

DataImport[®]

File Translation Utility

Version 6



User Guide

- ◆ Installation
- ◆ Introduction
- ◆ Tutorial
- ◆ Fitting DataImport to Your Needs
- ◆ Mask Reference
- ◆ Translate Reference
- ◆ Utilities Reference
- ◆ Task Commander Reference
- ◆ Appendix
- ◆ Index



Copyright

© 1986-2001 Spalding Software, Inc. All rights reserved. This manual and the software described in it are copyrighted with all rights reserved. This publication may be reproduced for educational purposes by licensed users only.

Trademarks

DataImport® is a registered trademark of Spalding Software Inc. Brand names and product names are trademarks or registered trademarks of their respective companies.

Printed in the USA

SPALDING SOFTWARE INC.
154 Technology Parkway
Suite 250
Norcross, GA 30092
770-449-0594
www.spaldingsoftware.com

Contents

Chapter 1: Installation	1
Single User Installation	1
Network Installation	3
How the Number of Users are Controlled	3
License and Registration Information	4
What's new in DataImport 6.0	5
Upgrading to DataImport 6.0	6
Technical Support	6
 Chapter 2: Introduction	 7
DataImport for Windows	7
Supported Output File Types	7
How It Works	8
Exploring DataImport	9
The DataImport Program Suite	9
DataImport Mask Window	10
 Chapter 3: Tutorial	 11
Running DataImport	11
Loading a File To Be Translated	12
Creating a Mask for Data Extraction	13
Choosing Data by Highlighting	14
Extracting Columns of Data	14
Specifying the Type of Data in a Column	18
Extracting Specific Lines of Data	19
Extracting Non-Columnar Data	21
Report Titles and Headings	23
Translating Data	26
Choosing an Output File Type	26
Running a Translation	27
Saving Masks for Reuse	28
Using the Output	28
 Chapter 4: Fitting DataImport to Your Needs	 31
Input and Output	31
Input Files	31
Output Files	33
Cleaning-Up Input Files	34
Special Characters	35
Blank Lines	35
Page Ejects	35

<i>Duplicate Lines</i>	35
Extracting Data	35
Columnar Data	36
Default Line Treatment	40
Titles and Headings	40
Reorganizing Data	41
Resequencing Data Columns	41
Unstacking Multiple Lines of Data	41
Getting Data from Multiple Lines into the Same Cell	43
Pulling Data out of Headers and Footers	43
Extracting Data from Forms	46
Filling Blank Column Cells	48
Transpose Rows and Columns	49
Recognizing Data Types and Formats	50
Numeric	50
Text	51
Date	52
Time of Day	52
Name Parse	53
Address Parse	53
Signed Overpunch Numbers	53
Code Page Settings	54
Performing Calculations	55
Formulas in Columns	55
Inserting Formula Rows	55
Working with Database Files	56

Chapter 5: DataImport Mask Reference 59

Running the Mask Application	59
File Menu	61
Search Menu	64
Column Menu	65
Tag Menu	69
Include Menu	75
Exclude Menu	77
Line Menu	80
Unstack Menu	82
Options Menu	83

Chapter 6: DataImport Translate Reference 95

Running the Translate Application	95
---	----

Chapter 7: DataImport Utilities Reference 97

Running the Utilities Application	97
Process Types	98
ASCII -> EBCDIC	99
Comma Separated Values	99
dBase convert	100
dBase header	100
EBCDIC -> ASCII	100
Fixed length	100

Line split by length.....	101
Parse spaces	101
Records per File Split	102
Statistics.....	102
Tab expansion	102
Unstack	103
Chapter 8: DataImport Task Commander Reference	105
Running the Task Commander Application.....	105
Appendix A: Supported Output File Formats	109
Output Formats.....	109
Output File Types.....	109
Output File List	110
ASCII (ASC).....	112
Alpha (DBF)	112
Clarion (DAT).....	112
Clipper (DBF)	112
Columnwise DIF (DIF).....	112
Comma Separated Value (CSV)	112
dBase II, III, IV (DBF)	112
Excel 2.1, 3.0, 4.0, 5.0, 7.0, 97, 2000, XP (XLS).....	113
Fixed length file (FXD)	113
FoxPro (DBF).....	113
HTML Tables (HTM).....	113
Lotus 1-2-3 1A, 2.0, 3.0, 4.0, 5.0 (WK*).....	113
Mailing Label (LBL)	114
Microsoft Access 1.1, 2.0, 3.0 (97), 4.0 (2000), XP (MDB)	114
Microsoft Word Merge File(WRD).....	114
Print Image (PRN).....	114
Quattro (WKQ)	114
Quattro Pro (WQ1).....	114
Quattro Pro 5.0 for Windows (WB1)	114
Standard Data Format (SDF).....	115
Sylk (SLK)	115
Symphony 1.0, 1.1 (WRK, WR1)	115
Tab Separated Variable (TSV).....	115
User-Defined Delimited (UDD).....	115
WordPerfect 5.0, 5.1 (W5*).....	116
xBase applications (DBF).....	116
NVL Named Value.....	116
XML Extensible Markup Language.....	116
Appendix B: Command Line Use	117
Translate Command Line.....	117
Translate Command Line Example 1.....	119
Translate Command Line Example 2.....	119
Translate Command Line Example 3.....	119
Translate Command Line Example 4.....	120
Utilities Command Line.....	120
Task Commander Command Line.....	124

Appendix C: Customizing the Dictionary File	125
Default Dictionary	125
Editing the Default.dic file	125
Appendix D: Frequently Asked Questions	127
DataImport Questions	127
Index	131

Chapter 1: Installation

This chapter describes how to install DataImport on a single computer or on a network and how to get technical support. This chapter also lists the new features in this version and provides information about upgrading from previous versions.

DataImport requires a PC running Windows 95, 98, NT, 2000, XP with a minimum of 64MB RAM available and 32MB of hard disk space. Both single user and multi-user versions of DataImport can be run from a network (LAN) server.

Single User Installation

To use DataImport, you must first install the program on your hard drive using the supplied installation program called SETUP. This program walks you through the installation procedure by asking you where you want to install the program files, copying the program files to your hard drive and creating a new program group.

To begin the installation, insert the DataImport CD. The DataImport CD should auto-run; however, if auto-run has been disabled select **Start > Run**. In the Run Dialog box type **X:\Setup.exe** (X: being the drive letter of your CD-ROM) and click OK.

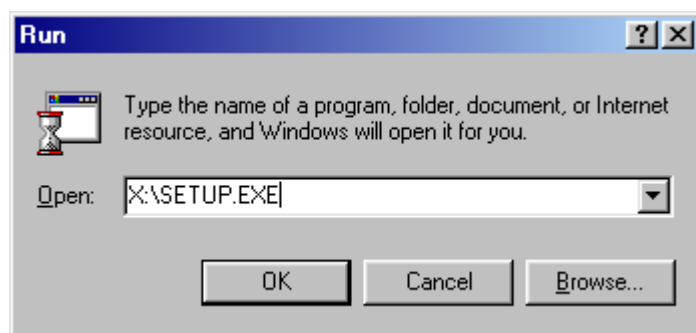


Figure 1-1 Running the Setup program for DataImport

The DataImport for Windows Setup screen appears. Follow the on-screen instructions.

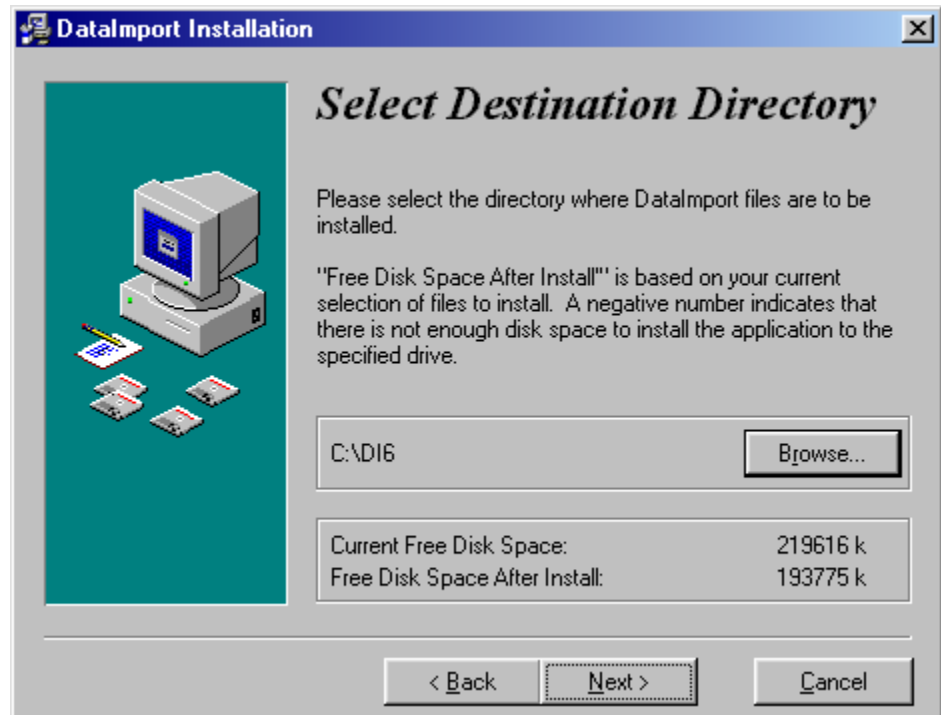


Figure 1-2 Choosing the destination directory for Datalport

To accept the default directory and install Datalport, click Next. If you want to install Datalport to a different directory, type in the new directory and click Next.

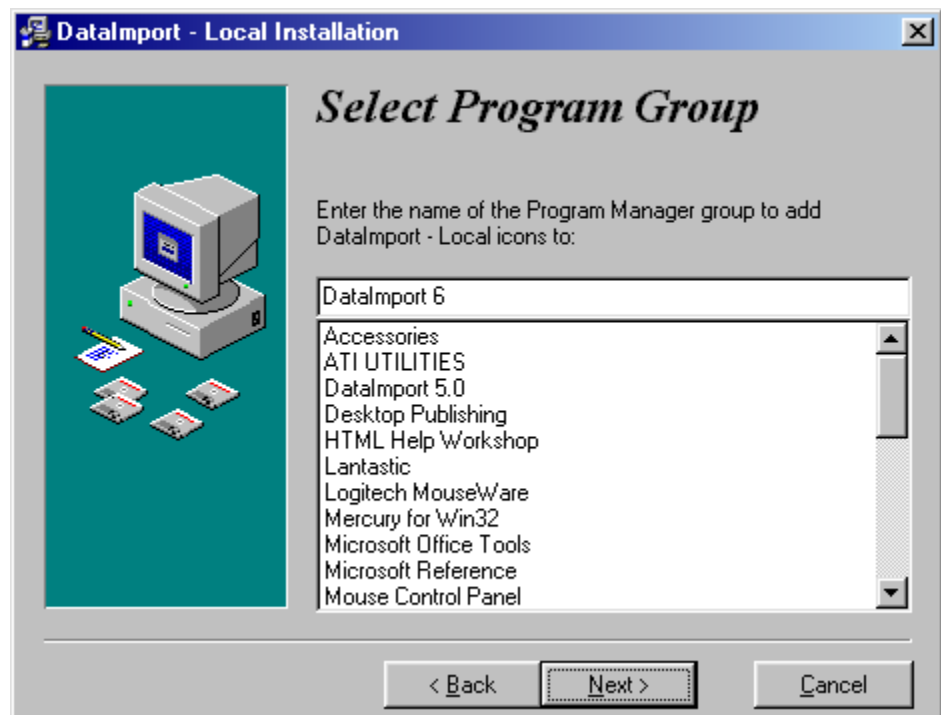


Figure 1-4 Select Program Group

By default, the program group DataImport 6 will be created. You can change the name or direct the installation to a different program group, otherwise click Next.

Follow the on-screen instructions to complete installing DataImport.

NOTE: The Setup program writes a log of the installation process called INSTALL.LOG to the directory where DataImport is located. This log lists what files were copied to your hard disk and where the files are located.

If installing a fully licensed version, see the section later in this chapter titled **License and Registration Information**.

Network Installation

***Note:** Network installation must be done from a workstation not on the server itself.*

Both the single user version and the multi-user version of DataImport can be run on a network. The multi-user version of DataImport will allow multiple users to share a single copy of the program files.

Both versions of the software automatically keep track of the number of concurrent users, and will reject users if the maximum number of licensed users is exceeded. As soon as a user exits from the software, a slot is made available for another user.

Installation on a network is similar to installation on a stand alone PC. Use the Setup program located on the CD-ROM to install the software to the server. The network installation must be done from a workstation not on the server itself. The procedure is to run setup from the CD-ROM installing the program files to a network drive. Then install DataImport on the workstation using the setup on the network drive, not the CD.

If the setup program detects that you are installing the program onto a remote network drive on a server, it will copy all the files from the installation disk onto the server. When installing to a network you will have the option of installing to the network drive, or installing to the network drive and installing the software on the workstation at the same time.

To install DataImport on a workstation run the Setup program on the server. This will install any necessary .OCX and .DLL files to the workstations and will create a Program Group pointing to the server's programs. A DataImport INI file will be created in the workstation's local Windows directory the first time the user runs the program.

Do not set the READ-ONLY attribute of the following files on the server: DIWNUMUS.EXE and DIWLOCK.EXE. Users must have Read and Write access to these files. If these files do not exist, they will be created the next time the software is run. The other .EXE files can be protected by setting their READ ONLY attributes.

If installing a fully licensed version, see the section later in this chapter titled **License and Registration Information**.

How the Number of Users are Controlled

When the user runs DataImport from a workstation, the application checks the number of users currently accessing the program. If the execution request does not exceed the maximum number of concurrent users, then the application will load as usual. If the request exceeds the maximum number of users, the following message appears:

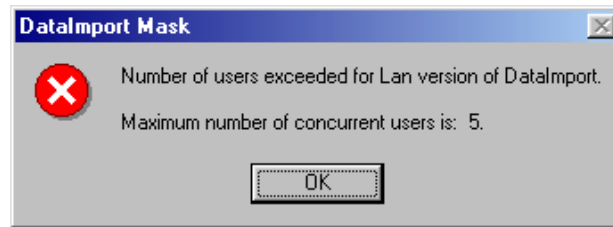


Figure 1-6 Network version error message when exceeding maximum users

While accessing files on the network, Input Files can be shared by multiple users. However, Output Files are locked whenever a user is translating and Mask Files are locked whenever a user is saving the mask.

If the software is trying to access a file, and permission is denied due to the file being locked by another user, the software retries several times at different time intervals before informing the user that the file is locked. The user can at that time try to access the file again, or can wait and access the file later.

License and Registration Information

When DataImport is first installed, it is installed as a test drive version. The software must be unlocked using the serial number and license key supplied with the DataImport software package. The serial number and license key can be found on the envelope containing the DataImport disk.

To unlock the software, after installing, run the DataImport program by clicking on the DataImport icon. The DataImport Menu Panel will be displayed.

Click Register Now to unlock the full power of DataImport

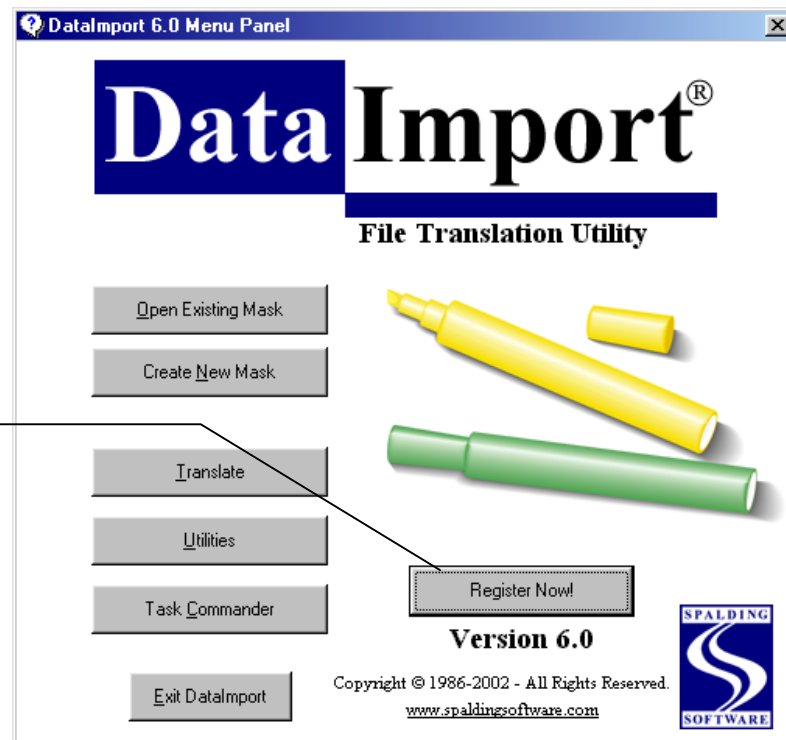


Figure 1-6 The DataImport Menu Panel

Click on the Register Now button to open the License Information dialog box.

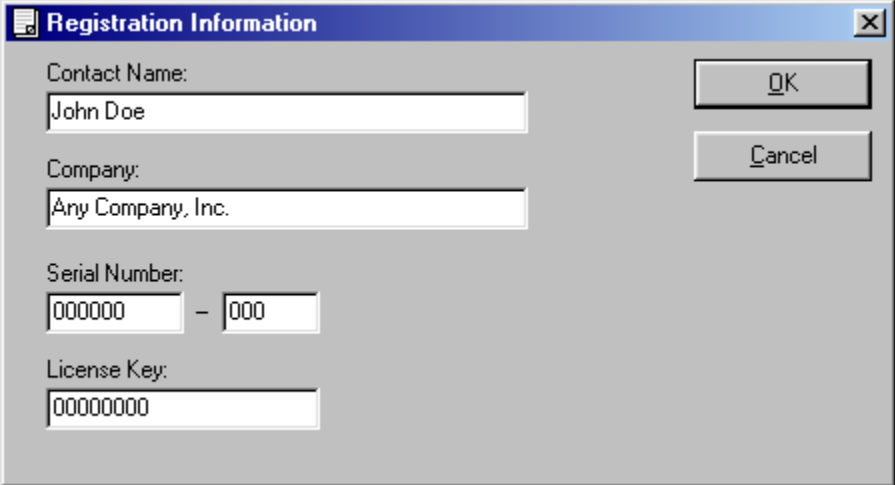
A screenshot of a Windows-style dialog box titled "Registration Information". It contains four input fields: "Contact Name:" with the text "John Doe", "Company:" with the text "Any Company, Inc.", "Serial Number:" with two sub-fields containing "000000" and "000" separated by a hyphen, and "License Key:" with the text "00000000". On the right side of the dialog are two buttons: "OK" and "Cancel".

Figure 1-7 The DataImport License Information Dialog Box

Enter your user information in the appropriate boxes. DataImport is now ready to use.

Be sure to send in your registration card so we can contact you about future upgrades to the software.

What's new in DataImport 6.0

The following is a list of new features and improvements in DataImport 6.0:

- A convenient start up Menu Panel allows instant access to all DataImport applications.
- Attractive colors indicating the data selected for translation that imitate the most common highlighting markers.
- Translates to all versions of Microsoft Excel and Access through XP.
- Translates to XML Tables.
- Footer Reference Points that allow pulling data up from a summary line at the bottom of a report to the detail lines above it.
- Line tags can occur before their associated reference point.
- The Mask window can now display the first 65,536 lines of a file.
Note: that if the output file type supports more than 65,536 lines, all of the lines will be translated.
- Preview a translation while in the mask application to quickly visually verify that the desired data is being extracted.
- Additional buttons have been added to the toolbar for quicker access to common features.

Upgrading to DataImport 6.0

- **Search > Replace** has been added to the search menu; it is the equivalent of **Exclude > Characters > Define** in previous versions. Either menu selection can be made in this version.
- Masks from versions 4.0 and 5.0 can be opened in version 6.0; however, once a mask has been saved in version 6.0 it can not be opened in version 4.0 or 5.0.

Technical Support

If you have problems with installation and use of the program, please call the support phone number on the first page of this manual. Before calling for support:

- From the mask application menu select **File > View Mask Settings**. Review this page to show all of the selections you have made. Often times you will discover the problem. If not, have this in hand when you call.
- Try to duplicate the problem, step by step, to see exactly what happened and when the problem occurred.
- Know your version of Windows.
- Know your DataImport Version (This can be found in the Help About dialog box.)
- Be at your computer when you call. Have your manual and your DataImport serial number handy.

Chapter 2: Introduction

This chapter introduces you to the many benefits of using DataImport and briefly describes the program's primary features. It also supplies answers to commonly asked questions about DataImport and its capabilities.

DataImport for Windows

The data files used by PC software products such as spreadsheets and databases have special file formats that are unique to each product. The files are encoded in a way that contains not only the data, but descriptions concerning the arrangement and use of the data.

Some PC software products provide importing capabilities for data files that are not in their special format. However, these capabilities are often so limited that they are not of practical use, especially in cases where the file is not specifically intended for use in another application such as a spreadsheet or database.

DataImport extracts data from plain text reports saved to file and translate it into the specialized file formats of spreadsheets and databases, as well as many other PC file formats. The reports might have come from an application on the PC or from a mainframe computer. With DataImport you can convert these reports into useful file formats such as Excel, Lotus 1-2-3, Access, HTML Tables, or dBase files.

Supported Output File Types

The following table is a listing of the formats that DataImport can create. Be sure to check the README file for last minute additions.

DataImport Translation Formats

ASCII (ASC)	Named Values (NVL)
Clarion (DAT)	Print Image (PRN)
Columnwise DIF (DIF)	Quattro (WKQ)
Comma Separated Variable (CSV)	Quattro Pro (WQ1)
dBase II, III, IV (DBF)	Quattro Pro 5.0 for Windows (WB1)
Excel 2.1, 3.0, 4.0, 5.0, 95, 97, 2000, XP (XLS)	Standard Data Format (SDF)
Fixed length file (FXD)	Sylk (SLK)
HTML Tables (HTM)	Symphony 1.0, 1.1 (WRK, WR1)
Lotus 1-2-3 1A, 2.0, 3.0, 4.0, 5.0 (WK*)	Tab Separated Variable (TSV)
Mailing Label (LBL)	User-Defined Delimited (UDD)
Microsoft Access 1.1, 95, 97, 2000, XP (MDB)	WordPerfect Merge File (W5*)
Microsoft Word Merge File (WRD)	XML 1.0

Figure 2-1 Output translation capabilities of DataImport.

How It Works

DataImport's visual interface displays your original file in a window. This file is called the input file. You simply point at the rows and columns you want to import and mark the rows or columns using "point-and-pick" operations like those in Excel and other programs. You select the portions of a file you want to translate. No special knowledge of file structures or command languages is needed to use DataImport.

DataImport "translates" the original data you select into the file formats required by your target application software, such as WKS, HTM, WR1, WK3, WKQ, XLS, or DBF. The file created by DataImport is called the output file.

Your selections and instructions for translating the data from a report are saved in a "mask" file that can be reused later. Using "masks" saves you time, particularly if you need to prepare the same, or similar, reports on a regular basis. You can even automate multi-step translations by using DataImport's Task Commander or incorporating them into a batch file.

DataImport does more than simply extract data from one file and put it into another. It distinguishes between numeric and non-numeric information, and handles both appropriately. When DataImport creates a spreadsheet file, it automatically sets up the numbers to include commas, currency symbols, and percent signs. It handles names, numbers, dates, and times.

Exploring DataImport

This section provides a quick introduction to DataImport, including the DataImport program group and the DataImport Mask application window.

The DataImport Program Suite

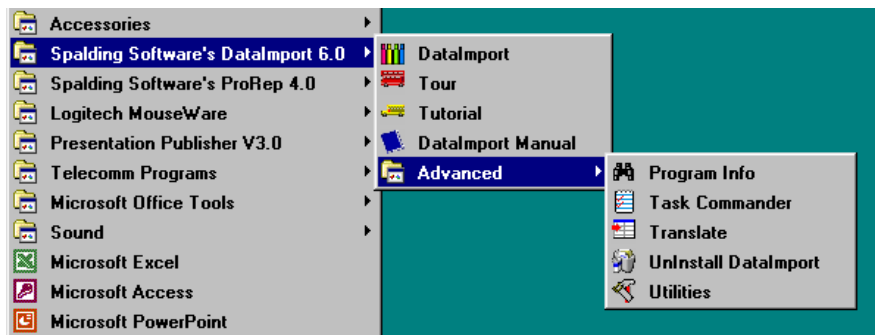


Figure 2-2 DataImport Program Suite

The DataImport Setup program creates this program suite on the Windows Programs Menu.

DataImport	This application encompasses the main features of DataImport. This program is where you create masks and translate files.
Tour	Provides a brief overview of the functionality of DataImport.
Tutorial	A step by step guide for using DataImport.
Manual	A complete user's guide to the DataImport product.
Translate	This application lets you directly access the translation engine of DataImport. Once you have defined a mask, you can quickly re-use masks with this application.
Utilities	This application provides special use utilities for reformatting and reorganizing Input Files.
Task Commander	This application automates the DataImport translate and utilities processes, allowing you to write scripts for repetitive multi-step translation jobs.

DatImport Mask Window

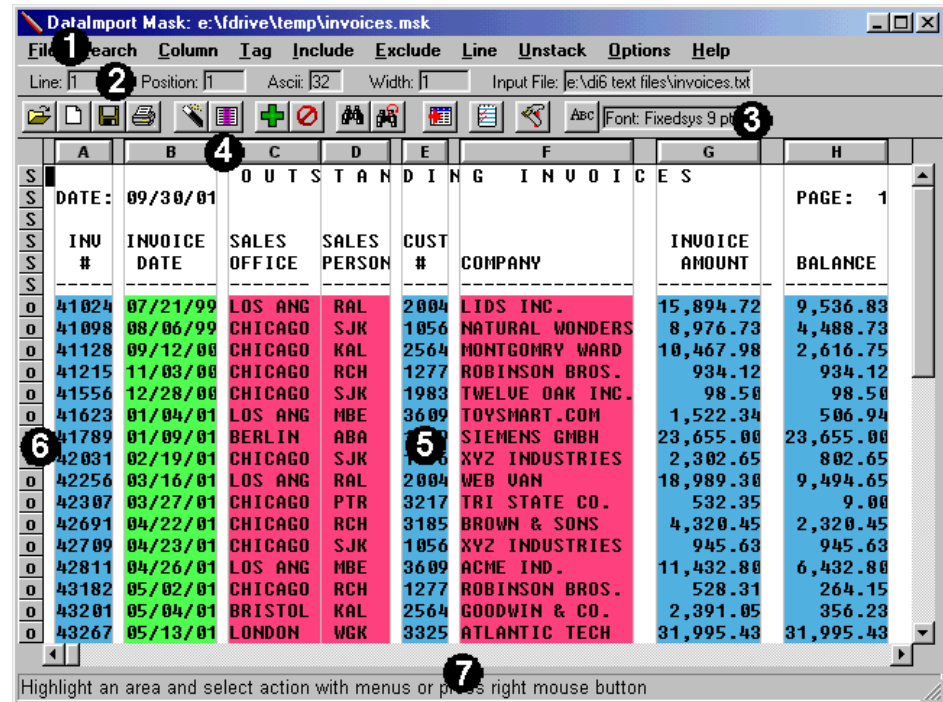


Figure 2-3 DatImport Mask Application

- Menu Bar** This section of the Mask window provides access to menu commands, just like other Windows applications.
- Information Bar** This section of the window shows the current line and character position of the cursor, the ASCII code of the selected character, the width of the current selection and the name of the Input File.
- Button Bar** This section contains buttons for several of the more frequently used actions, including Save Mask, Translate, and Column Define.
- Column Control Bar** This section controls the definition of columns. Drag the cursor across the Control Bar to create a column. Press a column button to change its settings. Drag the left or right edge of a column button to change its size.
- Input File window** This section is a scrolling text window that displays the current Input File and any mask definitions.
- Line Control Bar** This gray section controls the treatment of lines. Click or drag the cursor over the buttons of the Line Control Bar to change their definition.
- Prompt Line** This bar prompts you with information on how to use DataImport.

The next chapter gives a quick example of how you use DataImport to get you up and running quickly.

Chapter 3: Tutorial

This chapter covers the basics of using DataImport to get you up and running as quickly and productively as possible. The chapter shows you how to load a file, select the data you want to extract, select an output format and run a translation to extract your data.

The first step in using DataImport to translate a file is to define a “mask” for your data file that tells DataImport what information you want to extract and how you want it extracted. The masking process takes place while your data file or report (Input File) is displayed in a window.

As you select the data from your report, DataImport confirms your selections by displaying the selected portions in different colors. The colors of the masked data indicate how it will be extracted. For example, data with a blue background will be translated as numbers, data with a pink background will be translated as text (character data) and data with a green background will be translated as dates.

After the Mask is defined, it can be immediately used to perform a translation and extract your data to a spreadsheet or database file. The mask can also be saved for repeated translations of a similar report.

Running DataImport

To begin the mask definition process, you must first run the DataImport Mask application. From the desktop, double click on the **DataImport icon** as shown below.



Figure 3-1 DataImport Icon

The DataImport menu panel is displayed.

When using a new file for the first time, click the Create New Mask button.

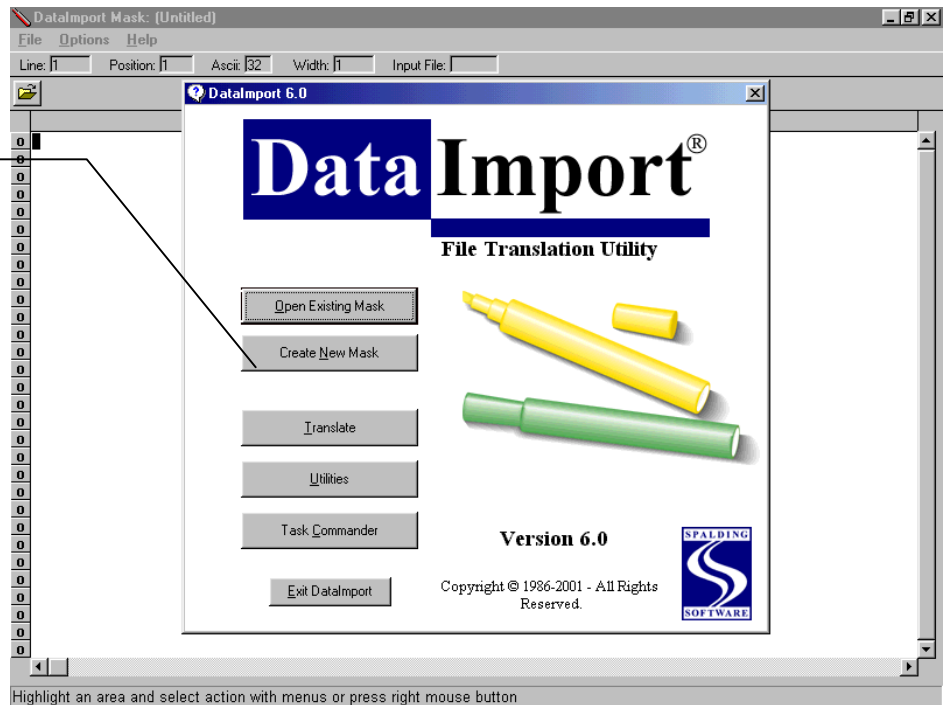


Figure 3-2 DataImport Menu Panel

Click the Create New Mask button to begin using DataImport on a new file.

Loading a File To Be Translated

The first step to defining a mask is to load a report, or Input File, into the Mask window. Let's start by loading a sample report named INVEST.PRN as our Input File:

Select **File > Load Input**. The following dialog box will appear:

The file INVEST.PRN is in the c:\DIW, or the directory where DataImport is installed.

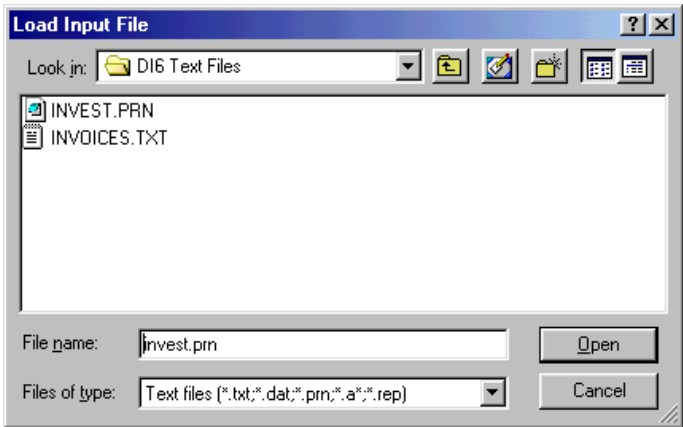


Figure 3-3 Load Input File dialog box

The Input File is loaded into the Mask window as shown below:

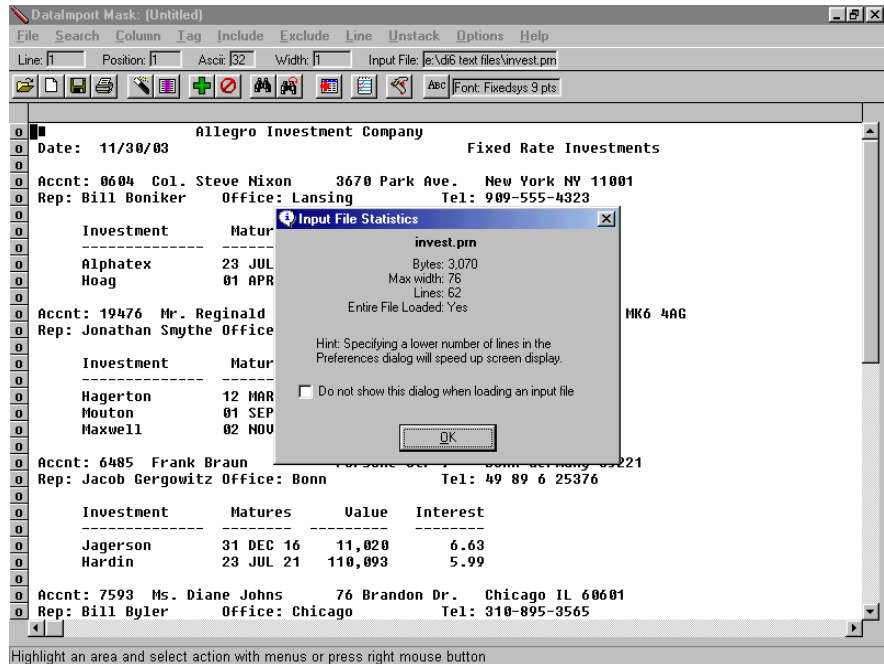


Figure 3-4 Load Input File as initially displayed in the Mask window.

Now we are ready to define the data we want to extract.

Creating a Mask for Data Extraction

Our example Input File is an Investment Report. The report is organized by Customer with a listing of each customer's investments.

Investor

Investor's
Investments

Allegro Investment Company				
Date:	11/30/83	Fixed Rate Investments		
Accnt:	0604 Col. Steve Nixon	3670 Park Ave.	New York NY 11001	
Rep:	Bill Boniker	Office: Lansing	Tel: 909-555-4323	
Investment	Matures	Value	Interest	
Alphatex	23 JUL 15	234,020	7.62	
Hoag	01 APR 09	10,090	5.91	
Accnt:	19476 Mr. Reginald Cook	221 B. Baker St	London England MK6 4AG	
Rep:	Jonathan Smythe	Office: London	Tel: 44 08 6 65051	
Investment	Matures	Value	Interest	
Hagerton	12 MAR 22	109,432	4.68	
Mouton	01 SEP 28	436,690	5.67	
Maxwell	02 NOV 35	36,690	6.90	
Accnt:	6485 Frank Braun	Porsche Str 9	Bonn Germany 85221	
Rep:	Jacob Gergowitz	Office: Bonn	Tel: 49 89 6 25376	
Investment	Matures	Value	Interest	
Jagerson	31 DEC 16	11,020	6.63	
Hardin	23 JUL 21	110,093	5.99	
Accnt:	7593 Ms. Diane Johns	76 Brandon Dr.	Chicago IL 60601	
Rep:	Bill Byler	Office: Chicago	Tel: 310-895-3565	

Figure 3-5 *Loaded Input File*

Let us assume that we want to put the investment data into a spreadsheet. In this spreadsheet, we would like each investment on a separate line. We also want the customer's name and phone number on the investment lines. Therefore, when we sort the investments by maturity date the customer contact information will also be on the same line. The following sections will show you how to create a mask that will extract just the data we want quickly and accurately.

