Understanding Data Converter

You use the functions of Data Converter first to set up the conversion information and save it in a Conversion Information File. You then can load conversion information files, run conversions and check conversion results. Use the buttons below to explore the various functions of Data Converter.

Main Window Buttons

The buttons at the base of the main window take you through the main sequence you need to follow to convert files: invoke the Conversion Wizard, preview the conversion, do the conversion, review any errors.

{button ,JI(`',`MDHELP_WIZ')} **Conversion Wizard**

Invokes the Conversion Wizard to help you assemble the conversion information.

{button ,JI(`',`MDHELP_PREVIEW')} **Preview**

Converts the first 100 records and displays the results in the Preview window. Works for all types of files.

{button ,JI(`',`MDHELP_CONTROL_CONV')} **Convert**

Executes the conversion.

{button ,JI(`',`MDHELP_ERRDSP')} **Display Errors**

Invokes a text editor (default is Notepad) to display the Conversion Error File. If there is no error file, this button is disabled.

Menus

{button ,JI(`',`MDHELP_CONTROL_FILEMENU')} **File**

{button ,JI(`',`MDHELP_CONTROL_DSPMENU')} **View**

{button ,JI(`',`MDHELP_CONTROL_TOOLMENU')} **Tools**

{button ,JI(`',`MDHELP_CONTROL_OPTIONMENU')} **Options**

{button ,JI(`',`MDHELP_CONTROL_HELPMENU')} **Help**

{button Limitations and Precautions,JI(`',`MDHELP_CAUTION_CONTROL')}

Executing the Conversion

Click on the Convert button to start the conversion.

Data Converter displays a progress indicator containing:

- name of the input file
- number of bytes converted
- total number of bytes in the file
- number of conversion errors, and
- percentage progress figure.

You can interrupt the conversion process by clicking the **Abort** button.   Abort discards the contents of the partially created output file.

At the end of the conversion process, Data Converter displays the numbers of input records, output records and conversion errors.

File Menu

{button File,JI(`',`MDHELP_CONTROL_FILEMENU')}{button
View,JI(`',`MDHELP_CONTROL_DSPMENU')}
{button
Tools,JI(`',`MDHELP_CONTROL_TOOLMENU')}
{button
Options,JI(`',`MDHELP_CONTROL_OPTIONMENU'
)}{button
Help,JI(`',`MDHELP_CONTROL_HELPMENU')}

Menu Function	**Explanation**
New Setup	Invokes the Conversion Wizard to help you assemble the conversion information for a new conversion process.
Open	Opens a Conversion Information File. Conversion information files can also be opened by dragging them from the Windows Explorer and dropping them on Data Converter. The layout definition file referred to in the conversion information file must be available when the information file is opened.
Save	Updates the conversion information file with any changes made.
Save As	Saves the conversion information to a new conversion information file.
Exit	Exits Data Converter. If you have not saved all your changes Data Converter prompts whether you want to save those changes.

View Menu

{button File,JI(`',`MDHELP_CONTROL_FILEMENU')}{button View,JI(`',`MDHELP_CONTROL_DSPMENU')}{button Tools,JI(`',`MDHELP_CONTROL_TOOLMENU')}{button Options,JI(`',`MDHELP_CONTROL_OPTIONMENU')}{button Help,JI(`',`MDHELP_CONTROL_HELPMENU')}

Menu Function	Explanation
Toolbar	Switches display of the toolbar on and off.
Status bar	Switches display of the status bar on and off.

Tools Menu

{button File,JI(`',`MDHELP_CONTROL_FILEMENU')}{button View,JI(`',`MDHELP_CONTROL_DSPMENU')} {button Tools,JI(`',`MDHELP_CONTROL_TOOLMENU')} {button Options,JI(`',`MDHELP_CONTROL_OPTIONMENU')}{button Help,JI(`',`MDHELP_CONTROL_HELPMENU')}

Menu Function	Explanation
Layout Definition	Starts the Layout Definition Utility.

Options Menu

{button File,JI(`',`MDHELP_CONTROL_FILEMENU')}{button View,JI(`',`MDHELP_CONTROL_DSPMENU')}{button Tools,JI(`',`MDHELP_CONTROL_TOOLMENU')}{button Options,JI(`',`MDHELP_CONTROL_OPTIONMENU')}{button Help,JI(`',`MDHELP_CONTROL_HELPMENU')}

Menu Function	**Explanation**
Environment Setup	Invokes the Data Converter Environment Setup dialog.

Help Menu

{button File,JI(`',`MDHELP_CONTROL_FILEMENU')}{button View,JI(`',`MDHELP_CONTROL_DSPMENU')} {button Tools,JI(`',`MDHELP_CONTROL_TOOLMENU')} {button Options,JI(`',`MDHELP_CONTROL_OPTIONMENU')}{button Help,JI(`',`MDHELP_CONTROL_HELPMENU')}

Menu Function	Explanation
Help Topics	Displays the help topics.
About Data Converter	Displays the version information

Environment Setup

Use this dialog to configure the Data Converter environment.

Item	Explanation
Work Folder	Specify the work folder. By default the system's work folder is used. (Specified in the environment variable TEMP.)
Log File for Batch Execution	Specify the full path for the Log File that is output when executing in Batch mode. Default is to create the file Mdport.log in the work folder.
Program for Displaying Conversion Errors	Specify the full path for a program to display the Conversion Error File . Notepad is used by default.

Creating User Defined Conversion Tables

Data Converter support ASCII and EBCDIC code sets and modifications or additions can be made through User Defined Conversion Tables. Character code conversions can be performed either using code sets that are not supported, or using mappings different than the standard mappings by creating your own conversion table. If desired all single byte code sets can be changed by defining 256 code conversion patterns in the User Defined Conversion Table, but if this is done, the user must specify “ASC” or “EBC” portion of the code system definition.

Format for User Defined Conversion Tables

User defined conversion tables are text files, that you create with any standard text file editor.

The conversion table consists of three types of lines:

Comment lines - these have a "*" in column 1

Code system definition lines - these have a "#" in column 1

and

Code mapping lines - these have a space in column 1.

Code system definition lines give ID's of the source and target code systems. When you convert files you enter the Code ID to specify that a particular file is in the defined code.

Code mapping lines follow each code system definition line. They give byte to byte (or double-byte to double-byte) mappings.

Format for Code System Definition Lines

```
# Code-system-name-1  Code-system-name-2
```

Where:

# is specified in column 1.

Code-system-name can be "ASCI" or 'EBCD"

ASCI: ASCII Code

EBCD: EBCDIC Code

Code-system-name-1 is the code used in the input file.

