

# FUJITSU

## Data Converter Online Manual

SIMPLE DEVELOPMENT & MAINTENANCE SUPPORT PROGRAM LIBRARIES FOR APPLICATION SYSTEM



**FUJITSU**



August 2015

---

## 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 various file formats, for example between a COBOL data file and a CSV (Comma Separated Values) format file.

It can also convert between different character code systems, such as between ASCII and EBCDIC, or ASCII and Unicode.

## To Use Help

- Please use a browser that supports HTML3.2 (Netscape Navigator V4.03 or newer, or Internet Explorer V3.02 or newer).
- For simplicity, this online manual refers to either NetCOBOL or COBOL generically as "COBOL". The "Data Converter - Readme" topic defines the relationship between "NetCOBOL" or "COBOL" and Data Converter.

## Export Regulation

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

## Trademarks

The registered trademarks used within this online manual are as follows.

- NetCOBOL is a trademark or registered trademark of Fujitsu Limited or its subsidiaries in the U.S.A. and other countries or in both.
- Microsoft, Windows, Windows Server, Visual C++, and Visual Basic are the registered trademarks of Microsoft Corporation in the U.S.A. and other countries.
- UNIX is a registered trademark in the U.S.A. and other countries, licensed exclusively through X/Open Company Limited.
- Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners. Oracle Solaris might be described as Solaris, Solaris Operating System, or Solaris OS.
- Other company names and product names are the trademarks or registered trademarks of the respective companies.

## Abbreviations

In this online manual, the following product abbreviations are used:

Microsoft Windows - All releases	→	"Windows"
----------------------------------	---	-----------

## Main Features

Function	Outline
<b>Code Conversion</b>	Data Converter supports ASCII, EBCDIC, and Unicode systems. Other codes can be processed with the User Option.
<b>Data Conversion</b>	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 set up in the <a href="#">Invoking the Layout Definition Utility</a> .
<b>File Organization Conversion</b>	Data Converter supports the conversion between XML files, <a href="#">COBOL Files</a> , binary files, text files, and CSV format files.
<b>Preview</b>	A preview function allows you to verify that the data conversion has been successful.
<b>Integration with Data Editor</b>	Data Converter is integrated with Data Editor so that data can be passed quickly and conveniently from Data Editor to Data Converter. Since Data Editor already has all of the record information, you only need to specify information in the last two steps of the Conversion Wizard.
<b>I/O of variable-length COBOL file</b>	Data Converter supports I/O of variable-length COBOL files.
<a href="#">Multi Layout</a>	Data Converter can handle multi format files.

The following are typical applications of Data Converter.

- Data transfer utility for platform migration as in downsizing etc.
- Analysis and confirmation of files from other platforms
- Data exchange between various DBMS and spreadsheet software through CSV format
- Data exchange between 16 bit and 32 bit platforms
- Data exchange between platforms with different bitwise data alignment
- Transferring data between COBOL data files and XML document files

## File Menu

[File](#) [View](#) [Tools](#) [Options](#) [Help](#)

---

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 their source directory and dropping them into Data Converter.  Note: 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 of your changes, Data Converter prompts whether you want to save those changes.

## View Menu

[File](#) [View](#) [Tools](#) [Options](#) [Help](#)

---

<b>Menu Function</b>	<b>Explanation</b>
Toolbar	Switches display of the toolbar on and off.
Status Bar	Switches display of the status bar on and off.

## Tools Menu

[File](#) [View](#) [Tools](#) [Options](#) [Help](#)

---

<b>Menu Function</b>	<b>Explanation</b>
Layout Definition	Starts the Layout Definition Utility. → <a href="#">Invoking the Layout Definition Utility</a>

## Options Menu

[File](#) [View](#) [Tools](#) [Options](#) [Help](#)

---

### Menu Function

### Explanation

---

[Preferences](#)

Invokes the Data Converter **Preferences** dialog.



## Help Menu

[File](#) [View](#) [Tools](#) [Options](#) [Help](#)








---

<b>Menu Function</b>	<b>Explanation</b>
Data Converter Help	Displays the online manual.
About Data Converter	Displays the Data Converter version information.