Choosing Data by Highlighting

The primary type of tool used in the Mask window is a “highlighting marker” or highlighter. In DataImport, you use the highlighter to mark an example of the data you want to extract.

Highlighting, or selecting, is done by moving the cursor (highlighter) to the beginning of an area where an action is to take place, holding down the left mouse button, dragging the highlighter to the end of where the action is to take place and releasing the mouse button.

Extracting Columns of Data

The primary way of extracting data with DataImport is to define columns over the data you want to extract. There are several methods you can use to create columns. By way of example, we will introduce each of these techniques in this section.

Defining Columns Using the Menu Bar

The first column of data we want to extract is the Investment Name. Move the Highlighter over the first letter ‘A’ of the investment Alphatex.

Highlight the area on the line that contains the investment name. Press the left mouse button and drag the highlighter to the end of the area that can contain the investment names and release the button. Remember some names may be longer than Alphatex. Make sure to highlight a long enough area. Your screen should look like the one shown below:

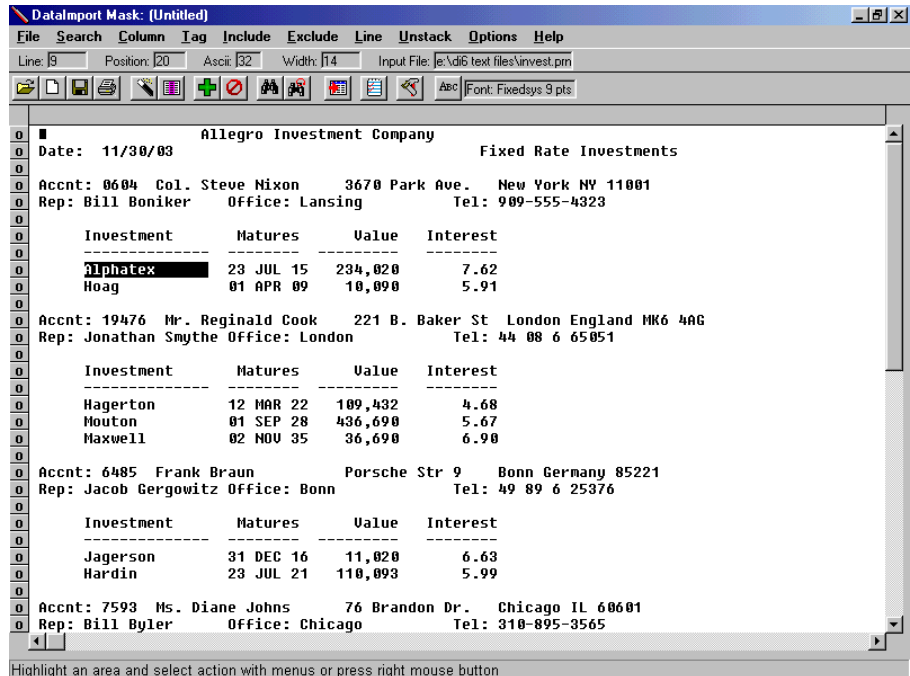


Figure 3-6 Defining a data column by highlighting the investment name

Now that one of the investments is highlighted, select **Column > Define**. The Input File is redisplayed with column A defined, and the Column Settings dialog box appears. Your screen should look like the one shown below:

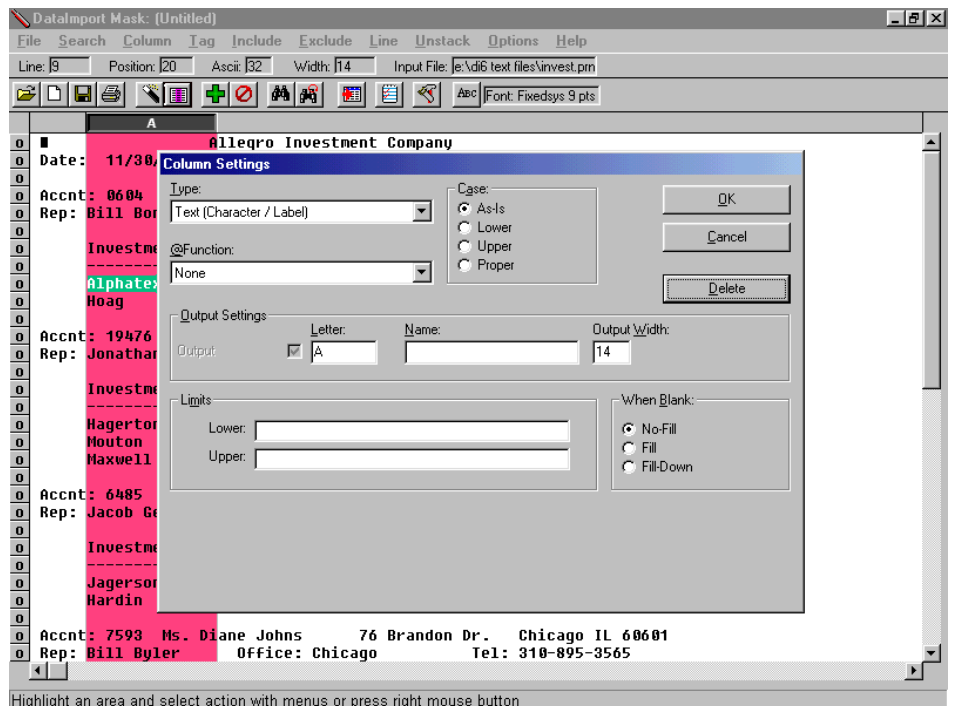


Figure 3-7 Column Settings dialog box is displayed while defining column A

This dialog box is used to define the settings of a column, such as its sequence and data type. Since Alphetex is text, select the Text (Character/Label) option.

Once you've defined the column, your screen should look like this:

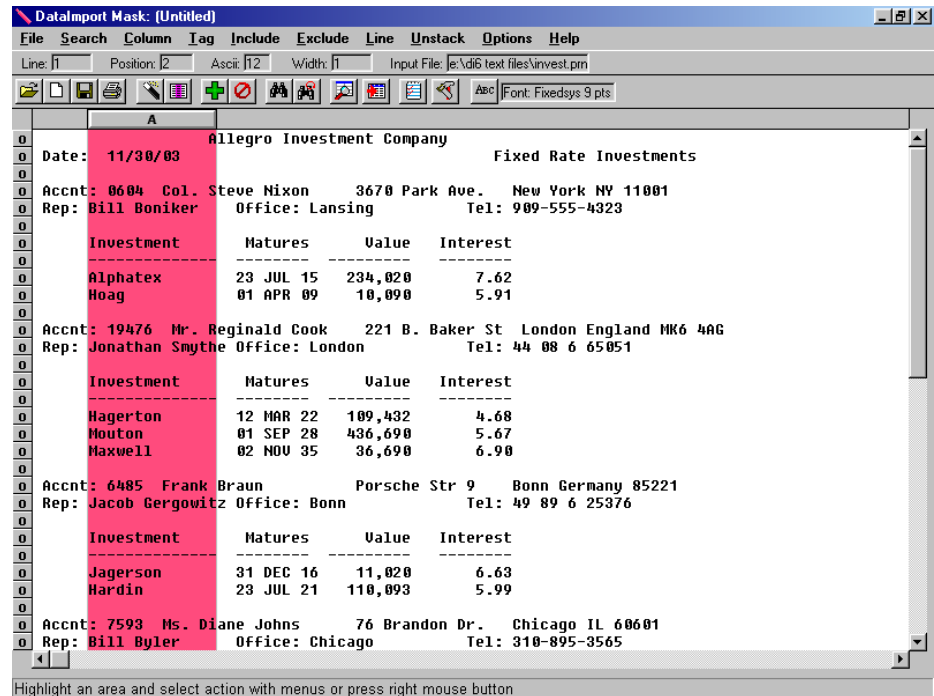


Figure 3-8 Mask window with column A defined

DataImport displays the data within the defined column with a background color. This coloring allows you to easily see what data will be extracted. For now, do not worry that the text on lines other than the detail investment lines are also highlighted.

Defining Columns Using the Popup Menu

The second column of data we want to extract is the date that the investment matures. Highlight any maturity date such as “23 JUL 15” on the Alphetex investment line. Make sure your highlighting does not start in the first column you created.

Click the right mouse button. A popup menu will appear as shown below:

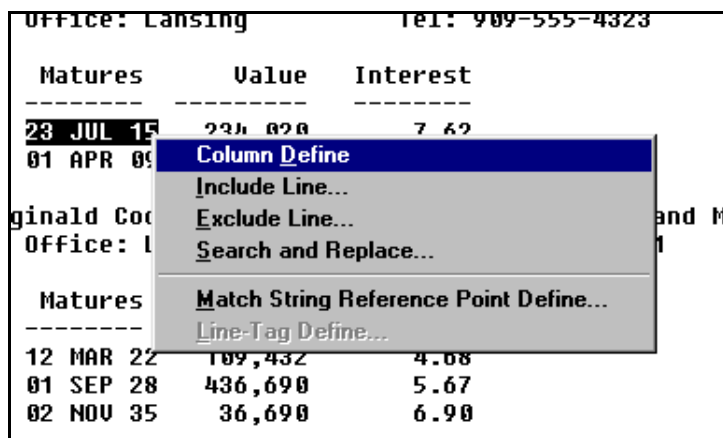


Figure 3-9 Highlighter popup menu

From the popup menu, select **Column Define**. The Input File is redisplayed with column B defined and the Column Settings dialog box displayed. For now, do not change any of the settings. Press OK to accept the current column settings.

Once you have defined the column, your screen should look like the one shown below:

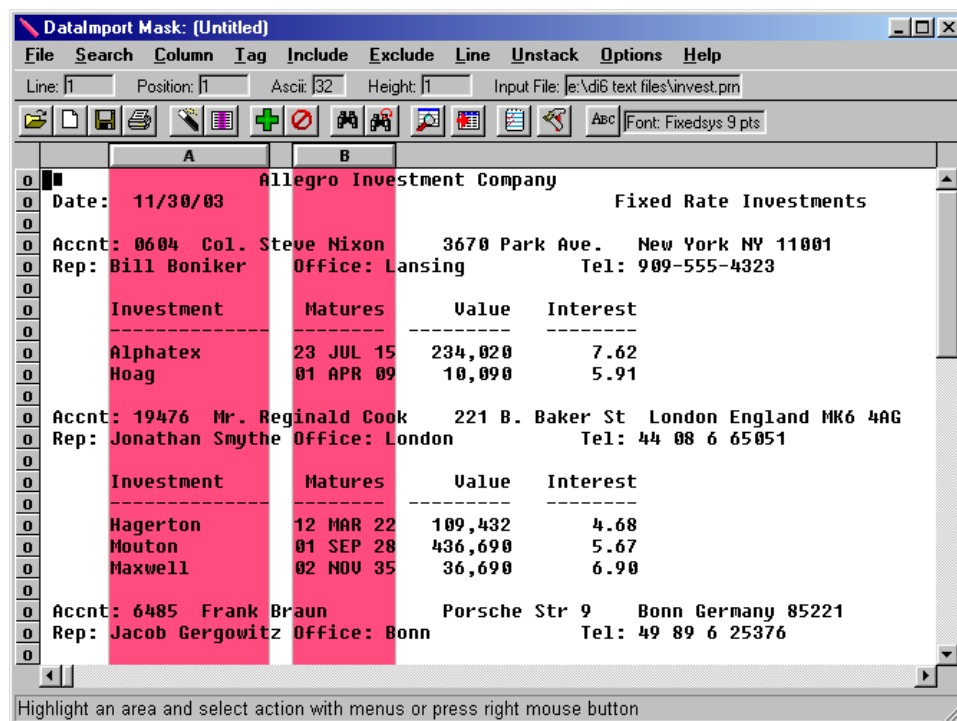


Figure 3-10 Mask window with columns A and B defined

Defining Columns Using the Column Control Bar

A quicker way to create a column is by using the Column Control Bar. This is the bar above the Input File window that shows the location of each column as a button labeled with the column letter.

The next column of data we want to extract is the Value column. Move the cursor into the Column Control Bar, this changes the cursor from a highlighter to a double-headed arrow. In the Column Control Bar, move the cursor to the beginning position of the Value column drag the cursor to the desired column width.

Column Control Bar.

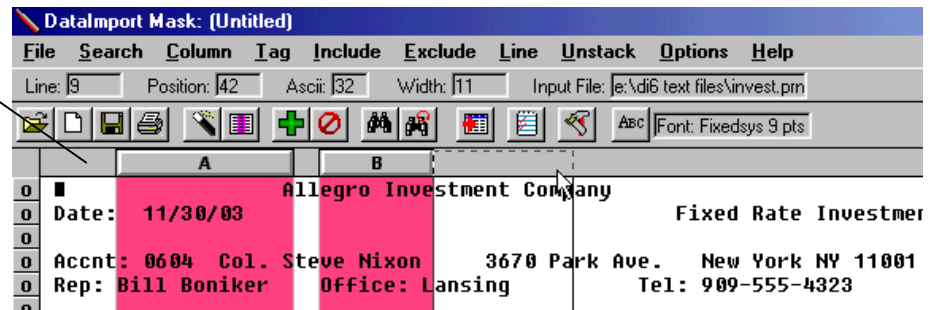


Figure 3-11 Using the Column Control Bar to define a column

Column C is now defined. Remember that only data with a background color other than white will be translated.

Now that you know how to define columns, use any of these methods to define a column for the interest rate. After creating all of these columns, your screen should look like this

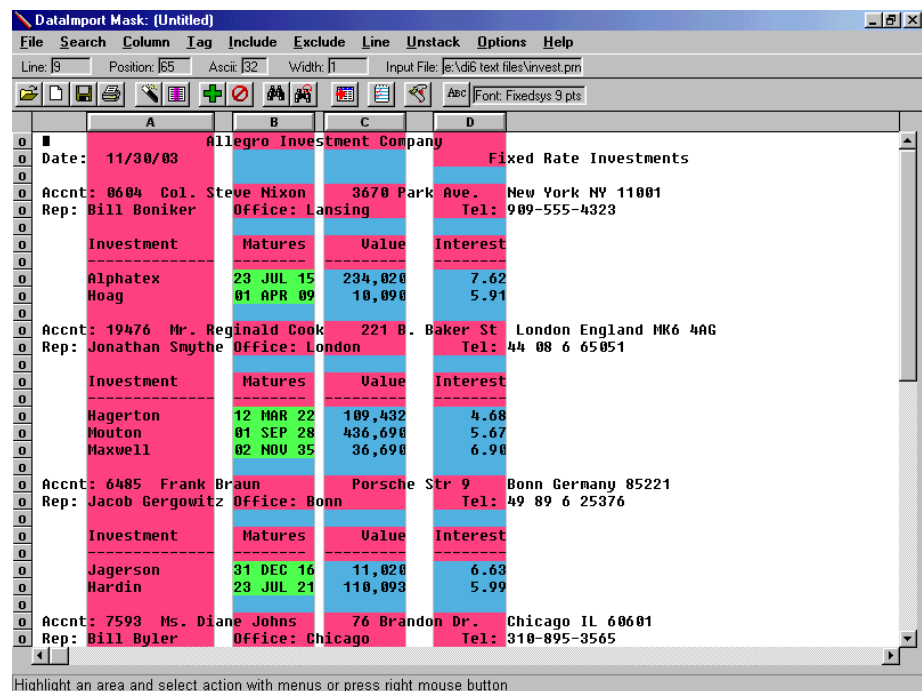


Figure 3-12 Mask window with all investment detail columns defined

Specifying the Type of Data in a Column

The colors of the masked data indicate how the data from the report will be translated: blue for numeric, pink for text, and green for dates. These colors indicate

what formatting will be applied to your data when it is extracted to a spreadsheet or database. Selecting the correct column type in DataImport will make your data easier to handle in your spreadsheet or database.

The maturity dates in column B of this report are currently displayed with a pink background. This coloring tells us the dates will be translated into a spreadsheet as text and not a date. Therefore, we would not be able to perform calculations in the spreadsheet with this data, such as calculating how long until the investment matures.

To change the column type, click the column's button and select the type of data from the pull down list labeled type. In the following example, the column type is being set as date.

Click on the column button to change the settings.

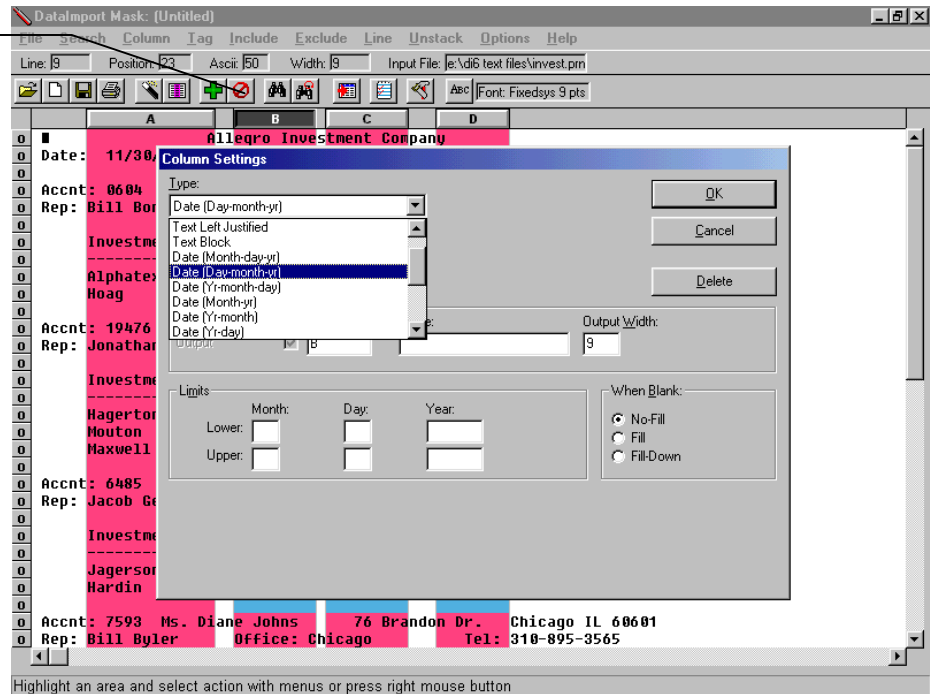


Figure 3-13 Changing column B's type to a Date format

The data in column B should be translated as dates, so change column B's type to Date. The data in column B that DataImport recognizes as dates is now displayed with a green background. The background color of data that cannot be recognized as dates is not changed.

Extracting Specific Lines of Data

At this point we have defined the columns of data on the investment detail lines. We now need to tell DataImport to extract only one line on this report for each investment detail line. Currently in our example report, every line on the report is selected for translation. There are two ways that we know this; there are background colors on each line within the defined columns, and a lowercase "o," for output, in the Line Control Bar at the left of each line.

There are many ways to select which lines or rows in the Input File are translated by DataImport. Usually lines are selected for output by either including or excluding lines that contain “matching” criteria.

We only want to include the investment lines, so we will find a common character or string of characters that is unique to these lines. One such common character is the decimal point in the Interest column. To include only those lines with the decimal point, highlight the decimal point, select **Include > Lines > Define**, and make the settings on the dialog box. When an include line match string is defined, all lines that do not include the match string are automatically changed to Skip lines.

To use “Wild Cards” for the include line match string, select one or more of these characters.

Select where the include line match string is located on the

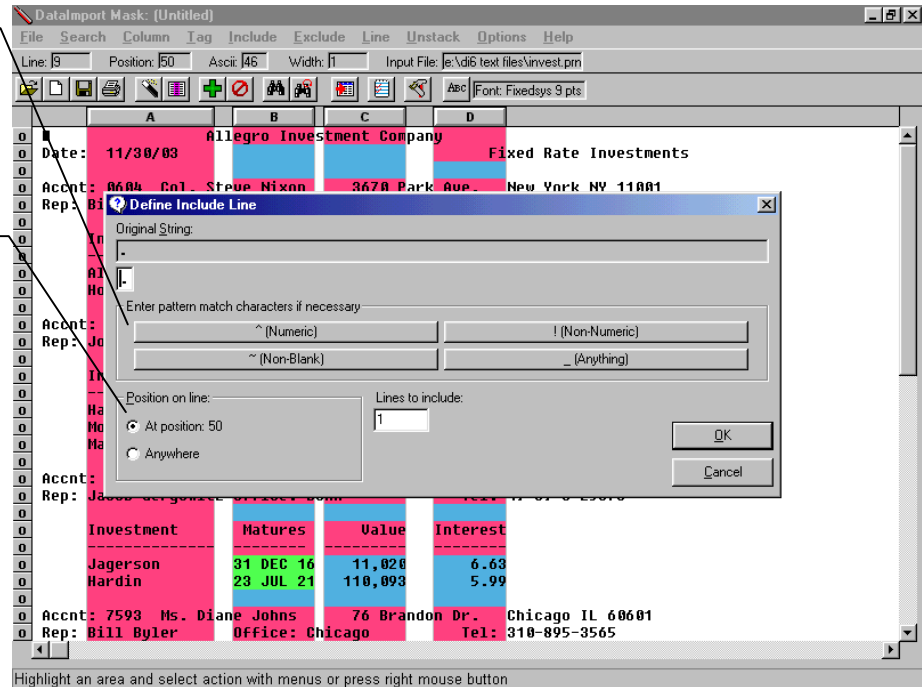


Figure 3-14 Include Line dialog box with “.” text string highlighted

Once you have defined the Include Line, your screen should look like the one below:

I is the include line indicator.

s is the skip line indicator.

The selected (included) investment lines contain a decimal (.) character at position 50.

	A	B	C	D
s	Allegro Investment Company			
s	Date: 11/30/83			Fixed Rate Investments
s	Acct: 0604 Col. Steve Nixon	3670 Park Ave.	New York NY 11001	
s	Rep: Bill Boniker	Office: Lansing	Tel: 909-555-4323	
s	Investment	Matures	Value	Interest
I	Alphatex	23 JUL 15	234,020	7.62
I	Hoag	01 APR 09	10,090	5.91
s	Acct: 19476 Mr. Reginald Cook	221 B. Baker St	London England MK6 4AG	
s	Rep: Jonathan Smythe	Office: London	Tel: 44 08 6 65051	
s	Investment	Matures	Value	Interest
I	Hagerton	12 MAR 22	109,432	4.68
I	Houton	01 SEP 28	436,690	5.67
I	Maxwell	02 NOV 35	36,690	6.90
s	Acct: 6485 Frank Braun	Porsche Str 9	Bonn Germany 85221	
s	Rep: Jacob Gergowitz	Office: Bonn	Tel: 49 89 6 25376	
s	Investment	Matures	Value	Interest
I	Jagerson	31 DEC 16	11,020	6.63
I	Hardin	23 JUL 21	110,093	5.99
s	Acct: 7593 Ms. Diane Johns	76 Brandon Dr.	Chicago IL 60601	
s	Rep: Bill Byler	Office: Chicago	Tel: 310-895-3565	

Figure 3-15 Lines containing a decimal point at position 50 are defined as Include Lines

Notice that only the lines with the decimal point at position 50 now have colored backgrounds within the defined columns. This coloring indicates which data will be translated to the Output File. Notice that these lines also have an uppercase **I** in the Line Control Bar on the left edge of the Input File window indicating that they are Include Lines.

All of the other lines on the report have lost their colored backgrounds, which means that they will not be translated to the Output File. These lines are automatically excluded from translation because DataImport automatically changed the default line treatment to “Skip” after we defined an Include Line match string. The application of the default Skip Line treatment is indicated by a lowercase **s** in the line control bar.

The Include Lines function is an easy way to pick only the data you want from an Input File, instead of sifting through the data in your target application. DataImport has another function called Exclude Lines which allows you to exclude specific data lines from translation while including all others. See Chapter 4: Fitting DataImport to Your Needs, “Extracting Data.”

Extracting Non-Columnar Data

As we mentioned earlier, we will also need the client’s name, phone number and location output on each investment line. This data is in a non-columnar arrangement above the sets of investment lines. In order to accomplish this, we will use Line Tags and Reference Points.

Defining Reference Points

A reference point is a positional anchor, or surveyor's mark, which DataImport uses to locate data that changes within a form or other non-columnar area on a report. One such anchor in our investment report is "Acctt:".

Highlight the letters "Acctt:" then select **Tag > Define Match String Reference Point**. A dialog box will appear.

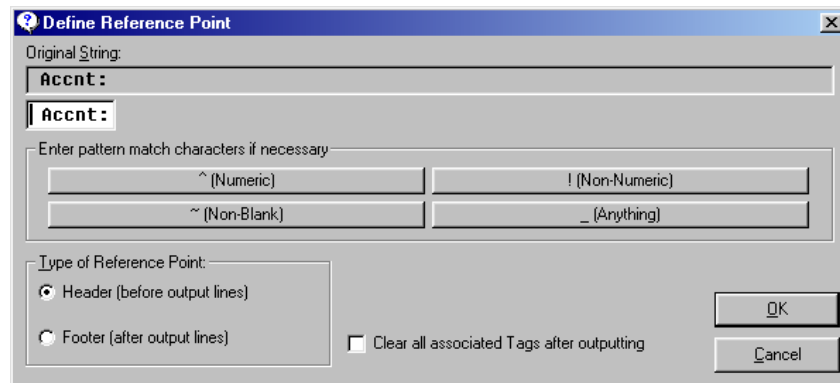


Figure 3-16 Define Reference Point Dialog Box

The reference point "Acctt:" should now be shown as black text with a yellow background. Now that we have our reference point we can begin to select information about the customer that we want output on each investment line.

Reference Point

Reference Point

Reference Point

Investment	Matures	Value	Interest
Alphonex	23 JUL 15	234,020	7.62
Hoag	01 APR 09	10,090	5.91
Hagerton	12 MAR 22	109,432	4.68
Mouton	01 SEP 28	436,690	5.67
Maxwell	02 NOV 35	36,690	6.90
Jagerson	31 DEC 16	11,020	6.63
Hardin	23 JUL 21	110,093	5.99

Figure 3-17 All occurrences of "Acctt:" defined at Reference Points.

Defining Line Tags

Line tags are the non-columnar data that we want to get into our spreadsheet. When translated, Line Tag information is output on each extracted line. For this report, we will be defining several tags: the client's name, city, state, zip, and telephone number.

Highlight the character string "Steve Nixon" and enough blank spaces to the right to select the longest name that will occur. Select **Line > Tag > Define**. In the Tag Settings dialog box, set the type to **Name Parse (first last)**. You will then see several checkboxes to indicate how you want the parts of the names handled.

Selected columns parse names into separate fields for first name and last name.

The Tag Settings dialog box is shown with the following settings:

- Type: Name Parse (Last, First)
- Case: As-Is (selected), Lower, Upper, Proper
- @Function: None
- Output Settings:

	Letter:	Name:	Output Width:
Prefix	<input type="checkbox"/>		19
First	<input checked="" type="checkbox"/> F		19
Middle	<input type="checkbox"/>		19
Last	<input checked="" type="checkbox"/> E		19
Suffix	<input type="checkbox"/>		19
- Reference Point: Accent
- Line-Tag occurs before Reference Point: ☐

Figure 3-18 Tag Settings Dialog Box with columns E (first name) and F (last name) defined.

*You can also click the right mouse button to display the shortcut menu and then select **Line Tag Define***

"Steve Nixon" and all other client names should now be shown as pink text with a yellow background. The client's first and last names will be output to columns E and F on each investment line. The next data to select is the City/State/Zip data. This is done in much the same way as names. Repeat these steps for the telephone number, and set the tag type to Text. When you are done, your screen should look like the one below.

	A	B	C	D
Date:	11/30/83	Allegro Investment Company		
Accnt:	0604 Col. Steve Nixon	3670 Park Ave.	New York NY 11001	
Rep:	Bill Boniker	Office: Lansing	Tel: 909-555-4323	
Investment	Matures	Value	Interest	
Alphatex	23 JUL 15	234,020	7.62	
Hoag	01 APR 09	10,090	5.91	
Accnt:	19476 Mr. Reginald Cook	221 B. Baker St.	London England MK6 4AG	
Rep:	Jonathan Smythe	Office: London	Tel: 44 08 6 65051	
Investment	Matures	Value	Interest	
Hagerton	12 MAR 22	109,432	4.68	
Mouton	01 SEP 28	436,690	5.67	
Maxwell	02 NOV 35	36,690	6.90	
Accnt:	6485 Frank Braun	Porsche Str 9	Bonn Germany 85221	
Rep:	Jacob Gergowitz	Office: Bonn	Tel: 49 89 6 25376	
Investment	Matures	Value	Interest	
Jagerson	31 DEC 16	11,020	6.63	
Hardin	23 JUL 21	110,093	5.99	
Accnt:	7593 Ms. Diane Johns	76 Brandon Dr.	Chicago IL 60601	
Rep:	Bill Byler	Office: Chicago	Tel: 310-895-3565	

Figure 3-19 Mask screen with all Line Tags defined.

During translation, each time the reference point “ACCNT:” is encountered, the data in the line tags for the investor’s name and address is updated in memory. When an include line is encountered, the data on the include line in columns A through D is written to a line in the output file along with the values contained in memory from the line tags.

Report Titles and Headings

Now that we have defined the data we want to extract, we also want to extract information that is not strictly data. For instance, we want the report title and column heading information from the report.

Currently, all lines in our report except those with the decimal point in the interest rate at position 50 have a Skip line treatment; these lines will not be output when a translation is performed. By defining special treatments—called Line Treatments—we can output the report title and column headings.

Extracting Report Titles

The title information from the top of the report should be translated into our Output File as entire lines; not just the parts of these lines that fall within the defined columns. To translate these lines as titles, define them as Title lines. Move the Highlighter over the Line Control Bar on the leftmost edge of the window. When the cursor is over the Line Control Bar it changes to a left pointing hand as shown below. Select lines 1 and 2 by clicking and dragging vertically on the Line Control Bar.

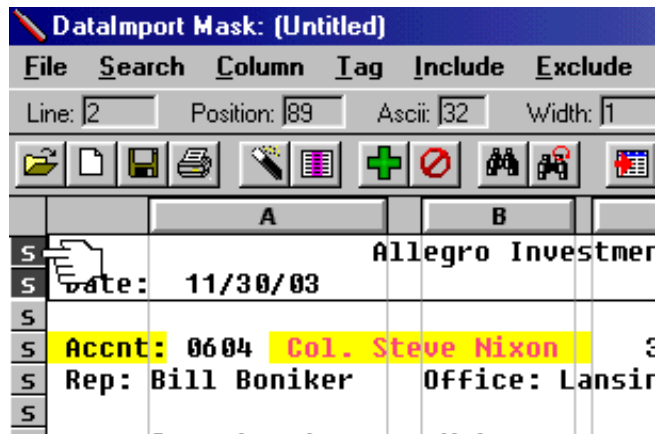


Figure 3-20 Selecting lines 1 and 2 using the Line Control Bar

After clicking, selecting, and releasing the line control buttons on the of the window, the Line Treatment pop up menu is displayed.

Line Treatment
pop up menu.

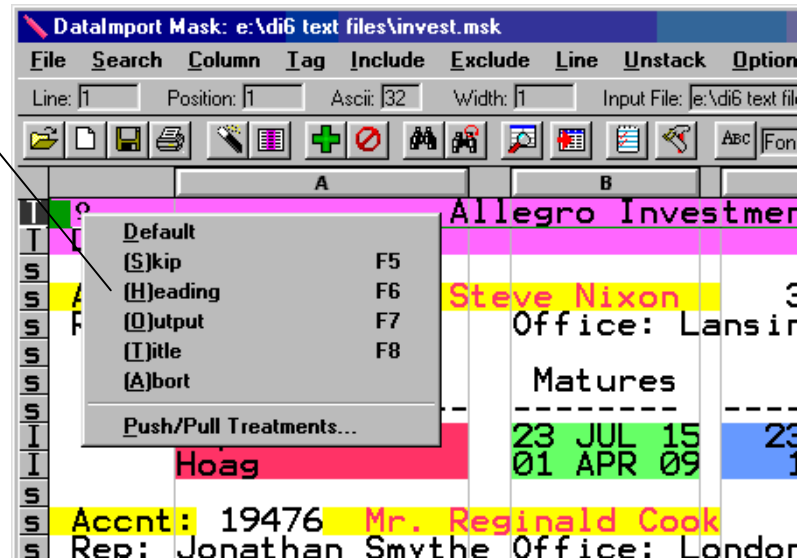


Figure 3-21 Selecting Title on the Line popup menu

From the **Line** popup menu, select **(T)itle**.

The Input File is redisplayed with the first two lines defined as Title as shown below:

Title lines are indicated with a "T"

Line	Position	Ascii	Width	Input File
1				File: Terminal 12 pts
2				Input File: e:\v66 text files\invest.pri
3				File: Terminal 12 pts
4				Input File: e:\v66 text files\invest.pri
5				File: Terminal 12 pts
6				Input File: e:\v66 text files\invest.pri
7				File: Terminal 12 pts
8				Input File: e:\v66 text files\invest.pri
9				File: Terminal 12 pts
10				Input File: e:\v66 text files\invest.pri
11				File: Terminal 12 pts
12				Input File: e:\v66 text files\invest.pri
13				File: Terminal 12 pts
14				Input File: e:\v66 text files\invest.pri
15				File: Terminal 12 pts
16				Input File: e:\v66 text files\invest.pri
17				File: Terminal 12 pts
18				Input File: e:\v66 text files\invest.pri
19				File: Terminal 12 pts
20				Input File: e:\v66 text files\invest.pri
21				File: Terminal 12 pts
22				Input File: e:\v66 text files\invest.pri
23				File: Terminal 12 pts
24				Input File: e:\v66 text files\invest.pri
25				File: Terminal 12 pts
26				Input File: e:\v66 text files\invest.pri
27				File: Terminal 12 pts
28				Input File: e:\v66 text files\invest.pri
29				File: Terminal 12 pts
30				Input File: e:\v66 text files\invest.pri
31				File: Terminal 12 pts
32				Input File: e:\v66 text files\invest.pri
33				File: Terminal 12 pts
34				Input File: e:\v66 text files\invest.pri
35				File: Terminal 12 pts
36				Input File: e:\v66 text files\invest.pri
37				File: Terminal 12 pts
38				Input File: e:\v66 text files\invest.pri
39				File: Terminal 12 pts
40				Input File: e:\v66 text files\invest.pri
41				File: Terminal 12 pts
42				Input File: e:\v66 text files\invest.pri
43				File: Terminal 12 pts
44				Input File: e:\v66 text files\invest.pri
45				File: Terminal 12 pts
46				Input File: e:\v66 text files\invest.pri
47				File: Terminal 12 pts
48				Input File: e:\v66 text files\invest.pri
49				File: Terminal 12 pts
50				Input File: e:\v66 text files\invest.pri
51				File: Terminal 12 pts
52				Input File: e:\v66 text files\invest.pri
53				File: Terminal 12 pts
54				Input File: e:\v66 text files\invest.pri
55				File: Terminal 12 pts
56				Input File: e:\v66 text files\invest.pri
57				File: Terminal 12 pts
58				Input File: e:\v66 text files\invest.pri
59				File: Terminal 12 pts
60				Input File: e:\v66 text files\invest.pri
61				File: Terminal 12 pts
62				Input File: e:\v66 text files\invest.pri
63				File: Terminal 12 pts
64				Input File: e:\v66 text files\invest.pri
65				File: Terminal 12 pts
66				Input File: e:\v66 text files\invest.pri
67				File: Terminal 12 pts
68				Input File: e:\v66 text files\invest.pri
69				File: Terminal 12 pts
70				Input File: e:\v66 text files\invest.pri
71				File: Terminal 12 pts
72				Input File: e:\v66 text files\invest.pri
73				File: Terminal 12 pts
74				Input File: e:\v66 text files\invest.pri
75				File: Terminal 12 pts
76				Input File: e:\v66 text files\invest.pri
77				File: Terminal 12 pts
78				Input File: e:\v66 text files\invest.pri
79				File: Terminal 12 pts
80				Input File: e:\v66 text files\invest.pri
81				File: Terminal 12 pts
82				Input File: e:\v66 text files\invest.pri
83				File: Terminal 12 pts
84				Input File: e:\v66 text files\invest.pri
85				File: Terminal 12 pts
86				Input File: e:\v66 text files\invest.pri
87				File: Terminal 12 pts
88				Input File: e:\v66 text files\invest.pri
89				File: Terminal 12 pts
90				Input File: e:\v66 text files\invest.pri
91				File: Terminal 12 pts
92				Input File: e:\v66 text files\invest.pri
93				File: Terminal 12 pts
94				Input File: e:\v66 text files\invest.pri
95				File: Terminal 12 pts
96				Input File: e:\v66 text files\invest.pri
97				File: Terminal 12 pts
98				Input File: e:\v66 text files\invest.pri
99				File: Terminal 12 pts
100				Input File: e:\v66 text files\invest.pri

Figure 3-22 Lines 1 & 2 defined as Title

Title Lines are displayed with a red background color, and an uppercase T on the Line Control Bar. When translated into a spreadsheet, Title Lines are output into the first column (usually column A) as a single long label that contains the entire line from the report.