Code-system-name-2 is the code used in the output file.

Format for Code Mapping Lines

```
 Character-code-1:Character-code-2
```

Where:

Space is specified in column 1.

Character-code-1 and Character-code-2 are separated by a ":" (colon) with no spaces.

Character-code is a 2 or 4 digit hexadecimal code (2 digits for single byte character sets, 4 digits for two byte character sets)

Character-code-2 can have the value "NO". This means that no corresponding code is available.

If there are multiple code mapping lines for the same character-code-1 value, the mapping defined in the last of these lines is used.

Example of a User Defined Conversion Table (SAMPLE.CNV)

```
* Conversion from EBCDIC to ASCII,
* Substitute two values to the standard mapping
# EBCD  ASCI
 00:5F
 5B:NO
* Conversion from ASCII to in-house code set by substituting
*  all byte mappings in character set "EBCDIC"
# ASCI  EBCD
 00:00
 ...
 20:40
 21:44
 ...
* (Defines mappings for all byte values)
```

Overview of Preview

Data Converter provides a Preview function to help you check the conversion setup before running a full conversion.

Preview converts the first 100 records of the input file then displays the results. Note: this function only works with binary, text and CSV format files, and will not work for sequential, indexed and relative files. The display window has configurable column gridlines so you can easily check item positions and also gives the option of viewing the data in hex.

The Preview menus are described in the following topics:

Menus

{button ,JI(`',`MDHELP_PREVIEW_EXITMENU')} File menu

{button ,JI(`',`MDHELP_PREVIEW_OPTIONMENU')} Option menu

{button ,JI(`',`MDHELP_PREVIEW_HELPMENU')} Help menu

{button Limitations and Precautions,JI(`',`MDHELP_CAUTION_PREVIEW')}

Preview Function - File Menu

{button File,JI(`',`MDHELP_PREVIEW_EXITMENU')}{button Option,JI(`',`MDHELP_PREVIEW_OPTIONMENU')} {button Help,JI(`',`MDHELP_PREVIEW_HELPMENU')}

The File menu has a single function - Exit.   It returns you to the Data Converter main window.

Preview function - Option Menu

{button File,JI(`',`MDHELP_PREVIEW_EXITMENU')}{button Option,JI(`',`MDHELP_PREVIEW_OPTIONMENU')}
{button Help,JI(`',`MDHELP_PREVIEW_HELPMENU')}

Menu Function	Explanation
HEX Display	Displays the data in hex underneath the characters.
Column Gridlines	Controls the density of the column lines. Select from lines every 10, 5 or 1 characters, or no gridlines.
Display Conversion Errors	Displays any errors that occurred in converting the first 100 records using the conversion error display function.

Preview function - Help Menu

{button File,JI(`',`MDHELP_PREVIEW_EXITMENU')}{button Option,JI(`',`MDHELP_PREVIEW_OPTIONMENU')} {button Help,JI(`',`MDHELP_PREVIEW_HELPMENU')}

The Help function invokes the Preview help.

Layout Definition Utility

The Layout Definition Utility helps you create the layout definition files used by Data Converter. To do this it analyzes COBOL library record definitions, and lets you edit the item information and the data format. The topics below guide you through the layout Definition Utility processes and functions.

Creating Layout Definition Files

{button ,JI(`',`MDHELP_LAYOUT_START')} Invoking the Layout Definition Utility

{button ,JI(`',`MDHELP_LAYOUT_COPYIN')} Analyzing COBOL Libraries

{button ,JI(`',`MDHELP_LAYOUT_DATATYPE')} Specifying the Data Format

{button ,JI(`',`MDHELP_LAYOUT_ITEMINF')} Understanding Item Information

{button ,JI(`',`MDHELP_LAYOUT_ITEMEDIT')} Editing Item Information

Menu Descriptions

{button ,JI(`',`MDHELP_LAYOUT_FILEMENU')} File menu

{button ,JI(`',`MDHELP_LAYOUT_EDITMENU')} Edit menu

{button ,JI(`',`MDHELP_LAYOUT_SETMENU')} Layout menu

{button ,JI(`',`MDHELP_LAYOUT_DSPMENU')} View menu

{button ,JI(`',`MDHELP_LAYOUT_HELPMENU')} Help menu

{button Limitations and Precautions,JI(`',`MDHELP_CAUTION_COPY')}

Invoking the Layout Definition Utility

You can invoke the Layout Definition Utility in any of the following ways:

- Selecting "Layout Definition" from the Start Programs menu
- Selecting Layout Definition from the Data Converter Tools menu
- Selecting Update Layout Definition File from the Conversion Wizard Step 1.
- Opening the **LAYOUT.EXE** application from Windows Explorer

Layout Definition Utility - Analyzing COBOL Libraries

Although you can enter the whole layout definition yourself it is generally better to start by having the Layout Definition Utility analyze a COBOL COPY Library.   To do this use the Create from COBOL Library function on the File menu.   This displays an Open file dialog in which you select the COBOL library to be analyzed.

The Layout Definition Utility analyzes the data descriptions in the COBOL library and automatically populates the layout for both input and output files.   You can then make any edits that are required and save the layout to a Layout Definition File.

{button Limitations and Precautions,JI(`',`MDHELP_CAUTION_COPY')}

Layout Definition Utility - Specifying the Data Format

You need to select which of two data formats apply to the input and output files. The two formats are:

- Data File Format
- CSV Format

Precaution on Changing the Data Format

If you change between Data File Format and CSV Format layout definition information is lost. When you convert from CSV Format to Data File Format you need to input the positions and item lengths for all the items. It is often easier to re-analyze the COBOL COPY library again than to change the data format.

Layout Definition Utility - Understanding Item Information

The table below lists all the information that the Layout Definition Utility handles for each item. Item Extension Information is indicated by a "*".

A number of the labels for the information columns are truncated in the display, but note that the utility displays the full label and contents of the cell in focus in a row above the column headings.

Select Extended Display on the View menu to display the Extension Information.