## Toolbar



## Button Menu

-  File-New Setup
-  File-Open
-  File-Save
-  File-Save As
-  Tools-Layout Definition
-  Options-Preferences
-  Help-Data Converter Help

## Layout Definition Utility - File Menu

[File](#) [Edit](#) [Layout](#) [View](#) [Options](#) [Help](#)

Menu Function	Explanation	See Also
New from COBOL library	Utility automatically generates the item definition from a COBOL COPY Library.	→ <a href="#">Analyzing COBOL Library</a>
New	Clears the display of the current layout definition in preparation for creating 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 to a new file. Checks the layout definition for correctness and asks you whether or not to save the file if it has errors.	
Save in Old Version Format	Saves the layout definition file in a format that can be used by the previous version.	
Generate COBOL Library	Creates a COBOL library containing information from the layout definition. Select an object (input side / output side) to create from <a href="#">Generate COBOL library</a> .	→ <a href="#">An Example of Generating a COBOL Library</a>
Generate DTD	Generate DTD if the layout definition being edited is in XML format.	→ <a href="#">An Example of generating DTD</a>
Replace Layout from Existing Layout	Overwrites information on an existing layout definition file.	
Replace Layout from COBOL file	Overwrites information on an existing COBOL file.	
Exit	Closes the Layout Definition Utility.	

## Layout Definition Utility - Edit Menu

[File](#) [Edit](#) [Layout](#) [View](#) [Options](#) [Help](#)

---

<b>Menu Function</b>	<b>Explanation</b>
Cut	Cuts the selected line(s) to the copy buffer.
Copy	Copies the selected line(s) to the copy buffer.
Paste	Inserts the contents of the copy buffer in front of 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.
Insert new layout	Inserts a new layout before the current layout.
Add new layout	Adds a new layout after the current layout.
Edit layout condition	Displays the Layout Definition Template.
Delete layout	Deletes the current layout.

## Layout Definition Utility - Layout Menu

[File](#) [Edit](#) [Layout](#) [View](#) [Options](#) [Help](#)

---

<b>Menu Function</b>	<b>Explanation</b>
<a href="#">Item Definition</a>	Displays the Item Definition Template for editing the information in the current line.
<a href="#">Special File Encoding</a>	Sets treatment of Null Area, Null string terminators, and big endian behavior.
<a href="#">Data Format</a>	Sets and changes the data format of the input and output.
<a href="#">Operations on All Items</a>	Provides a list of functions to apply to all of the items.
<a href="#">COBOL Library Analysis Option</a>	Sets various options for analyzing COBOL libraries.

---

**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 Definition Extension information.

→ [Understanding Item Definition Information](#)

[File](#) [Edit](#) [Layout](#) [View](#) [Options](#) [Help](#)

---

**Menu Function Explanation**

---

[Preferences](#) Displays the Preferences option selection screen of the layout definition function.

## Layout Definition Utility - Help Menu

[File](#) [Edit](#) [Layout](#) [View](#) [Options](#) [Help](#)


















---

<b>Menu Function</b>	<b>Explanation</b>
Data Converter Help	Displays the online manual.
About Data Converter	Displays the version information.





**Button Menu**

-  File-New from COBOL library
-  File-New
-  File-Open
-  File-Save
-  File-Save As
-  Edit-Cut
-  Edit-Copy
-  Edit-Paste
-  Edit-Insert
-  Edit-Delete
-  Edit-Append
-  Edit-Insert new layout
-  Edit-Add new layout
-  Edit-Edit layout condition
-  Edit-Delete layout
-  Options-Preferences
-  Help-Data Converter Help

## Data Converter Preview function - File Menu

[File](#) [Options](#) [Help](#)

---

The **File** menu has a single function - **Exit**. It returns you to the Data Converter main window or will exit from the Layout Definition tool.