Extracting Column Heading Information

We also want the column headings on lines 7 and 8 to be translated into our Output file one time. Use the Line Control Bar to apply Heading treatments to the first occurrence of these lines.

The Input File is redisplayed with lines 7 and 8 defined as Heading Lines.

Column heading lines are indicated with a "H"

Line	Position	Ascii	Width	Input File
1				File: Terminal 12 pts
2				Input File: e:\v66 text files\invest.pri
3				File: Terminal 12 pts
4				Input File: e:\v66 text files\invest.pri
5				File: Terminal 12 pts
6				Input File: e:\v66 text files\invest.pri
7				File: Terminal 12 pts
8				Input File: e:\v66 text files\invest.pri
9				File: Terminal 12 pts
10				Input File: e:\v66 text files\invest.pri
11				File: Terminal 12 pts
12				Input File: e:\v66 text files\invest.pri
13				File: Terminal 12 pts
14				Input File: e:\v66 text files\invest.pri
15				File: Terminal 12 pts
16				Input File: e:\v66 text files\invest.pri
17				File: Terminal 12 pts
18				Input File: e:\v66 text files\invest.pri
19				File: Terminal 12 pts
20				Input File: e:\v66 text files\invest.pri
21				File: Terminal 12 pts
22				Input File: e:\v66 text files\invest.pri
23				File: Terminal 12 pts
24				Input File: e:\v66 text files\invest.pri
25				File: Terminal 12 pts
26				Input File: e:\v66 text files\invest.pri
27				File: Terminal 12 pts
28				Input File: e:\v66 text files\invest.pri
29				File: Terminal 12 pts
30				Input File: e:\v66 text files\invest.pri
31				File: Terminal 12 pts
32				Input File: e:\v66 text files\invest.pri
33				File: Terminal 12 pts
34				Input File: e:\v66 text files\invest.pri
35				File: Terminal 12 pts
36				Input File: e:\v66 text files\invest.pri
37				File: Terminal 12 pts
38				Input File: e:\v66 text files\invest.pri
39				File: Terminal 12 pts
40				Input File: e:\v66 text files\invest.pri
41				File: Terminal 12 pts
42				Input File: e:\v66 text files\invest.pri
43				File: Terminal 12 pts
44				Input File: e:\v66 text files\invest.pri
45				File: Terminal 12 pts
46				Input File: e:\v66 text files\invest.pri
47				File: Terminal 12 pts
48				Input File: e:\v66 text files\invest.pri
49				File: Terminal 12 pts
50				Input File: e:\v66 text files\invest.pri
51				File: Terminal 12 pts
52				Input File: e:\v66 text files\invest.pri
53				File: Terminal 12 pts
54				Input File: e:\v66 text files\invest.pri
55				File: Terminal 12 pts
56				Input File: e:\v66 text files\invest.pri
57				File: Terminal 12 pts
58				Input File: e:\v66 text files\invest.pri
59				File: Terminal 12 pts
60				Input File: e:\v66 text files\invest.pri
61				File: Terminal 12 pts
62				Input File: e:\v66 text files\invest.pri
63				File: Terminal 12 pts
64				Input File: e:\v66 text files\invest.pri
65				File: Terminal 12 pts
66				Input File: e:\v66 text files\invest.pri
67				File: Terminal 12 pts
68				Input File: e:\v66 text files\invest.pri
69				File: Terminal 12 pts
70				Input File: e:\v66 text files\invest.pri
71				File: Terminal 12 pts
72				Input File: e:\v66 text files\invest.pri
73				File: Terminal 12 pts
74				Input File: e:\v66 text files\invest.pri
75				File: Terminal 12 pts
76				Input File: e:\v66 text files\invest.pri
77				File: Terminal 12 pts
78				Input File: e:\v66 text files\invest.pri
79				File: Terminal 12 pts
80				Input File: e:\v66 text files\invest.pri
81				File: Terminal 12 pts
82				Input File: e:\v66 text files\invest.pri
83				File: Terminal 12 pts
84				Input File: e:\v66 text files\invest.pri
85				File: Terminal 12 pts
86				Input File: e:\v66 text files\invest.pri
87				File: Terminal 12 pts
88				Input File: e:\v66 text files\invest.pri
89				File: Terminal 12 pts
90				Input File: e:\v66 text files\invest.pri
91				File: Terminal 12 pts
92				Input File: e:\v66 text files\invest.pri
93				File: Terminal 12 pts
94				Input File: e:\v66 text files\invest.pri
95				File: Terminal 12 pts
96				Input File: e:\v66 text files\invest.pri
97				File: Terminal 12 pts
98				Input File: e:\v66 text files\invest.pri
99				File: Terminal 12 pts
100				Input File: e:\v66 text files\invest.pri

Figure 3-23 Mask window with all relevant data defined.

Heading Lines are displayed with a pink background color within defined columns, and with an uppercase **H** on the Line Control Bar. When translated into a spreadsheet, Heading Lines are formatted as labels and are placed as individual cells into their respective columns.

Translating Data

We have selected the rows and columns of data from the report that we want translated into our spreadsheet. Now we need to select an output file type and extract the data to a file.

Choosing an Output File Type

DataImport can translate data from the Input File into Output Files of many different types. In this example, we will select the Output File type Excel 97/2000.

Select **File > Define Output File**.

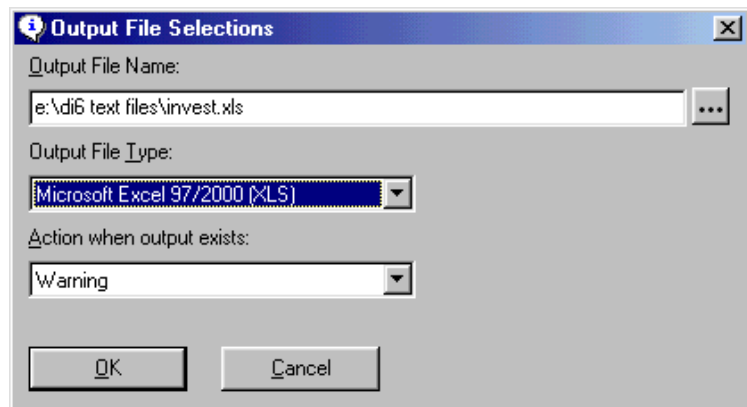


Figure 3-24 Output File Selections dialog box

Click on the arrow to the right of the Output File Type option's drop down list box.

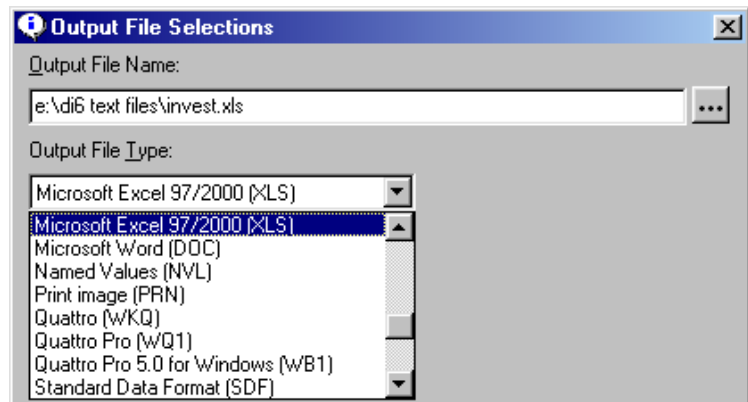


Figure 3-25 Output File type drop down list

Select **Excel 97/2000[XLS]** or select the type of file that your software requires.

Press OK to accept the current selections.

DataImport can create files in nearly 40 formats. See Appendix B: Supported Output Formats for more information.

Running a Translation

Now translate the file into the Output File in the format you chose:

Select **Files > Translate**. The Translation Parameters dialog box is displayed:

Output files can automatically be opened in your spreadsheet or database when the translation is completed by checking this box.

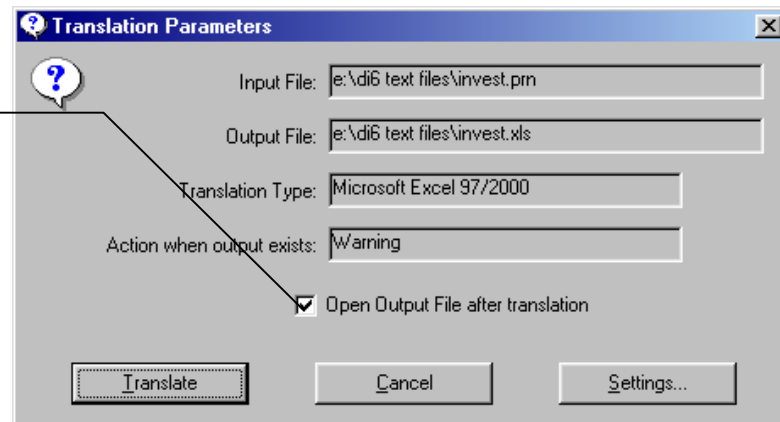


Figure 3-26 Translation Parameters dialog box.

Click **Translate** to begin the translation process. The Translation Progress window is displayed:

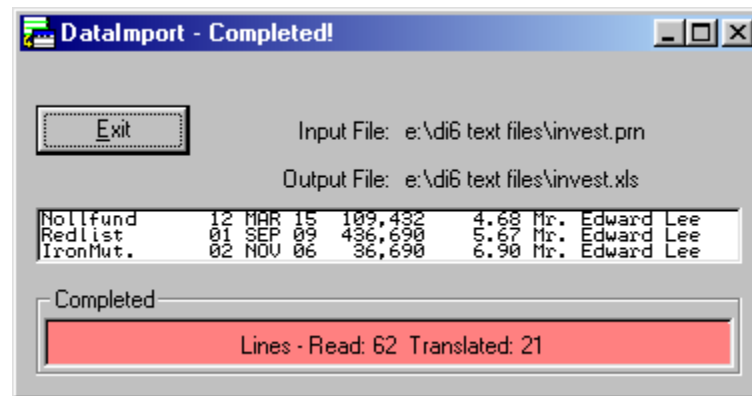


Figure 3-27 Translating Progress window

As the report is translated, the selected lines and columns of data are displayed in the Translating window. When finished translating, the title of the window changes to DataImport - Completed! Information about the number of lines read from the Input File and the number of lines written to the Output File is also displayed.

Click **Exit** to return to the Mask window.

If you are eager to see the results, switch to your spreadsheet and load the file you just created. Then, please come back for a few final words.

Saving Masks for Reuse

If you get the same report every month or every week, you do not have to define a mask each time you convert the report into your application's file format; you can save it for later use. For example, we will save the mask we have defined for the Investment Report. In the future we can then use the saved mask to translate new versions of this report.

Select **File > Save Mask As**. The following dialog box will appear:

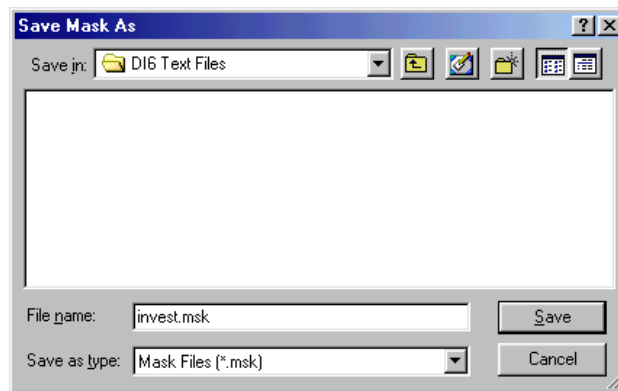


Figure 3-28 Save Mask As dialog box

The name of the Mask File defaults to the name of the Input File with the extension MSK. The name, drive, and directory of the Mask File can be changed using the options in this dialog box.

Press OK to save the Mask File.

The Summary Info dialog box will appear. Here you can enter information about the file, as well as the author.

The quickest way to perform another translation using this saved Mask is to select the Translate icon from the DataImport program group. In the Translate application window, select the Mask File to be used and translate the Input File. In the Translate window, you can also temporarily or permanently change the Input File name, Output File name, or translation type saved in the Mask File.

If you will be performing multiple translations on a regular basis, this process can be automated using DataImport's Task Commander.

Using the Output

You can use the spreadsheet file, or any other file format, created by DataImport in your software just as if you had key entered all of the translated data. Following is the spreadsheet that DataImport translated from the example Investment Report loaded into Microsoft Excel:

	A	B	C	D	E
1		Allegro Investment Company			
2	Date: 11/30/03				Fixed Rate Investme
3	Investment	Matures	Value	Interest	
4					
5	Alphatex	07/23/2015	234,020	7.62	Col. Steve Nixon New
6	Hoag	04/01/2009	10,090	5.91	Col. Steve Nixon New
7	Hagerton	03/12/2022	109,432	4.68	Mr. Reginald Cook Long
8	Mouton	09/01/2028	436,690	5.67	Mr. Reginald Cook Long
9	Maxwell	11/02/2035	36,690	6.90	Mr. Reginald Cook Long
10	Jagerson	12/31/2016	11,020	6.63	Frank Braun Bonn
11	Hardin	07/23/2021	110,093	5.99	Frank Braun Bonn
12	Silver	10/12/2018	154,456	8.68	Ms. Diane Johns Chic
13	Adfund	01/01/2022	496,835	3.87	Ms. Diane Johns Chic
14	Ifund	07/02/2009	537,337	8.99	Ms. Diane Johns Chic
15	Laslo	02/24/2024	63,366	7.87	Peter Parker Atlai
16	Engstrom	03/12/2012	17,592	3.89	Peter Parker Atlai
17	Carvex	07/27/2028	234,020	7.62	Mr. Jerry Hilyer Phoe
18	4Trust	04/11/2032	10,090	5.91	Mr. Jerry Hilyer Phoe
19	Nollifund	03/12/2015	109,432	4.68	Mr. Edward Lee Mars
20	Redlist	09/01/2009	436,690	5.67	Mr. Edward Lee Mars
21		11/02/2006	36,690	6.90	Mr. Edward Lee Mars

Figure 3-29 Excel with translated data

Notice that all data is properly formatted: dates are formatted as dates, numbers as numbers and columns are set to their proper width.

Chapter 4: Fitting DataImport to Your Needs

This chapter identifies specific types of problems in extracting data from input files and how to use DataImport to solve these problems. It is recommended that you read the chapters titled “Introduction” and “Tutorial” first to understand the basic operation of the program.

Input and Output

This section discusses how to get your data into DataImport and out to your spreadsheet, database or analysis program.

DataImport works best with Input Files in ASCII text format. If your data application allows it, save a copy of your data to an ASCII text file for best results. If this is not possible, look for a printing option in your data application that will print to a text file, rather than to the printer. These types of options are usually called “print to disk” or “print to file.”

DataImport can create output files for most spreadsheet and database programs including Excel, Lotus 1-2-3, Quattro Pro, Access, and dBase compatible applications.

Input Files

The Input File contains the data you want to translate. The file can be any ASCII file and is typically a print file or output from an application on a mainframe, minicomputer, LAN or stand alone PC.

There is no limit to the number of lines or records in an input file. However, information can only be *displayed* from the first 65,536 lines of a file. With DataImport you can view and translate lines or records as long as 16,384 characters from the Input File. Characters beyond the first 16,384 characters are ignored.

NOTE: If you have a file that is longer than this maximum line width and you need to view or translate characters beyond the 16,384 limit, use the DataImport Utilities application and select the Line Split by Length process. Similarly, if you need to view lines beyond the 65,536 limit, select the Records per File Split process in the Utilities application. Lines beyond the 65,536 limit *are processed* by

DataImport—even though you may not see them in the Input File window—with all mask definitions except for manually applied Line Treatments.

DataImport can handle files that have their information formatted many different ways. Files that have their data in a columnar format are easier to work with. Utilities are provided that convert several types of non-ASCII and/or non-columnar files into ASCII columnar files.

The Input File can contain printer control codes (ASCII characters 0 through 31). The control codes can be removed by using the Mask application's **Exclude > Characters > All Special Characters** command.

Loading an Input File

The Input File containing the information to be translated can be selected from either the Mask application or the Translation application.

In the Mask application select **File > Load Input**. Select a file from the File Name list box or type the name of the file to be loaded including the full path under File Name. Click OK to load the file.

After an Input File is chosen, it is displayed in the Mask window. If the file does not load or loads with a lot of garbage characters, the Input File is probably not an ASCII text file. Check the output settings in the program from which you obtained your data to make sure it is set to an ASCII text format and try outputting the data again.

If the new data file still has a high percentage of garbage characters, try excluding these characters with the **Exclude > Characters > All Special Characters** command and see if the text looks right.

If the Input File is still not usable, read the section below to see if you can use the DataImport Utilities application to translate the file.

Converting Files into a Type DataImport Can Use

Prior to displaying a file for mask definition, you may want to convert the file using the functions available in the DataImport Utilities program. These functions make mask definition easier and more effective for certain types of files. The following types of conversions are available:

Comma Separated Values Converts a comma separated value file to a fixed length column or file. Users can also specify a separator and string delimiter with this process by typing in the ASCII code of the character.

dBase convert Converts a dBase II, III, or IV data file into an ASCII columnar file with the database field names above each column.

EBCDIC -> ASCII Converts a file that has been downloaded from an IBM mainframe or midrange computer that has EBCDIC encoding into an ASCII encoded file.

Fixed length Breaks up a file with fixed length records without record separators into a fixed length file with record separators.

Line split by length Splits the Input File vertically by producing two or more files, each with a shorter specified part of each line.

Parse spaces Converts a space separated variable file into a file with fixed length fields.

Records per file split Splits a file with a large number of records into several files, each with a specified number of lines.

Tab expansion Expands tabs by inserting spaces to align the data in columns.

Unstack Makes single-line items from multiple lines that logically go together.

Chapter 7: DataImport Utilities Reference provides a complete description of these functions, with illustrations.

Output Files

DataImport places the extracted data in an Output File whose name, file type, and location on the disk is specified by the user. For more information and a list of supported output formats, see Appendix B: Supported Output File Formats.

Choosing an Output File Type

DataImport can translate a file into many different file formats. The correct file extension is automatically appended to each Output File.

The Output File type can be defined from either the Mask or Translate applications. In the Mask application, select **File > Define Output File** and then select the output type from the Output File Type: pull-down menu. In the Translate application, select the output type from the Output File Type pull-down menu.

The available translation types are displayed in the pull-down menu. The currently selected type is highlighted at the top of the menu. Examine the documentation accompanying your target software application to determine the file type and/or file extension required.

NOTE Keep in mind that most programs can read earlier versions of their file formats. In most cases, a file format with a version number equal to or less than your software version will work, unless you are combining or appending files. In this case, you must output your data as the same version as the existing output file. You can also output it as an earlier version and append it from within the software.

Choosing an Output File Name

The Output File name defaults to the Input File name plus the extension of the file format you have chosen. You can change the name of the Output File to any name supported by the operating system. DataImport adds the file extension based on the type of Output File you want created when the translation is performed.

When a mask is saved, DataImport saves the name of the last specified Output File with the mask. This name is used again when the mask is used for translation later unless you change it then.

The Output File name can be specified in the Mask application by selecting the **File > Define Output File** command. You may also specify the Output File name in the Translation application by selecting the Output File Name option.

NOTE: If you are performing a file append or combining files, the output file name must have the same name as the existing output file. See the next section for more information about appending and combining.

Translating to Existing Output Files

The output of a DataImport translation is usually to a new file, however, you can direct the output to an existing file. DataImport can simply replace the data in the existing file with data from the current translation and—in some cases—append the new data to the end of the existing file or combine the new data with the data in the existing file.

You set DataImport to automatically Append, Combine or Replace by selecting **File > Define Output File** and selecting one of these options from the **Action when output exists** list.

Appending Data to an Existing File

If the Output File type is a spreadsheet or a database, DataImport can append the data from the current translation to the end of the existing file.

Combining Data into an Existing Spreadsheet File

For spreadsheet output types, DataImport can combine the data from the current translation into the existing spreadsheet file at a specific row and column address. To set the starting address for a file combine, select **Options > Global**, type in the Starting Cell Address and press OK.

DataImport's combine option works in much the same way as the File Combine option in Excel and other spreadsheet programs.

Database File Considerations

In addition to data, database files contain a database field structure that specifies the field names, field lengths, and field types. This information is written to the database file when it is created.

If a database file exists when you translate, DataImport uses the existing field structure in the file—even if it is different from the column settings in the mask. If the file does not exist, DataImport automatically creates a structure using the column settings defined in the mask.

IMPORTANT!: If you have used DataImport to originally create the database and you make a change to the mask, you need to delete the database file that DataImport created before translating again. If you do not delete the first database, the data field structure will remain intact and the changes in the mask will not be applied to the database file.

Cleaning-Up Input Files

In many cases, your Input Files may not be a simple ASCII text file, especially if they are print to disk files from a mainframe, midrange or PC. These files often contain printer codes, control codes or other special characters. *Special characters* are non-alphanumeric characters with ASCII codes from 0 through 31. This section discusses how to remove these characters from the Mask display and your Output Files.

NOTE: All clean-up functions described in this section do not in any way change the actual content of the Input File. These functions simply suppress the display of specific characters or text in the Mask window and tell DataImport to ignore this information when translating a file.

Special Characters

An Input File can contain any number of special characters or “garbage characters” that make a file harder to read and are not needed in the Output File. All single character control codes can be removed with the Exclude Characters All Special Characters command. This option removes all ASCII characters with codes of 0 through 31 except for the escape character (ASCII 27). Escape characters are not removed so that printer control codes can be identified and excluded more easily.

Blank Lines

To remove all blank lines in an Input File, select **Exclude > Blank Lines**.

Page Ejects

A *page eject* or the ASCII 12 character is a standard printer control code for causing a printer to feed to the top of the next page. To remove all page eject characters, select **Exclude > Page Ejects**.

Duplicate Lines

Some programs print a line, perform a carriage return without a line feed, and print the line again. This results in “bold” print that is used for emphasizing titles and headings on reports. The **Exclude > Duplicate Lines** command removes the second line of print from this style of report or any line that is exactly the same as the preceding line.

Escape Sequences

Escape sequences are a string of two or more characters beginning with the escape character (ASCII 27) that provide control instructions to a printer. To remove an escape sequence, or any other string of characters, select the sequence with the highlighter and select **Exclude > Characters > Define**.

First Position Carriage Control

Carriage control characters are another type of printer control that is included in reports created by some programming languages on certain computers. FORTRAN, for example, uses the first character position of each line to indicate line feeds and form feeds to the printer. To ignore carriage control characters in the first position of every line, select **Options > Global**, and in the First positions to exclude field, type the number of characters to exclude (usually 1) and press OK.

Extracting Data

DataImport provides many facilities for extracting data from both columnar reports and forms. If your data is not columnar, also see “Unstacking Multiple Lines of

Data” on page 41 and “Putting Header or Form Information into Columns“ on page 43 for information on organizing data in columnar format.

This section discusses how to use DataImport’s Masking processes to extract data and other information from computerized reports, including defining columns, extracting groups of information, excluding lines and extracting specific lines.

Columnar Data

DataImport is effective in extracting data arranged in columnar format and provides many facilities for extracting this data efficiently. One of the primary tools that DataImport provides is the capability of including specific data lines in the translation process. These features allow you to easily extract only the data you want without extensive processing in your spreadsheet or database.

Defining Columns

Columns are used to define the positions, or cells, within each line of the Input File that will be translated to the Output File. Data is translated only if included within the defined range of a column, or a line tag as explained later in this chapter. A column encompasses all lines in the Input File for the range specified. A maximum of 256 columns can be defined in any one mask and columns cannot overlap.

As you learned in Chapter 3: Tutorial, there are several ways of defining columns. You can create a column in several ways manually, or use DataImport’s AutoColumn feature to automatically define columns. You can also add columns or remove columns from an existing mask.

After a column is defined, the type of data in the column can be specified. The output sequence of columns can also be defined, see “Resequencing Data Columns” on page 41.

Manual Column Definition

Columns may be defined by highlighting sample data that indicates the width of a column and either using the **Column > Define** command. Columns can also be defined by dragging the cursor over the Column Control Bar. Columns cannot overlap each other.

Automatic Column Definition

You can automatically define columns by simply selecting a line in the file for DataImport to use as a pattern to establish column positions. This method disregards all previous column specifications and defines columns for the entire length of the selected line.

To define columns using this method, place the cursor on the line to be used as the pattern, then select the **Column > Auto Define All** command. The Input File will then be re-displayed with the new column definitions.

DataImport can also be set to automatically define columns when an Input File is first displayed on the Mask Screen by selecting **Options > Preferences** and checking the Automatically define columns option. If the resulting columns are inappropriate (the complexities of some file structures may produce undesirable column definitions), you can manually modify the resulting automatic column definitions.

Removing Columns

All columns or a specific column can be removed from the mask. To remove all columns from the mask, select the **Column > Undo All** option. To remove a specific column, click within the column and select the **Column > Undo** command.

Including Data Lines

In some cases you might want all lines in the file, in other cases, you may want to only include specific lines of data. You may want data from particular regions or information about a certain product.

There are three ways to include lines for translation to an Output File. By specifying that all lines default to being output by selecting **Options > Global**, and setting Default Line Treatment to Output Lines. Lines containing a specified string of characters can be automatically output. Or, a particular line can be manually included.

Including Lines with a Match String on the Line

DataImport can include all lines for translation that contain a specified match string. The match string can be a specific string of characters, or a pattern match string that contains wildcard characters. This feature (usually used in Global Skip Line Mode), is useful when selecting specific lines that all contain common information, or a range of lines starting with a line containing the specified string of characters.

Lines included in this way are translated the same way as Output Lines. A line already treated as an Output Line, Title or Heading that does not contain the specified string is not affected by the use of the Include Line feature; they are always output.

To include lines, after the occurrence of a match string reference point, examine the lines that are to be output and identify a string of characters or a pattern unique to these lines. Highlight a text string to cause the line to be included in the translation. Select **Include > Lines > Define**. The Define Include Line dialog box appears. If necessary, type in new characters or pattern match characters under the Original String field.

If the match string should be exactly what you selected, do not make any changes to the text. If the match string should be different, change the text as necessary. If the match string should be a general pattern, use the wildcard characters to define the type of characters allowed in each position.

The following wildcard characters are used to define the type of characters allowed in each position of a Match String. All other characters in the pattern match string require that character at that position.

^	(caret)	Any number (0 through 9)
!	(exclamation)	Any character except 0 through 9
~	(tilde)	Any character except blank
_	(underscore)	Any character including blank

Figure 4-1 Pattern Match wildcard characters

Select At position if the lines should be included only when the string is found at the same character position as the original match string. Select Anywhere if the line should be included if the string is found at any position in the line. Press OK to apply the Include command.

The file will be re-displayed with the lines to be included marked with an “I” in the Line Control Bar on the left side of the Input File window.

Using Pause and Resume Match Strings to Include a Variable Number of Lines.

Some reports list a varying number of detail lines for a region, product or sales office with the name of the group shown only above or on the first line of that group. There is no unique text on each line that can be used as a match string for the Include Line feature. In this case, you can include these lines by using the Pause and Resume feature.

Pause and Resume commands start or stop translation of a file when match strings are encountered. When a Pause match string is encountered in an input file, that and all lines after it will be skipped until a Resume match string is encountered.

To include a variable number of lines in the Output File, identify a character string that identifies the beginning of the lines that you want to include and then use the **Include > Resume > Define** command to insert a resume in translation at these points. In the dialog box, check the Begin in Pause Mode box if the lines to be included are not at the very beginning of the report.

The Mask window will display the Input File with the new resume definition. Resumed lines will have color highlighting.

To stop extracting lines, highlight a character string that identifies the start of the lines that you do not want to include and select **Exclude > Pause > Define** this inserts a pause in translation at these points.

The Mask window will display the Input File with the new pause definitions. Paused lines will not have any color highlighting and a “P” will appear in the left margin for any paused line. Only one Pause and one Resume definition can be defined in a Mask.

Manually Including Lines

Occasionally you may need to include lines that do not share a common text string. In this case DataImport allows you to manually Output lines.

To manually output lines, highlight a range of lines and select the **Line > (O)utput** command. The line will be re-displayed with color highlighting and a “O” will appear in the Line Control Bar to the left of the line.

Excluding Data Lines

In some cases, you may want to ignore or exclude specific lines from being translated. You may not be interested in data from particular regions or information about a certain product. The DataImport Mask application allows you to exclude this information from your output files using the Exclude functions.

To exclude lines from output, use one of the following methods. Setting DataImport to skip all lines unless otherwise included by selecting **Options > Global** and setting the Default Line Treatment to Skip. Lines containing a specified

string of characters can be automatically excluded. Lines can be set to be skipped individually based on their line position in the input file. Or, lines that contain a specific range of values within a column can be excluded from translation.

Excluding Lines with a Match String on the Line

DataImport can be instructed to exclude all lines from translation that contain a specified match string. The *match string* is a specific string of characters, or a pattern match string that contains wildcard characters. This feature (usually used in Global Output Lines Mode), is useful when excluding lines that contain common information from translation into the Output File. This feature can also be used to exclude recurring page titles and headers.

To exclude lines, examine the lines that are to be ignored and highlight a string of characters or a pattern of numbers or letters unique to these lines, and select **Exclude > Lines > Define**. The Define Exclude Line dialog box appears. If necessary, type in new characters or pattern match characters in the area under the Original String field.

If the match string should be exactly what you selected, do not make any changes to the text. If the match string should be different, change the text as necessary. If the match string should be a general pattern, use the wildcard characters to define the type of characters allowed in each position.

Select At position if the lines should be excluded only when the string is found at the same character position as the original match string. Select Anywhere if the line should be excluded if the string is found at any position in the line.

The file will be redisplayed with the lines to be excluded marked with an “E” in the Line Control Bar on the left side of the Mask window.

Excluding Lines with Column Limits

Lines can be excluded from translation that contain values that are lower or higher than the specified limits in a column. After a column is defined, limits can be set by entering an upper limit, a lower limit, or both. How DataImport prompts for limits is dependent on the column type; numeric, text, date, or time.

To define a column limit, click in the column and select **Column Settings**. The Column Settings dialog box appears. Type in an upper limit, a lower limit or both in the Limits fields and press OK to apply the limit.

Lines with values in the column that are higher or lower than the limits are displayed without a background color. An uppercase “E” also appears in the Line Control Bar to indicate that the line is Excluded.

Numeric data, dates and times are excluded based on their numeric value. Text is excluded based on its alphanumeric ASCII value order. This order starts with numbers, then uppercase letters, then lower case letters. For example a limit between an upper limit of 1 to a lower limit of Z is valid, but a range from an upper limit of A to a lower limit of 9 is not.

If the column type is changed, the limits for the column are eliminated. To eliminate or undo the test for limits in a column, delete the limits from the Limits fields for the column.

Manually Skipping Lines

Occasionally you may need to exclude lines that do not share a common text string. In this case DataImport allows you to manually skip specific lines.

To manually skip lines, highlight the range of lines to be skipped and select the **Line (S)kip** command. The line will be redisplayed without any color highlighting and an “S” will appear in the Line Control Bar to the left of the line.

Aborting Translation Specified Line

Some Input Files may be very long and you may only need to translate the first section of the data or you may want to test the results before translating the whole file. In this case DataImport allows you to define an artificial end of file that will stop the translation when it reaches a particular line.

To define an end of file, click on the line below the last one you want to translate and select the **Line (A)abort** command. The line will be re-displayed without any color selection, an “A” will appear in the left margin of the Mask window for that line and a small “a” will appear on every line following it.

Default Line Treatment

DataImport has two modes for the default treatment of lines. One mode assumes that the data on all lines is to be translated unless the line is specifically Excluded or Skipped. This mode is called Global Output Line Mode. The other mode assumes that no lines are to be translated unless they are specifically included. This mode is called Global Skip Line Mode.

In Global Output Mode, Include Lines have precedence over Exclude Lines. That is, if an include match string occurs in an Exclude Lines’ range, the Include Lines will be translated. In Global Skip Line Mode, Exclude Lines have precedence over Include Lines; if an exclude match string occurs in an Include Lines’ range, the Exclude Lines will not be translated.

To set the default line treatment, select **Options > Global**, and under Default Line Treatment mark either the Output lines or Skip lines option.

Titles and Headings

Some reports contain a title or headings that do not belong in data columns but which you may want to include in your spreadsheet. DataImport allows you to define lines as titles or headings that are written to spreadsheet Output Files. These line definitions are called line treatments.

To define a line as a Title or Heading, select the lines to be defined and select the **Line > (T)itle** or the **Line > (H)eading** command.

Heading The data within the columns on each Heading Line is translated as text (non-numeric). The line is displayed with a pink background and an “H” appears in the left margin. Notice that lines defined as headings include only data in defined column ranges, not data for the entire line. To include the entire line, see the Title treatment below.

Title The entire line is translated as a single text string (non-numeric) and column definitions are ignored. The entire line is displayed with a red background and a “T” is displayed in the left margin.

*To keep repetitive titles and headings from being output, see the previous section titled **Excluding Lines with a Match String on the Line**.*

To restore the default setting for a range of lines that have been defined with Titles, Headings or other Line Treatments, select the lines to be restored and select **Line Default**.

Reorganizing Data

Some Input Files may not have data organized in a way that is appropriate for your type of spreadsheet or database. Rows and columns may be switched, different types of data may be stacked on top of one another or columns may simply not be in the right order. This section discusses some data organization problems and how to use DataImport to solve these problems.

Resequencing Data Columns

When a mask is defined, DataImport orders the columns according to the sequence in which they are defined on the screen. If this sequence does not put the information in the most usable order, you can resequence the columns. For example, you could switch positions of column C and column A. The new sequence would be C, B, A, D.. This option is useful when the Output File is an existing database whose structure orders information differently from the Input File. Columns can also be skipped (e.g., A, D, E, K).

To resequence a column, click in the column to be resequenced, select **Column > Settings** and type the new column letter in the Letter field. If the letter is already in use by another column the letters of the two columns are switched. Line Tags can be resequenced in a similar way.

Unstacking Multiple Lines of Data

Some reports stack multiple line sets of data on top of one another. For instance, in a sales report for different regions, the report might list the region on the first line along with the daily sales and then list the monthly and yearly sales for that region are moved onto the next two lines. The easiest way to handle this is if the daily, monthly, and yearly data are on the same line so that they can be put into their own unique columns. The Unstack function allows you to do this automatically.

In order to unstack data, you use a text match string to identify the first line of each set of lines and then define how many lines of data are stacked. Select a text string which identifies the first line of each set of stacked lines. Select **Options > Unstack > Define**. In the Lines to unstack field, enter the number of lines to unstack. To apply the unstacking function, press OK.

The following example illustrates how a report file with stacked lines looks before and after it is converted.

Each branch has two lines of data. One line for the month and one for the year.

Branch	Period	Sales
NEW YORK	MONTH	12,935
NEW YORK	YEAR	31,221
LONDON	MONTH	9,264
LONDON	YEAR	24,698
PARIS	MONTH	8,573
PARIS	YEAR	35,745
MUNICH	MONTH	14,684

Figure 4-2 Stacked Input File with columns and headings defined

In this example, the Unstack command was defined with “MONTH” as the match string and 2 Lines to unstack. The screen below shows the results of the Unstack command.

After unstacking, there is one line per branch with both the month and year data.

Branch	Period	Sales
NEW YORK	MONTH	12,935
NEW YORK	YEAR	31,221
LONDON	MONTH	9,264
LONDON	YEAR	24,698
PARIS	MONTH	8,573
PARIS	YEAR	35,745
MUNICH	MONTH	14,684

Figure 4-3 Unstacked data

Notice that the yearly data has been moved into new columns to the right of the monthly data and that the Heading information has been duplicated in the new columns. To complete the mask definition, two new columns should be defined for the new yearly PERIOD and SALES.

If the Input File is a report with a heading at the top of each page that can sometimes occur within a block of lines, it may be necessary to define a mask and perform a translation to a print image file (.PRN) to remove the headings before unstacking the file.

Unstacking is also very useful for preparing a text file of names and addresses created with a word processor for translation into a spreadsheet or database file. Unstacking such a file can produce separate columns for name, street address, and city/state/zip code. Only one Unstack command is allowed per Mask.