Type	Meaning
Lv	Displays the repetition levels of 0 ~ 7. When there are no repetitions the value of the repetition level is 0. Note that the repetition levels are different from the COBOL data description level numbers. Refer to Defining repeating items for a full description of repetition levels.
Repeat	Displays the number of repetitions. Lines with repetition counts only have the repetition level and repetition count. Refer to Defining repeating items for a full description.
Item-name	The name of the item. Item names do not affect the conversion so you can edit or abbreviate names if you wish.
Attribute	Displays the Attribute of the item. Attributes are divided into character and numeric types. Layout Definition generates attributes based on the COBOL data description. You cannot convert between character and numeric attributes.
Sign	(For numeric attributes items) For external decimal data (attribute "Z") the sign can take one of the External decimal sign types. For other numeric fields displays "S" for fields containing signs.
Offset	(For data file format) Displays the offset of the item within the record. The position starts from 1.
Item Length	Displays the length of the item in bytes.
Total digits	(For numeric attribute items in data file format) Displays the total number of digits for numeric items. The total digits figure is the sum of the number of digits in the integer part and the number of digits after the decimal point.
Decimal	(For numeric attribute items in data file format) Displays the number of digits after the decimal point.
Quote(*)	(For CSV format) Displays "Y" for items that have quotes. By default character type items are enclosed in quotes and numeric type items are not in quotes.

Input check(*) Enter “Y” to have the input data checked for match with the item attributes. It does not affect the output file, only the messages that are written to the error file.

No conversion(*) Enter “Y” by items that you want to be output as they are, with no conversion.

Fixed output(*) Enter “Y” for variable length items that are to be output using the specified length.

Attribute types M, R, and V have variable length and normally only output sufficient bytes to show the significant data. If you specify Fixed Output these fields are padded with spaces, or truncated, so that the bytes output match the specified length.

Type R fields have leading spaces inserted or are truncated from the left.

Type M fields have trailing spaces added and are truncated from the right.

Type V fields can have lengths specified for both total digits and decimal digits. Decimal digits are counted from the decimal point - spaces are added to fill to the specified length. Total digits are then counted from the right and the spaces inserted before the integer digits or the integer is truncated.

If no length is specified with this option checked then the length is taken to be zero, so make sure you specify lengths with this option checked.

Zero suppress(*) (Valid for numeric character strings with fixed output)

Enter "Y" to specify that leading zeros in numeric character strings (V)are output as spaces.

Null(*) (For character items)

Enter "I" to suppress conversion errors when input alphanumeric items contain nulls (0x00).

Layout Definition Utility - Editing Item Information

Layout definition files contain a lot of information about each item in the records to be converted. You can edit the Item Information in one of two ways:

Using the Item Definition Template

The Item Definition Template is a dialog box giving you access to all the basic information for each item. It does not give access to Item Extension Information so do not use the template for editing items with extension information. Even if you do not want to edit the extension information, using the template will cause the extension information to be lost.

The template is a good way of editing items that do not have extension information. It has the advantage over direct editing that it provides you with a full description of each item and drop down lists of attribute codes. The template also checks your edits for consistency when you press the OK button, and informs you of any errors so you can immediately correct them.

Invoke the Item Definition Template dialog by one of the following methods:

- Double clicking on an item in the table

- Selecting an item and pressing the F8 key

- Selecting an item and selecting Item Definition from the Layout menu.

Direct Editing Using the ENTER key

You can edit information directly in the table by putting focus on a cell and pressing Enter. The cell changes to an edit field so you can enter values directly in the cell. Pressing Enter again confirms your changes; pressing Esc cancels any changes.

The Layout Definition Utility does not check your input for consistency or correctness until you save the the Layout Definition File. It is therefore advisable to make only a few changes by direct editing or to save the file frequently if making numerous direct editing changes.

You must use direct editing to change Item Extension Information. Display extended information by selecting Extended Display from the View menu.

Attribute Descriptions

Items are divided into character and numeric types.

CSV format files can contain only "M" and "V" attributes.

An "*" in the meaning column marks the attributes in which the internal format differs according to the Endian of the supporting system.

Type	Display Character	Common Description	Meaning
Char	**X**	Alphanumeric char	String of 1 byte characters
	C	Char type	Character string terminated by a null byte
	R	Variable length string	Character string with data length in the first 2 bytes(*)
Num	**Z**	External decimal	COBOL numeric display format
	P	Internal decimal	COBOL packed decimal format
	B	Binary	COBOL binary format
	5	COMP-5	COBOL85 binary format (*)
	S	Short type	16 bit binary (*)
	L	Long type	32 bit binary (*)
	V	Numeric string	Numeric in character form (Signs and the decimal point are included as characters)
Other		No character	Use for addition, deletion of items

Attribute Descriptions

Items are divided into character and numeric types.

CSV format files can contain only "M" and "V" attributes.

An "*" in the meaning column marks the attributes in which the internal format differs according to the Endian of the supporting system.

Type	Display Character	Common Description	Meaning
Char	**X**	Alphanumeric char	String of 1 byte characters
	M	Mixed character	String of 1 byte (and, for Japanese working, 2 byte) characters.
	C	Char type	Character string terminated by a null byte
	R	Variable len. Str.	Character string with data length in the first 2 bytes(*)
Num	**Z**	External decimal	COBOL numeric display format

	P	Internal decimal	COBOL packed decimal format
	B	Binary	COBOL binary format
	5	COMP-5	COBOL85 binary format (*)
	S	Short type	16 bit binary (*)
	L	Long type	32 bit binary (*)
	V	Numeric string	Numeric in character form (Signs and the decimal point are included as characters)
Other		No character	Use for addition, deletion of items

External decimal sign types

External decimal items can have the following sign types:

Symbol	**Meaning**
S	Same as "T”.
L	LEADING sign. Stored in the high order 4 bits of the first byte.
T	TRAILING sign. Stored in the high order 4 bits of the last byte.
LS	LEADING SEPARATE sign. Stored in the first byte.
TS	TRAILING SEPARATE sign. Stored in the last byte.

Layout Definition Utility - Defining Repeating Items

The Layout Definition Utility uses the repetition level (labeled "Level") column and "Repeat" column to communicate repeating groups of items, usually defined in COBOL OCCURS clauses.

The Layout Definition Utility only displays elementary items. To indicate that a group of items is repeated the utility inserts a repetitive definition line immediately before the repeating group. This line only contains a level number and a repetition count.

The repetition level indicates the level of nesting of repeating groups. The repetitions column gives the number of times the group is repeated. The level number is incremented in the line after the repetition line.

The utility supports up to 7 levels of repetition nesting.

Example

[Data description of COBOL]

```
01   ABC-REC.
   03   ITEM1     PIC X(10).
   03   GROUP1.
      05   ITEM2   PIC X(3).
      05   ITEM3   PIC X(1).
   03   GROUP2   OCCURS 10.
      05   ITEM4   PIC X(5).
      05   ITEM5   PIC X(5).
      05   GROUP2-1   OCCURS 5.
         07   ITEM6   PIC 9(2).
      05   ITEM7   PIC X(2).
   03   ITEM8   PIC X(20).
   03    ITEM9     OCCURS   5     PIC
X(8).
   03   ITEM10   PIC X.
```

[Layout definition]

Level	Repetitions	Item name
0		ITEM1
0		ITEM2
0		ITEM3
0	10	
1		ITEM4
1		ITEM5
1	5	
2		ITEM6
1		ITEM7
0		ITEM8
0	5	
1		ITEM9
0		ITEM10

Layout Definition Utility - Item Definition Template

Use the Item Definition Template dialog to edit the Item Information.