## Data Converter Preview function - Option Menu

[File](#) [Options](#) [Help](#)

---

<b>Menu Function</b>	<b>Explanation</b>	
HEX Display	Displays the data in hexadecimal underneath the ASCII characters.	
Column Grid Lines	Controls the spacing 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.	<a href="#">→Displaying Conversion Errors</a>

[File](#) [Options](#) [Help](#)

---

The Help function invokes the Preview help.

## Checking Conversion Using the Provided Sample Data

The following describes the procedure for using the samples provide with this product. Use these samples to check the operation of Data Converter after installation, etc. Samples include data values not matching item attributes to trigger the conversion error display function.

### 1. Activating Data Converter



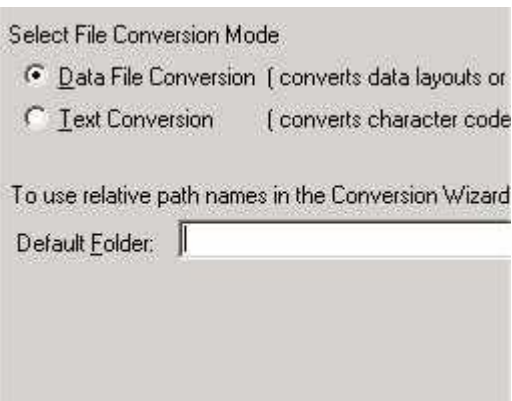
From the **Start** menu (or Program manager), click the **Data Converter** icon to start Data Converter.

### 2. Activating the Conversion Information Wizard



Select **New Setup** from the **File** menu or from the toolbar or select Conversion Wizard button to activate the Conversion Wizard.

### 3. Conversion Information Wizard(1/5)



Specify the file conversion mode and the default folder.

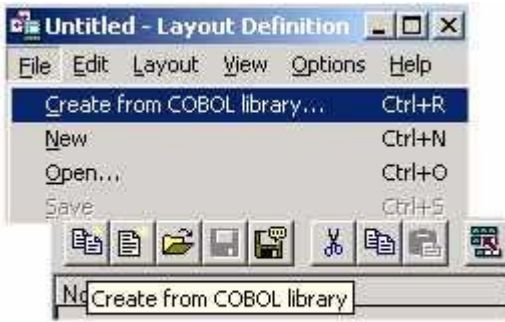
Select **Data File Conversion** as the file conversion mode and specify the folder that stores sample data (DATA1.DAT) as the default folder.

### 4. Conversion Information Wizard(2/5)



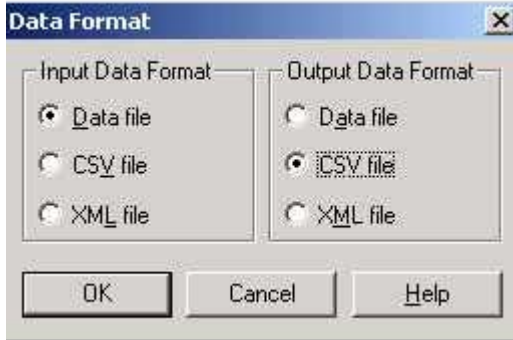
Specify a layout definition file. To create a new layout definition file, select **Update** to activate the Layout Definition Program.

### 5. Layout Definition Function (Create from COBOL library)



Click on the **Update** button to create a new Layout Definition File. Select **Create from COBOL library** from the File menu to create it from COBOL library(DATA1.CBL)

## 6. Layout Definition Function (Specifying a Data Format)



When the analysis of COBOL library is complete, the **Data Format** dialog is displayed.

Specify the input and output data formats in this dialog.

Select **Data file** as the **Input Data Format** and **CSV file** as the **Output Data Format** and click **OK**.

## 7. Layout Definition Function (Checking, Saving, and Terminating a Layout Definition)

Item No.	Item Name	Attribute
1	ORDERNO	X
2	PRODUCTNO	X
3	PRODUCTNAME	X
4	PRICE	Z
5	SALESNUMBER	Z
6	SALES	Z
7	SALESDATE	Z
8	CUSTOMERNAME	X
9	CUSTCODE	X

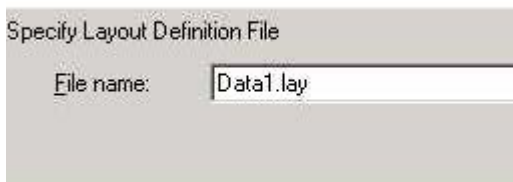
The result of the analysis is displayed in the list.