Getting Data from Multiple Lines into the Same Cell

Sometimes reports contain multiple lines for a single field, such as this inventory report:



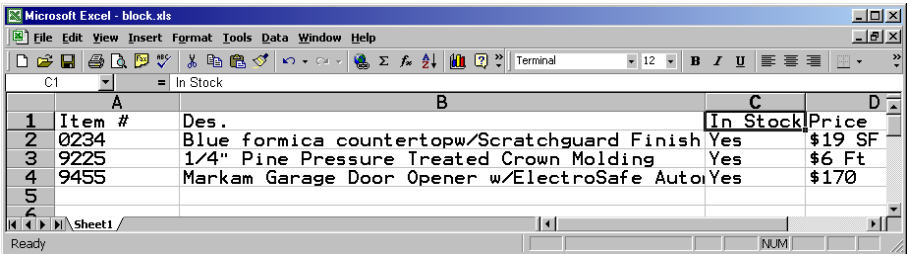
The screenshot shows the DataImport Mask application window. It displays a table with four columns: Item #, Des., In Stock, and Price. The data is as follows:

Item #	Des.	In Stock	Price
0234	Blue formica countertop w/Scratchguard Finish	Yes	\$19 SF
9225	1/4" Pine Pressure Treated Crown Molding	Yes	\$6 Ft
9455	Markam Garage Door Opener w/ElectroSafe Automatic Shutoff	Yes	\$170

Figure 4-4 Input File with Multiple Data Lines per Field

Note that the description field contains multiple lines of data. To get multiple lines of data into the same cell, use the column type *Text Block*. Text Block instructs DataImport to keep adding data from multiple lines in a column into the same cell (or field) during translation until the next line is to be output or a blank cell in the column is encountered. You can also specify the Block to be a fixed number of lines.

When outputting a text block, make sure that you increase the column width in the Column Define dialog box to be wide enough to contain the longest resulting unstacked block of text.



The screenshot shows the Microsoft Excel application window with a file named 'block.xls'. The data is displayed in a table with four columns: Item #, Des., In Stock, and Price. The data is as follows:

Item #	Des.	In Stock	Price
0234	Blue formica countertopw/Scratchguard Finish	Yes	\$19 SF
9225	1/4" Pine Pressure Treated Crown Molding	Yes	\$6 Ft
9455	Markam Garage Door Opener w/ElectroSafe Auto	Yes	\$170

Figure 4-5 Translated file with Text Block

Pulling Data out of Headers and Footers

Reports often list information in headers, footers or elsewhere on the page. This data is not listed in columns and is often preceded by a repeated title. For instance, on an invoice report, the title "Region:" would appear on every page followed by the name of the region like "Northeast" or "South".

Outstanding Invoices Over 120 Days				Page 001
Region:	Inv#	Date	Amount	
South				
Office: Orlando				
Customer: JP Enterprises	173 Oak St.			Miami
	003242	10/20/01	\$4,092	
	100459	07/30/01	\$2,010	
Customer: ABC Inc.	64 Megs Ln.			Tampa
	002344	03/13/01	\$3,493	
	002934	05/29/01	\$1,993	
	003942	07/01/00	\$2,097	
Office: Atlanta				
Customer: Larry's Pawn	394 Ponce De Leon.			Decatur
	014993	01/20/01	\$999	
	013455	04/30/00	\$450	
Outstanding Invoices Over 120 Days				Page 002
Northeast				
Office: Bangor				
Customer: Bruce Ranyawn	139 Lazy Ln.			Bethel
	019932	10/03/00	\$1,300	
	012394	09/21/00	\$3,954	
Customer: Royal Snow	120 Marsten Ave.			Salem
	001843	03/06/01	\$2,029	

Figure 4-6 Input file with Page and Section Headings

You may want to put this type of information into a database or spreadsheet as single uniform rows or records. DataImport allows you to place this header information in columns by defining positional anchors or Reference Points on a sheet which point to fields of data or Line Tags.

A *Reference Point* is a positional anchor which DataImport uses to locate data that changes within a form or other report. There can be up to 100 Reference Points within a given Mask. In the example above, the word "Region" would serve as a reference point to data that changes from page to page. In this case, the changing data is the location or region of the invoices. A reference point can also be a top of form, or assumed to occur every specified number of lines.

A *Line Tag* is a field of data positioned at fixed places on a page which is located with a Reference Point. In the example above, the different regions ("South" and "Northeast") would be a Line Tag which would be located with the "Region" Reference Point.

Line Tagging works much the way it sounds: Data from Line Tags is output on each extracted data line, thus "tagging" each line with information. Each Line Tag you define is output as a column in addition to the columns that have a button on the line control bar.

The Line Tag function updates the information each time it encounters a Reference Point with which it is associated. In the example above, the Line Tag for the region information is updated each time DataImport encounters the Reference Point "Region:" text string in the Input File. The Line Tag function allows DataImport to "read" certain data from the Input file by looking for a keyword (the Reference Point) and then looking down, and to the right and left of the keyword for specific information (the Line Tag).

You tell DataImport where to find Line Tag information by first defining a match string that is used as a Reference Point and then defining the relative position of the Line Tag information to the Reference Point. Information in each Line Tag column is repeated on each output line until the Reference Point match string is encountered on another line, at which point the Line Tag information is refreshed.

For example, if the Input File contains a date in the heading of each page, the date can be output on every line. By defining a Reference Point and then the date as a Line Tag, every time the program encounters another occurrence of the Reference Point during translation, the next date will be output as a Line Tag.

Before a Line Tag can be defined, at least one Reference Point must be defined . You can define up to 100 Reference Points. There is no limit to the number of defined Line Tags. When a Line Tag is defined, it is associated with the selected Reference Point. The reference point can occur before or after the associated line tag.

The screenshot shows the DataImport Mask software interface. The main window displays a translated input file with defined line tags. The interface includes a menu bar (File, Search, Column, Tag, Include, Exclude, Line, Unstack, Options, Help) and a toolbar. The main area shows a table with columns A, B, and C. The data is organized into sections for 'South' and 'Northeast' regions, each with 'Outstanding Invoices Over 120 Days' and 'Page 001' and 'Page 002' headers. The data is organized into sections for 'South' and 'Northeast' regions, each with 'Outstanding Invoices Over 120 Days' and 'Page 001' and 'Page 002' headers. The data is organized into sections for 'South' and 'Northeast' regions, each with 'Outstanding Invoices Over 120 Days' and 'Page 001' and 'Page 002' headers.

Figure 4-7 Input file with all relevant data defined

Make sure to define the Line Tag with a selection that is as wide as the largest information that will occur in that position. For example, if the first Line Tag is “South”, make sure to select some space after “h” so that the “east” at the end of “Northeast” does not get cut off. Line Tags are assigned column letters sequentially in the order they are defined. Their sequence can be changed.

Options that can be selected for a normally defined column can also be selected for a column defined by use of Line Tags. This includes selecting the Type, Column Letter, Name, and @Function. To set the properties for a Line Tag column, click inside the text of the Line Tag you want to define and select **Tag > Line-Tag Define**. The Tag Settings dialog box will appear. Set the column definitions for the Tag as you would for a normal column and press OK to apply the new definitions.

Below is the translated input file in Excel.

Outstanding Invoices Over 120 Days					
Inv#	Date	Amount			
003242	10/20/01	\$4,092	Orlando	South	JP Enterprises
100459	07/30/01	\$2,010	Orlando	South	JP Enterprises
002344	03/13/01	\$3,493	Orlando	South	ABC Inc.
002934	05/29/01	\$1,993	Orlando	South	ABC Inc.
003942	07/01/00	\$2,097	Orlando	South	ABC Inc.
014993	01/20/01	\$999	Atlanta	South	Larry's Pawn
013455	04/30/00	\$450	Atlanta	South	Larry's Pawn
019932	10/03/00	\$1,300	Bangor	Northeast	Bruce Ranyawn
012394	09/21/00	\$3,954	Bangor	Northeast	Bruce Ranyawn
001843	03/06/01	\$2,029	Bangor	Northeast	Royal Snow

Figure 4-8 Translated invoice data in Excel format

Extracting Data from Forms

Data in form reports is usually located on a specific line and at a specific character position on each page of the report. This type of report looks like a computerized version of an insurance form; there is a line or block for your last name, first name, previous doctor, insurance company, policy number and so on. This type of report may not have any data in columns, which makes it difficult to format records for a database.

DataImport provides a function to “read” these types of forms and put the data they contain into columnar format. This function is accomplished with a combination of Reference Point and Line Tag commands.

Reference Points and Line Tags offer a way to “point” to data in a form. In a sense, they allow you to give DataImport directions to the location of data in a form. As with any directions, DataImport needs landmarks to find its way and locate the correct data. Reference Points serve as these landmarks and Line Tags are “directions” from a Reference Point to a place where data is located.

For example, if an insurance form report like the Input File below has a field titled “LAST NAME” at the top of each page of a form report followed by the last name, the last name information for each form can be extracted to a column.

Field	Value
LAST NAME	LOWE
FIRST NAME	CURTIS
DOCTOR	I.R. QUACK
INS COMPANY	AETNA LIFE AND CASUALTY
POLICY#	555-A555
PATIENT#	2349-23555
LAST NAME	LARRY
FIRST NAME	MELMAN
DOCTOR	DR. RAHVADASHI
INS COMPANY	PRUDENTIAL
POLICY#	123x5-02
PATIENT#	3933-91234

Figure 4-9 Form type report

By defining the text “LAST NAME” as a Reference Point and the last name of the patient as a Line Tag, the program will output the last name of the patient into a cell on each row that it outputs. When DataImport encounters another occurrence of the Reference Point “LAST NAME” it will update its current tag information and continue to tag each data line with the new information.

Let’s say that you have a form-type report like the one pictured above and you want to put this information in a spreadsheet where you have a Last Name column, a First Name Column and a Patient # column. DataImport will allow you to put this information into columnar format.

To output each patient as a row of a spreadsheet, change to Global Skip Line Mode, define a column that will capture the last piece of data on each form (Patient #) and then define an Include Line to extract the last line of data for each patient. Then use Reference Points and Line Tags to capture the rest of the data for the patient. Define a Reference Point at the beginning of each patient and then define what data to extract with the **Tag > Line-Tag Define** command.

In the following example, the default line treatment was set to Global Skip Lines. Then, to capture the last unit of data on each form, a column was defined to capture the Patient # and then an Include Line was defined with a match string of “PATIENT #” to create a single data line for each form. Notice the “I” in the Line Control Bar to the left of “Patient #.”

A Reference Point and two line tags were then added to extract the Last Name and First Name of the patients to separate columns. The Reference Point was created using the match string “LAST NAME” and two Line Tags were created: one with the selection “LOWE” and one with the section “CURTIS”. These selections were made larger than the name in order to capture longer names that may appear on the forms. The resulting mask is shown below:

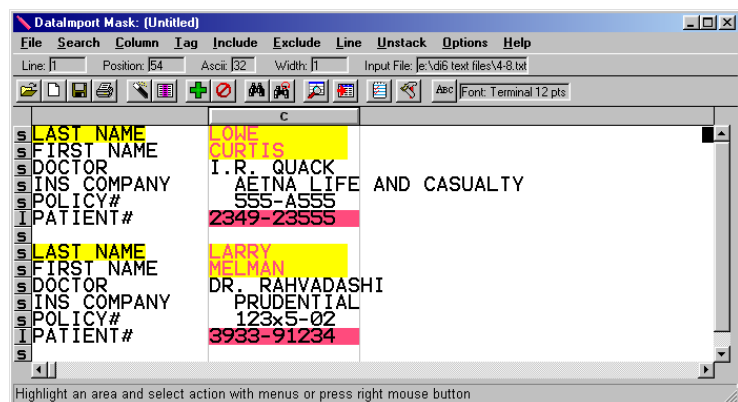


Figure 4-10 Mask for outputting form information to columns

After translating, the output of this mask is shown below:

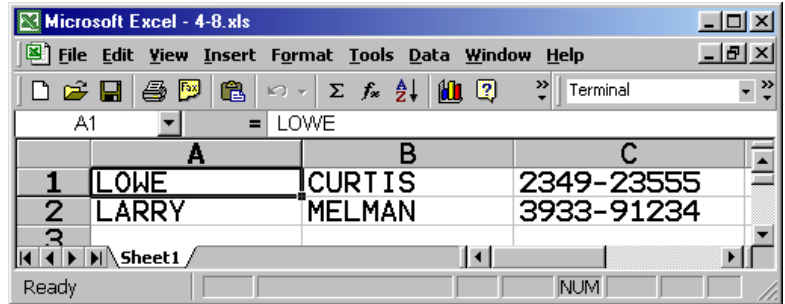


Figure 4-11 Output from Mask pictured above.

Line Tags are always included into any outputted data lines (Include or Value Lines) that occur *after* or on the same line as the Reference point, so it is important to define the Reference Point before or on the same line as the Included Line on the form.

Options that can be selected for a normally defined column can also be selected for a column defined by use of Line Tags. This includes selecting the Type, Column Letter, Name, and @Function. To set the properties for a Line Tag column, click inside the Line Tag you want to define and select **Tag > Line-Tag Define**. The Tag Settings dialog box will appear. Set the column definitions for the Tag as you would for a normal column and press OK to apply the new definitions.

Filling Blank Column Cells

Some Input Files do not repeat information in a column unless it changes. For example, in the sales report below, the column indicating the region in which the person works is included only for the first salesperson in each region.

(Col A)	(Col B)	(Col C)
Region	Salesperson	Units Sold
Southwest	John F.	10
	Joan K.	14
	Terri Y.	15
Northeast	Jim B.	16
	Jill S.	12
	Tim R.	14

Figure 4-12 Report with blanks in column A

To tell DataImport to fill all the blanks in the column with the most current information, click in column A, select **Column Settings**. When Blank, mark the Fill-down option. Below is the resulting output:

Figure 4-13 Column Settings dialog box with fill down selected

(Col A)	(Col B)	(Col C)
Region	Salesperson	Units Sold
Southwest	John F.	10
Southwest	Joan K.	14
Southwest	Terri Y.	15
Northeast	Jim B.	16
Northeast	Jill S.	12
Northeast	Tim R.	14

Figure 4-14 Report output with Fill-down applied to column A

Transpose Rows and Columns

Some reports may print data that should be in columns on a line or may print data in columns that should be on lines. For output used with a spreadsheet, DataImport can transpose columns and lines. That is, data displayed as columns in the Mask Screen will be output as lines, and vice versa.

To output rows as columns and columns as rows, select **Options > Global**, mark the Transpose rows and columns option and press OK. The display will not change, but the data output to a spreadsheet file will be transposed.

If this option is selected, column widths defined in the mask are not used during translation. The Output File will be displayed using the spreadsheet program's default column width.

Recognizing Data Types and Formats

Data in computerized reports can come in many different styles. Depending on the person or system that produced the report, data formats for numbers, dates and times can vary widely. You may receive reports from the United States, Japan, Germany or Australia that use different date, currency and decimal formats. DataImport can recognize a wide variety of data types and formats—assuming it knows what to look for.

The way that DataImport recognizes data is the Type setting for the definition of the Column. Whenever you define a column, DataImport defaults to the column type specified in the Preferences. You can change this definition for a specific column by clicking in the column, selecting **Column > Settings**, and choosing an option from the Type pull-down menu.

There are several data types you can define in DataImport. These are discussed below.

Numeric

When you define a column with this type, the program will attempt to treat all the data in a cell of this type as numeric values. If DataImport cannot translate the cells as numeric values, it will translate them as text. Data that will be translated as numbers is displayed in a blue color.

DataImport understands how numbers are represented on reports and translates them correctly into the Output File. It also understands that negative numbers can be represented several ways, either with a minus sign before or after the number, with parentheses, or with a CR or DR next to the number (credits and debits).

DataImport understands that a percent sign (%) indicates the number should be divided by 100. It even handles subtotals that are marked with asterisks. DataImport can also translate numbers that are expressed in scientific notation.

Computer systems in the U.S. use the dollar sign (\$) to express currency, a period (.) to indicate the decimal point, and a comma (,) as the thousands separator. DataImport uses these settings as defaults but it can be instructed to use other symbols for recognizing currency, thousands and decimals, as explained below.

Currency DataImport can recognize any currency format. By default the currency setting is set to US dollars (\$). The following are some of the pre-set symbols that can be recognized as currency symbols:

Symbol (currency name)	
\$	(dollar)
€	(euro)
¢	(centavo)
£	(pound)
¥	(yen)
A\$	(Australian dollar)
C\$	(Canadian dollar)
Dkr	(Danish Krone)
DM	(German mark)
Fr.	(French Franc)

Gld	(Guilder)
L	(Lira)
NKr	(Norwegian Krone)
p	(peseta)
SFr	(Swiss Franc)
SKr	(Swedish Krona)

To select the current currency symbol, select **Options > International.** and under Number Format, either select the currency symbol from the Currency pull-down menu or type in the currency symbol.

Thousands Notation The symbol for separating thousands in the U.S. is the comma; also used in other countries are the period and the space. DataImport supports all three of these notations.

To select the thousands symbol, select **Options > International.** and under Number Format, select the thousands notation symbol from the Thousands pull-down menu.

Decimal Points The symbol for separating the fractional or decimal part of a number from the integer portion of a number can be defined as either the period or the comma.

To select the current decimal symbol, select **Options > International.** and in the Number Format field select the decimal symbol from the Decimal pull-down menu.

Text

This type instructs DataImport to translate the cells in the columns as the same characters that are in the Input File. When another column type, such as numeric or date, are selected and DataImport cannot translate that data into the requested type, DataImport will output the data as text. Data that will be translated as text is displayed in a pink color.

There are three kinds of Text selections.

Text (Character/Label) text instructs DataImport to translate the data in the column as text characters. This is the most commonly used text type.

Text (Left Justified) text instructs DataImport to remove spaces from the beginning of text.

Text Block (unstack) instructs DataImport to keep adding data from multiple lines within a column into the same cell (or field) during translation until the next line is to be output or a blank cell in the column is encountered. You can also specify the Block to be a fixed number of lines.

Case There are four case settings, three of which affect the capitalization of text when they are output. The following table illustrates the effects of these options on sample text:

Case Setting	Input	Output
(As-Is)	Miami	Miami
(Lowercase)	Miami	miami
(Uppercase)	Miami	MIAMI
(Proper)	MIAMI	Miami

Figure 4-15 Case Settings and their effects

Date

This type instructs DataImport to translate the cells in the column as dates. In spreadsheets, the date is the number of days since January 1, 1900. The order of the month, day and year of the date must be specified. Eight formats are supported. Dates do not need to contain separators. If the program cannot translate the cells as dates, it will attempt to translate them as numeric values, then as text. Data that will be translated as dates is displayed in a green color.

Date Custom The custom format is used to recognize some types of complex that do not contain separators and is applied when you specify the Date (Custom) setting for a column.

To define the custom date format, select **Options > Dates** and in the Custom Date Format enter the custom date. Type in the format string using the letters D, M and Y as positional indicators for the day, month and year. The table below shows some examples of custom date formats and how they interpret dates without separators.

Type this:	To recognize this:	As this:
YYMMDD	011231	December 31, 2001
MMYYDD	120131	December 31, 2001
DDMMYY	311201	December 31, 2001
YYYYMMDD	20011231	December 31, 2001

Figure 4-16 Custom date examples and results. .

Two Digit Years For dates that contain two digits for the year, the cutoff date that divides 19xx from 20xx can be defined. In some files, the date 11/25/55 could mean 1955 or 2055. To define the two digit year that is the cutoff between 19xx and 20xx, select **Options > Dates**, and in the Year for 19XX field type the cutoff date for interpreting two-digit years.

Month Names DataImport recognizes the standard U.S. spellings of the names of the months, but it can also be instructed to recognize different spellings, such as the German spelling of October: "Oktober". To specify the spellings of the month names, select **Options > International** and in the Month Name field type in the month spellings you want DataImport to recognize. To save your custom Month spelling definitions for use in future masks, press the Save as defaults button. If you want to use these saved month spellings at a later date, simply select **Options > International** and press the Load defaults button.

Time of Day

This type instructs DataImport to translate the cells in the column as the time of day. The time is translated to a decimal number between 0 and 1. 0 indicates midnight, and 0.5 indicates noon. Spreadsheets will show this number as a time. If the program cannot translate the cells as times, it will attempt to translate them as numeric values, then as text. Data that will be translated as the time of day is displayed in a yellow color.

Name Parse

This type instructs DataImport to translate the data in the cells of the column as names, which are split into separate text columns (or fields) during translation. Names can be parsed into prefix, first, middle, last and suffix, i.e. Mr./John/Van Kamp/Jr. Data that will be translated as Name Parse is displayed in a pink color.

Name Parse uses a DEFAULT.DIC file to scan for all included prefixes, suffixes and beginning of last names. You can edit this file or create your own dictionary. See Appendix C: Customizing the Dictionary File for more information.

Address Parse

This type instructs DataImport to translate parts of an address. Addresses can be parsed into City, State, and Zip/Postal code, i.e. Atlanta/Georgia/30301. This data type can also be customized, and one or all of the address elements can be selected. Address Parse uses a DEFAULT.DIC file to scan for all state abbreviations. You can edit this file or create your own dictionary. See Appendix C: Customizing the Dictionary File for more information.

Signed Overpunch Numbers

Some computer systems use special notation to indicate positive and negative numbers. By assigning special characters to either the first or last digit of a number, a program can indicate whether a number is positive or negative. This helps conserve file space; rather than outputting the minus sign, it is “coded” into the number as a signed overpunch technique.

To specify the characters used to indicate the sign of the number, select **Options > Signed Overpunch**. DataImport provides three options for designating the characters that are used to indicate that a number is positive or negative:

0-9, }-R 0 → 9 as positive, } → R as negative

{-I, }-R { → I as positive, } → R as negative

Custom (user-defined)

One of the first two options translates the majority of Input Files using signed overpunch correctly. Selecting the Custom option displays all 20 possible digits and the current character assignment. To create a custom format, select the Custom option under Character Set and then in the Characters field, change the characters to match those used in the Input File. To save your custom Signed Overpunch definitions as the default for all new masks, press the Save as defaults button. If you want to use this saved character set at later date, simply select **Options > Signed Overpunch** and press the Load defaults button. The custom character set will also be saved when you save the Mask.

DataImport can use either the leading (first position) or trailing (last position) digit of a number as the signed overpunch. To check or change the setting, select **Options > Signed Overpunch** and mark either the Trailing or Leading options in the Position field.

Code Page Settings

ASCII stands for American Standard Code for Information Interchange. It is a specification that determines how bytes that are numbers are converted to characters. The ASCII specification specifies characters 0 through 127. The characters are control codes, numbers (0-9), lowercase letters (a-z), uppercase letters (A-Z), and punctuation common to the English language. Character Numbers 128 through 255 are not defined.

When IBM and Microsoft developed MS/PC-DOS for the IBM PC they developed extensions to the standard ASCII characters that define characters numbered from 128 to 255. The Extended ASCII character set contains line-drawing characters, symbols, and a small set of punctuated letters used in non-English languages. Punctuated letters are those that are made up of a letter and a diacritical mark, for example, á, Ç, ú, Ü, and Ä.

In version 3.3, IBM and Microsoft added National Language Support to DOS because the original Extended ASCII character set did not contain all the punctuated letters that would be necessary to support other languages. They developed a series of Code Pages to support the other languages. Code Pages are essentially a specification of what character to display for a given byte. All Code Pages share the first 128 characters (the original ASCII specification), but have different characters for the range of 128-255. Most of the Code Pages kept the same line-drawing characters. You can look in the back of your MS-DOS User's Guide to see the characters that are defined in each Code Page.

The Supported Code Pages are:

- 437 - US (Extended ASCII)
- 850 - Multi-Lingual (Latin 1)
- 852 - Latin II
- 860 - Portugal
- 862 - Hebrew
- 863 - Canadian-French
- 865 - Nordic
- Windows ANSI

Microsoft Windows does not use Code Pages, instead it uses fonts. Most fonts use the ANSI character set, which is not the same as any of the previous DOS Code Pages. The ANSI character set uses the ASCII codes for the first 128 characters. The last 128 characters contain symbols and a full set of punctuated letters, both uppercase and lowercase. Courier and FixedSys are examples of fonts that use the ANSI character set. The Terminal font supplied with most versions of Windows does not use the ANSI character set, it uses the Code Page 437 character set.

To assist in reading files created on systems using a different Code Page than the standard US (Code Page 437), you can set what Code Page rules will work correctly. Also the Code Page defines how translation to Lotus products will be conducted. Lotus products use either the Lotus International Character Set (LICS) or Lotus Multi-Byte Character Set (LMBCS). With the Code Page set correctly in the mask, DataImport will do the necessary code Page translations to ensure that the output file has the correct characters.

Performing Calculations

Depending on the type of analysis or report you are creating, you may need to perform calculations on the data you extract from a report. DataImport provides functions to automatically output rows with Sum, Average or Count formulas.

DataImport can include formulas only in spreadsheet Output File types. Formulas are output as rows that can replace existing rows in the Input File, or they can be output as additional rows. These *Formula Rows* can be output either when the information in a specific column changes, or when a specific character match string is found.

Formulas in Columns

In order to properly insert Formulas in a spreadsheet, DataImport needs to know what type of calculation should be performed for each column. You define the type of calculation you want in the settings of a column.

To define the type of calculation to perform on a column, click in the column, select **Column > Settings** and select a formula type from the @Function pull-down menu.

Inserting Formula Rows

After you have defined the formulas you want in each column, you must then specify when DataImport should output a Formula Row. A *Formula Row* is a row DataImport inserts in an Output file which contains formula cells that calculate a Sum, Average or Count of the cells above.

Formula Rows can be inserted when the data in one column changes or when a match string is found. DataImport can also replace a line in the Input File with a Formula row.

Inserting Formulas at a Column Change

You may want to insert a Formula Row every time the data in a particular column changes. DataImport allows you to insert a Formula Row into the Output File each time the character contents of a non-blank cell in a specified column changes. This option is useful when the Input File is a list of records in a sorted order without subtotals.

For example, let's say you have a report that lists all of the data from the Southwestern region on consecutive lines and the first column (A) lists SOUTHWEST for each entry. The first column then changes to NORTHWEST and lists all the data from the Northwest region. You want to get the totals for each region—SOUTHWEST, NORTHWEST, etc.—so you need to insert a Formula Row each time the name in the first column changes.

To insert a Formula Row based on a change in the column, select **Options > Formula Rows > Column Change**, type in the Column Letter—in this case, “A”—and press OK. DataImport will now insert a formula each time the data changes in the column you specified (in the example above, when SOUTHWEST changes to NORTHWEST and when NORTHWEST ends or changes to something else).

The column to be tested for a change in contents can be a normally defined column or a Line Tag column. A dashed line is output before, and a blank line is output

after, the formula line. This formatting makes it easy to determine where DataImport has inserted formulas.

Inserting Formula Rows with Match Strings

DataImport allows you to output a Formula Row each time a specified match string is encountered during translation. The match string can be defined to require an exact match, or a pattern match using wildcard characters.

This option is useful when a report has groups of information without subtotals that is either on different pages, or is separated by headings. DataImport can be instructed to insert formula cells in rows before each new page or heading.

To insert a Formula Row based on a match string, select the text to be used as a match string and select **Options > Formula Rows > Insert on Match**. If the match string is always the same character sequence, press OK. To define a pattern of characters, edit the Original String by including the appropriate wildcard characters. If the string must occur at the same line position as the original string, then in the Position on line field, mark the At Position option, otherwise, select Anywhere.

Replacing Lines with Formula Rows

A Formula Row can replace the cell contents of an Input File line each time a specified match string on the line is encountered during translation. The match string can be defined to require an exact match, or allow for a pattern match using wildcard characters.

Many reports already contain totals. The **Options > Formula Rows > Replace on Match** command can be used to replace the literal totals in an Input File with the formulas for the totals. This change can facilitate “what if” analysis in spreadsheets by showing the new total after a change is made to a detail line.

To replace a line with a Formula Row based on a match string, select the text to be used as a match string and select **Options > Formula Rows > Replace on Match**. If the match string is the exact same character sequence occurring at the same line position, press OK. To define a pattern of characters, edit the Original String by including the appropriate wildcard characters. If the string must occur at the same line position as the original string, then in the Position on line field, mark the At Position option, otherwise, select Anywhere.

Working with Database Files

Relationship Between Columns and Fields

A field in a database serves much the same function as a column does in DataImport. The information in a defined column will be sent to the corresponding field in the database file. For example, data in column A will go to the first field, column B's data will go to the second, etc.

Assigning Columns to Database Fields

When creating a new database or mail merge Output File, the DataImport column names are used as the field names. The column names initially default to be the column letters. The column names can be changed by clicking the mouse button

anywhere in the column, selecting **Column > Settings** and typing a new name in the Field Name field.

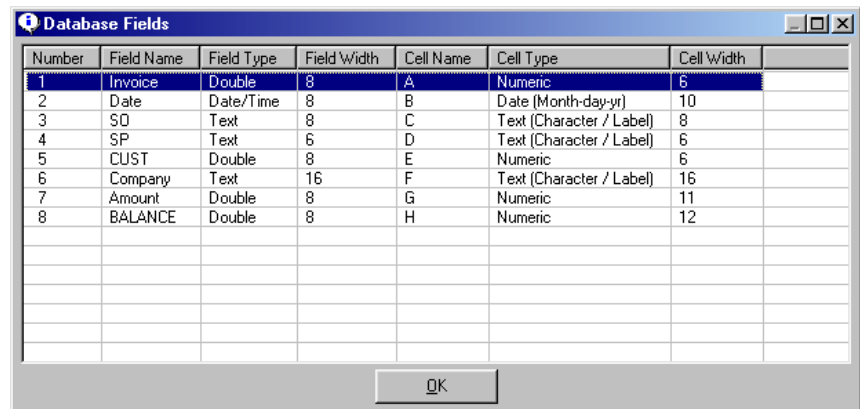
After the column name is entered, the column indicator bar above the Input File display window will include the column letter and the column name. For example, if column B is named “CITY”, the column indicator bar will display “B:CITY” above the column.

If the output database already exists, then you can only select from the available field names in the existing output file. This feature prevents existing database files from being corrupted with conflicting fields and data. To create a database with a new structure, select **File > Define Output File** type in a different name for the new database and press OK, or delete the existing database file.

Displaying the Database Structure

DataImport offers two ways of displaying the structure of an existing database file. Both methods show field name, length, type, and the column letter that corresponds to the field in the mask.

When outputting to an existing database file in the Mask application, you can display the database structure of the existing file by choosing the **File > Show Database Fields** command. A dialog box will appear showing the current field names, field types, field lengths and their relationship to the existing column definitions:



Number	Field Name	Field Type	Field Width	Cell Name	Cell Type	Cell Width
1	Invoice	Double	8	A	Numeric	6
2	Date	Date/Time	8	B	Date (Month-day-yr)	10
3	SO	Text	8	C	Text (Character / Label)	8
4	SP	Text	6	D	Text (Character / Label)	6
5	CUST	Double	8	E	Numeric	6
6	Company	Text	16	F	Text (Character / Label)	16
7	Amount	Double	8	G	Numeric	11
8	BALANCE	Double	8	H	Numeric	12

Figure 4-16 Database structure displayed with the **File > Show Database Fields** command in the Mask application.

The DataImport Utilities application can also extract the structure of a dBase format database file from the header information. To output the structure of the database to a file, use the dBase header function.

Specifying Table Names Access (MDB) format databases use Tables within files. You can specify an existing table from a file or specify a table name. If no table name is specified, DataImport will use “Table1”. To specify the table name for an Access Output File, select **File > Define Output File**, make sure a Microsoft Access output type is defined and specify a table name in the Table Name field.

Translating to an Existing Database

Caution: Except for Microsoft Access, any indexes associated with an existing database must be regenerated following output to the file. DataImport does not

update indexes when it performs a database translation. DataImport does update Access indices.

DataImport retains all characteristics of a database structure during a translation and only outputs information that is associated with a field. If more columns have been defined than existing fields in the database, then the information in the columns not associated with fields is not output. If fewer columns have been defined than fields in the database, some fields will remain blank.

When instructed to perform a translation into a database file, DataImport verifies that the Output File exists. If the file exists, you can specify that DataImport either append to the file by keeping the current records and adding new records, or completely replace the records already in the file.

You can set DataImport to automatically Append or Replace by choosing **File > Define Output File** and selecting one of these options from the Action when output exists menu.

Translating to a New Database

If a database file does not exist when a file is translated to a database Output File type, then DataImport creates the database structure by setting the field names, field length and field type according to the current definitions in the mask.

The default database field names written into the new database structure are the Mask's column letters (i.e., A, B, C, D,...). You can specify the Field Name of a column (data fields) as any valid field name (i.e., NAME, ADDRESS, etc.) by clicking in a column, choosing **Column > Settings**, and typing in a field name in the Field Name text field.

IMPORTANT!: If you are using DataImport to create the structure of the database and you redefine the mask to include more or fewer columns, change a field type or column width after translating, delete the database file and associated structure created by DataImport before proceeding. If you do not delete the first database, the structure will remain intact and DataImport will not alter the structure even if you change the mask. Alternately, you can change the name of the output file to a new name that does not exist.

Creating Records from Report Files

Some reports contain information in a heading at the top of the page that needs to be included on each line. Columns containing this heading information can be added to each line using the Mask application's Line Tag feature.

Some reports leave the column blank if it contains the same information as the preceding line. Information from previous lines can be automatically copied into the blank cells on each line in a column by choosing **Column > Settings** and marking the Fill-down option for each column.

Chapter 5: DataImport Mask Reference

The contents of the selected input file are displayed in the mask application. The Mask application allows the user to select the data to be extracted, format it should be converted to, and order it should be written to the output file.

Running the Mask Application

The Mask Application can be run from the DataImport Menu Panel.

Either of these buttons will run the Mask Application.

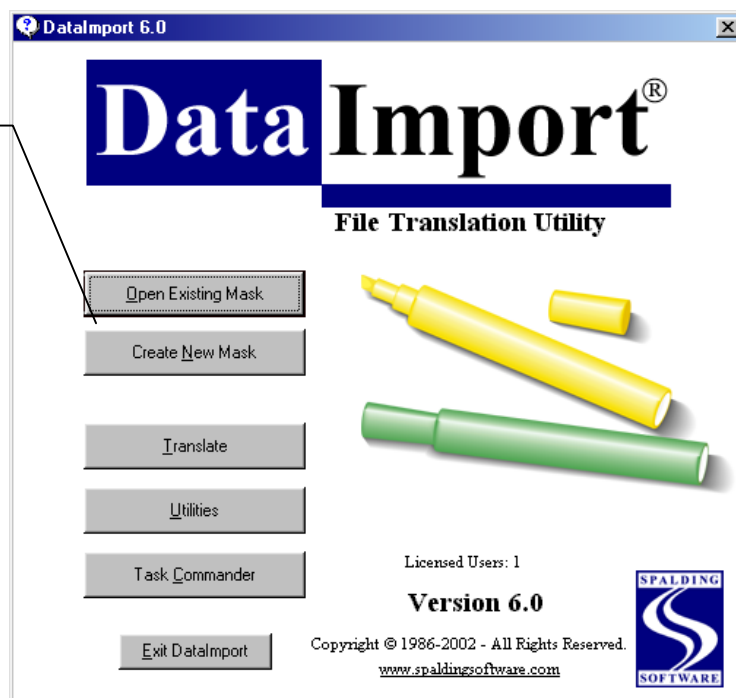


Figure 5-1 The DataImport Menu Panel

The Mask application can also be run from the Mask button in the other DataImport Applications.

Load Input File

Look in:

- ins_mrch.pri
- INVEST.PRN
- INVOICES.TXT
- UNSTACK.txt
- working.avi

File name:

Files of type:

The Input File containing the data to be extracted is displayed in the mask window.