Do not use the Item Definition Template for items with extended information as the template initializes the extension information fields.

Automatic Update of Positions

When you change the length of an item you affect the positions of all the following items in the record. The Layout Definition Utility prompts you when you change item lengths, asking whether you want the positions updated automatically. If you do not have the positions updated automatically you must edit them yourself.

The automatic update function does not update repetition levels and numbers of repetitions. Also it cannot be used when the item whose length has been changed has a repetition level greater than 1.

Layout Definition Utility - Special Codes

Use this dialog to specify special behaviors.

Item	Explanation
Layout	Specify the layout name.(treated as comment only)
Record	Specify the record name.(treated as comment only)
Null display area for all item	(For data file format) If selected inserts a 2 byte Null Display Area before each item. The actual item length becomes 2 bytes longer than the length in Item Information. When you change the selection you must also change the positions of each item. The All items function can be used to update the positions..
Use binary as big endian	Specifies the Endian of binary numeric on data. Selecting this item affects the following: Internal display of COMP-5, short, and long items. The data length added to variable length items (attributes R and Y) Recommended to check this when you are targeting SPARC machines.
Take NULL in the input data as string terminator	(For input data file format files) When the input items are character strings (item attributes X, N, M, R), take null characters in the input data as indicating the end of the string. Conversion errors will occur if this box is not checked and nulls are encountered in the above character strings. "C" attribute items always take null as the string terminator.

Layout Definition Utility - All Items

The All Items function provides a list of functions that can be applied to all the items in the layout definition table.

Item	Explanation
Setup Menu	Select a function from the list. See the descriptions in the explanation box for details of what each function does.
Setup Object	Select whether the target of the function is the input or output definitions or both. Input/Output selections are disabled according to what makes sense for the function.
Explanation	Displays a description of the function selected from the Setup Menu.

Layout Definition Utility - File Menu

{button File,JI(`',`MDHELP_LAYOUT_FILEMENU')}{button Edit,JI(`',`MDHELP_LAYOUT_EDITMENU')}
{button
Layout,JI(`',`MDHELP_LAYOUT_SETMENU')}
{button View,JI(`',`MDHELP_LAYOUT_DSPMENU')}
{button
Help,JI(`',`MDHELP_LAYOUT_HELPMENU')}

Menu Function	Explanation
Create from	
COBOL Library	Utility automatically generates the item definition from a COBOL COPY Library. See Analyzing COBOL Libraries
New	Clears the display of the current layout definition ready to create a new Layout Definition File.
Open	Opens an existing Layout Definition File.
Save	Saves edits to the Layout Definition File. Checks the layout definition for correctness and asks you whether or not to save the file if it has errors.
Save As	Saves the Layout Definition File and edits to a new file. Checks the layout definition for correctness and asks you whether or not to save the file if it has errors.
Exit	Closes the Layout Definition Utility.

Layout Definition Utility - Edit Menu

{button File,JI(`',`MDHELP_LAYOUT_FILEMENU')}{button Edit,JI(`',`MDHELP_LAYOUT_EDITMENU')}
{button
Layout,JI(`',`MDHELP_LAYOUT_SETMENU')}
{button View,JI(`',`MDHELP_LAYOUT_DSPMENU')}
{button
Help,JI(`',`MDHELP_LAYOUT_HELPMENU')}

Menu Function	Explanation
Cut	Cuts selected line(s) to the copy buffer.
Copy	Copies selected line(s) to the copy buffer.
Paste	Contents of the copy buffer are inserted before the current line.
Insert	Inserts a new line before the current line.
Delete	Deletes the selected line(s).
Append	Adds a new line after the last line.

Layout Definition Utility - Layout Menu

{button File,JI(`',`MDHELP_LAYOUT_FILEMENU')}{button Edit,JI(`',`MDHELP_LAYOUT_EDITMENU')} {button Layout,JI(`',`MDHELP_LAYOUT_SETMENU')} {button View,JI(`',`MDHELP_LAYOUT_DSPMENU')} {button Help,JI(`',`MDHELP_LAYOUT_HELPMENU')}

Menu Function	**Explanation**
Item Definition	Displays the Item Definition Template for editing the information in the current line.
Special Codes	Sets treatment of Null Area, Null string terminators and big endian behavior.
Data Format	Sets the input and output data formats.
All Items	Provides a list of functions to apply to all the items.

Layout Definition Utility - View Menu

{button File,JI(`',`MDHELP_LAYOUT_FILEMENU')}{button Edit,JI(`',`MDHELP_LAYOUT_EDITMENU')}
{button
Layout,JI(`',`MDHELP_LAYOUT_SETMENU')}
{button View,JI(`',`MDHELP_LAYOUT_DSPMENU')}
{button
Help,JI(`',`MDHELP_LAYOUT_HELPMENU')}

Menu Function	Explanation
Toolbar	Displays or hides the toolbar.
Status Bar	Displays or hides the status Bar.
Reset Cell Sizes	Resets the cell width of the item definition table and resizes the window.
Extended Display	Displays or hides the Item Extension Information.

Layout Definition Utility - Help Menu

{button File,JI(`',`MDHELP_LAYOUT_FILEMENU')}{button Edit,JI(`',`MDHELP_LAYOUT_EDITMENU')}{button Layout,JI(`',`MDHELP_LAYOUT_SETMENU')}{button View,JI(`',`MDHELP_LAYOUT_DSPMENU')}{button Help,JI(`',`MDHELP_LAYOUT_HELPMENU')}

Menu Function	Explanation
Help Topics	Displays the Topic search dialog box.
About Data Converter	Displays version information.

The glossary of terms used is given below.   Click an item to see the explanation.