Select **Save** from the **File** Menu or toolbar, or **Save As** and enter a new name. (The extension is ".LAY").

The sample layout definition file (DATA1.LAY) is analyzed and saved in this manner.

After saving the data, terminate the layout definition function and return to the Data Converter's Conversion Information Wizard (Step 2 of 5).

## 8. Conversion Information Wizard(2/5 After making the layout definition file)



Specify the file name saved by the layout definition function for the layout definition file name.

## 9. Conversion Information Wizard(3/5)

Specify input information.

Specify **ASCII** as the code system, the sample data (DATA1.DAT) as the file name, and **Binary or Text File** as the file format.

Specify **Fixed Length** as the record attribute and **116** as the record length.

#### 10. Conversion Information Wizard(4/5)

Specify output information.

Specify **ASCII** as the code system, any file name as the file name (DATA1.CSV by default), **Variable length** as the record attribute, and **New line code** (CR+LF) as the delimiter of records.

Additionally, click on **Setup** to bring up the **CSV Information Setup** dialog and set up the CSV format for output.

#### 11. Specifying CSV Information

Specify CSV information settings.

From the reference settings, select **Oracle**. The general loader format of Oracle will be selected.

Click OK and go to Conversion Information Wizard Step 5 of 5.

#### 12. Conversion Information Wizard(5/5)

Specify other information on conversion.

This information can be changed as required. It is recommended that the default settings be used for a sample.

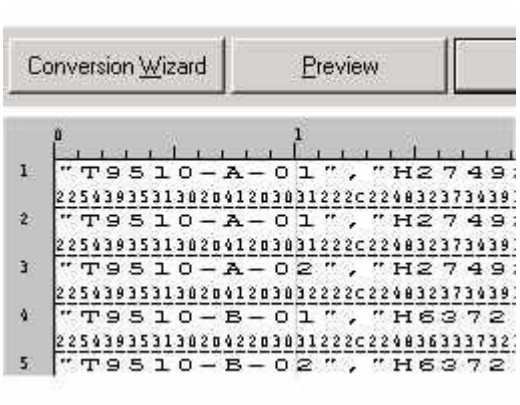
Click **Finish** to complete the Conversion Information Wizard.

#### 13. Preview

Click the **Preview** button to bring up the preview screen.

Check that the data is in CSV format.

Select **Exit** from the **File** menu to close the Preview screen.



#### 14. Executing Conversion



Click the **Convert** button to start the actual conversion processing.

When the conversion processing is complete, a dialog will display the input and output record counts and the error occurrence count.

The results displayed for the sample data are as follows :

Number of input records: 16  
 Number of output records: 16  
 Number of errors: 0

#### 15. Saving a Conversion Information File



Select **Save As** from the toolbar to save the conversion file (with the extension of "CIF").

#### 16. Terminating Data Converter



Select **Exit** from the **File** menu to terminate Data Converter.



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. Since 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 set up the conversion information.

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

Conversion mode: Select either [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 and options, and, if appropriate, a user-defined conversion table.

When you finish the Conversion Wizard process, you return to the Data Converter main window where all of the details are 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 Converter provides a preview function that displays the results of the conversion of the first 100 records. You can verify that all of the conversion details are correct before executing 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 Error** button to view the [Conversion Error File](#).

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

If the output file is targeted at a UNIX or mainframe machine, use appropriate 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.

Data Converter is used first to set up the conversion information and then to save it in a conversion information file. You → [Conversion Information file](#) then can then run conversions and check conversion results. Use the links below to explore the various functions of Data Converter. The buttons at the base of the main window take you through the main sequence you need to follow to convert files: invoking the Conversion Wizard, previewing the conversion, running the conversion, and reviewing any errors.