Dataport Mask: (Untitled)

File Search Column Tag Include Exclude Line Unstack Options Help

Line: 11 Position: 11 Ascii: 32 Width: 11 Input File: c:\diw32\invest.prm

Font: Fixedsys 9 pts

Allegro Investment Company

Date: 11/30/83 Fixed Rate Investments

Acct: 0604 Col. Steve Nixon 3670 Park Ave. New York NY 11001

Rep: Bill Boniker Office: Lansing Tel: 909-555-4323

Investment	Matures	Value	Interest
Alphonatex	23 JUL 15	234,020	7.62
Hoag	01 APR 89	10,090	5.91

Acct: 19476 Mr. Reginald Cook 221 B. Baker St London England MK6 4AG

Rep: Jonathan Smythe Office: London Tel: 44 08 6 65051

Investment	Matures	Value	Interest
Hagerton	12 MAR 22	109,432	4.68
Mouton	01 SEP 28	436,690	5.67
Maxwell	02 NOV 35	36,690	6.90

Acct: 6485 Frank Braun Porsche Str 9 Bonn Germany 85221

Highlight an area and select action with menus or press right mouse button

DatalImport 6.0 User's Guide

File Menu

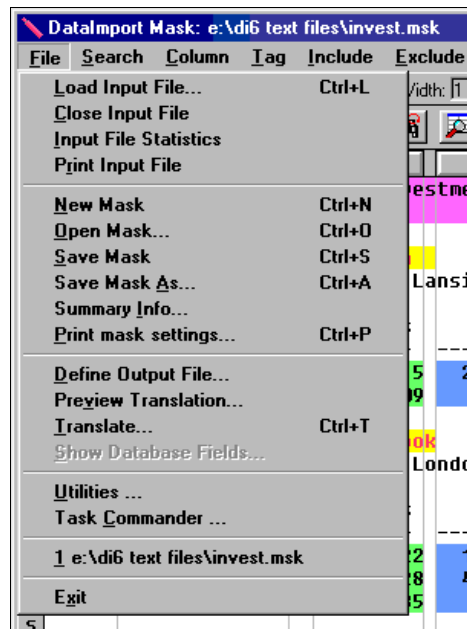


Figure 5-4 File menu selection.

By default the first 1,000 lines of the input file are loaded into the Mask window. The number of lines to be loaded is specified by selecting Preferences from the Options menu. The fewer lines that are loaded the faster the Mask window is updated.

File > Load Input File

Selects and then loads an input file to be translated. After an input file is chosen, it is displayed in the Mask window.

File > Close Input File

Closes the input file and removes it from the Mask window.

File > Input File Statistics

Displays information about the currently loaded input file. This includes the number of bytes, number of lines and the character width of the longest line in the file.

File > Print Input File

Prints the currently loaded input file.

File > New Mask

Clears memory of all mask columns, line treatments, settings and options. This deletes the current mask from memory and restores all mask selections to the default settings. It does not delete or change any Mask Files saved on disk.

File > Open Mask

Loads a previously saved Mask File.

File > Save Mask

Saves the current mask in memory to a Mask File. The Mask File is saved in the current mask directory.

File > Save Mask As

Saves the current mask to a Mask File with a new name and/or directory location.

File > Save Mask As is used when saving a Mask File for the first time, when naming a new version of an existing mask or for saving the current mask to a different directory. An extension of .MSK is automatically added to the Mask File name.

File > Summary Info

Displays a brief description and the author of the current mask and allows changing this information. This information is saved in the mask file.

File > View Mask Settings

Lists settings defined in the current mask. This option is very useful, particularly when the mask contains a lot of Include Lines, Exclude Lines and other settings. The report lists all settings, file names and column definitions. The mask settings can be printed from the viewer

File > Define Output File

Selects a name and file format for an output file. The Action when output exists option controls the procedure for saving output to a file that exists. The default is Warning, which allows the user to decide each time a file is translated.

File > Preview Translation

Displays in a table view a 200 line sample of the data that will be translated.

File > Translate

Translates the input file using the current mask and output settings. Usually used at the conclusion of a masking session to generate output for a spreadsheet or database.

File > Show Database Fields

Displays the database record structure along with a comparison to the columns in DataImport. This option is available only when the type of translation is a database and the Output File exists. Information is displayed about the existing database file that can help match fields in the records with the proper columns in DataImport. The example below illustrates how the database record structure is displayed.

If your Mask is not selecting the information that you expected, try viewing the mask settings and reviewing the listing.

*If your output file type is a database like Access, dBase or please be sure to read the section titled **Working with Database Files** in Chapter4.*

Search Menu

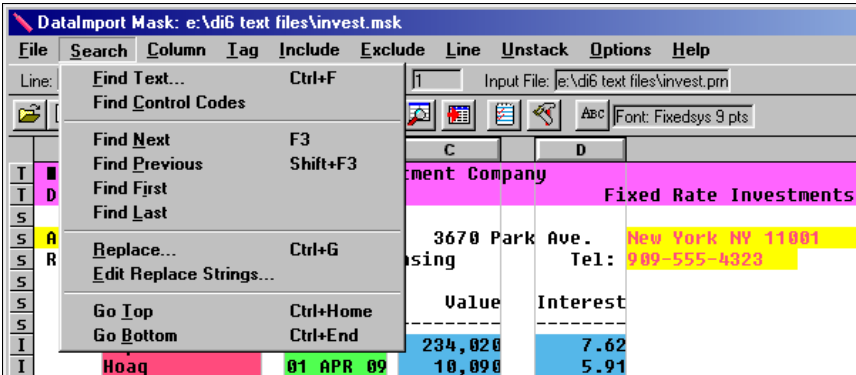


Figure 5-6 Search menu

Search > Find Text

Searches for a specified text string within the current input file. The search will find the specified text as a single word or within a word. For example, a search for “on” will find “on”, “iron” and “control.”

Search > Find Control Codes

Searches for a control code within the current input file.

Control codes are characters that have an ASCII value less than 32. These codes are typically used to control printer functions on older computer systems.

To find a series of control codes, use the **Find > Next** command to proceed through a number of these codes. Exclude the display and translation of these characters by using the **Exclude > Characters > All Special Characters** command. To exclude a specific control code, highlight the character and use the **Search > Replace** command.

Search > Find Next

Searches for next instance of current Find match string. This command is useful for locating multiple instances of a text string in a large input file. The shortcut key for this command is <F3>.

Search > Find Previous

Searches for previous instance of the Find match string. This command uses the currently defined criteria for the search.

Search > Find First

Searches for instance of the Find match string closest to the beginning of the Input File. This command uses the currently defined criteria for the search.

Search > Find Last

Searches for instance of **Find** match string closest to the end of the Input File.

This command uses the currently defined criteria for the search.

Search > Replace

Finds specific characters in the input file and replace them with user defined characters. **Search > Replace** is identical to the **Exclude > Characters > Define** command.

Search > Edit Replace Strings

Edits the Replace String set up in the **Search > Replace** menu. After a replace function has been setup DataImport will execute that function each time the input file is loaded.

Search > Go Top

Searches for the beginning of the input file.

Search > Go Bottom

Searches for the end of the input file.

Column Menu

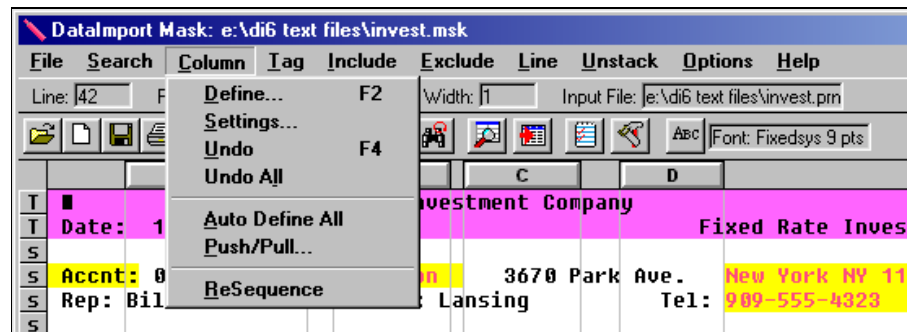


Figure 5-7 Column Menu

Column > Define

Defines a new column based on the position of the currently highlighted string of characters.

Columns cannot overlap. The data within the defined column is extracted to the same column in a spreadsheet, or the same field in all records of a database. When a column is defined, the Column Settings dialog box appears.

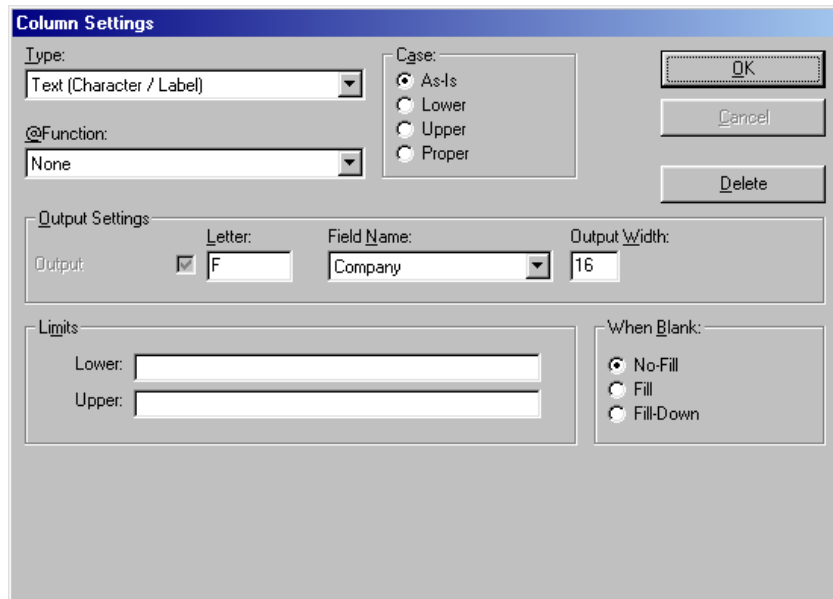


Figure 5-8 Column Settings Dialog Box

Type indicates the kind and/or format of data to be extracted. When a column is initially defined, the column's data type defaults to the column type specified in the **Options > Preferences** menu.

Numeric attempts to format data as numbers. In addition to numeric digits, it also recognizes symbols used to format numbers, such as decimal points, thousands separators, currency symbols, and negative indicators. To change these symbols, select **Options > International** and select the appropriate symbols. If non-numeric data is encountered in the cell, DataImport formats the data as Text.

Text (Character/Label) formats the data as alphanumeric text.

Text Left Justified is the same as Text Character/Label except that it removes blank spaces from the left of the text.

Text Block formats multiple lines of data into the same cell. Text Block keeps adding data from multiple lines within the same column into the same cell (or field) during translation until the next line is to be output or a blank cell in the column is encountered. You can also specify the Block to be a specified number of lines.

Date (Month-day-yr) attempts to recognize data as a date that is printed in month day year order into the date format of spreadsheets and databases. Date settings can be modified by selecting **Options > International** and **Options > Dates**.

DataImport will attempt to format this data as dates first, then text.

Date (Day-month-yr) Similar to Date Month-day-yr.

Date (Yr-month-day) Similar to Date Month-day-yr.

Date (Month-yr) Similar to Date Month-day-yr.

Date (Yr-month) Similar to Date Month-day-yr.

Date (Yr-day) Similar to Date Month-day-yr.

Date (Day-yr) Similar to Date Month-day-yr.

Date Custom will format data as a custom date defined by the user in the **Options > Dates** dialog box as a format without separators.

Time attempts to format data as time, then numeric, and text last.

Signed Overpunch attempts to recognize data in the input file that is signed overpunch and translate the data as numeric. The signed overpunch strings are determined by **Options > Signed Overpunch**.

Name Parse (Last, First) attempts to recognize the individual parts of a name that consists of a last name followed by a comma and the remaining parts of the name. During translation it outputs the data into separate columns for the Prefix, Last Name, Middle Name, First Name, and Suffix. You can have one, several, or all of these fields defined as separate columns.

Name Parse (First Last) attempts to recognize the parts of the name in their natural order, such as a name would appear in an address. It outputs the data into separate columns as described above.

Address Parse outputs data into separate columns for the City, State, and Zip/Postal Code.

Case determines how text data will be capitalized. There are four options; As-is, Lower, Upper, and Proper (which capitalizes the first encountered character and the first character after each blank space).

Implied Decimals is a specified number of decimal places that can be applied to numeric data in the input file. For example, the number 34596 with Implied Decimals set to 2 would be translated as 345.96. If the number already has a decimal point, it is not changed.

Letter controls the sequence in which the columns will be output, regardless of their actual position in the Input File. Data can be output to any column in a spreadsheet. Columns can be skipped. The column's sequence defaults to the order in which they are defined or selected. For example, if a new column is defined between existing column A and existing column B, it will default to column C.

Name specifies the name of the database field to which the column will be output. If the output database file exists, an option menu lists the field names in the existing database. If the database does not exist, the user can type in the name of the new field or let it default to the letter of the column. A name can also be specified when outputting to other file types. Optionally the names of the columns can also be output to the first row of the spreadsheet file.

Output Width specifies the character width of the translated column in the output file.

When Blank controls how DataImport deals with blank cells or records in existing files.

No-Fill does not write anything into the cell.

Fill writes blank data into the cell.

Fill-down writes the same data from the last filled cell above it.

Some reports do not duplicate information from line to line if the information stays the same. Only the first occurrence of the information is included; successive cells contain blanks until the contents change. The following is an example of this type of report:

W E E K L Y S A L E S R E P O R T			
CITY	SALES PERSON	CUSTOMER	AMOUNT
ATLANTA	JOHN DOE	18321	10,245
		24356	6,322
BOSTON	SAM JONES	17623	28,995
	MARY SMITH	16590	2,056
		08842	18,545

Figure 5-9 Input File with blank cells

If the columns city and salesperson are defined with the Fill-down option, the data will be translated with the following format:

W E E K L Y S A L E S R E P O R T			
CITY	SALES PERSON	CUSTOMER	AMOUNT
ATLANTA	JOHN DOE	18321	10,245
ATLANTA	JOHN DOE	24356	6,322
ATLANTA	SAM JONES	17623	28,995
BOSTON	MARY SMITH	16590	2,056
BOSTON	MARY SMITH	08842	18,545

Figure 5-10 Output File with blank cells filled-down

@Function defines the mathematical formula to be calculated with the column data. The inclusion of formulas is controlled with the **Options > Formula Rows** commands.

Limit defines a range of data to be extracted. For example, a range of dates from March 15, 1992 to December 30, 2001 or a range of dollar amounts between 1 million and 10 million. Limit also works with labels by allowing you to select alphabetical ranges. For example, part numbers starting with DX to parts starting with DZ.

Column > Settings

Displays the column settings dialog box for the column in which the cursor rests. Column settings can also be changed by pressing the column control button at the top of the column or by double clicking in the column. The width of a column can be changed by dragging the left or right edge of a column control button.

Column > Undo

Removes the column in which the cursor rests.

Column > Undo All

Removes all columns. Use this command to clear all columns from a mask without removing other mask definitions. Use **File > New Mask** to remove all mask settings.

Column > Auto Define All

Defines columns automatically based on the patterns found in the current line that the cursor is positioned on. The Input File is re-displayed with the new column positions. All previously defined column positions are disregarded. New columns are defined for the entire length of the line used as the pattern.

DataImport can be set to automatically define columns when an Input File is loaded. Turn AutoColumn function on or off by choosing **Options > Preferences**, and changing the Automatically define columns option.

Column > Push/Pull

Moves all columns to the right of the current cursor location a specified number of character positions in either horizontal direction. Use this command to ‘push or pull’ some or all of the mask columns to the left or right.

Column > Resequence

Automatically sequences columns in their natural order left to right. Note: if line tags are in use, they become the first columns.

Tag Menu

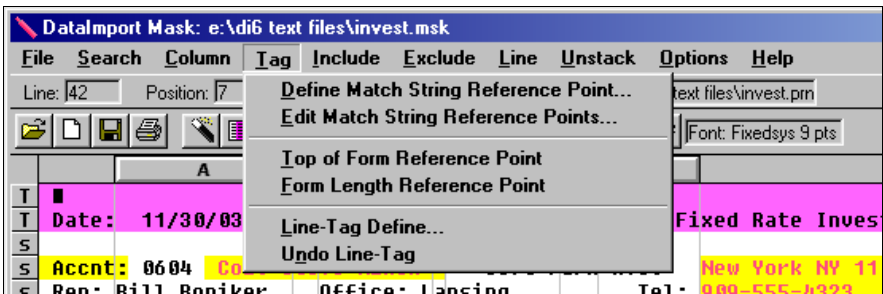


Figure 5-11 Tag Menu

Tag > Define Match String Reference Point

Creates a Reference Point for Line Tags based on selected text. Highlight the text to be used as a reference point, select Tag > Define Match String Reference Point, the following dialog is displayed.

“Wild Cards” can be used if the text changes from line to line.

A reference point can occur before or after the output line.

Check this box if you only want each line tag output once.

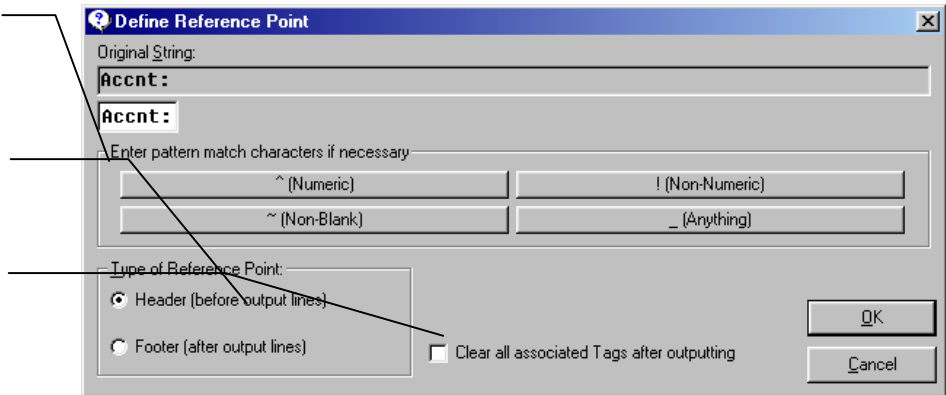


Figure 5-12 Define Reference Point Dialog Box

Match String Reference Points are used in conjunction with Line Tags to extract information from forms and other reports where information is located at specific positions on each page. They are also used to output information from page headings to each detail line. Reference Points serve as a fixed position from which

DataImport uses to find information that is located in a physical relationship to the Reference Point.

Header Reference Point

For example, on a form that repetitively prints the text “Month” at a certain position on each line that is then followed by a date such as “FEB 2045”, the text “Month” would serve as a header Reference Point as shown below.

Line tag data to be included on each detail line.

“Month” is a header reference point.

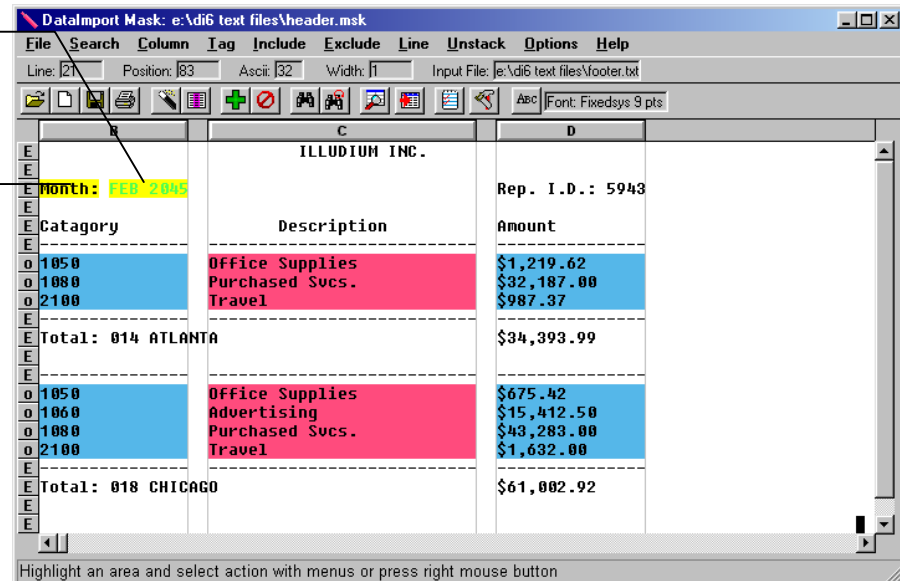


Figure 5-13 Mask with Header Reference Point Defined

Following is the file after translation to Excel.

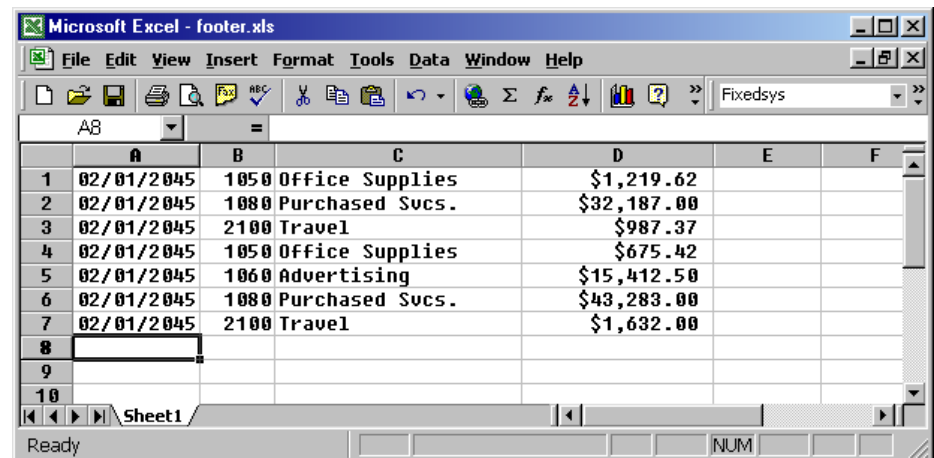


Figure 5-14 Output file with a Header Reference Point

During translation, the occurrence of a Reference Point causes the Line Tags associated with it to be refreshed. The Reference Point match string can be defined to require an exact match, or a pattern match using wildcard characters.

Reference Points are displayed as a black text on a yellow background. Up to one hundred Reference Point match strings can be defined for a mask. Only one Reference Point is allowed on a line.

A Reference Point must be defined before any tags to be associated with it can be defined. When a Line Tag is being defined, the Reference Point it is associated with is selected.

Footer Reference Point

A footer reference point works in a similar manner as a header reference point with a few exceptions. DataImport reads ahead in the input file to find a footer reference point before the data is output. In the following example, the text “Total” is the reference point with two line tags attached to it. These line tags will create columns A and B in the output file.

“Total” is a footer reference point.

Line Tag data to be included on each detail line.

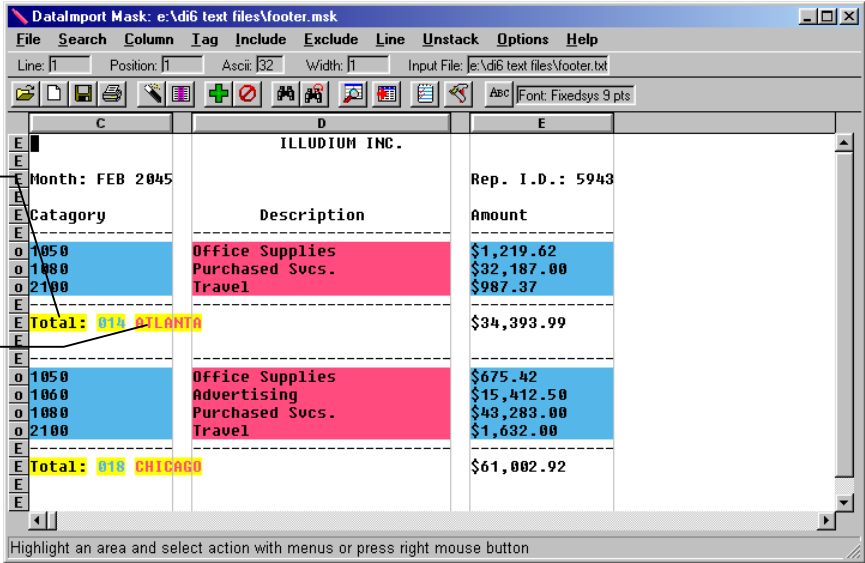


Figure 5-15 Mask with Footer Reference Point Defined

In the above example, the occurrence of the reference point is after the output lines. A footer reference point will produce an output file that looks like the following.

Results of pulling footer data up onto the lines above it.

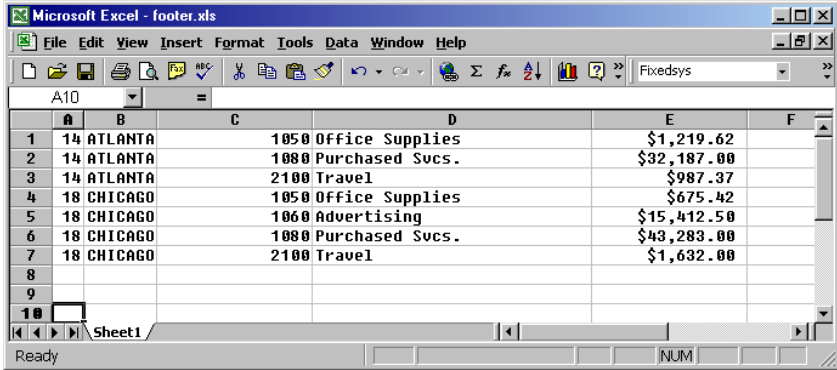


Figure 5-16 Output File with a Footer Reference Point

Caution: If the same reference point, “Total,” is set up as a header reference point the result will be an output file with jumbled data as shown in figure 5-13.

Results of failing to properly specify a reference point as occurring in a footer.

	A	B	C	D	E	F
1			1050	Office Supplies	\$1,219.62	
2			1080	Purchased Svcs.	\$32,187.00	
3			2100	Travel	\$987.37	
4	14	ATLANTA	1050	Office Supplies	\$675.42	
5	14	ATLANTA	1060	Advertising	\$15,412.50	
6	14	ATLANTA	1080	Purchased Svcs.	\$43,283.00	
7	14	ATLANTA	2100	Travel	\$1,632.00	
8						
9						
10						

Figure 5-17 Output File with an Incorrect Header Reference Point

As the above example shows the data in columns A and B has not been output correctly. This is because a header reference point is only for use when it occurs before the output lines.

The example below shows both header and footer reference points being used in the same mask.

Month is a header reference point.

Total is a footer reference point

D	E	F
	ILLUDIUM INC.	
Month: FEB 2005		Rep. I.D.: 5943
Catagory	Description	Amount
1050	Office Supplies	\$1,219.62
1080	Purchased Svcs.	\$32,187.00
2100	Travel	\$987.37
Total: 14 ATLANTA		\$34,393.99
1050	Office Supplies	\$675.42
1060	Advertising	\$15,412.50
1080	Purchased Svcs.	\$43,283.00
2100	Travel	\$1,632.00
Total: 14 CHICAGO		\$61,002.92

Figure 5-18 Mask with both Header and Footer Reference Points

When output the data from the line tags associated with the reference points will be output in columns A, B, and C as shown in figure 5-15. When using header and footer reference points the mask scans for all of the reference points before outputting the data on the output lines.

	A	B	C	D	E	F
1	14	ATLANTA	02/01/2045		1050 Office Supplies	\$1,219.62
2	14	ATLANTA	02/01/2045		1080 Purchased Svcs.	\$32,187.00
3	14	ATLANTA	02/01/2045		2100 Travel	\$987.37
4	18	CHICAGO	02/01/2045		1050 Office Supplies	\$675.42
5	18	CHICAGO	02/01/2045		1060 Advertising	\$15,412.50
6	18	CHICAGO	02/01/2045		1080 Purchased Svcs.	\$43,283.00
7	18	CHICAGO	02/01/2045		2100 Travel	\$1,632.00
8						
9						

Figure 5-19 Output File with both a Header and a Footer Reference Point

The output file in figure 5-15 shows the line tag data extracted using the footer reference point “Total” in columns A & B and the line tag data extracted using the header reference point “Month” in column C.

Tag > Edit Match String Reference Point

Allows deleting existing Reference Points and editing of the match strings.

Tag > Top of Form Reference Point

Creates a Reference Point at the top of each page. This is useful when your input file has one form per page.

Tag > Form Length Reference Point

Creates a Reference point every specified number of lines. This is useful when you have a set number of lines per form.

Tag > Line-Tag Define

Creates a Line Tag based on the position of the highlighted string of characters. If the cursor is positioned within an existing Line Tag; the tag’s settings are displayed.

Tag Settings

Type: Text (Character / Label)

@Function: None

Case: ☒ As-is, ☐ Lower, ☐ Upper, ☐ Proper

Output Settings: Output ☒ Letter: E Name: Output Width: 19

Reference Point: Accnt

☐ Line-Tag occurs before Reference Point

Buttons: OK, Cancel, Delete

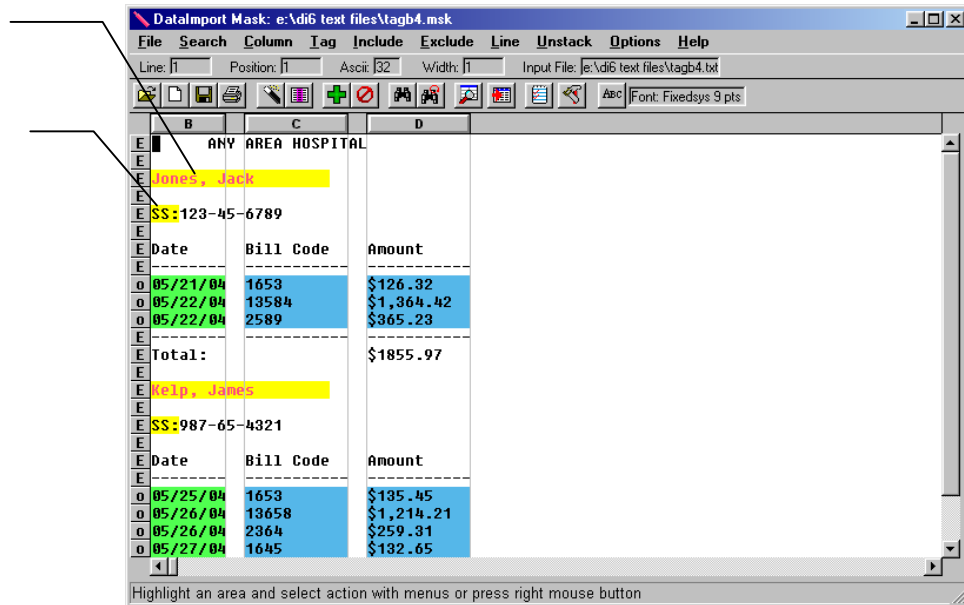
Figure 5-20 Line Tag Settings Dialog Box

At least one Reference Point must be defined before a line Tag can be defined. Select the reference point for the line tag and whether or not the line tag occurs before the reference point. See **Column > Define** for further explanation of the options in the tags setting dialog box.

The Line Tag function inserts a new column into the Input File. Line Tags repeat the same information on each output line until an associated Reference Point is encountered, at which point the Line Tag information is updated. At each occurrence of a Reference Point, only the information for the Line Tags associated with that Reference Point are updated.

Line Tag

A reference point occurring after its associated line tag.



B	C	D
ANY AREA HOSPITAL		
Line Tag: Jones, Jack		
SS: 123-45-6789		
Date	Bill Code	Amount
05/21/04	1653	\$126.32
05/22/04	13584	\$1,364.42
05/22/04	2589	\$365.23
Total:		\$1855.97
Line Tag: Kelp, James		
SS: 987-65-4321		
Date	Bill Code	Amount
05/25/04	1653	\$135.45
05/26/04	13658	\$1,214.21
05/26/04	2364	\$259.31
05/27/04	1645	\$132.65

Figure 5-21 Reference Point occurring after the line tag.

Line Tags are displayed with a yellow background and a foreground color that indicates the type of data defined, with blue for values, pink for labels, green for dates and orange for times.

A reference point occurring before its associated line tags.

Line Tag

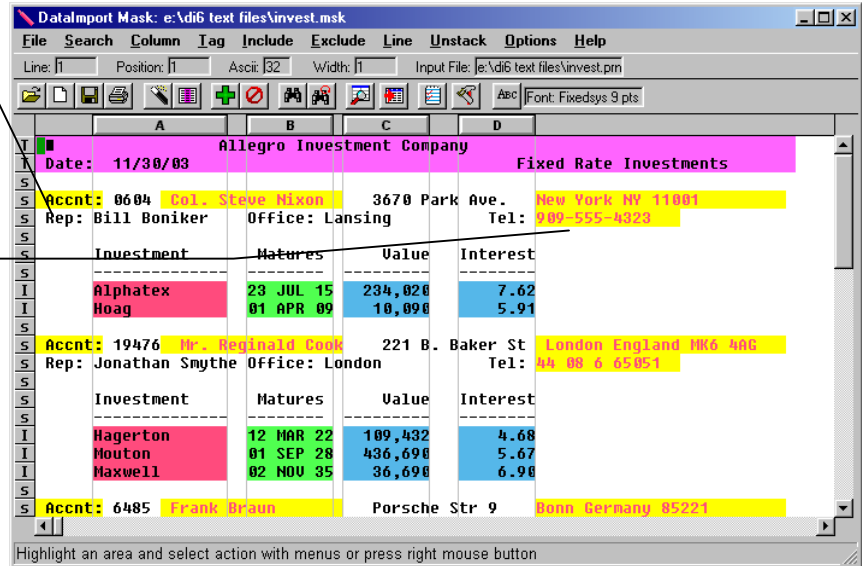


Figure 5-22 Reference Point occurring before the line tag

Tag > Undo Line Tag

Removes the line tag in which the cursor rests.

Include Menu

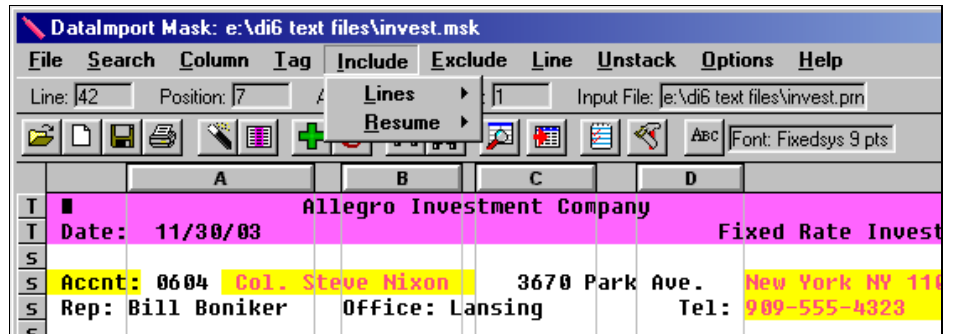


Figure 5-23 Include menu

Include > Lines > Define

Defines a specified number of lines to be included in the output file based on the occurrence of a specified character match string. Highlight the text string to trigger the include, select **Include > Lines > Define**, a dialog box will be displayed.

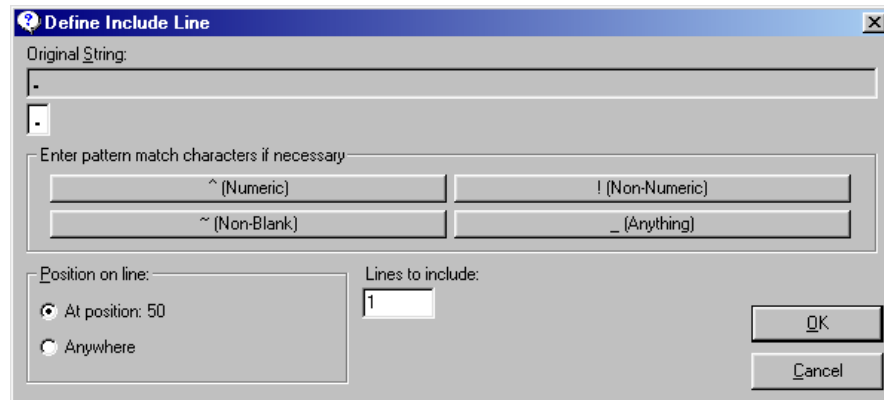


Figure 5-24 Define Include Line Dialog Box

The text string field under the Original String defines what characters must be present on a line in order for a line to be included. Use the special pattern matching characters as wildcards for searches:

- ^ single numeric character (0–9)
- ! single non-numeric character (AaZz\$%^&*” “)
- ~ any single character excluding a blank space (AaZz\$%^&*””)
- _ any single character or blank space (0–9,AaZz\$%^&*””)

Position on line controls where the text string can occur on a line.

At position indicates the text string must occur at the same line position as the original text string.

Anywhere indicates the text string can occur at any position on a line.

Lines to include controls how many lines are included for each occurrence of the Include match string. For values greater than one, additional lines are included immediately below the line where the Include text string occurs.

The lines to be included are displayed with background colors indicating how the cells will be translated. The line containing the Include match string is displayed with an uppercase “I” in the left margin. All additional lines in the associated range are displayed with a lowercase “i” in the left margin.

Lines that are specifically set as Skip, as indicated by an uppercase “S” in the left margin, will not be included during translation. Output, Title and Heading Lines will be translated.