{button ,PI(`',`WORD_IPPANFILE')}   Binary File

{button ,PI(`',`WORD_COBOLFILE')}   COBOL85 File

{button ,PI(`',`WORD_OPTIONCODE')}   Code conversion option

{button ,PI(`',`WORD_ERRFILE')}   Conversion error file

{button ,PI(`',`WORD_CONTROLFILE')}   Conversion information file

{button ,PI(`',`WORD_COPY')}   COPY library

{button ,PI(`',`WORD_CSVTYPE')}   CSV format

{button ,PI(`',`WORD_DATAFILEMODE')}   Data file conversion

{button ,PI(`',`WORD_DATATYPE')}   Data file format

{button ,PI(`',`WORD_ENDIAN')}   Endian

{button ,PI(`',`WORD_ITEMEXTINF')}   Item extension information

{button ,PI(`',`WORD_LAYOUTFILE')}   Layout definition file

{button ,PI(`',`WORD_NULLDSP')}   Null display area

{button ,PI(`',`WORD_POWERGEM')}   PowerGEM Plus

{button ,PI(`',`WORD_TEXTMODE')}   Text conversion

{button ,PI(`',`WORD_CVTABLE')}   User defined conversion table

COBOL85 Files

Data Converter directly supports the following COBOL85 file organizations:

Sequential
Relative
Indexed

Data Converter does not directly support Line Sequential files but they can be handled as text files.

It also supports data files created using COBOL85 programs on mainframes.

COPY Library

A COBOL source file which, for Data Converter's purposes, contains a record description

CSV Format

Comma-separated Values format. Data is stored in text form, with data items separated by commas. Data Converter lets you configure other characters to be used as separators.

CSV format files are generally used for exchanging data with spreadsheets and relational databases.

You select whether a file is in CSV format or Data File Format in the Layout Definition Utility.

PowerGEM Plus

A Fujitsu Windows product that combines resource management and integration of tools.

Binary File

Term used to describe files that are neither COBOL format files nor text files. For example:

- Files transferred from mainframe and UNIX machines.

Endian

Term used to describe the internal format of binary numbers. There are two formats Big Endian and Little Endian. The format used is determined by the machine and operating system.

Big Endian

Numbers are stored so that the higher address bytes contain the least significant digits. Big Endian is found on machines such as UNIX SPARC machines.

Little Endian

Numbers are stored so that the higher address bytes contain the most significant digits. Little Endian is found in Windows Intel compatible machines.

User Code Conversion Option

Data Converter can use a code conversion table that you create.　With this option you can handle files using code systems other than those supported directly by Fujitsu.

Item Extension Information

Term used to describe additional information that guides conversion of items and can be displayed and edited in the Layout Definition Utility.

Data File Format

Files in which data items are distinguished only by their byte position within records. Files created by COBOL programs are in this format. The alternative is CSV format in which data items are variable length and separated by commas.

Data File Conversion

Use data file conversion mode for all conversions other than converting text files. Select data file conversion mode for conversions involving CSV (comma-separate values) files.

Text Conversion

Use text conversion mode to convert files containing only text. Data Converter supports conversion of fixed or variable length format text files, in ASCII or EBCDIC codes, with CR/LF or LF as variable length record terminators.

Null Display Area

A 2 byte area added at the beginning of items and used by some software products. When the bytes contain 0xFFFF (hexadecimal) the following data is treated as null; when the bytes contain 0x0000 (hexadecimal) they are not treated as null.

Conversion Error File

When conversion errors occur Data Converter stores information about the errors in a text file with a .ERR extension. You can check the error information with a text file editor such as Notepad. Data Converter lets you view the file by clicking on the Display Errors button.

Conversion Information File

You save the conversion information in a file with a .CIF extension.   Data Converter reloads the conversion information when you select the file using the File, Open function.

User Defined Conversion Table

A text file that contains mappings of byte values so that Data Converter can be used to convert files using character codes other than standard ASCII or EBCDIC.

Layout Definition File

File containing the details and mappings of input data items to output data items.   It is created by the Layout Definition Utility and its extension is ".lay".

Using the Conversion Wizard

You use the Conversion Wizard to set up all the information required for the conversion.

{button ,JI(`',`MDHELP_WIZ_INIT')}   Step 1 - Common Information
{button ,JI(`',`MDHELP_WIZ_LAYOUT')}   Step 2 - Layout Definition Information
{button ,JI(`',`MDHELP_WIZ_FILE')}   Steps 3 & 4- Input/Output File Information
{button ,JI(`',`MDHELP_WIZ_OTHER')}   Step 5- Other Information

Conversion Wizard Step 1 of 5 - Common Information

{button Step1 ,JI(`',`MDHELP_WIZ_INIT')}{button Step2,JI(`',`MDHELP_WIZ_LAYOUT')}{button Steps3+4,JI(`',`MDHELP_WIZ_FILE')}{button Step5,JI(`',`MDHELP_WIZ_OTHER')}

Specify File Conversion Mode

Select Data File Conversion or Text Conversion for the file conversion mode.

Setup Default Folder

If you specify a default folder you can setup each file name by using a relative path name.

If you use a conversion information file at a later date and the files have been moved to another folder, you need only change the default folder name.   The other relative file names will be still be valid.

Conversion Wizard Step 2 of 5 - Layout Definition Information

{button Step1 ,JI(`',`MDHELP_WIZ_INIT')}{button Step2,JI(`',`MDHELP_WIZ_LAYOUT')}{button Steps3+4,JI(`',`MDHELP_WIZ_FILE')}{button Step5,JI(`',`MDHELP_WIZ_OTHER')}

Note : Conversion Wizard does not display this page for Text Conversion mode.

Specify Layout Definition File

Specify the name of a Layout Definition File.

Update Layout Definition File

Executes the Layout Definition Utility so that you can check or edit the layout definition.

Create New Layout Definition File

Create the Layout Definition File from a COBOL library.

Conversion Wizard Steps 3 and 4 of 5 - Input / Output File Information

{button Step1 ,JI(`',`MDHELP_WIZ_INIT')}{button Step2,JI(`',`MDHELP_WIZ_LAYOUT')}{button Steps3+4,JI(`',`MDHELP_WIZ_FILE')}{button Step5,JI(`',`MDHELP_WIZ_OTHER')}

Code System

Selects the file code system from ASCII or EBCDIC.

File Name

Specify the Input or Output file name.   Relative path names can be used if a Default Folder is set.

Data Format (For data file conversion)

Specify Binary or Text File or a COBOL85 File organization.

Note that CSV format is selected in the Layout Definition Utility, in the Layout, Data Format function.

Key Information (For indexed output files)

Key Information.

CSV Format Option (For CSV format files)

CSV Format Details (For CSV format)

Note that CSV format is selected in the Layout Definition Utility, in the Layout, Data Format function.