- [Common Information \(Step1\)](#)
- [Layout Definition File Information \(Step2\)](#)
- [Input/Output File Information \(Step3 & 4\)](#)
- [Other Information \(Step5\)](#)

## Executing the Conversion

---

Click on the **Convert** button to start the conversion. Data Converter will display a progress indicator containing:

- the name of the input file
- the number of bytes converted
- the total number of bytes in the file
- the number of conversion errors
- the percentage of progress

You may interrupt the conversion process by clicking the **Abort** button. Aborting the conversion will discard the contents of the partially created output file.

At the end of the conversion process, Data Converter will display the number of input records, output records, and conversion errors.

If the following message appears during conversion, check the conditional value in the data condition setting. For details, see [Record Extraction Specifications](#).

For notes, see [Notes on Record Extraction Function](#).

"Code conversion for an extraction condition failed.  
The code of the n-th extraction condition has not been converted.  
Normal extraction may not be assured with this condition".

n: Data condition serial number

Conversion accompanying record extraction is explained below.

If you have specified data conditions on Step 2 of the Data Converter Conversion Wizard, record extraction is performed. The processing subsequent to record extraction is the same as for normal conversion.

When conversion accompanying record extraction is completed, information on the conversion is output, including the number of input records, the number of output records, and the number of conversion errors.

Number of input records	Number of records in the input file.
Number of output records	Number of records that satisfy the data conditions.
Number of errors	Number of conversion errors in the records extracted through record extraction.

## Creating User Defined Conversion Tables

Data Converter supports ASCII, EBCDIC code, and Unicode character sets. Modifications or additions to these character sets can be made through user-defined conversion tables. Character code conversions can be performed using custom-designed or non-standard code sets by creating your own conversion tables. If desired, all single byte code sets can be changed by defining 256 code conversion patterns in the user-defined conversion table. Caution: if this is done, the user must specify the **ASCII**, **EBCD**, or **UCS2** portion of the code system definition.

### Format for User Defined Conversion Tables

→ [Conversion Wizard Step 5](#)

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 the ID 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**

- # is specified in column 1.
- Code system name can be **ASCII**, **EBCD**, or **UCS2**.

ASCII	ASCII Code
EBCD	EBCDIC Code
UCS2	UCS2 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**

- 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 ASCII  
00:5F  
5B:NO  
* Conversion from ASCII to in-house code set by substituting  
* all byte mappings in character set "EBCDIC"
```

# ASCII EBCD

00:00

...

20:40

21:44

...

\* (Defines mappings for all byte values)

## Previewing Conversion Results

---

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, and 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. The display window also provides the option of viewing the data in hexadecimal.

## Displaying Conversion Errors

Data Converter stores errors encountered during conversion in a conversion error file. The conversion error display function displays errors using an external viewer program (the default is a Notepad) specified by an option.

→ [Conversion Error File](#)  
→ [Preferences](#)

### Example of a Conversion Error File (SAMPLE.ERR)