In Global Output Lines Mode, Include Lines have precedence over Exclude Lines. That is, if an Include match string occurs in an Exclude Lines’ range, the Include Lines will be translated. In Global Skip Lines Mode, Exclude Lines have precedence over Include Lines.

Include Lines are usually used with the Global Skip Lines Mode option to select lines that have common information. Therefore, Global Skip Lines Mode is automatically activated when the first include line is defined.

Include > Lines > Edit

Removes specified Include Line definitions and allows editing of the previously defined match strings.

Include > Resume > Define

Restarts translation of rows after a Pause command when a specified match string is encountered. Use the Include Resume command in conjunction with **Exclude > Pause > Define** to extract blocks of information that occur over an indeterminate number of lines. Use the **Include > Line > Define** command to specify translation of blocks with a fixed number of lines.

The text string field under Original String defines what characters must be present on a line in order for translation to be restarted. Use the special pattern matching characters as wildcards for searches:

Position on line controls where the text string can occur on a line.

At position indicates the text string must occur at the same line position as the original text string.

Anywhere indicates the text string can occur at any position on a line.

Begin in Pause Mode controls whether or not the translation is assumed to be in pause at the beginning of a file. Marking this option essentially puts a Pause command at the beginning of the input file.

The Resume command restarts translation of rows of data after a Pause command has been defined in a previous row. Resume definitions are based on the occurrence of a text string in the Input File. For more information about the Pause command, see **Exclude > Pause > Define** on page 79.

Include > Resume > Undo

Removes the current Resume definition. Use this command to remove a previously applied Resume definition.

Exclude Menu

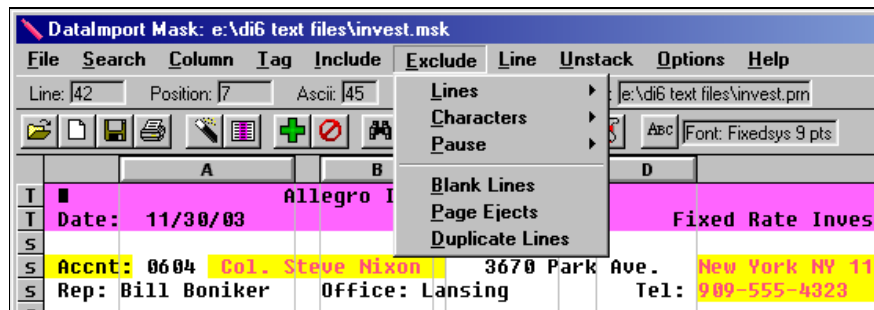


Figure 5-25 Exclude menu

Exclude > Lines > Define

Defines a specified number of lines to be excluded in the output file based on the occurrence of a specified character match string. Highlight the text to trigger the exclude, select **Exclude > Lines > Define**, the Define Exclude Line dialog box will be displayed.

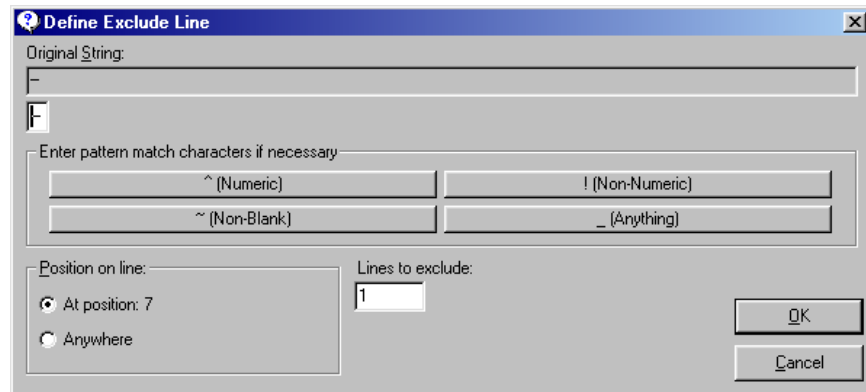


Figure 5-26 Define Exclude Line dialog box

The text string field under Original String defines what characters must be present on a line in order for the line to be included. Use the special pattern matching characters as wildcards for searches.

Position on line controls where the text string must occur on the line.

At position indicates the text string must occur at the same line position as the original text string.

Anywhere indicates the text string can occur at any position on a line.

Lines to exclude controls how many lines are excluded for each occurrence of the Exclude text string. For values greater than one, additional lines are excluded below the line where the Exclude match string occurs.

The lines to be excluded are displayed with a white background. The first line in an Exclude Line range is displayed with an uppercase “E” in the left margin. All additional lines in the range are displayed with a lowercase “e” in the left margin.

Exclude Lines are often used to eliminate the repetitive output of titles and headings after the first page of a report. On the first page of the report, the **Line (T)itle** and **Line (H)eading** commands can be used to indicate the page titles and column headings. The **Exclude > Lines > Define** command can then be used to exclude all subsequent occurrences of titles and headings.

Lines with a Heading or Title treatment will not be excluded during translation. Lines set as Output Lines—indicated by an uppercase “O” in the left margin—will not be excluded during translation.

In Global Skip Lines Mode, Exclude Lines have precedence over Include Lines. If an Exclude match string occurs in an Include Line’s range, the Exclude Lines will not be translated. In Global Output Lines Mode, Include Lines have precedence over Exclude Lines.

Exclude > Lines > Edit

Removes a specified Exclude Line definition and allows editing of the match string.

Exclude > Characters > Define

Excludes the highlighted character string from translation. Use this command to exclude a specific string of characters any time they are encountered during translation. Excluded characters can be optionally replaced with a different string of

characters. This command can be used to remove or replace printer control codes and “escape sequences”. This is the same as Search Replace.

Exclude Character String shows the originally selected character string to be excluded from translation.

Replacement String controls what characters are put in place of the excluded characters. The Replacement String is optional, but users may want to type the number of spaces equal to the length of the character string to preserve column and line spacing.

Position on line controls where the character string can occur on a line.

At position indicates the character string must occur at the same line position as the originally selected character string.

Anywhere indicates the character string can occur at any position on a line.

The Input File will be re-displayed, omitting the Excluded characters from all lines of the display. If defined, the replacement strings will be displayed.

Exclude > Characters > All Special Characters

Excludes all control characters, except the escape character (ASCII 27). ASCII character codes 0 through 31 in an Input File are generally formatting or control characters generated by the program that created the file. Such special characters interfere with the display of the file in the Mask window, causing misalignment of the columns.

Use this command to exclude all characters with ASCII codes 0 through 31. The escape character is not excluded automatically because many times the escape character is used to signal that one or more of the following characters are printer control codes. These are called “escape sequences.” Undesired escape sequences in the Input File can be removed by highlighting an occurrence of the the escape sequence and then choosing **Exclude > Characters > Define**.

Exclude > Characters > Edit

Removes a specified Exclude Characters definition and allows editing of the replacement string.

Exclude > Characters > Undo All Special

Removes a previous **Exclude > Characters > All Special Characters** command.

Exclude > Pause > Define

Suspends translation of any lines into the Output File when a match string is encountered.

Use the **Exclude > Pause** command in conjunction with **Include > Resume > Define** to extract blocks of information that occur over a variable number of lines. Use the **Include > Line > Define** command to specify translation blocks with a fixed number of lines.

The text string field under Original String defines what characters must be present on a line in order for translation to be suspended. Use the special pattern matching characters as wildcards for searches.

Position on line controls where the text string can occur on a line.

At position indicates the text string must occur at the same line position as the original text string.

Anywhere indicates the string can occur at any position on a line.

The Pause command stops translation of rows based on the occurrence of a text string in the Input File. The Pause definition suspends translation of the line on which the text string occurs and all lines thereafter until a Resume text string is encountered. Resume definitions are created using the **Include > Resume > Define** command.

Exclude > Pause Undo

Use this command to remove a previously applied Pause definition.

Exclude > Blank Lines

Skips empty lines in an Input File when translating. Use this command when the input file contains blank lines that you do not want in the output file.

Exclude > Page Ejects

Removes page ejects (form feed, ASCII character code 12) from data in the Input File during translation. This option should be selected if form feeds are present in the Input File and the translation is of spreadsheet, database, or interchange format. When printing, some software programs insert a form feed character as a page eject indicator and an end of line indicator. When DataImport excludes a form feed, it replaces it with an end of line indicator.

The Input File will be re-displayed, omitting all form feed characters from the display.

Exclude > Duplicate Lines

Removes lines that are exactly the same as the preceding line. Some programs on mainframe computers print a line, perform a carriage return without a line feed, and print the line again. This results in double striking or “bold” print used for emphasizing titles and headings on reports. This option removes the second line of print.

Line Menu

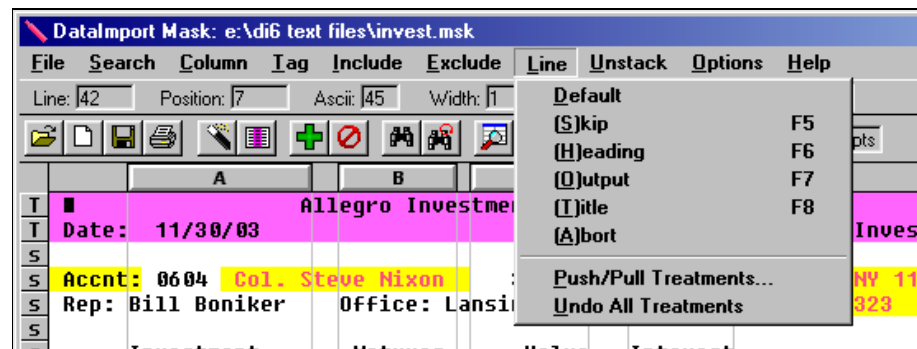


Figure 5-27 Line Menu

Line > Default

Resets line treatment to the default mode. The selected lines are set to the system default. The Line Control Bar will display a small “o” for each line if the mask is set to Global Output Line Mode or a small “s” if the mask is in Global Skip Line Mode. For more information about Global modes, see “Options Global” on page 88.

Use this command to remove treatments from a line that has been defined as Output, Skip, Heading or Title. All lines in a new mask default to the Global Mode setting. Line treatment defaults to the Global Mode setting until another line treatment (Output, Skip, Heading, Title, etc.) is selected or Include and Exclude Lines are defined.

Line > (S)kip

Defines lines to be ignored during translation. Skip Lines are positional. For example, if line 5 is set to Skip, every translation using this mask will not output line 5 of the input file, no matter what text is on the line.

An “S” appears in the Line Control Bar on the left margin to indicate that the lines will be skipped.

Line > (H)eading

Translates text within columns on the selected lines as column headings. During translation, each intersection of a Heading Line and a column results in a cell being output. The cell type is always text. When Unstacking data (**Options > Unstack**), Heading Lines are duplicated over the unstacked data.

An “H” appears in the Line Control Bar on the left margin to indicate that the lines will be treated as Headers. The headings are displayed on the screen with a pink background.

Line > (O)utput

Defines lines to be included in the translation. Output Lines are positional, not associated with a particular format or character string. If lines 2 and 3 are marked as Output Lines, then lines 2 and 3 are always output during translation, no matter what data is on the line.

An “O” appears in the Line Control Bar on the left margin to indicate that the lines will be treated as Output. Numeric values are displayed on the screen with a blue background, labels with a pink background, and dates with a green background. Lines defined as Output Lines include only data in defined column ranges, not data for the entire line. During translation, each intersection of an Output Line and a column result in a cell being output. The formatting assigned to the cell is based on each column’s Type setting.

Line > (T)itle

Translates the entire line as text or a single long label. Titles are commonly used when the Input File is a report. Most reports contain information at the top of each page such as the report name and date of printing. Defining lines as Titles keeps this information intact. Column selection does not affect the translation of Title Lines.

*The Mask window's **Exclude > Lines > Define** Command also skips lines during translation based on the occurrence of a match string.*

Heading line treatment can be used to make sure that numbers that are in column headings are not translated as values.

A “T” appears the Line Control Bar on the left margin to indicate that the lines will be translated as Titles. Title Lines are displayed on the screen with a red background. Use this command to define lines that should be translated as a single long label in the first column of your spreadsheet. Titles are not translated into database output files.

The **Exclude > Line > Define** command can be used to suppress the output of page titles after the first page.

Line > (A)abort

When you are defining a new mask for a large Input File, you can test your mask by temporarily defining an Abort line.

Defines an artificial end-of-file at the selected line. Defining a line as Abort causes DataImport to act as if it has reached the end of the Input File. The information translated up to the Abort Line is saved in the Output File. An Abort definition supersedes all other line treatments on the initial Abort line and all lines following it. Remove an abort line by selecting the line and then choosing **Line > Default**.

An “A” appears the Line Control Bar on the left margin and all following lines are marked with an “a” to indicate that these lines will not be translated.

Line > Push/Pull Treatments

Moves the selected line treatments further up or down in the mask.

Insert Default Line Treatments is typically used to modify an existing mask when additional lines are inserted into the body of a report. Inserting line treatments into the mask moves all previously defined line treatments down without changing them. The line treatments that are inserted are set to the default line treatment.

Delete Line Treatments is typically used to modify an existing mask when lines are removed from the body of a report. Deleting line treatments from the mask moves all previously defined line treatments up without changing them. Line treatments on the selected lines are discarded.

Line > Undo All Treatments

Resets all line treatments to the default line treatment. The default line treatment—Output or Skip—is controlled by choosing **Options > Global** and selecting either the Output lines or Skip lines option under Default Line Treatment.

Unstack Menu

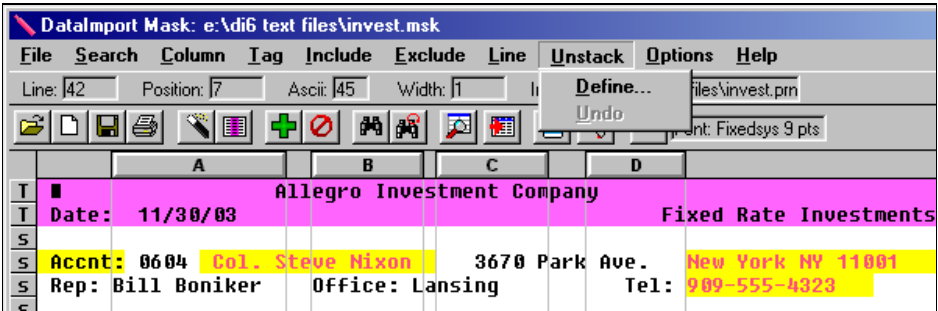


Figure 5-28 Unstack Menu

Unstack > Define

Turns sets of a specified number of stacked lines into single longer lines based on the occurrence of a match string. Only one unstack command is allowed per mask.

The following example illustrates how a report file with stacked lines looks before and after it is converted.

BRANCH	PERIOD	SALES
NEW YORK	MONTH	12,935
NEW YORK	YEAR	31,221
LONDON	MONTH	9,264
LONDON	YEAR	24,698
PARIS	MONTH	8,573
PARIS	YEAR	35,745
MUNICH	MONTH	14,684

Figure 5-29 Stacked Input File with columns and headings defined

In this example, the Unstack command was defined with 2 Lines to unstack with “MONTH” as the match string. The screen below shows the results of the Unstack command.

BRANCH	PERIOD	SALES
NEW YORK	MONTH	12,935
NEW YORK	YEAR	31,221
LONDON	MONTH	9,264
LONDON	YEAR	24,698
PARIS	MONTH	8,573
PARIS	YEAR	35,745
MUNICH	MONTH	14,684

Figure 5-30 Unstacked lines

To unstack lines only within a column, select the Text Block column type in the Column Dialog Box.

Notice that the yearly data has been moved into new columns to the right of the monthly data and that the Heading information has been duplicated over the new columns. To complete the mask definition, two new columns should be defined for the new yearly PERIOD and SALES columns.

If the Input File has a heading at the top of each page and a block of lines starts at the bottom of one page and carries over to the next page, it is necessary to perform two translations. The first translation is used to remove all of the headings from the file so that they do not interfere with unstacking. To do this, define columns for the desired data, exclude the headings (or include the desired lines), and translate to a print image (.PRN) file. The second translation uses the output of the first translation as a new clean input file.

The DataImport Utilities application will also unstack lines. It is useful when it is not possible to define a match string. This is often the case with address labels.

Unstack > Undo

Removes an unwanted or incorrect Unstack command.

Options Menu

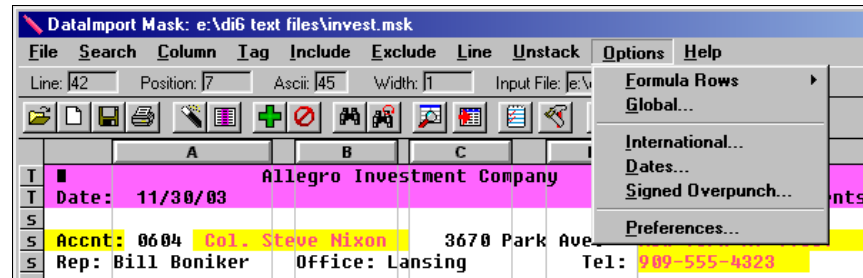


Figure 5-31 Options Menu

Options > Formula Rows > Column Change

Inserts a row of formulas based on a data change in a specified column. Select **Options > Formula Rows > Column Change**, the @Formula dialog box appears, type the column letter to be checked for data change, click OK to apply the Formula Row definition.

For example, the data in column A in the sample report below lists ATLANTA 4 times and then changes to SAN FRANCISCO. By defining a Formula Row Column Change based on column A, a subtotal formula line is inserted after the last ATLANTA for the ATLANTA data. The formulas are inserted and calculated again after the last SAN FRANCISCO for that group of data.

The screenshot shows the Datalport Mask application window with a report titled "WEEKLY SALES REPORT". The report has four columns: CITY, SALES PERSON, ORDERS, and SALES. The data is grouped by city, with ATLANTA appearing four times followed by SAN FRANCISCO. Subtotal lines are inserted after the last ATLANTA and after the last SAN FRANCISCO. The font is set to Terminal 12 pts.

CITY	SALES PERSON	ORDERS	SALES
ATLANTA	JANE DOE	125	5,622.87
ATLANTA	FORD PREFECT	117	5,089.45
ATLANTA	PAUL ATREIDES	246	11,070.65
ATLANTA	LAZARUS LONG	187	8,393.37
SAN FRANCISCO	DIRK GENTLY	29	1,419.65
SAN FRANCISCO	BILLY BATSON	310	13,950.00
SAN FRANCISCO	ZACHARY SMITH	49	2,112.98

Figure 5-32 Input File without subtotals

If all columns on this report are defined, and the CITY column (column A) is declared for the **Formula Rows Column Change** option, subtotals are inserted at the end of each CITY's listing. Below is the resulting report, in spreadsheet form. The spreadsheet actually contains the formulas, as indicated in cell B16.

Microsoft Excel - atform2.xls

File Edit View Insert Format Tools Data Window Help

A17 =

1	W E E K L Y	S A L E S R E P O R T			
2	C I T Y	S A L E S P E R S O N	O R D E R S	S A L E S	
3					
4	ATLANTA	JANE DOE	125	5,622.87	
5	ATLANTA	FORD PREFECT	117	5,089.45	
6	ATLANTA	PAUL ATREIDES	246	11,070.65	
7	ATLANTA	LAZARUS LONG	187	8,393.37	
8					
9			675	30176.34	
10					
11	SAN FRANCISCO	DIRK GENTLY	29	1,419.65	
12	SAN FRANCISCO	BILLY BATSON	310	13,950.00	
13	SAN FRANCISCO	ZACHARY SMITH	49	2,112.98	
14					
15					
16			388	17482.63	
17					
18					
19					

Sheet1

Ready

Figure 5-33 Output File with formulas for subtotals inserted

Notice that a solid dashed line is output before the formula row. After the formula row, a blank row is output. This formatting makes it easy to determine where DataImport has inserted formulas.

If no formula is defined for a column, its cell in the formula row is blank. To define or change an @Formula for a column, use the **Column > Define** command.

The column to be tested for a change in contents can be a normally defined column or a Line Tag column. DataImport calculates a Formula Row Column Change only if the Output File type is a spreadsheet.

Applying this function replaces a previous Formula Row definition. To check which (or if a) formula row definition has been set, look for a check mark next to an option on the **Option > Formula > Row** submenu.

Options > Formula Rows > Insert on Match

Inserts a row of formulas based on the occurrence of a specified text string. For example, the report below lists sales by region and then prints END REGION: after each region. By defining END REGION: as a match string, you can use the **Formula Rows > Insert on Match** function to insert @Formula subtotals before each occurrence of that text string. On the following report, the region name appears after the list of cities in the region.

CITY	SALES PERSON	ORDERS	SALES
ATLANTA	JANE DOE	125	5,622.87
BIRMINGHAM	FORD PREFECT	117	5,089.45
CHICAGO	PAUL ATREIDES	246	11,070.65
NEW YORK	LAZARUS LONG	187	8,393.37
END REGION: EAST			
SAN DIEGO	DIRK GENTLY	29	1,419.65
DENVER	BILLY BATSON	310	13,950.00
PORTLAND	ZACHARY SMITH	49	2,112.98
END REGION: WEST			

Figure 5-34 Input File without subtotals by region

If all columns on this report are defined, and the match string is defined as “END REGION:” for the **Formula Rows Insert on Match** command, subtotals are inserted before the match string. Following is the resulting report, in spreadsheet form. The spreadsheet actually contains the formulas, as shown in cell B18.

	A	B	C	D	E	F
1	WEEKLY	SALES	REPORT			
2						
3	CITY	SALES PERSON	ORDERS	SALES		
4						
5	ATLANTA	JANE DOE	125	5,622.87		
6	BIRMINGHAM	FORD PREFECT	117	5,089.45		
7	CHICAGO	PAUL ATREIDES	246	11,070.65		
8	NEW YORK	LAZARUS LONG	187	8,393.37		
9						
10			168.75	30176.34		
11						
12	SAN DIEGO	DIRK GENTLY	29	1,419.65		
13	DENVER	BILLY BATSON	310	13,950.00		
14	PORTLAND	ZACHARY SMITH	49	2,112.98		
15						
16			129.333	17482.63		
17						
18						

Figure 5-35 Output File with formulas inserted for each region

Notice that a solid dashed line is output before the formula row. After the formula row, a blank row is output. This formatting makes it easy to determine where DataImport has inserted formulas.

In the Define @Formula Match String dialog box, the text string field under Original String defines what characters must be present on a line in order for a formula row to be inserted. Use the special pattern matching characters as wildcards for searches.

Position on line controls where the text string can occur on a line.

At position indicates the text string must occur at the same line position as the original text string.

Anywhere indicates the text string can occur at any position on a line.

DataImport inserts formulas during translation only if the Output File type is a spreadsheet. This command replaces the selection of any other Row Formula option.

The type of formula that is output is dependent upon each column @Formula setting. If no formula is defined for a column, its cell in the formula row is blank. To define or change an @Formula for a column, use the **Column > Define** command.

Options > Formula Rows > Replace on Match

Replaces a line where a specified text string occurs with a row of formulas. For example, the report below lists sales by region and then prints END REGION: after each region. By defining END REGION: as a match string, you can use the **Formula Rows Replace on Match** function to insert @Formula subtotals that will replace the lines where the text string occurs.

CITY	SALES PERSON	ORDERS	SALES
ATLANTA	JANE DOE	125	5,622.87
BIRMINGHAM	FORD PREFECT	117	5,089.45
CHICAGO	PAUL ATREIDES	246	11,070.65
NEW YORK	LAZARUS LONG	187	8,393.37
END REGION: EAST			
SAN DIEGO	DIRK GENTLY	29	1,419.65
DENVER	BILLY BATSON	310	13,950.00
PORTLAND	ZACHARY SMITH	49	2,112.98
END REGION: WEST			

Figure 5-36 Input File without subtotals by region

If all columns on this report are defined, and the match string is defined as “END REGION:” for the **Formula Rows Replace on Match** command, subtotals are written on the line where the match string occurs. Following is the resulting report, in spreadsheet form. The spreadsheet actually contains the formulas, as shown in cell B16.

	A	B	C	D	E	F
1	W E E K L Y	S A L E S	R E P O R T			
2						
3	CITY	SALES PERSON	ORDERS	SALES		
4						
5	ATLANTA	JANE DOE	125	5,622.87		
6	BIRMINGHAM	FORD PREFECT	117	5,089.45		
7	CHICAGO	PAUL ATREIDES	246	11,070.65		
8	NEW YORK	LAZARUS LONG	187	8,393.37		
9			168.75	30176.34		
10						
11	SAN DIEGO	DIRK GENTLY	29	1,419.65		
12	DENVER	BILLY BATSON	310	13,950.00		
13	PORTLAND	ZACHARY SMITH	49	2,112.98		
14			3	129.333	17482.63	
15						
16						
17						
18						
19						

Figure 5-37 Output file with formulas replacing the end region line

In the Define @Formula Match String dialog box, the text string field under Original String defines what characters must be present on a line in order for it to be replaced by a formula row. Use the special pattern matching characters as wildcards for searches.

Position on line controls where the text string can occur on a line.

At position indicates the text string must occur at the same line position as the original text string.

Anywhere indicates the text string can occur at any position on a line.

DataImport inserts formulas during translation only if the Output File type is a spreadsheet. This command replaces the selection of any other Formula Row option.

The type of formula that is output is dependent upon each column @Formula setting. If no formula is defined for a column, its cell in the formula row is blank. To define or change an @Formula for a column, use the **Column > Define** command.

Options > Formula Rows > Display Current Settings

Shows the current setting for the Formula Rows function. A check mark on the Formula Rows submenu indicates which of the Formula Row functions is in use (if any).

Options > Formula Rows > Undo

Removes unwanted or incorrect Formula Row definitions.

Options > Global

Controls global settings for the current mask.

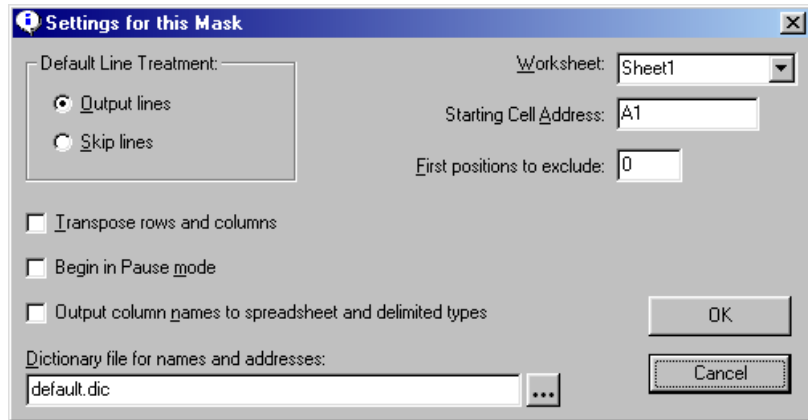


Figure 5-38 Global Setting Dialog Box

Default Line Treatment controls the default starting point for translation of data.

Output lines option or Global Output Lines Mode assumes that—by default—all data within columns of all lines will be translated, so the user must specify what lines of data *should not* be translated. Use the **Exclude > Line**, **Line > Skip** and **Exclude > Pause** commands to prevent lines from being translated.

In Global Output Line Mode, Include Lines have precedence over Exclude Lines. That is, if an Include match string occurs in an Exclude Lines' range, the Include Lines will be translated.

Skip lines option or Global Skip Line Mode assumes that—by default—no lines will be translated, so the user must specify what lines of data *should* be translated. Use the **Include > Line**, **Line > Output** and **Include > Resume** commands to select line for translation.

In Global Skip Lines Mode, Exclude Lines have precedence over Include Lines. That is, if an Exclude match string occurs in an Include Lines' range, the Exclude Lines will be not be translated.

Transpose rows and columns translates rows as columns and columns as rows. Selecting this option only changes spreadsheet format output files; no changes are made to the Input File or the display in the Mask window.

Begin in Pause mode automatically inserts a Pause command at the beginning of the Input File. Use this option if you plan to use a Resume and Pause command combination to select lines of data for translation.

For more information about Pause and Resume, see **Include > Resume > Define** on page 77 and **Exclude > Pause > Define** on page 79.

Output column names to spreadsheet and delimited types will output the column names. With spreadsheet data types, the column names are output into the cells on the first row of the output file. With comma separated variable data types, the column names are output as the first record with separators surrounding each column name

Dictionary file for names and addresses selects the dictionary file to use when parsing names and addresses. The default dictionary file is DEFAULT.DIC. You can create your own custom dictionary file to handle different prefixes, suffixes, etc.

Starting Cell Address starts the output of a spreadsheet translation at a cell address other than A1. This option is useful when combining the output of a translation into an existing spreadsheet.

First positions to exclude removes a specified number of character positions at the beginning of each line. This option is typically used to remove carriage control characters.

Beginning of line carriage control characters are included in reports created by some programming languages on certain computers. FORTRAN, for example, uses the first character of each line to indicate carriage control.

Beginning of line carriage control is used on some types of mini and mainframe computers to tell the printer when to perform line feeds and page ejects. This type of carriage control is not used on personal computers.

Options > International

Defines settings for recognition and translation of currency, dates, decimals and ASCII code pages. Any changes made to the International settings are saved in the definition of the current Mask. Settings saved using the Save as defaults button in the International Settings dialog box become the new system defaults. To load the system defaults into the current mask, press the Load defaults button.

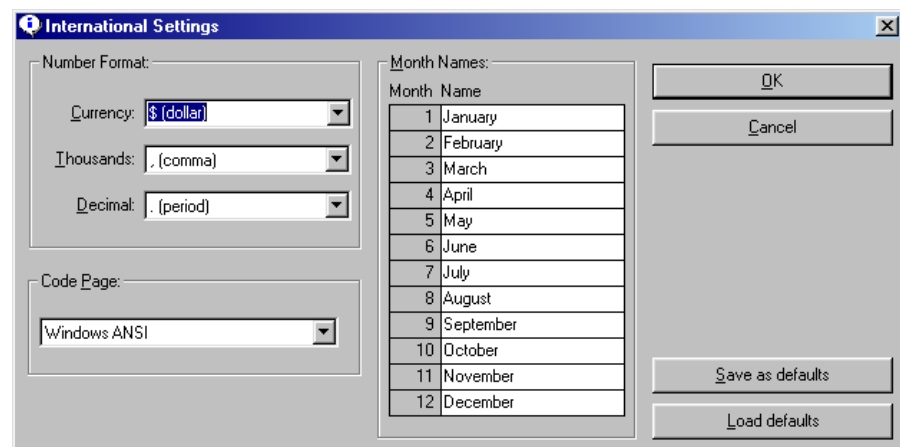


Figure 5-39 International Settings Dialog Box

Number Format options control how Numeric format data is translated to an output file by DataImport.

Currency defines the character(s) DataImport recognizes as the currency symbol. The default is “\$” (U.S. dollar). The selected currency symbol is used in all translations until changed.

Thousands defines the character DataImport recognizes as the symbol used to separate thousands. The default is “,” (comma).

Decimal defines the character DataImport recognizes as the symbol that separates the fractional or decimal part of a number from the integer part of the number. The default is “.” (period).

Code Page changes the code page that is used for interpreting ASCII characters whose values are above 127. By default, MS-DOS and DataImport use the U.S. Code page (437) to display and interpret the ASCII characters above 127.

If your computer uses a different code page than the default, or if your Input File was created on a computer that uses a different code page, you may have to change this setting for DataImport to translate correctly. The selected code page is used for all translations until another is selected.

Month Names changes the spellings of names DataImport recognizes as months. The spelling of one or more months can be changed. The default setting is U.S. English spellings.

DataImport recognizes the entire month-name or only the first part and is not upper/lower case sensitive. For example, the text September, Sept and SEP are all recognized as the month “September”.

Options > Dates

Defines Custom date format and two digit year translation.

Custom Date Format defines a non delimited custom date for use in Column Definitions. The date format defined here will appear in the Column Settings dialog box in the Type option menu.

To define a custom date, type in Y for year, M for month and D for day. Each letter represents one digit of a date number. The default custom date is “YYMMDD”.

Year for 19XX defines how a two-digit year is interpreted by DataImport. If the two digits (XX) are a number equal to or greater than the number defined here, the year is translated as 19XX. If the digits are less than the number defined here, the year is translated as 20XX. For example, with the Year for 19XX option set to 50, “51” would be interpreted as 1951 and “12” would be interpreted as 2012.

Options > Signed Overpunch

Defines how signed overpunch characters that occur in some raw data files are translated. Any changes made to the Signed Overpunch Settings are always saved when the current Mask is saved. Settings saved using the Save as defaults button in the Signed Overpunch Settings dialog box become the system defaults. To load the system defaults into the current mask, press the Load defaults button.

Characters shows the current interpretation of ASCII characters for signed overpunch data.

Value indicates the output value of overpunch characters.

Char shows which characters are interpreted as overpunch characters. Choosing Custom from the Character Set options makes the Char characters editable.

ASCII shows the ASCII value of the characters in the Char field.

Character Set defines the set of characters that are translated as signed overpunch characters. Three options are displayed; **0-9,-R** and **{-I,-R}** and **Custom**. One of the first two options will translate most Input Files correctly. If the Input File uses a scheme other than these two, select Custom. Choosing Custom makes the Characters Char field editable. Change the characters as appropriate and press the Save as defaults to save the custom translation set for use in future mask definitions.

Translation tables will vary depending on the application and system that created the file containing the signed overpunched numbers. Almost all use the characters “}” through “R” to indicate the negative digits. Most systems use “{” through “I” for the positive digits, but some systems (primarily IBM System 38, 36, and AS400) use “0” through “9”.

Position defines where DataImport looks for signed overpunch characters in a number.

Options > Preferences

Defines the system default settings for masks and application settings for DataImport.

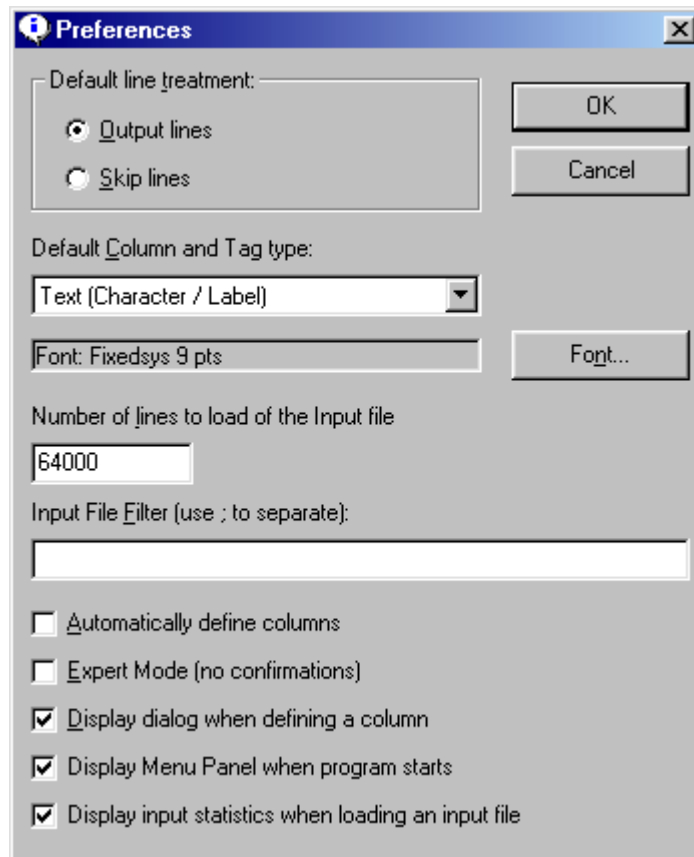


Figure 5-40 Preferences Dialog Box

Default Line Treatment controls the Output/Skip Lines Mode system default for translation of data. Default is Output Lines. This setting does not change the setting for the current mask. It is used as the default when starting a new mask.

Output lines option or Global Output Lines Mode assumes that—by default—all data within columns of all lines will be translated, so the user must specify what lines of data *should not* be translated. Use the **Exclude > Line**, **Line > Skip** and **Exclude > Pause** commands to prevent lines from being translated.

In Global Output Line Mode, Include Lines have precedence over Exclude Lines. That is, if an Include match string occurs in an Exclude Lines’ range, the Include Lines will be translated.

Skip lines option or Global Skip Line Mode assumes that—by default—no lines will be translated, so the user must specify what lines of data *should* be translated. Use the **Include > Line, Line > Output** and **Include > Resume** commands to select line for translation.

In Global Skip Lines Mode, Exclude Lines have precedence over Include Lines. That is, if an Exclude match string occurs in an Include Lines' range, the Exclude Lines will be not be translated.