Specify Record Attributes

Specify the record attributes from one of the combinations listed below:

Type of File	Attributes	New Line Code
Fixed length data file format	Fixed length	No (Record length is the same as in the layout definition)
Windows/DOS CSV format	Variable length	New line code (CR+LF)
Windows/DOS text	Variable length	New line code (CR+LF)
UNIX text and CSV	Variable length	New line code (LF)
Mainframe source	Fixed length	No (Record length : 80)

Conversion Wizard Step 5 of 5 - Other Information

{button Step1 ,JI(`',`MDHELP_WIZ_INIT')}{button Step2,JI(`',`MDHELP_WIZ_LAYOUT')}{button Steps3+4,JI(`',`MDHELP_WIZ_FILE')}{button Step5,JI(`',`MDHELP_WIZ_OTHER')}

Code Conversion Specification Information

If required enter the name of a User Defined Conversion Table file that specifies the code conversion to be used. The topic: User Defined Conversion Table gives details of the file format.

Error File Information

File Name

Enter the full path name of a Conversion Error File. Relative path names can be used if a Default Folder is set.

By default the error file has an extension of ".err".

Conversion Error Limit

Specify the maximum number of error messages to be written to the Conversion Error File. The maximum can be in the range 1 to 999. The default is 100.

Continue even if Errors Exceed the Limit

When this box is checked Data Converter runs the conversion to the end regardless of the number of errors.

When this box is not checked Data Converter displays a dialog box when the limit is reached, or in Batch mode. Data Converter aborts the conversion without displaying a message.

1 Byte System Substitute Code

When Data Converter encounters an unsupported character code in the input file, it outputs the character specified in the field. Specify the value using two hex digits.

The default substitute code is underscore (_).

Conversion Wizard - Key Specification

For indexed files use this dialog box to specify the keys.

Keys

Keys are specified using the following format:

```
offset-1,length-1 [/offset-2,length-2] ...
```

Offset and lengths are numbers of bytes.   Offset start from 1.

Where keys have multiple parts use a "/" (slash) to separate the parts.

Example   "1,10/15,5"

(This is a two part key, the first part is 10 bytes long starting at the first byte, the second part is 5 bytes long starting at the 15th byte).

Duplicate Key

Check this box when duplicate keys are allowed.

Conversion Wizard - CSV Information Setup

Specify how Data Converter should handle CSV (comma-separated values files).

Item	Explanation
Quotes	Specifies the quote marks used for character strings.
Delimiter	Specifies the delimiter between CSV fields. The default is comma "," which is the standard CSV delimiter.
Output quoted data using double quotes	Specifies that quotes within character strings should be made into double quotes. For example the string: 'Label is "ABC".', would be output as: "Label is ""ABC""."
Output null character as double quotes	Specifies that character fields containing null should be output as a double quote. For example, the following data: 02 ITEM1 PIC 9(3) VALUE 100. 02 ITEM2 PIC X(5) VALUE LOW VALUES. 02 ITEM3 PIC 9(3) VALUE 100. Would be output as: 100,"",100, with this box checked, and 100,,100 with the box not checked.
Output item header in first record	Only valid for output files. If checked Data Converter outputs the item names in the first record of the CSV file. This makes the CSV file convenient to use in a spreadsheet.

Conversion Wizard - Layout Definition File Creation

This dialog box is displayed when you request the Conversion Wizard to create a layout definition file automatically.

After you specify the COBOL file to analyze, Data Converter prompts you for the remaining information required to create the Layout Definition File.   The created layout definition file can be viewed and modified using the Layout Definition Utility.

Items	Explanation
Input Data	Select the input file format from Data File Format or CSV Format.
Output Data	Select the output file format from Data File Format or CSV Format.
Specify new layout definition file name	Enter a name for the Layout Definition File to be created.

Displaying Conversion Errors

Data Converter stores errors encountered during conversion in a Conversion Error File . Data Converter also provides a function to display the contents of the conversion error file using an editor specified in Environment setup. You display the errors by pressing the Display Errors button.

Example of a Conversion Error File (SAMPLE.ERR)

```
*** Data Converter conversion error file ***

Input File : V:\HOME\MDPORT32\TEST\DATA1.DAT

Layout definition file : V:\HOME\MDPORT32\TEST\DATA1.LAY

Conversion error count: 2

[Code Conversion Error] Number of occurrences indicated in ( ) (*) more
than 100

0000( 1)


[Error Items] Item No:Item name:Attribute(Number of errors occurred; "*" is
more than 100)

3:Product name:N( 1)

5:Sales volume:Z( 1)


[Error Details] Record number-Position  Error contents  Item No:Item
name:Attribute

3-23 Code Conversion Error(0x0000) 3:Unit price:Z

4-46 Numeric part error 5:High sales:Z
```

{button Limitations and precautions,JI(`',`MDHELP_CAUTION_ERRDSP')}

Code Conversion Error

List of character codes that caused code conversion errors.

Format

Hexadecimal-character-code (*Number-of-occurrences*)

Error Items

List of items that contained conversion errors. (Data file conversion only)

Format

Item-No- in-layout-definition : Item-name :Item-attribute (Number-of-errors)

Error Details

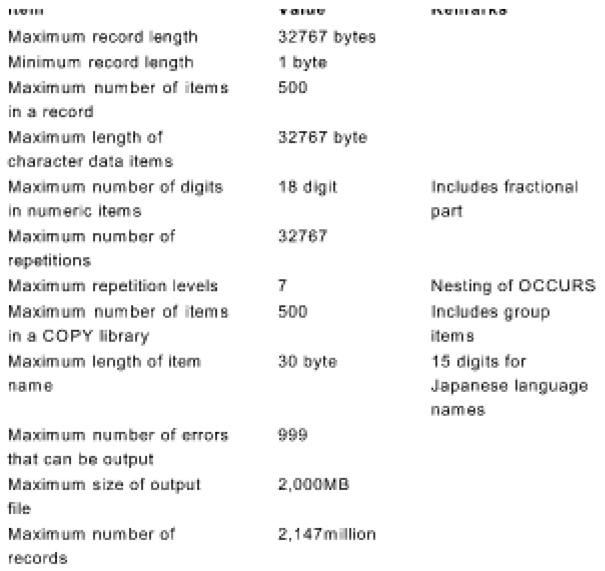
Provides a detailed breakdown of each conversion error in the order they occurred. The number of errors reported is determined by the maximum specified in the Conversion Wizard.

