F7 [60](#)
- Shift-F8 [62](#)
- Shift-PrtSc [146](#)
- Shift-Tab [10](#), [59](#)

Show MARC subfield delimiters as
Changing for Output Files [48](#)

Software Licence [5](#)

Software Registration Form [5](#)

spanned format MARC files [31](#)

Special MARC field names

MARC Output files [48](#)

STAFF.DBF [19](#)

STAFF.DBT [19](#)

STAFF.STR [19](#)

STAR [1](#), [21](#)

conversion from MARC [198](#)

Creating an output file [42](#)

Format of output file [42](#)

Importing from [33](#)

input format [17](#)

Output file [48](#)

Output format [17](#)

Using *DUMP to write records out [33](#)

Start of Record process code(s) [43](#)

Starting number

Specifying from command line [141](#)

Starts With [26](#)

STATUS

Special MARC field [32](#), [48](#)

Stored String

Assigning in the Command Line [142](#)

Text Begin [124](#), [125](#)

Text Recall [126](#)

Stored strings

Text Begin [124](#)

Text End [125](#)

Text Recall [126](#)

Text Store [127](#)

STR

INMAGIC Structure file [47](#)

subfield

misunderstood [119](#), [120](#), [138](#)

Subfields

Adding a marker [120](#)

Deleting excess markers [138](#)

Limiting which to process [119](#)

Substitute Table [122](#)

Summary Screen

Switching from View Output [57](#)

Updating statistics [57](#)

Support - email address

support@folland.com [15](#)

Symbol

Tagged file option [44](#)

System Errors [145](#)

Tab [10](#), [59](#)

As a field separator [23](#)

Tabbed output

Command line option [141](#)

Tag [24](#)

Tagged file

output format [43](#)

Tagged files [21](#)

Definition [17](#)

Field names [37](#)

Input file [24](#)

Output format [41](#)

The Data Magician

copyright screen [9](#)

starting [9](#)

version [9](#)

Type

Input Fields [37](#)

Typeover mode [60](#)

Unrecognized field names [57](#)

Unspanned

MARC export format [48](#)

unspanned MARC files [31](#)

Up Arrow [59](#)

Upgrades [5](#)

UTLAS

to INMAGIC [198](#)

VGA [141](#)

Viewing output file [57](#)

Web site

<http://www.folland.com> [15](#)

Wildcards [67](#)

Changing [129](#), [130](#), [156](#)

in replace text [68](#)

Signal End of Record [26](#)

Signal Start of Record [26](#)

Symbols allowed [129](#), [130](#)

Windows

Icon file (DM.ICO) [6](#)

Usage notes [6](#)

Windows 3.x

Using The Data Magician with [6](#)

Windows 95

Using The Data Magician with [6](#)