Default Column and Tag Type sets the default data type when defining a new Column or Line Tag.

Font determines the default display font and font size used to display the input file.

Name sets the display font for the Input File window. The display font can only be a mono-spaced font and the option menu will only display the available fonts of this type. Default is Fixedsys.

Size sets the size of the current display font. The available fonts have a fixed number of sizes and the option menu will only display the available sizes of the selected font.

Number of Lines to Load sets the number of lines to load into the Mask screen when opening an input file. This number can be from 100 to 65,536. The fewer lines loaded, the faster the screen is updated after Include/Exclude Line selections.

Input File Filter is used to display a list of files with specified extensions when loading or opening an input file. Use the *.XXX format and a “;” (semicolon) as a separator, example: *.PRN;*.TXT;*.DOC. Once the filter is defined, it is available in the Load Input File dialog box in the List Files of Type option menu. Default is none.

Automatically define columns applies the **Column Auto Define All** command automatically if there are no columns defined in the current Mask when an Input File is loaded. Default is off (unmarked).

Expert Mode eliminates all confirmation prompts when executing commands. For example, when Expert mode is on, DataImport will not ask for confirmation when deleting a column. Default is off (unmarked.)

Display dialog when defining a column displays the Column Settings dialog box when defining a column. Default is on (marked.)

Display Menu Panel when program starts DataImport Version 6.0 added a start up Menu Panel to the software package. The default is on (marked.)

Display input statistics when loading an input file displays the number of lines and characters when the file is loaded. Default is on (marked.)

Chapter 6: DataImport Translate Reference

The DataImport Translate application is used to quickly translate files by using existing masks. The application can be used to translate multiple Input Files using one mask, translate a single Input File using multiple masks, or to make input or output file changes to a mask.

Running the Translate Application

The translate application can be opened from the DataImport Menu Panel.

The DataImport Translate application can be opened from the DataImport Menu Panel.

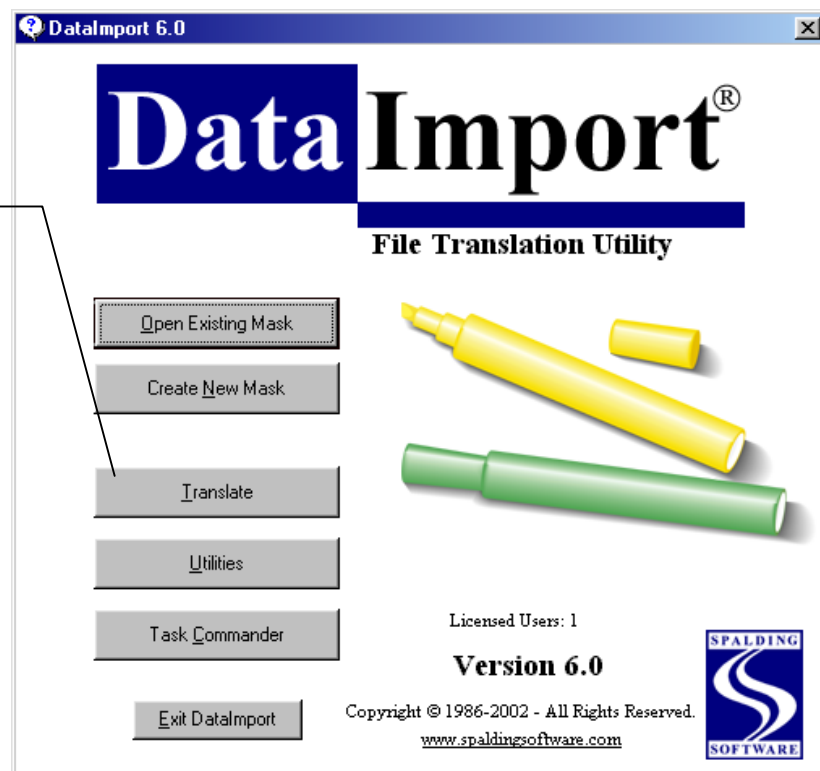


Figure 6-1 The DataImport Menu Panel

The translate application can also be opened from the Advanced section of the DataImport program group or by selecting the translate button in the Mask, Utilities, or Task Commander applications.

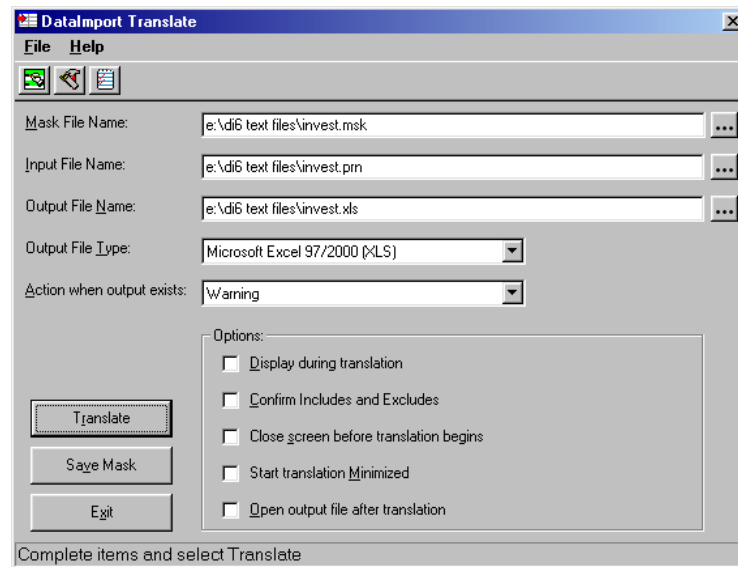


Figure 6-2 DataImport Translate application window

Mask File Name controls which mask is used to translate the Input File.

Input File Name controls which file is the source of data to be translated.

Output File Name defines the name of the file to which the translated data is written.

Output File Type defines the type of spreadsheet, database, or other file to which the translated data is written.

Action when output exists specifies what action to take when an file of the same name as the Output File Name exists.

Translate performs a translation with the current settings.

Save Mask saves the current mask with the defined output parameters.

Exit closes the Translate application.

Options specify the information displayed during and after a translation.

Display During Translation displays translated data during translation.

Confirm Includes and Excludes prompts for manual confirmation of each Include and Exclude line as they are encountered during translation. This option used to verify that the mask will include or exclude the correct lines.

Close screen before translation begins closes the DataImport Translate application dialog box when translation starts.

Start translation Minimized hides the translate dialog box until the translation is complete.

Open output file after translation automatically opens the output file in the proper application when the translation is complete.

Chapter 7: DataImport Utilities Reference

The DataImport Utilities application provides many useful tools for creating files that the Mask program can read and for reformatting data for use in other applications.

Running the Utilities Application

The utilities application can be opened from the DataImport Menu Panel.

The DataImport Utilities application can be opened from the DataImport Menu Panel.

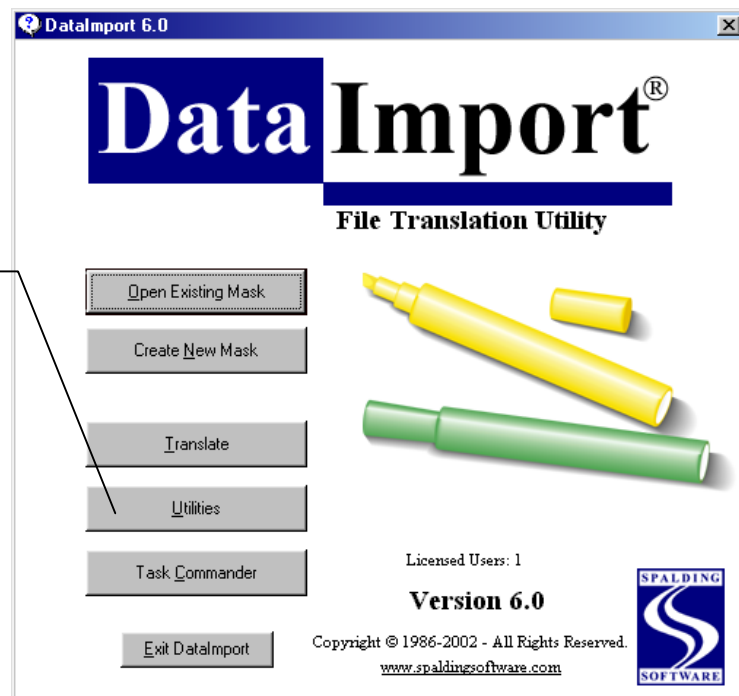


Figure 7-1 The DataImport Menu Panel

The utilities application can also be opened from the Advanced section of the DataImport program group or by selecting the utilities button in the Mask, Utilities, or Task Commander applications.

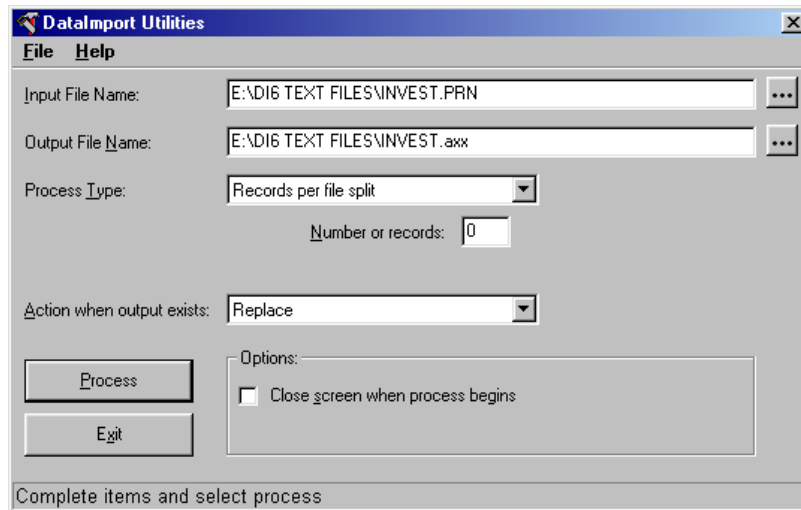


Figure 7-2 DataImport Utilities application window

Input File Name controls which data file is processed. Change the Input File by pressing the [...] button at the end of the field, selecting a new file and pressing the OK button.

Output File Name defines the name of the file output from the process. Change the Output File by pressing the [...] button at the end of the field, selecting a new file and pressing the OK button. The extension of the Output file is automatically set by the Utilities application. The extension is in the form Axx, where xx is a number beginning with 1. If one file is output the extension is A1. If three files are output the extensions are A1, A2, and A3.

Process Type specifies what type of process to perform on the Input File. For more information about available process types, see the next section.

Action when output exists specifies what action to take when an Output File of the same name as the Output File Name exists.

Process runs the utility process with the current settings.

Exit closes the Utilities application.

Close screen when process begins closes the DataImport Utilities application window when the process starts.

Process Types

The DataImport Utilities application can process Input Files for use in the Mask application and reorganize data in many ways. The following options are available in the Process Type option menu of the Utilities application:

ASCII -> EBCDIC
 Comma Separated Values
 dBase convert
 dBase header
 EBCDIC -> ASCII

Fixed length
 Line split by length
 Parse spaces
 Records per File Split
 Statistics
 Tab expansion
 Unstack

ASCII -> EBCDIC

Converts a file that is encoded in ASCII to a file that is encoded in EBCDIC. Some mini and mainframe computers encode their characters using EBCDIC. PCs encode their characters using ASCII. To upload a file that is encoded in ASCII to a computer that encodes its files in EBCDIC use this process.

Comma Separated Values

Converts a delimited text file such as a comma separated value file into a file with fixed length fields. The default field separator is a comma (ASCII 44) and the default string delimiter is quotation marks (ASCII 34). Different field separators and string delimiters can be selected.

Before actually creating the Output File, DataImport Utilities reads the entire Input File to determine the column widths necessary in the Output File. Each column is defined one character position wider than the widest data string in that column.

Files formatted with commas between fields and quotation marks around numbers should be converted to a column oriented file using this option before the mask is defined.

Comma Separated Value Input File

```
"NEW YORK",1034,968,23653
"LONDON",576,2349,9413
"ROME",1439,2008,12537
```

Output File

```
NEW YORK    1034    968    23653
LONDON      576     2349   9413
ROME        1439    2008   12537
```

Figure 7-3 Comma delimited to columnar conversion

The numbers in this example are left justified within columns. The numbers will be right justified if the Input File is translated into a spreadsheet or database file.

This option can also be used to convert files that use tabs or any other character to separate fields in a record.

dBase convert

Creates an ASCII columnar file usable by DataImport from a dBase II, III, or IV database file.

At the top of each column is the dBase field name; space limitations may prevent the entire field name from being displayed. Use the Function Header process type to output the file structure to review the truncated field names.

dBase header

Outputs the dBase II, III, or IV file structure contained in the database file's header record.

When the Go option is selected, the file structure is written to the Output File. This file can be viewed using the Mask application. The file structure can also be output directly to a printer by specifying the Output File as either LPT1, LPT2 or LPT3.

The dBase III file: SALES.DBF contains 137 records. Each record is 43 bytes long.						
#	Field name	Field type	Length	Decimals	DI	Column
-	-----	-----	-----	-----	-----	-----
1	BRANCH	Character	15	0	A	
2	WEEKSALES	Numeric	10	2	B	
3	YEARSales	Numeric	10	2	C	
4	UPDATED	Date	8	0	D	

Figure 7-4 dBase file header

EBCDIC -> ASCII

Converts a file that is encoded in EBCDIC to a file that is encoded in ASCII.

Some mini and mainframe computers encode their characters using EBCDIC. PCs encode their characters using ASCII. To use a file that is encoded in EBCDIC use this process.

NOTE Most PC to Host emulation and file transfer software handles the translation between ASCII and EBCDIC automatically.

Fixed length

Breaks up a fixed length record file that does not have record separators into a sequential file with record separators.

Report files, such as those downloaded from a mainframe, utilize a carriage return and/or a line feed character (ASCII codes 13 and 10, respectively) to indicate the end of each line or record in the file. DataImport requires these record separator characters to function properly.

Most database management systems and many other software programs use data files with fixed length records without record separators. These files are called random or direct access files. These programs recognize the length of each field in

the record and do not include record separator characters to save file space. Frequently, the first information in the file describes the fields in the file, their length and type. Users can skip this information by specifying a number of characters, and thereby eliminate the header from the Output File.

The following files illustrate how a data file without record separators looks before and after conversion. In this example, the record length is 18 and the number of characters to skip at the beginning of the file is 29.

Input File Without Record Separators

002OFFICE C 10SALES N 8.2NEW YORK 12935.45LONDON
9264.32ROME 7194.39TOKYO 15778.56

Output File

NEW YORK	12935.45
LONDON	9264.32
ROME	7194.39
TOKYO	15778.56

Figure 7-5 Adding record separators

Line split by length

Splits the Input File vertically by producing two or more Output Files with shorter record lengths. The records in the Input File are divided into shorter records, using the length specified, then written to the Output Files.

This utility is useful when an Input File contains records whose length is greater than 2048 characters; DataImport can display and translate files whose record lengths are shorter than 2048 characters. This utility can facilitate translation of extremely wide files.

For example, if the Input File SALES.DAT, containing records 5,000 characters wide, is split into files whose record length is 2,000, three Output Files are created. The first Output File created is named SALES.A1 and contains the first 2,000 characters of each record from the SALES.DAT Input File. The second file created is named SALES.A2 and contains the next 2,000 characters of each record of the Input File, and the third named SALES.A3 contains the remaining 1,000 characters of each record of the Input File.

Parse spaces

Converts a space separated variable file into a file with fixed length fields. Test and sampling instrumentation and software often create files of this format. There are two parameters that can be specified:

Column width is the width of columns to use for the parsed fields.

Skip Characters is the number of character positions at the beginning of each line to output without parsing. This parameter is optional.

In the following example, the width of the columns is set to 6 positions each, and the number of characters to not parse at the beginning of each line is 19 positions.

Space Separated Variable Input File

001	08/13/92	12:32	12	14	23	12	9.876
002	08/13/92	13:10	5	14.6	7		
001	08/14/92	12:40	5	34	9.987	12	98
002	08/14/92	13:12	9	12	6.875		

Output File

001	08/13/92	12:32	12	14	23	12	9.876
002	08/13/92	13:10	5	14.6	7		
001	08/14/92	12:40	5	34	9.987	12	98
002	08/14/92	13:12	9	12	6.875		

Figure 7-6 Space delimited to columnar conversion

Records per File Split

Splits the Input File horizontally by producing two or more Output Files with fewer records per file.

This function has two useful applications:

- Only the first 65,536 records (lines) of a file can be displayed in the Mask application. By splitting extremely long files, all records can be displayed. Remember, there is no limit to the number of lines that can be translated using a mask.
- Spreadsheets have limitations on the number of lines allowed in a worksheet. By splitting a long file, the shorter files can be translated into several usable worksheets.

For example, if the Input File ORDER.DAT, contains 7,000 records and the file is record-split into files with 3,000 records per file, DataImport creates three Output Files. The first Output File created is named ORDER.A1 and contains the first 3,000 records from the ORDER.DAT Input File. The second file created is named ORDER.A2 and contains the next 3,000 records, and the third file created is named ORDER.A3 and contains the remaining 1,000 records.

Statistics

Displays statistics about the current Input File. The statistics displayed by this process include the length of the longest line, the number of lines, and whether the file contains tab characters.

This option is often selected to determine if the use of any of the other utility functions are necessary prior to opening the Input File in the Mask application.

Tab expansion

Expands tab characters (ASCII character 9).

For example, if a user defines a tab stop as 8, numbers preceded by a tab character will be aligned in columns on the 9th, 17th, 25th, etc. character positions. The two files below illustrate how this utility works. The first file is shown with tabs displayed as a “→” character. The second file shows how the file looks when the tabs are expanded.

Input File Before Tabs Are Expanded

```
NEW YORK→1,034→968→23,653
LONDON→576→2,349→9,413
ROME→1,439→2,008→12,537
```

Output File After Tabs Are Expanded

```
NEW YORK 1,034 968 23,653
LONDON 576 2,349 9,413
ROME 1,439 2,008 12,537
```

Figure 7-7 Expanding tabs with the Tab expansion process

The numbers in this example are left justified within the columns. The numbers will be right justified when DataImport translates this file as a spreadsheet or database file.

Unstack

Reorganizes data blocks where two or more sets of data are mixed in a single column.

Use this command to separate columns with multiple data sets—for instance, yearly, monthly and daily sales—into multiple columns with one data set each.

Lines to unstack is the number of lines in each group of data lines.

Lines to Skip is the number of lines to ignore at the beginning of the file.

In the following example, the number of Lines to unstack is 2 and the number of Lines to Skip at the beginning of the file is 4.

Stacked Input File

```
SALES REPORT
MONTH: MARCH   XYZ CORPORATION
BRANCH         PERIOD SALES
-----
NEW YORK       MONTH 12,935
                YEAR  31,221
LONDON         MONTH  9,264
                YEAR 24,786
```

Unstacked Output File

```
NEW YORK       MONTH 12,935   YEAR  31,221
LONDON         MONTH  9,264   YEAR 24,786
```

Figure 7-8 Unstacking records

If the Input File is a report with a heading at the top of each page, it may be necessary to define a mask and perform a translation to remove the headings before unstacking the file.

Unstacking is also very useful for preparing a text file of names and addresses created with a word processor for translation into a spreadsheet or database file. Unstacking such a file can produce separate columns for name, street address, and city/state/zip code.

Chapter 8: DataImport Task Commander Reference

Running the Task Commander Application

The Task Commander application can be opened from the DataImport Menu Panel.

The DataImport Task Commander application can be opened from the DataImport Menu Panel.

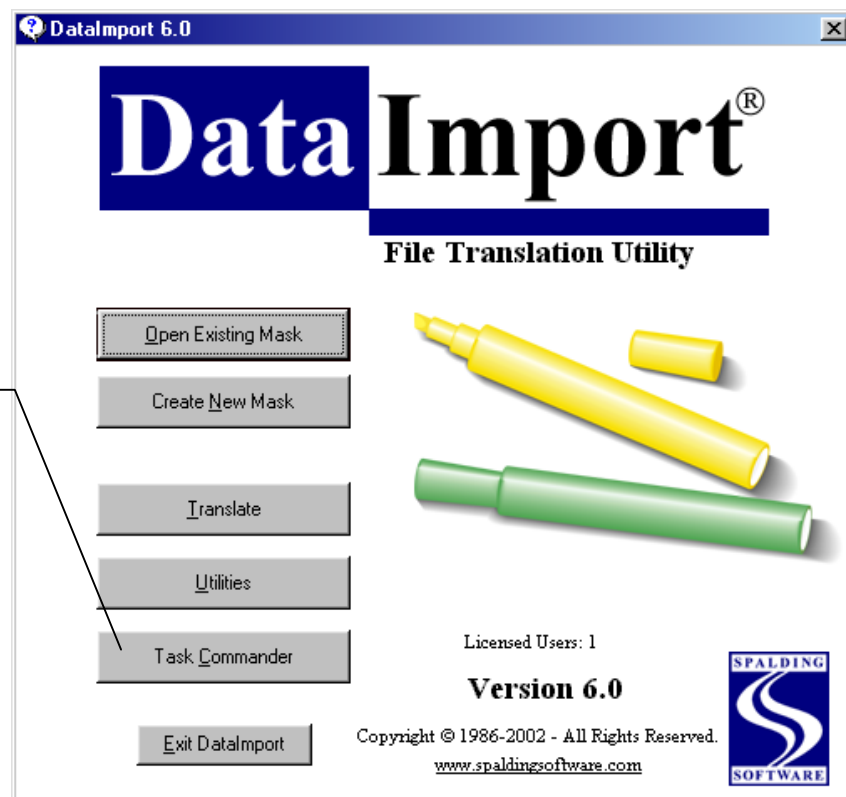


Figure 8-1 The DataImport Menu Panel

The Task Commander application can also be opened from the advanced section of the DataImport program group or by selecting the Task Commander button in the Mask, Utilities, or Task Commander applications.

This section of the User Reference describes operations available in the Task Commander application. Task Commander automates a series of translations and/or utilities within DataImport and saves this information in a Task file (.TSK). Task files can be run, created and edited from the Task Commander screen. Below is the main Task Commander screen.

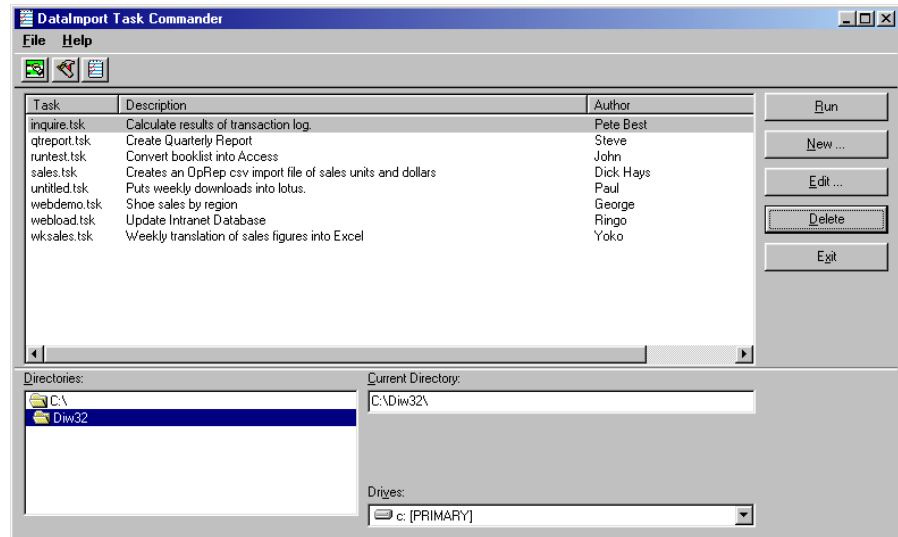


Figure 8-1 The Task Commander Main Window

The Task File list box lists all task files in the specified directory. This includes the file name, and the Author and Summary information, if given. There are several buttons on the right of the screen that control the Task Commander as shown below.

Run executes the task file(s) selected in the Task File list box.

New creates a new task file. This opens the Task File dialog box.

Edit changes an existing task file. This opens the Task File dialog box.

Exit closes Task Commander.

Task File Dialog Box

The Task Commander dialog box is where you create and edit task files. This is shown below.

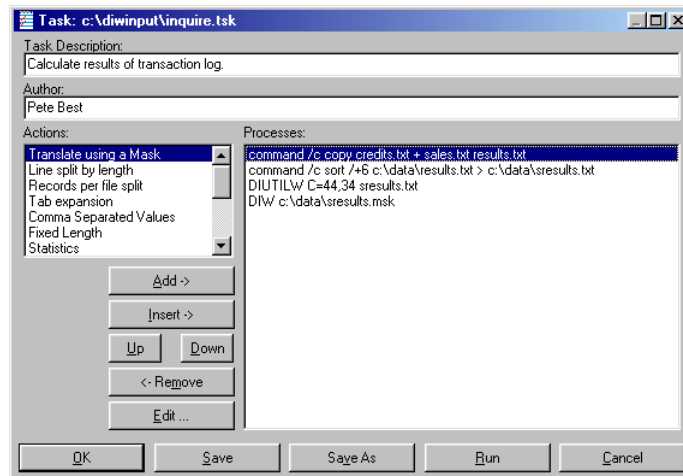


Figure 8-2 Task File dialog box

Task Description: This is where a description of the task file can be entered.

Author: This is where the Author can be entered.

Actions: Contains the list of actions that can be performed in a task. The possible actions consist of the Translate command and all of the Utilities process types (except Statistics), and running other non DataImport applications that support command line execution.

Processes: The Processes list box is where the 'script' of the task file is displayed. Multiple actions can be added to the Processes box.

Add Add copies an action from the Actions box to the Processes box.

Insert Inserts the selected Action before the selected Process.

Up Moves the selected Process up one step.

Down Moves the selected Process down one step.

Remove Removes an action or actions from the Processes box.

Edit Edits an action within the Processes box.

Ok Returns to the main Task Commander window. If you have not saved changes to the mask, you will be prompted to save it.

Save Saves the current task file.

Run Executes the task file.

Cancel Returns you to the main Task Commander window.

Based on the action that you select to add or edit, either the Translate parameter or the Utilities parameter window will be displayed. For more information on the Translate and Utilities commands, see Chapter 6; DataImport Translate Reference and Chapter 7: DataImport Utilities Reference.

Note that you can also run a task from a command line or create an Icon for running a task. For more information refer to Appendix G: Command Line Use.

Appendix A: Supported Output File Formats

Output Formats

DataImport can create output files for most standard spreadsheet and database programs including Excel, Lotus 1-2-3, Quattro Pro, Access and dBase compatible applications. The next section in this appendix lists the types of formats DataImport can read and write. Check the README file for output format additions, if you do not see the format you need.

Keep in mind that most programs can read earlier versions of their file formats. In most cases, a file format with a version number equal to or less than your software version will work, unless you are combining or appending files. In this case, you must output your data in the same format as the existing output file.

If the application you want to get data into is not on the list of supported output types, check your application's help system or manual for an import feature. Then use DataImport to create a file of the type your application can import. For example, WinFax Pro will import dBase files (DBF) and FedEx Ship will import comma separated ASCII (CSV) files. The most common types of files that applications can import are comma separated values (CSV - sometimes referred to as ASCII or text), tab separated values (TSV), dBASE (DBF) and Lotus version 1A (WKS).

Many database management software products use the dBase format as their native format. These products are often referred to as an xBase product and include FoxPro, Clipper and Alpha.

Output File Types

There are five main types of formats that DataImport supports: spreadsheets, databases, word processing merge data formats, text formats and interchange formats. Some formats may not support features and formats provided by DataImport. For example, formulas are only supported by the spreadsheet output types and field names can only be written to database files. If you are using advanced features of DataImport, read the documentation carefully to make sure that your chosen output format will support it. DataImport will not write anything to the output file that is not supported by the selected output format

Output File List

Product	Type	File Ext.	Combine	Append	Field Name	Table Name	Multi-sheet
ASCII	T	ASC		X			
Clarion	D	DAT		X	X		
Columnwise DIF	I	DIF					
Comma Separated Value	T	CSV		X			
dBase II	D	DBF		X	X		
dBase III	D	DBF		X	X		
dBase IV	D	DBF		X	X		
Excel 2.1	S	XLS	X	X			
Excel 3.0	S	XLS	X	X			
Excel 4.0	S	XLS	X	X			
Excel 5.0	S	XLS	X	X			X
Excel 7.0	S	XLS	X	X			X
Excel 97, 2000, XP	S	XLS	X	X			X
Fixed length file	T	FXD		X			
HTML Table	T	HTM					
Lotus 1-2-3 1A	S	WKS	X	X			
Lotus 1-2-3 2.0	S	WK1	X	X			
Lotus 1-2-3 3.0	S	WK3	X	X			X
Lotus 1-2-3 4.0	S	WK4	X	X			X
Lotus 1-2-3 5.0	S	WK4	X	X			X
Mailing Label	T	LBL		X			
Access 1.1	D	MDB		X	X	X	
Access 2.0	D	MDB		X	X	X	
Access 3.0/97	D	MDB		X	X	X	
Access 4.0 2000, XP	D	MDB		X	X	X	
Microsoft Word Merge	W	WRD		X	X		
Print Image	T	PRN		X			

Quattro	S	WKQ	X	X			
Quattro Pro	S	WQ1	X	X			
Quattro Pro 5.0 for Windows	S	WB1	X	X			X
Standard Data Format	T	SDF		X			
Sylk	I	SLK					
Symphony 1.0	S	WRK	X	X			
Symphony 1.1	S	WR1	X	X			
Tab Separated Variable	T	TSV		X			
User-Defined Delimited	T	UDD		X			
WordPerfect 5.0 Merge	W	W50		X	X		
WordPerfect 5.1 Merge	W	W51		X	X		
XML	T	XML			X		
Named Value	T	NVL			X		

Figure A-2 Output File formats and capabilities

Type Indicates the file type of the format: S=Spreadsheet, D=Database, T=Text, W=Word processing merge data documents, I=Interchange

File Ext. File name extension for output type.

Combine A mark in this column indicates DataImport can combine files of this format. Only spreadsheet formats allow this function.

Append A mark in this column indicates DataImport can append to files of this type.

Field Name A mark in this column indicates this format uses field names in its files. Databases and word processing merge files typically use these names.

Table Name A mark in this column indicates this format uses Table Names in the file. Microsoft Access formats 1.1 and 2.0 are currently the only formats that manage Table Names in this way.

Multi-sheet A mark in this column indicates this spreadsheet program can have multiple sheets per file. DataImport allows you to place data on a specific sheet using the Starting Cell Address field in the **Options Global** dialog box.

ASCII (ASC)

This text output format is an ASCII file, delimited with commas between fields (DataImport columns cells) and quotation marks surrounding non-numeric fields. This type of file is used with some languages like BASIC for data files. Most database management systems like dBase II or dBase III will read or import this

type of file. This format is similar to the Comma Separated Value (CSV) format detailed below.

Alpha (DBF)

This database program uses a version of dBase as its file format. Check your documentation for details.

Clarion (DAT)

This database output format is a the file type used by the Clarion database management program.

Clipper (DBF)

This database program uses a version of dBase as its file format. Check your documentation for details.

Columnwise DIF (DIF)

This data interchange output format arranges data in column-wise order. This file is used to transfer data between spreadsheet programs and other software.

Comma Separated Value (CSV)

This text output format separates items of data with a comma and encases text data with quotation marks. To output files with different separators, see Tab Separated Variable and User Defined Delimited formats below.

dBase II, III, IV (DBF)

These database output formats are standard database formats for dBase compatible programs. The following database translation types can be selected:

- | | |
|------------------|--|
| dBase II | dBase II database file, file extension: DBF |
| dBase III | dBase III and dBase III Plus database file, file extension: DBF This file can be read by many other dBase compatible software products including FoxPro and Clipper. |
| dBase IV | dBase IV database file, file extension: DBF |

Excel 2.1, 3.0, 4.0, 5.0, 7.0, 97, 2000, XP (XLS)

These spreadsheet output formats are used by the Microsoft Excel application. Lower versions of Excel will not load higher version worksheet files. Higher versions of Excel will load lower version worksheet files. The following spreadsheet translation types can be selected, the XLS file extension used for all versions of Excel.

- | | |
|------------------|---|
| Excel 2.1 | Microsoft Excel version 2.1 worksheet file. |
| Excel 3.0 | Microsoft Excel version 3.0 worksheet file. |
| Excel 4.0 | Microsoft Excel version 4.0 worksheet file. |

Excel 5.0, 7.0	Microsoft Excel version 5.0 and 7.0 worksheet files. (7.0 is Windows 95)
Excel 97/2000	Microsoft Excel version 8.0 workbook files.

Fixed length file (FXD)

This text output format is a fixed record format file. All fields are fixed width and all records are a fixed length. Records are not separated. This output type is useful for uploading to mainframes that do not use record separators.

FoxPro (DBF)

This database program uses a version of dBase as its file format. Check your documentation for details.

HTML Tables (HTM)

This format option allows you to create HTML Tables in accordance with HTML 2 standards.

Lotus 1-2-3 1A, 2.0, 3.0, 4.0, 5.0 (WK*)

This spreadsheet output format is used by the Lotus 1-2-3 spreadsheet program. The following spreadsheet translation types can be selected; the indicated file extension is used for each output option.

- Lotus 123 1A** Lotus 1-2-3 release 1 and 1A worksheet file, file extension: WKS This file can be read by all releases of Lotus 1-2-3 and by the Symphony spreadsheet program. This format is a commonly supported by spreadsheet programs and other software packages.
- Lotus 123 2.0** Lotus 1-2-3 release 2.x worksheet file, file extension: WK1
- Lotus 123 3.0** Lotus 1-2-3 release 3.x worksheet file, file extension: WK3
- Lotus 123 4.0** Lotus 1-2-3 release 4.x worksheet file, file extension: WK4
- Lotus 123 5.0** Lotus 1-2-3 release 5.x worksheet file, file extension: WK4

Mailing Label (LBL)

This text output format produces a mailing label style file. Each line of a column (cell) is output as a separate line in the Output File.

Label output type is useful for translating an Input File that contains names in one column and addresses in another column into a file that has the name on the first line, address on the next line, etc. If there are five columns defined, the data in column A will appear on every fifth line. To add blank lines between labels to control spacing, define columns on the right side of the mask at a location that does not contain data.

Microsoft Access 1.1, 2.0, 3.0 (97), 4.0 (2000), XP (MDB)

These database output formats are used by the Microsoft Access application. Lower versions of Access will not load higher version database files. Higher versions of Access will load lower version database files. The following database translation types can be selected; the MDB file extension used for all versions of Access.

Microsoft Word Merge File(WRD)

This word processor output format is the data document file format used by Microsoft Word for DOS for print merges. This file format can also be used by Microsoft Word for Windows.

Print Image (PRN)

This text output file format is a DOS print image file. This file type contains no special formatting and can be read by word processing software. Text in the file is arranged in columns without use of tabs.

Quattro (WKQ)

This spreadsheet output format is used by the Quattro spreadsheet program.

Quattro Pro (WQ1)

This spreadsheet output format is used by the Quattro Pro spreadsheet program.

Quattro Pro 5.0 for Windows (WB1)

This spreadsheet output format is used by the Quattro Pro 5.0 for Windows spreadsheet program.

Standard Data Format (SDF)

This text output format produces a Standard Data Format file. All fields are a fixed width and all records are a fixed length. Each record is separated by a carriage return and line feed (ASCII codes 13 and 10).

Sylk (SLK)

This interchange output format produces a Symbolic Link file. Many of Microsoft's products read SYLK files, including MultiPlan and Microsoft Chart.

Symphony 1.0, 1.1 (WRK, WR1)

This spreadsheet file format is used by the Symphony spreadsheet program. The following database translation types can be selected:

Symphony 1.0 Symphony release 1.0 worksheet file, file extension:
WRK

Symphony 1.1 Symphony release 1.1, 1.2 and 2.x worksheet file, file extension: WR1

Tab Separated Variable (TSV)

This text output format separates variable fields or cells with the tab character (ASCII code 9). Records are separated by a carriage return and line feed (ASCII codes 13 and 10). This output type is used by many Apple Macintosh applications. It is also useful for preparing tables to be loaded into word processors.

User-Defined Delimited (UDD)

This text output file type separates records or cells with one or more user selected ASCII characters. Non-numeric fields, or labels, are also surrounded by one or more user selected ASCII characters. Records are separated by a carriage return and line feed (ASCII codes 13 and 10).