Format

Input-record-number : Position-within-the-record Error-contents Item-No.-in-Layout-definition **:** *Item-name* **:** *Item-attribute*

Data Limits and Unsupported File Formats

Data Size Limits



Item	Value	Remarks
Maximum record length	32767 bytes	
Minimum record length	1 byte	
Maximum number of items in a record	500	
Maximum length of character data items	32767 byte	
Maximum number of digits in numeric items	18 digit	Includes fractional part
Maximum number of repetitions	32767	
Maximum repetition levels	7	Nesting of OCCURS
Maximum number of items in a COPY library	500	Includes group items
Maximum length of item name	30 byte	15 digits for Japanese language names
Maximum number of errors that can be output	999	
Maximum size of output file	2,000MB	
Maximum number of records	2,147million	

Unsupported File Formats

Data Converter does not support:

- Mainframe variable length record files.
- Files with multiple record format (multi-format)

Conversion Information File - Precaution

Data Converter stores full path names for the Layout Definition, Input and Output files in the Conversion Information File. Consequently you will generate errors if the layout definition file does not exist when the Conversion Information File is saved, or if these files are moved to other folders.

Precautions for Handling COBOL85 Files

Take the following precautions when using COBOL85 Files:

- Data Converter does not take exclusive control of COBOL85 files. Therefore do not update the input files from other applications while running Data Converter.
- Data Converter reports an error and aborts the conversion process if duplicate key values are found for output indexed files defined with no duplicates.
- Data Converter picks up the organization, record attributes and record lengths from the header records of COBOL85 indexed files. If the information from the file does not match the information defined in the Conversion Information File, Data Converter ignores the Conversion Information File information and continues with the conversion.

Limitations and Precautions for the Preview Function

- Preview does not work on COBOL structured files i.e. those files with sequential, relative or indexed organizations.
- Preview can only interpret ASCII character values. If the output file is in a character code other than ASCII you need to preview the values in Hex.
- Preview uses only “Courier New” font.
- Non-character codes, other than new line character, are displayed as “.” (period).

Limitations and Precautions for the Conversion Error Display Function

- The function is only enabled if the conversion error file exists.
- The program for viewing the conversion error file is activated from the command line. If the program does not have a command line interface it will not work.

Limitations for Layout Definition Library Analysis

- The analysis expects correct COBOL syntax. Incorrect syntax may produce unpredictable results.
- The maximum number of items in the library, including group items, is 500. If a library contains more than 500 items you need to reduce the number of items by combining items containing details that are not significant to the conversion (for example adjacent character items might be combined into a single item).
- The COPY library should start with an 01 level item.
- When there are multiple 01 level items, only the first 01 level item is analyzed.
- Analysis does not support the following features. You need to edit these out of the COPY library.
    - Items having level numbers other than 0149.
    - SYNCHRONIZED(SYNC) clause.
    - OCCURS DEPENDING ON clause.
- Analysis ignores REDEFINES items. Only the redefined item descriptions are used.
- Analysis interprets the following descriptions as numeric items:
    - Pointer items
    - Boolean items
    - Index items
    - Floating point items
- Numeric edited items and alphanumeric edit items are interpreted as alphanumeric item attributes.
- Analysis ignores the following clauses:
    - KEY IS
    - INDEXED BY
    - JUSTIFIED
    - BLANK
    - VALUE
    - CHARACTER TYPE
    - PRINTING POSITION
    - BASED ON
- Data item names can be no longer than 30 characters.

Supported Character Conversions

Data Converter has built-in support for conversions between ASCII and EBCDIC code sets.   Adjustments to the standard conversions can be made using User Defined Conversion Tables.

When converting character fields (for example fields with attribute X) Data Converter checks that the characters are valid.   These checks are performed whether or not the input and output code systems are different.   Data Converter reports conversion errors when non-character codes are found in input character fields.   ASCII tab characters (0x09), and EBCDIC values 0x28 and 0x29, do not cause errors.

You can switch off the code checking for a field by specifying "Y" in the "No Conversion" extension information in the Layout Definition utility.

See Creating User Defined Conversion Tables for more information on user defined conversion tables.

Size Errors when Converting Character Strings

For character string conversions, Data Converter ignores trailing spaces when reporting size errors. Data Converter only reports a size error if the string that remains, after the trailing spaces are removed, is too big for the target field.

For CSV files a size error is reported only if the string length is greater than the maximum allowed.

Rules for Converting Numeric Items

Signs in CSV Format

CSV Output Files

Numeric items (attribute "V") with a sign ("S" in the sign column) have “+” or “-” placed at the beginning of the items depending on the sign of the input item.

CSV Input Files

Data Converter checks for "+" and "-" signs at the beginning and end of the field and sets the sign of the output number accordingly.   If there is no sign character the output is set to positive.

Values Output when Numeric Errors are Detected

Numeric errors occur when input numeric items do not contain valid values.

When the output numeric field contains characters (types Z and V) Data Converter will read the lower 4 bits and output a decimal digit based on that value (mod 10) (i.e. the remainder when the value is divided by 10).   So, if the input field contains "A" (0x41), the output is the digit 1; if the input is "L" (0x4C), the output digit is 2 (0xC divided by ten leaves a remainder 2).

For all other output numeric types Data Converter outputs zero when the input data is invalid.

Digit Size Errors

Data Converter reports digit size errors if there are insufficient digits in the integer part of the output number to hold the digits from the input number.

Overview of Batch Mode

You can run Data Converter in batch mode.

Batch mode indicates that the execution of the conversion process is without user intervention. If specified it is possible to execute the process in background. Data Converter stores the execution results in a **Log File** and sets the command **Return Code.**

To prepare for executing a conversion in batch mode, create a Conversion Information File.

{button ,JI(`',`MDHELP_BATCH_EXEC')} Running Data Converter in Batch Mode

{button ,JI(`',`MDHELP_BATCH_LOGFILE')} Examining the Log File

Running Data Converter in Batch Mode

Execution Commands

You can execute the conversion with or without the display of a progress indicator. You specify which mode you want through a command line option. Data Converter displays an error message box when there is an error in command line options.

Executing with a Progress Indicator - Command Format

Dataconv.exe /x conversion-information-file-name

The conversion information file is the file created by Data Converter. It is stored with a .CIF extension.

You can interrupt the execution by clicking on the Abort button in the progress indicator window.

Executing without a Progress Indicator - Command Format

Dataconv.exe /b conversion-information-file-name

The process cannot be aborted.

Return Codes

Data Converter returns one of the following values at the end of the conversion:

Return Code	Explanation
0	Normal End (without conversion errors)
1	Normal End (with conversion errors)
2	Abnormal End due to Log File access error
3	Abnormal End (check Log File for details)

Note: A return code is not assured if the process is terminated forcefully.

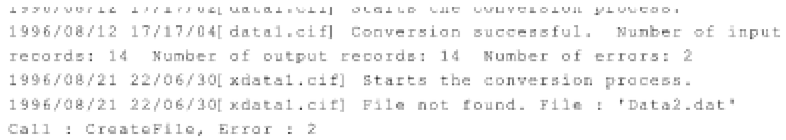
Examining the Log File

The **Log File** is a text file created by Data Converter to record the results of executing in batch mode. It is written to the file name and folder specified in Environment Setup.