```
*** Data Converter conversion error file ***
Current Folder : V:\HOME\MDPORT32\TEST
Input File : DATA1.DAT
Layout definition file : 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; "*" 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
```

[Code Conversion Error]	List of character codes that caused code conversion errors. For Unicode, UCS2 Big Endian is displayed regardless of type (UCS2/UTF8) or Endian. <b>Format</b> Hexadecimal-character-code (number-of-occurrences)
[Error Items]	List of items that contained conversion errors (data file conversion only) <b>Format</b> 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. For Unicode, displays Type (UCS2/UTF8) and Endian like the input code. <b>Format</b> Input-record-number : Position-within-the-record Error-contents Item-No.-in-Layout-definition : Item-name : Item-attribute

## Using Data Converter - Displaying Conversion Errors - Examples of Typical Conversion Errors

Typical conversion errors and their causes are explained below.

### Area overflow error - usually because of area length changes

After data file conversion involving code conversion of alphanumeric character items, Japanese items, mixed items, variable-length character string items, or variable-length national items, the input data length and output data length of the same item may not match.

Cause of area length changes	Action
1 Conversion from ASCII/EBCDIC to Unicode (UCS2 format) was performed. →In the UCS2 format, all characters are represented by 2-byte codes.	
2 Conversion from the CSV/XML format to the data format was performed. →In the CSV/XML format, the length of each input item is variable. Therefore, the data length after conversion may not match the output area length. If the data length after conversion is shorter than the output area length, the non-data area is padded with null characters (or one-byte spaces for some item attributes). If the data length after conversion is longer than the output area length, extra characters are removed and an error is output.	Use the layout definition function to increase the area length of each output item.

### Code conversion error - code conversion error/incorrect conversion

Cause of code conversion errors	Action
3 Attempts to convert LOW-VALUE (0x00) or HIGH-VALUE (0xFF) or extended ASCII code result in a conversion error. →Because Data Converter does not recognize LOW / HIGH-VALUES as characters, a conversion error occurs and the LOW / HIGH-VALUES are replaced with specific alternate characters.	Create a mapping table in a user-defined file. Example: ----- # ASCII ASCII 00:00 80:80 81:81 FF:FF 0000:0000 -----



## Errors in numeric part - causes of errors in numeric part

### Causes of errors in numeric part

- Non-numeric data exists in an external/internal decimal data area.

Attribute	Sign	COPY clause	Internal representation (hex.)	Value	Error
External decimal	No	9(4)	31323334	1234	No
External decimal	Yes	S9(4)	31323344	+1234	No
External decimal	No	9(4)	3132333A	0	Yes
Internal decimal	No	9(4) COMP3	01234F	1234	No
Internal decimal	Yes	S9(4) COMP3	01234D	-1234	No
Internal decimal	No	9(4) COMP3	012B4F	0	Yes

- A non-numeric character or symbol exists in a CSV/XML-formatted (numeric) data area.

Attribute	Sign	COPY clause	Value	Error
External decimal	No	9(4)	ABCD	Yes
External decimal	Yes	S9(4)	\$1234	Yes

## Errors in sign part - causes of errors in sign part

When data file conversion involves conversion of external or internal decimal items, the sign part is strictly checked. Before the sign part can be checked, you must enable the Perform Input Check option when defining layout definition items.