To define the delimiter character(s) for UDD format, select File Define Output Format. From the Output File Type pull-down menu and select User-Defined Delimited. The Field Separator and String Delimiter options appear. In the Field Separator field, type the character to indicate the end of a cell or record data and the beginning of the next cell or record. In the String Delimiter field, type the character to use to set off text or alphanumeric strings in data cells or records.

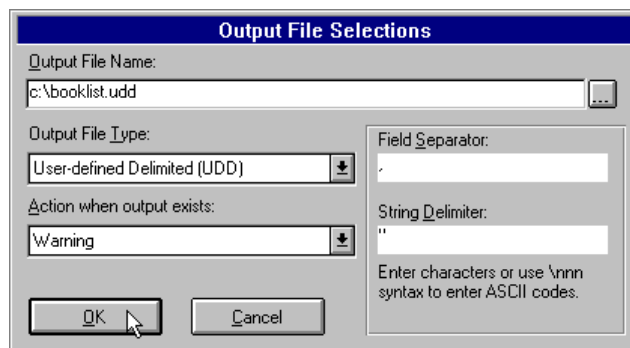


Figure A-2 Defining the Field Separators and String Delimiters for the User-defined Delimited output format.

WordPerfect 5.0, 5.1 (W5*)

This text output format is a secondary merge file for WordPerfect. The following database translation types can be selected:

WordPerfect 5.0 WordPerfect 5.0 secondary merge document, file extension: W50

WordPerfect 5.1 WordPerfect 5.1 secondary merge document, file extension: W51

xBase applications (DBF)

These database programs use versions of dBase as their file formats. Check your documentation for details.

NVL Named Value

Follows the standard for NVL.

XML Extensible Markup Language

Follows the standard for XML 1.0.

Appendix B: Command Line Use

DataImport Translate, Utilities, and Task Commander programs can be run as batch operations, allowing you to further automate your translations.

The DataImport Task Commander allows the execution of a series of DataImport Translate, DataImport Utilities and non-DataImport applications that can be run from a command line. With the DataImport Task Commander, you can run batch processes without knowledge of the command line. Task Commander automatically creates the command lines for you.

You can also achieve some batch-like functionality by creating an icon or a shortcut for a DataImport translation or utilities process. To achieve more extensive batch processing functionality in Windows, you will need to use an add-on utility like WinBatch by Wilson WindowWare (800-762-8383) or another third-party utility.

Translate Command Line

Once a Mask has been defined and saved to disk, the translation can be performed using command line controls.

Syntax:

**DIW32 mask[, [input], [output], [type], [display], [confirm]] [/A]
[/C] [/M]**

The command line parameters that follow the **DIW** command are positional and separated by commas. If a parameter is skipped, a comma must be used to hold its place. The switches, **/A** and **/C**, are not positional and are not separated by commas.

mask Mask File name, including the path if necessary. This is the only required parameter. If no other parameters are specified, the parameters specified when the mask was created (or last saved) will be used in the translation.

input Input File name, including the path and extension if necessary.

output Output File name, including the path if necessary. If an extension is specified, it will be used rather than the file extension
DataImport normally uses based on the type of translation.

type Type of translation to be performed. Any of the following types can be specified:

WKS Lotus 1-2-3 release 1 and 1A

WK1 Lotus 1-2-3 release 2.x
WK3 Lotus 1-2-3 release 3.x
WK4 Lotus 1-2-3 release 4.x and 5.x
WRK Symphony release 1.0
WR1 Symphony release 1.1, 1.2 and 2.x
WKQ Borland Quattro
WQ1 Borland Quattro Pro
WB1 Borland Quattro Pro 5.0 for Windows
XLS Microsoft Excel version 2.1
XLS3 Microsoft Excel version 3.0
XLS4 Microsoft Excel version 4.0
XLS5 Microsoft Excel version 5.0 and 7.0.
XLS8 Microsoft Excel version 8.0 (97/2000/XP)
DBF dBase III
DBF2 dBase II
DBF3 dBase III
DBF4 dBase IV
MDB1 Microsoft Access 1.1
MDB Microsoft Access 2.0
MDB3 Microsoft Access 3.0 (97)
MDB4 Microsoft Access 4.0 (2000/XP)
DAT Clarion
DIF Columnwise DIF
CDIF Columnwise DIF
SLK SYLK or Symbolic Link
PRN Print Image
HTM HTML Table
XML XML Table
ASC Comma separated with quotes around strings
CSV Comma Separated Variable
SDF Standard Data Format
FXD Fixed record format without delimiters
TSV Tab separated variables
UDD User-defined delimited
TXT ASCII text file
LBL Mailing label format
WRD Microsoft Word data document
W50 Word Perfect 5.0 secondary merge file
W51 Word Perfect 5.1 secondary merge file
NVL Named Value

display Specifies whether the output is to be displayed on screen during translation: **Y** for yes, **N** for no. The default is Yes.
confirm Specifies whether Include Line and Exclude Line treatments must be confirmed manually during the translation. **Y** for yes, **N** for no. The default is No.
/A Appends the output of the translation to the end of an existing Output File. See the description of the Mask command **Files Define Output File.** for more information.
/C Combines the output of the translation into an existing spreadsheet Output File. See the description of the Mask command **Files Define Output File.** for more information.

/M Minimizes the status box during translation.

The only required parameter is the Mask File name. If no other parameters are specified, the parameters defined when the mask was created or last saved are used.

To specify some parameters and not others, include the intervening commas as place holders. This is necessary to indicate to DataImport which of the options you want to use. For example, if you want to use the name of the Input File stored with the mask, but want to change the name of the Output File, you would place two commas before the name of the new Output File. Otherwise, DataImport would interpret the file name as an Input File. Commas, however, are unnecessary as place holders before the /A and /C parameters.

If a parameter is not specified on the command line and the parameter has not been specified in the mask, the translation cannot proceed. In such cases when DataImport aborts the translation, a message is displayed on the screen to indicate the missing or invalid parameter(s).

The following four examples illustrate the ways translations can be initiated using a command line.

Translate Command Line Example 1

To perform a translation from the command line using the following parameters:

Mask File name	MYMASK
Input File name	DIDEMO.TXT
Output File name	SALESDAT
Translation type	XLS
Display on	Y (Yes)
Confirm include/exclude	Y (Yes)
Append to existing file	/A

The command line should read:

DIW MYMASK,DIDEMO.TXT,SALESDAT,XLS,Y,Y/A

Translate Command Line Example 2

To perform a translation from the command line, use the following parameters:

Mask File name	MYMASK
Input File name	as specified in the mask
Output File name	as specified in the mask
Translation type	XLS
Display on	default to yes
Confirm include/exclude	default to no

The command line should read:

DIW MYMASK,,,XLS

Translate Command Line Example 3

To perform a translation from the command line, use the following parameters:

Mask File name	MYMASK
----------------	--------

Input File name	as specified in the mask
Output File name	as specified in the mask
Translation type	as specified in the mask
Display on	default to yes
Confirm include/exclude	default to no

The command line should read:

DIW MYMASK

Translate Command Line Example 4

To perform a translation from the command line, use the following parameters:

Mask File name	MYMASK
Input File name	ORIGINAL.DAT
Output File name	GOOD
Translation type	as specified in the mask
Display on	default to yes
Confirm include/exclude	default to no
File-combine	/C

The command line should read:

DIW MYMASK,ORIGINAL.DAT,GOOD/C

Utilities Command Line

The running of a utility can be initiated from a command line.

Syntax:

DIUTIL32 option[=v1[,v2]] input [output] [/W]

The **option**, **input** file name and **output** file name parameters are positional and separated by spaces.

option Option or process to be performed. DataImport offers eleven options or utilities.

L	Line Split by Length
R	Records per File Split
T	Tab expansion to ASCII columnar
C	CSV (Comma Separated Variable) to columnar
F	Fixed length records
S	Statistics
U	Unstack
D	dBase to ASCII columnar
H	Header of dBase to ASCII file
A	ASCII to EBCDIC
E	EBCDIC to ASCII
P	Parse space delimited to columnar

The options and examples of their uses are described below.

v1 & v2	Required and/or optional values. Whether these values are used depends on the utility option specified.
input	Input File name, including the path and extension if necessary.
output	Output File name, without an extension, but including any drive and path specifications necessary to tell DataImport where to find the existing file or where to place the new file. If an Output File name is not specified, the Output Files will have the same path and name as the Input File. DataImport provides unique, sequential file extensions (A1 through A99), as some options produce more than one Output File.
/W	Includes a Warning if DataImport detects that the Output File name specified already exists. Stops the program and prompts whether to proceed. If not specified, any file(s) having the same name as the Output File will be overwritten. DataImport offers twelve options or utilities. Descriptions of these options and examples of their use are provided below:
L=v1	<p>Line Split by Length Splits the Input File vertically by producing two or more Output Files with shorter record lengths. The value v1 is the maximum record length in the Output Files. See the description of the DataImport Utilities process Line Split by Length for more information.</p> <p>For example, to split the file INFILE.DAT into files with a record length of 80 and output the data into files with the name of OUTFILE, the command line would read as follows:</p> <p>DIUTIL32 L=80 INFILE.DAT OUTFILE</p> <p>The number of files created depends on the length of the Input File and the number of characters specified as the maximum length for each Output File. The first Output File is named OUTFILE.A1, the second OUTFILE.A2, etc.</p>
R=v1	<p>Records per File Split Splits the Input File into two or more files with fewer records in each Output File. The value v1 is the maximum number of records in the Output Files. See the description of the Utilities process Records per File Split in the Utilities Reference for more information.</p> <p>For example, to split the file INFILE.DAT into files with no more than 8192 records per file, the command line would read as follows:</p> <p>DIUTIL32 R=8192 INFILE.DAT</p> <p>The number of files created depends on the length of the Input File and the number of records specified as the maximum for each Output File. The first Output File is named INFILE.A1, the second INFILE.A2, etc.</p>
T=v1	<p>Tabs Expands tab characters by the value indicated by v1. This value sets the number of spaces to use as tab stops. See the description of the Utilities Screen option Function Tabs for more information.</p>

For example, to expand the tabs in the file INFILE.DAT with tab stops of 8, the command line would read as follows:

DIUTIL32 T=8 INFILE.DAT

C[=v1[,v2]] Comma Separated Converts a comma separated (or user-defined separated) file into an ASCII columnar text file. The value v1 specifies the field separator. The value v2 specifies the string delimiter. If no optional values are supplied, the comma character (ASCII: 44) will be used as the field separator, and the quote character (ASCII: 34) will be used as the string delimiter.

See the description of the Utilities Screen option Comma Separated Values for more information.

For example, to convert the semicolon and quote delimited file INFILE.DAT, the command line would read:

DIUTIL32 C=59,34 INFILE.DAT

F=v1[,v2] Fixed Converts a fixed length record file that does not have record separators into a sequential file with record separators. The value v1 is the length of each record in characters. The optional value v2 is the number of characters to skip at the beginning of the Input File before outputting records. This option is useful when the first part of the file contains header information or other data that should not be translated. See the description of the Utilities process **Fixed Length** in the Utilities Reference for more information.

For example, to convert the file INFILE.DAT into sequential records with a length of 16 and to skip the first 18 bytes in the file, the command line would read as follows:

DIUTIL32 F=16,18 INFILE.DAT

S Statistics Displays statistics about the Input File. The statistics include the length of the longest line, the number of lines, and whether the file contains tab characters. See the description of the Utilities process **Statistics** in the Applications Reference for more information.

For example, to display statistics about the file INFILE.DAT, the command line would read as follows as follows:

DIUTIL32 S INFILE.DAT

U=v1[,v2] Unstack Unstacks a file containing multiple lines that logically go together, but are on separate lines. The value v1 specifies the number of lines to be combined into a single line. The optional value v2 is the number of lines to skip at the beginning of the file before combining lines. This option is useful if the first part of the file contains header information or other data that should not be translated. See the description of the Utilities process **Unstack** in the DataImport Utilities Reference for more information.

For example, to unstack the file INFILE.DAT by combining each pair of lines in the Input File, skipping the first 5 lines, the command line would read:

DIUTIL32 U=2,5 INFILE.DAT

- D dBase** Creates a sequential file that is usable by DataImport from a dBase II, III or IV data file. See the description of the Utilities process **dBase Convert** in the DataImport Utilities Reference for more information.

For example, to convert the dBase file INFILE.DBF to a sequential file, the command line would read as follows:

DIUTIL32 D INFILE.DBF

- H Header** Outputs the dBase file structure contained in the database file's header record. See the description of the Utilities process **dBase Header** in the DataImport Utilities Reference for more information.

For example, to output the structure of the dBase file INFILE.DBF, the command line would read as follows:

DIUTIL32 H INFILE.DBF

- A ASCII TO EBCDIC** Converts a file whose characters are encoded in ASCII (used by PC's) into a file encoded in EBCDIC (used by IBM midrange and mainframe computers). See the description of the Utilities process **ASCII->EBCDIC** in the DataImport Utilities Reference for more information.

For example, to convert the ASCII file INFILE.DAT to an EBCDIC file, the command line would read as follows:

DIUTIL32 A INFILE.DAT

- E EBCDIC TO ASCII** Converts a file whose characters are encoded in EBCDIC into a file encoded in ASCII, that is usable by DataImport. See the description of the Utilities process **EBCDIC->ASCII** in the DataImport Utilities Reference for more information.

For example, to convert the EBCDIC file INFILE.DAT to an ASCII file, the command line would read as follows:

DIUTIL32 E INFILE.DAT

- P=v1[,v2]Parse space delimited** converts a space separated variable file into a file with fixed length fields. The value v1 specifies the width of columns to use for the parsed fields. The optional value v2 specifies the number of character positions at the beginning of each line to write to the Output File without parsing.

For example, to convert the space separated file INFILE.DAT into a columnar file with the first 20 characters exactly as in the in the Input File and the remaining characters in columns that are 10 wide, the command line would read as follows:

DIUTIL32 P=10,20 INFILE.DAT

Task Commander Command Line

The DataImport Task commander can be run from the command line.

Taskfile Taskfile is the name of the file created and saved in Task Commander. Taskfiles always have an extension of TSK.

Syntax:

DITASK32 taskfile

Appendix C: Customizing the Dictionary File

Default Dictionary

DataImport comes with a default dictionary file (DEFAULT.DIC) which is used when defining columns and line tags with a type of Name Parse and/or Address Parse. The dictionary file contains common prefixes, suffixes and beginnings of last names. It also contains the names and abbreviations for all of the states of the United States, the provinces of Canada, and several countries. If the data you are working with uses different or additional prefixes, provinces, etc., you can edit the default.dic file or a copy of the default.dic file. The name of the dictionary file to be used in a mask is specified from the mask's **Options Global** dialog box.

Editing the Default.dic file

Using an editor such as Windows Notepad or DOS Edit, open the default.dic or other .dic file. Make the changes to the appropriate section. For example, if you wanted to include the Provinces of Australia and their abbreviations, you would locate the section of the file named [State/Province] and enter the following data:

Queensland
Qld.
New South Wales
N.S.W.
Western Australia
W.A.
Southern Australia
S.A.
Northern Territory
N.T.

Example

Following is an example of the sections and contents of a dictionary file. The section names in brackets must be **exactly** as shown.

```
[prefix]
Mr. & Mrs.
Mr.
Mr
Ms.

[suffix]
Jr.
Jr
M.D.
M. D.

[begin_last_name]
Van Der
Von
St.

[begin_cities]
St.
Los
Las
New

[State/Province]
Alabama
AL
Alaska
AK
Alberta
AB
```

Figure C-1 Sample Dictionary File

Appendix D: Frequently Asked Questions

DataImport Questions

I'm translating into a database and I've changed the column type in the mask from numeric to character (text). I open the file in my database management software and the field type has not changed! What's wrong?

You have already translated the file, the database exists. Database files contain both the data records and a structure that defines the layout of the fields in the records. This structure includes field names, types, and widths. As a safety precaution, DataImport never changes this structure. Therefore even if DataImport created the structure, changing column types and widths in the mask will not change the structure, even if the replace option is selected. Replace will only replace the data records, not the structure. To solve this problem, either change the output file name to a new name, or delete the existing database file. See the section in Chapter 4 titled "Working with Database Files".

I've translated my data and nothing came out. Why?

There are several possible reasons. You must have at least one column or Line Tag defined. You must also be in Global Output Line Mode, or have at least one line whose type is indicated in the mask application as either Output or Include.

Why is DataImport removing leading 0's from numeric data?

You have selected numeric as the column's type. Leading 0's (zeros) are stripped when translating to a number. To keep the leading 0's in numbers like zip codes and social security numbers, select text as the column type.

How big of a file can DataImport translate?

DataImport can translate files of unlimited lengths. DataImport can extract data from the first 16,348 characters of each line or record.

Is there some way to automate a series of translations and/or utility processes?

Sure, use the DataImport Task Commander. It is a batch file utility that is explained in Chapter 8 of this manual.

Why don't I see all of my file in the Mask window. What will happen when I translate?

DataImport by default loads 1000 lines of the file. This can be changed by selecting **Options > Preferences** and changing the setting. The Mask screen can load the first 65,536 lines of the file. Regardless of how many lines are loaded into the mask screen, DataImport will translate the entire file.

Why do I get an error message when I translate my 70,000 line file to Excel?

The Excel format is limited to 65,536 lines. If your files are larger than this, you should probably be using a database. You could also use the DataImport Utilities to create a series of input files that contain fewer lines per file by selecting the record per file split process type.

I have Line Tags defined. They are coming out repeatedly on multiple lines when I translate. I only want them to come out one time for each set of data.

You are probably in Global Output Line Mode. Try including just one line for each set of data in your file while in Global Skip Mode. See the "Include Lines Define" option in Chapter 5.

I'm trying to unstack sets of 3 lines to make one line that contains all of the information from these lines. Some times one or two of these lines appear at the bottom of a page and the remaining lines appear at the top of the next page under the page heading. The headings are getting in the way.

This will require two masks and two translations. First, define a mask that extracts all of the data on every line into one or more columns. For now, don't worry that each item of data is not in its own column. You can use **Column > Auto Define All** to do this very quickly. In this mask, use the **Exclude > Lines** feature to exclude the page headings. Select the Output file type as print image (.PRN) and translate. The resulting file will look just like the original without the page headings.

Start a new mask and load the print image file that you just created as the Input file. This new Input file will not include the interrupting page headings. Now you are ready to apply the unstack feature and select the columns of information that you want. Remember, that you can automate this two step process by using DataImport's Task Commander.

When I try to run the Mask program, I get the message "Not a valid Win32 application". The Utilities, Translate, and Task Commander all work normally? What's going on?

Your mask program is probably corrupt. Reinstall the software or contact Spalding Software for replacement files.

When translating to an existing Excel file, I get the message 'CANNOT OPEN FILE xxxxxxx.XLS'.

The file you are translating into probably already exists and is open in Excel or another application. Switch to Excel or other application and close the file.

When I try to append or file combine to an Excel file I get the message 'CANNOT FILE COMBINE OR APPEND TO A NONEXISTANT SHEET'.

DataImport cannot append or file combine new data into a sheet in Excel that does not already exist. DataImport cannot create new sheets in an existing Excel file. To solve this problem, use Excel to setup these sheets in advance. DataImport can append or combine to any existing sheet in an Excel file. DataImport can also create a sheet of any valid name when creating a new Excel file or replacing a file of the same name.

When using the DataImport Utilities on my file (LENNY.PRN) to create a new file in a different format, the dialog box indicates it is outputting the file to LENNY.AXX. After running the utility I can't find any file called LENNY.AXX.

Some of the utilities create multiple files, and the letters XX refer to the numeric sequence of the files the Utility is creating. Therefore if the utility you are running is creating one file, it would be named Lenny.a1. If there were multiple files they would be named lenny.a2, lenny.a3, etc.

When I select that the column type as Text Block, I don't see the data from the second line etc. in the cell of the output file. What gives?

Your output width (in your text block column's define dialog box) defaults to the size of the column you create. This would be fine for any other type of column, but you are unstacking multiple lines of text here! You need to change the output width to accommodate the column width required to hold the additional line(s) of text. Column width is limited to a maximum of 255 characters.

How do I extract different types of data from an import file into separate sheets in an Excel spreadsheet?

First, in Excel create a workbook (file) with all of the worksheets that will be needed. Second, in DataImport, for each worksheet in the excel workbook create a separate mask. In each mask select the appropriate Excel worksheet from the drop down list. Finally, translate using each mask.

Note: The Task Commander can be used to automate the process of running multiple masks.

Index

\$

\$ (dollar) 50

¢

¢ (centavo) 50

£

£ (pound) 50

¥

¥ (yen) 50

€

€ (euro) 50

A

A\$ (Australian dollar) 50

Abort
translation at a line 40

Access
output format 114
Table Names 57

Address parse
columns 53

Alpha 4
output format 112

Appending to an existing file 34

ASC 112

ASCII
output format 112

ASCII -> EBCDIC
process 99

ASCII characters

removing 34

AutoColumn

applying automatically 93

Automating DataImport 117

Translate application 117

Utilities application 120

B

Batch programming 117

Blank cells
filling 48, 67

Blank lines
removing 35

C

C\$ (Canadian dollar) 50

Calculations 55
defining 55
inserting 55
inserting on change 83
inserting on match 85
replacing numbers with formulas 56
replacing on match 86

Cells
Blank, filling 67

Character delimited files
user-defined 115

Characters
excluding first position 35

Clarion
output format 112

Clipper 112
output format 112

Code page
control 90

Column
Auto Define All 68
Define. 65
menu 65
Push/Pull 69
Resequence 69
Settings. 68
turning dialog box off 93
Undo 68
Undo All 68

Column Control Bar 10
using 17

Column settings
dialog box 16

Columnar data
extracting 36

Columns

- Address parse 53
- and fields 56
- blank cells 48
- calculations 55
- database field names 56
- date 52
- defining 14, 36, 65
- defining automatically 36
- defining manually 36
- defining type 18
- defining with Column Control Bar 17
- defining with menu bar 14
- defining with popup menu 16
- Fill down 58
- format options 66
- formulas 55
- from non-columnar data 47
- limits, excluding lines 39
- maximum 36
- Name Parse 53
- names 58
- numeric 50
- removing 37
- resequencing 41
- text 51
- time 52
- transposing with rows 49
- Columnwise DIF
 - output format 112
- Combining files 34
 - starting cell 89
- Comma Separated Variables
 - output format 112
- Command line
 - translation 117
 - translation, example 119
 - utilities 120
- Command line controls 117
- Commands
 - Mask application 59
 - Translate application 95
 - Utilities application 97
- Confirmation
 - turning off 93
- Control characters 78
 - excluding 35
- CSV 112
- Currency symbol 50
- Custom date
 - control 90

D

- DAT 112

- Data
 - extracting 35
 - reorganizing 41
 - selecting for translation 13
 - unstacking 41
- Data columns
 - extracting 14
- Data formatting 18
- Data Interchange Format
 - Columnwise 112
- Data sets
 - arranging 41
- Data types
 - numeric 50
 - recognizing 50
 - setting 50
 - time 52
- Database fields
 - showing 62
- Databases
 - Alpha 4 112
 - appending 58
 - changing structure 34
 - Clarion 112
 - Clipper 112
 - considerations for output 34
 - creating 58
 - dBase 112
 - existing 57
 - field names 56, 58
 - fields 56
 - FoxPro 112, 113
 - indexes 57
 - Microsoft Access 114
 - new 58
 - showing structure 57
 - structure 57, 58, 100
 - example 62
 - XBase 116
- DataExport/DLL
 - Installation 1
- DataImport
 - input formats list 8
 - output formats list 8
 - programs 9
 - running 11
 - uses 7
- DataImport Mask
 - Basics 11
 - commands 59
- DataImport Mask window
 - explained 10
- DataImport Program Suite 9
- DataImport Task Commander 105

- DataImport Translate
 - controls 95
- DataImport Utilities
 - controls 97
 - using 32
- Date
 - column 52
 - month names 52
 - two digit years 52
 - without separators 52
- Date format
 - applying to columns 18
- dBase
 - convert process 100
 - header process 100
 - output format 112
- DBF 112
- Decimal separator 51
- Defining
 - columns, example 14, 16, 17
- Defining Line Tags 22
- Defining Reference Points 21
- Deleting
 - characters 34
- Delimited text file 99
- Dialog boxes
 - turning off extra 93
- Dictionary file
 - names and addresses 89
- DIF
 - Columnwise 112
- Display Input Statistics
 - turning off 93
- Displaying
 - database structure 57, 62
 - input files 31
- DKr (Danish Krone) 50
- DM (German mark) 50
- Duplicate lines
 - removing 35

E

- EBCDIC -> ASCII
 - process 100
- End translation at line 40
- Escape sequences
 - excluding 35
- Excel
 - output format 113
- Exclude
 - Blank Lines 79
 - Characters All Special Characters 78
 - Characters Define. 78

- Characters Undo All Special 79
- Duplicate Lines 80
- Edit 79
- Lines Undo. 78
- menu 77
- Page Ejects 79
- Pause Define. 79
- Pause Undo. 79
- Excluding
 - blank lines 35
 - character sequences 35
 - characters 78
 - control characters 78
 - control codes 35
 - duplicate lines 35
 - escape sequences 35
 - line groups 38
 - page ejects 35, 79
 - printer carriage control 35, 89
 - special characters 35
- Excluding lines 38, 39
 - exact match 39
 - limits 39
 - pattern match 39
 - pattern match characters 37
- Extensible Markup Language
 - XML 116
- Extracting data 13, 35
 - columnar 36
 - columnar 14
 - form based data 46
 - example 46
 - groups of lines 38
 - lines 19

F

- Fields
 - and columns 56
- File
 - Define Output File. 62
 - Exit 63
 - Input File Statistics 61
 - Load Input File. 61
 - Mask Summary Info 62
 - menu, Mask application 61
 - New Mask 61
 - Open Mask 61
 - Preview Translation 62
 - Print Input File 61
 - Print Mask Settings. 62
 - Save Mask 62
 - Save Mask As. 62
 - Show Database Fields 62

- Task Commander 63
- Translate 62
- Utilities 63
- File Filter
 - custom 93
- File format
 - choosing 33
- Files
 - adding record separators 100
 - example 101
 - output 109
 - splitting 102
- Fixed length
 - output format 113
 - process 100
- Font
 - controls 93
- Footer Match String Reference Point 71
- Foreign
 - currency 50
 - month names 52
 - number formats 51
- Formatting data 18
- Forms 46
- Formula Row
 - defined 55
 - inserting 55
- Formulas 55
 - column change 83
 - example 84
 - defining 55
 - inserting at column change 55
 - inserting on match 56, 85
 - inserting on match, example 85
 - replacing numbers 56
 - replacing on match 56, 86
- FoxPro 112
 - output format 113
- Fr. (French Franc) 50
- Frequently Asked Questions 127
- FXD 113

G

- garbage characters
 - in input files 32
- Gld (Guilder) 51
- global settings 88
 - line treatment 88

H

- Headers
 - inserting into column 43

- Heading lines
 - defining 23
- Headings 40

I

- Icons
 - DataImport 9
- Include
 - Lines Define. 75, 77
 - Lines Undo. 76
 - menu 75
 - Resume Define. 76
 - Resume Undo. 77
- Including lines 37
 - exact match 37
 - individually 38
 - pattern match 37
- Information Bar 10
- Input File window 10
- Input files 31
 - adding record separators 100
 - example 101
 - cleaning up 34
 - comma delimited 101
 - dBase 100
 - expanding tabs 102
 - example 103
 - garbage characters 32
 - loading 12, 32
 - number of records 31
 - record size 31
 - space delimited 123
 - splitting files 102
 - splitting lines 101
 - statistics 102
 - test & sampling instruments 123
 - unstacking 41, 103
 - example 103
 - unstacking, example 41, 82, 103
- Installation
 - instructions 1
 - Network version 3

L

- L (Lira) 51
- LBL 113
- Learning
 - DataImport 11
- License Key 4
- Limits
 - to exclude data 39
- Line

- (A)bort 81
- (H)eading 80
- (O)utput 81
- (S)kip 80
- (T)itle 81
- Default 80
- Insert Treatments 81
- menu 80
- Undo All Treatments 82
- Line Control Bar 10
- Line split by length
 - process 101
- Line Tags
 - column definition 45, 48
 - defined 44
 - defining 21, 44, 47, 73
 - Defining 22
 - how they work 44
 - reference points 44
 - relation to included lines 48
 - using 46, 73
- Line treatments 40
 - abort 40
 - default 40
 - global setting 88
 - output 38
 - resetting 82
 - restoring default 41, 80
 - skip 40
- Line treatments
 - inserting 81
- Lines
 - (A)bort-ing 40
 - abort at line 81
 - blank, filling 67
 - blank, removing 35
 - column heading 40, 80
 - Default number of lines to load 93
 - default treatment 21, 80
 - excluding 38, 39
 - excluding blank 35
 - excluding duplicate 35, 80
 - excluding first characters 35, 89
 - excluding groups 38
 - extracting 19
 - global output lines mode 40
 - global skip lines mode 40
 - heading 25
 - including 19, 37, 40, 75, 77
 - including individually 38
 - inserting treatments 81
 - resetting treatments 82
 - skipping 80
 - skipping individual 40

- splitting 101
- title 24, 40, 81
- unstacking 41, 103
- Loading input files
 - custom filter 93
 - example 12
- Lotus 1-2-3
 - output format 113

M

- Mailing Label
 - output format 113
- Mask application 9
 - commands 59
 - running 11
- Mask files
 - printing settings 62
 - saving 28
- Mask window
 - explained 10
- Masking
 - example 13
- Masks
 - applying to files 26
- Match String Reference Point
 - Footer 71
- Match Strings
 - defined 39
 - for excluding lines 39
 - to include lines 37
- Maximum users
 - error 3
- MDB 114
- Memory requirements 1
- Menu Bar 10
- Menu Panel
 - turning off 93
- Microsoft Access
 - output format 114
 - Table Names 57
- Microsoft Word
 - output format 114
- Missing text
 - large files 31
- Month names
 - control 90
 - spellings 52

N

- Name Parse
 - column 53
- Named Value

- NVL 116
- naming
 - output files 33
- Negative notation
 - signed overpunch 53
- Network 3
- Network installation 3
 - client users 3
- Network version 3
 - checking users 3
- NKr (Norwegian Krone) 51
- Notation
 - currency 50
 - Date 52
 - decimals 51
 - signed overpunch 53
 - thousands 51
- Numbers
 - credits 50
 - debits 50
 - formats 50
 - negative 50
 - replacing with formulas 56
 - scientific notation 50
 - signed overpunch 53
- Numeric
 - column 50
- Numeric formats 50
- NVL
 - Named Value 116

O

- o
 - on the Line Control Bar 19
- Options
 - Dates. 90
 - Default Line Treatment 92
 - Formula Rows Column Change. 83
 - Formula Rows Display Current Settings 88
 - Formula Rows Insert on Match 85
 - Formula Rows Replace on Match 86
 - Formula Rows Undo 88
 - Global. 88
 - International. 89
 - menu, Mask application 83
 - Preferences. 91
 - Signed Overpunch. 91
- Order
 - columns 41
- Output files 33, 109
 - Alpha 4 112
 - appending 34
 - ASCII delimited 112

- choosing file name 33
- choosing file type 33
- choosing type 26
- Clarion 112
- Clipper 112
- combining 34
- database 57, 112
- databases 34
- Excel 113
- existing 34
- fixed length 113
- FoxPro 112, 113
- interchange 112
- list 110
- Lotus 1-2-3 113
- mailing labels 113
- Microsoft Access 114
- Microsoft Word 114
- Print image 114
- Quattro 114
- Quattro Pro 114
- Quattro Pro 5.0 114
- replacing 34
- Standard Data Format 114
- starting cell 89
- Sylk 114
- Symphony 115
- tab separated variables 115
- text 112
- types 109
- user-defined delimited 115
- using 28
- WordPerfect 115
- XBase 116
- Output files 31
- output formats
 - versions 33

P

- p (peseta) 51
- Page ejects
 - excluding 35
- Pages
 - blank, removing 35
- Parse spaces
 - process 101
- Pause
 - starting in 89
- Pause translation 38
- popup menus
 - using 16
- Positive notation
 - signed overpunch 53

- Precedence, line types 76, 78
- Preference
 - controls 91
- Print image
 - output format 114
- Printing
 - mask settings 62
- PRN 114
- Processes
 - ASCII -> EBCDIC 99
 - Comma Separated Value 99
 - dBase convert 100
 - dBase header 100
 - EBCDIC -> ASCII 100
 - Fixed length 100
 - Line split by length 101
 - Parse spaces 101
 - Records per File Split 102
 - Statistics 102
 - Tab expansion 102
 - Unstack 103
- Programs
 - DataImport 9
- Prompt Line 10

Q

- Quattro
 - output format 114
- Quattro Pro
 - output format 114

R

- Recognizing data types 50
- Records per File Split
 - process 102
- Reference Point
 - Form Length 73
 - Top of Form 73
- Reference Points
 - defined 44
 - defining 21
 - Defining 21
 - using 46
- Registering DataImport 4
- Registration Information 4
- removing
 - characters 34
 - columns 37
- reorganizing data 41
- Replacing lines
 - with formulas 56
- Reports

- form based 46
- Requirements
 - system 1
- Resequencing
 - columns 41
- Rows
 - transposing with columns 49

S

- SDF 114
- Search
 - Edit Replace Strings 65
 - Find Control Codes 64
 - Find First 64
 - Find Last 64
 - Find Next 64
 - Find Previous 64
 - Find Text. 64
 - Go Bottom 65
 - Go Top 65
 - menu 64
 - Replace 65
- Sequence
 - columns 41
- Serial Number 4
- SETUP.EXE 1
- SFr (Swiss Franc) 51
- Showing database structure 62
- Signed overpunch
 - characters 53, 91
 - controls 91
 - custom 53
 - explained 53
 - position 53
- Skipping lines
 - groups 38
 - individually 40
- SKr (Swedish Krona) 51
- SLK 114
- Special characters
 - defined 35
 - excluding 35
 - removing 34
- Spreadsheets
 - Excel worksheet 113
 - formulas 55
 - headings 40
 - Lotus worksheets 113
 - Quattro 114
 - Quattro Pro 114
 - starting cell 89
 - Symphony 115
 - titles 40

- Standard Data Format
 - output format 114
- Statistics 102
 - process 102
- styles
 - recognizing 50
- Support 6
- suppressing
 - characters 34
- Sylk
 - output format 114
- Symbolic Link
 - output format 114
- System requirements 1

T

- Tab expansion
 - process 102
- Tab Separated Variables
 - output format 115
- Table Names
 - Microsoft Access 57
- Tabs
 - expanding 102
 - example 103
- Tag
 - Define Match String Reference Point. 69
 - Line-Tag Define 73
 - menu 69
 - Undo Reference Point. 73
- Task Commander 106
- Task Commander Screen 106
- Task File Dialog Box 106
- Technical Support 6
- Text
 - column 51
- Text block 43
- Thousands separator 51
- Time
 - column 52
- Time format 52
- Title lines 40
 - defining 23
- Titles 40
 - inserting into column 43
- Tools
 - highlighters 14
- Translate application 9
 - automating 117
 - command line 117
 - examples 119
 - window 95
- Translating data 26

- Translations
 - pausing 38
 - running 27
- Transpose
 - rows and columns 89
- Transposing rows/columns 49
- TSV 115
- Tutorial 11
- two-digit year
 - control 91

U

- UDD 115
- Undo
 - columns 37
- Unlocking the Test Drive 4
- Unstack
 - Define 82
 - menu, Mask application 82
 - process 103
 - Undo 83
- User-Defined Delimited
 - output format 115
- Users
 - checking maximum 3
- Utilities 100
 - adding record separators 100
 - example 101
 - application 9
 - converting comma delimited files 101
 - converting dBase files 100
 - EBCDIC to ASCII 100
 - expanding tabs 102
 - example 103
 - file statistics 102
 - splitting files 102
 - splitting lines 101
 - unstacking
 - example 82, 103
 - untacking
 - example 103
 - using 32
- Utilities application 97
 - automating 120
 - processes 98
 - window 97

W

- W50 115
- W51 115
- WB1 114
- What's new in DataImport 6 5

- Windows 1
 - batch programming 117
- WK1 113
- WK3 113
- WK4 113
- WKQ 114
- WKS 113
- Word processors
 - Microsoft Word 114
 - WordPerfect 115
- WordPerfect
 - output format 115
- WQ1 114
- WR1 115
- WRD 114
- WRK 115

X

- XBase
 - output format 116
- XLS 113
- XML
 - Extensible Markup Language 116

Y

- Years
 - two digits 52