If a log file of the specified name already exists the new log entries are appended to it.   If it does not exist a new file is created.

Entries are written to the log file for both normal and abnormal termination.

Example of a Log File   (DataConv.log)



```
1996/08/12 17/17/02[data1.cif] Starts the conversion process.
1996/08/12 17/17/04[data1.cif] Conversion successful.  Number of input
records: 14  Number of output records: 14  Number of errors: 2
1996/08/21 22/06/30[xdata1.cif] Starts the conversion process.
1996/08/21 22/06/30[xdata1.cif] File not found. File : 'Data2.dat'
Call : CreateFile, Error : 2
```

Data Converter

Introduction

Data Converter is a tool for converting data files from one format to another. You define the input and output file formats and Data Converter performs the task of moving the data between the formats. Data Converter handles the following types of conversion:

Converting between different file organizations:

- Sequential,
- Indexed,
- Relative,
- Text file (line sequential),
- Binary,
- Comma-Separated Values (CSV).

Converting between different character code systems:

- ASCII
- ECBDIC

Converting between files with different record structures (e.g. with fields added or removed)

Main Uses

Data Converter provides great flexibility in converting files so it can be used in several different situations. Here are some of the main uses:

- Helping move files between platforms - mainframes, workstations and PCs by converting the character codes and, if necessary, file formats.
- Being an essential part of an enhancement program in which existing files need to be converted into new formats with new fields.
- Transferring data between COBOL data files and spreadsheets or relational databases by using CSV format.
- Setting up test files from a base set of test data.

Data Converter Help

This help system provides overall guidelines on how to convert your files and detailed descriptions of each of the constituent parts of Data Converter. Start with the topic on the conversion process, then consult the detailed descriptions as and when needed.

Main Features

Data Converter has the following features:

Code Conversion

Data Converter supports the ASCII and EBCDIC code systems. Other codes can be handled by using the User Option.

Data Conversion

Data Converter understands all COBOL data types such as packed decimal and binary data. It supports converting individual fields between different data types. Items are converted according to the descriptions setup in the Layout Definition Function.

File Organization Conversion

Data Converter supports the conversion between COBOL85 Files, binary files, text files, and CSV format files.

Layout Definition

The Layout Definition utility makes it easy to setup the input and output file structures by analyzing COBOL COPY libraries and displaying the field by field conversions that will be performed.

Conversion Wizard

The Conversion Wizard helps you specify all the information required to perform a file conversion by constructing the information step-by-step.

Preview

A preview function lets you check that the data conversion has been successful.

Error Log Review

During the conversion process Data Converter creates an error log that you can view from within the Data Converter tool.

Batch Mode Working

In many situations you need to perform file conversions regularly. Data Converter provides a batch mode interface so the conversions are performed without requiring your input.

Integration with Data Editor

Data Converter is integrated with Data Editor so that data can be passed quickly and conveniently from Data Editor to Data Converter. Because Data Editor already has all the record information you only need to specify information in the last two steps of the Conversion Wizard.

The Conversion Process

Follow these steps to convert files using Data Converter.

1. **Prepare input file for conversion.**

   If you want to convert files created on other machines such as UNIX workstations or mainframes you must first transfer them to the PC environment.

   Because most file transfer programs do not understand COBOL file organizations or field types it is best to transfer the files as binary files to ensure no character conversion is attempted.

2. **Run the Conversion Wizard to setup the conversion information.**

   Data Converter uses a set of information which it stores in a conversion information file (or conversion file for short). You enter and edit this information by running the Conversion Wizard. The key information required is:

   Select the **conversion mode** from Data File Conversion or Text Conversion.

   For data file conversion use the Layout Definition Utility to define the **input and output file layouts** and create a layout definition file. The Layout Definition Tool analyzes a COBOL COPY library to give the initial record layouts.

   Specify the **input file name and organization** details.

   Specify the **output file name and organization** details.

   Confirm the **error log name**, **options** and, if appropriate, a **user-defined conversion table**.

   When you finish the Conversion Wizard process you return to the Data Converter main window with all your details displayed.

3. **Save the conversion information.**

   It is best to save the conversion information in a Conversion Information File. This ensures that the information is available for future executions of the file conversion.

4. **Use Preview to check that the conversion is setup correctly.**

   Data Editor provides a Preview Function that displays the results of converting the first 100 records. Thus you can check that all the conversion details are correct before doing a full conversion run.

5. **Execute the conversion.**

   Click the Convert button to execute the conversion.

6. **Check for errors.**

   Data Converter reports the number of records converted and the number of errors in the conversion. If there are errors click on the Display Errors button to view the Conversion Error File.

7. **Transfer the output file to the target environment.**

   If the output file is targeted at UNIX or mainframe machine use your file transfer software to transfer the file to the target environment.

   As with the input file, it is best to transfer the files as binary files to ensure no character conversion is attempted.

Linking from Data Editor

You can invoke Data Converter from Data Editor. This is useful if you want to edit a file in one format but use it in another format.

Actions within Data Editor

1. Open the data file.
2. Apply any required edits (but note that deleted items are still passed to Data Converter).
3. Use the Select Items function if you only want to use data from selected fields.
4. From the Options menu select Link to Data Converter.
5. Select the output data format and choice of all or selected items in the Link to Data Converter dialog box.
6. Click on Start Converter.

Data Converter creates the necessary files such as the Conversion Information File, Layout Definition File and input file as temporary files in the Data Editor workfile folder.

Actions within Data Converter

You enter Data Converter at Step 4 of the Conversion Wizard.

1. Confirm the output file name and code system.
2. Select the desired output file format, and enter or adjust other information as appropriate.
3. Click Next to proceed to Step 5 and confirm or change details as required.
4. Click Finish. Data Converter displays the conversion information in its main window.
5. Click the Convert button to convert the data.
6. Note the name of the output file (format is LINnnnn.OUT, where nnnn is an arbitrary string of characters).
7. Close Data Converter. Data Converter deletes the temporary files containing the input data, conversion information and layout definition.

Precautions

- When Data Converter is invoked from Data Editor the File menu functions are not available.
- The information file and layout file are deleted when Data Converter is closed. To re-execute the conversion you need to start afresh from Data Editor.

Trademarks

Microsoft, Windows, MS-DOS, and MS are registered trademarks of Microsoft Corporation.
UNIX is the registered trademark of X/Open Company, Ltd for USA.
Intel is a trademark of Intel Corporation.