Attribute	Sign	COPY clause	Internal representation	Value	Error
External decimal	No	9(4)	31323334	1234	Any digit that is not 3 is erroneous.
External decimal	Yes	S9(4)	31323354	-1234	Any digit that is not 4 or 5 is erroneous.
External decimal	Yes	S9(4)	31323344	1234	
Internal decimal	No	9(4) COMP3	01234F	1234	Any digit that is not F is erroneous.
Internal decimal	Yes	S9(4) COMP3	01234C	+1234	Any digit that is not C or D is erroneous.
Internal decimal	Yes	S9(4) COMP3	01234D	-1234	

Note: The above external decimal examples assume the ASCII code system. The operational signs for the EBCDIC code system are as follows:

F: Zone bit
C: Positive operational sign
D: Negative operational sign

The record extraction function accepts only the records that satisfy the criteria defined for input file data items.

To open the Data Condition Setting screen:

1. In Step 2 of the Conversion Wizard, specify the name of the layout definition file.
2. In Step 2 of the Conversion Wizard, click the Setting button. The Data Condition Setting screen is displayed.

→ [Notes](#)

Using the Layout Definition Utility, you can create and edit the layout definition file used by the data file conversion.

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

- Select **Layout Definition** from the **Start > All Programs** menu.
- Start from the Data Converter conversion instruction.
- Open the LAYOUT.EXE application from Windows Explorer.

Although you can enter the whole layout definition yourself, it is generally better to start by having the Layout Definition Utility analyze an existing 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.

→ [Layout Definition File](#)  
→ [COBOL library](#)  
→ [Precautions - Analyzing a COBOL Library](#)

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.

In the Layout Definition Utility, you can specify the data format for the input and the output files. There are three options:

- [Data file](#)
- [CSV file](#)
- [XML file](#)

### **Precautions when Changing the Data Format**

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

### 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 in 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 click the **OK** button, and informs you of any errors so you can immediately correct them.

Invoke the item definition template as the template initializes the advanced settings fields.

- [Understanding Item Definition Information](#)
- [Preferences](#)
- [Item Definition Template](#)
- [Advanced Settings](#)
- [View Menu](#)

## Layout Definition Utility - Defining Repeating Items

The Layout Definition Utility uses the repetition level (**Lv**) column and the repetitions (**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 indicates 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) Group Item option is "Hidden".** [See COBOL Library Analysis Options](#)

### Data description in 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

Hierarchy	Number of 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

Specification of a group item takes the same form as a description of a repeated item, without the specification of the number of repetitions.

The items subordinate to a group item are identified by nesting the subordinate class below the group item definition line. A group item definition line must consist of only a class and an item name, and the following line must have a subordinate class. The maximum nesting depth of subordinate classes is limited to seven (7).

In order to display a group item created from an existing COBOL property on a layout definition screen, the group item must be set up as a "[COBOL Library Analysis Option](#)". Regardless of the setup, editing the group item (adding, deleting, etc.) is possible in the layout definition.

### Example:

#### Data Description in COBOL

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

#### Layout Definition

Level	Occurs	Item name
0		ITEM1
0		GROUP1
1		ITEM2
1		ITEM3
0		GROUP2
1		ITEM4
1		ITEM5
1		GROUP2-1
2		ITEM6
1		ITEM7



## Multi Layout Definition Function

In this manual, "multi format file" means that the file has various record formats; each record in a multi format file has its own format. In this section, you will see how to operate a multi format file using the Multi Layout Definition Function.

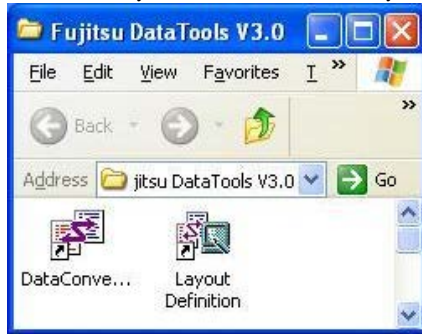
By specifying the extraction condition in the Multi Layout Definition Function, you can select specific records to which a created layout definition is applied. Using this function, you can also extract records from an input file.

### The procedure to make a Multi Layout Definition File

To make a multi layout definition file:

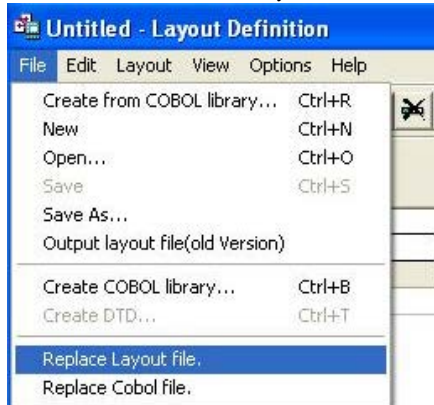
#### 1. Start the Layout Definition application

Click on the Layout Definition icon to start Layout Definition from the **Start** menu or Program Manager.



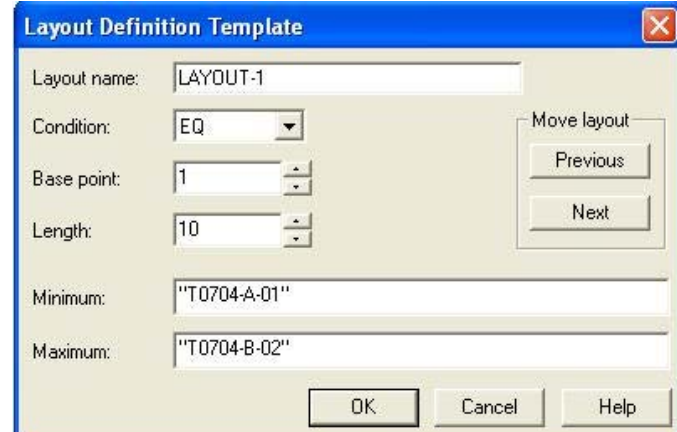
#### 2. Create a new Layout Definition File

Create a new layout definition file by reading an existing layout file or analyzing an existing COBOL library. After reading or analyzing the existing file, Data Converter creates all of the item definitions in the new layout definition file.



#### 3. Edit the multi layout

Specify the comparison information in the Layout Definition Template dialog box.



#### 4. Add a new layout

Click on the **Add new layout** button to add a new layout after the last layout.

Untitled - Layout Definition

File Edit Layout View Options Help

Layout: LAYOUT-2 Equal Pos=1 : Length=10 Comp

No. 1 LAYOUT-2 Item Name:ORDEF

	Lv	Repeat	Item-name	Attribute	Sign
1	0		ORDERNO	X	
2	0		PRODUCTNO	X	
3	0		PRODUCTNAME	X	
4	0		PRICE	Z	
5	0		SALESNUMBER	Z	S
6	0		SALES	Z	S

### 5. Insert a new layout

Click on the **Insert new layout** button to insert a new layout in front of the displaying layout.

Untitled - Layout Definition

File Edit Layout View Options Help

Layout: LAYOUT-9 Equal Pos=1 : Length=10 Comp "T0704-C-01" - "T0704-D-02"

No. 1 LAYOUT-9 Item Name:ORDERNO

	Lv	Repeat	Item-name	Attribute	Sign	Input (Data format)				Output (Data format)					
						Position	Length	Total digits	Decimal	Attribute	Sign	Position	Length	Total digits	Decimal
1	0		ORDERNO	X		1	10			X		1	10		
2	0		PRODUCTNO	X		11	10			X		11	10		
3	0		PRODUCTNAME	X		21	20			X		21	20		
4	0		PRICE	Z		41	5	5		Z		41	5	5	
5	0		SALESNUMBER	Z	S	46	4	4		Z	S	46	4	4	
6	0		SALES	Z	S	50	9	9		Z	S	50	9	9	
7	0		SALESDATE	Z		59	8	8		Z		59	8	8	
8	0		CUSTOMERNAME	X		67	40			X		67	40		
9	0		CUSTCODE	X		107	10			X		107	10		

Ready CAP NUM

### 6. Delete a layout

Click on the **Delete layout** button to delete the displaying layout.

Untitled - Layout Definition

File Edit Layout View Options Help

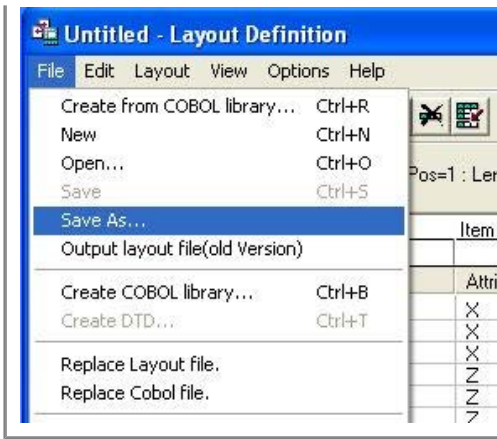
Layout: LAYOUT-9 Equal Pos=1 : Length=10 Comp "T0704-C-01" - "T0704-D-02"

No. 1 LAYOUT-9 Item Name:ORDERNO

	Lv	Repeat	Item-name	Attribute	Sign	Input (Data format)			
						Position	Length	Total digits	Decimal
1	0		ORDERNO	X		1	10		
2	0		PRODUCTNO	X		11	10		
3	0		PRODUCTNAME	X		21	20		
4	0		PRICE	Z		41	5	5	
5	0		SALESNUMBER	Z	S	46	4	4	
6	0		SALES	Z	S	50	9	9	
7	0		SALESDATE	Z		59	8	8	
8	0		CUSTOMERNAME	X		67	40		
9	0		CUSTCODE	X		107	10		

### 7. Save the Multi Layout Definition File

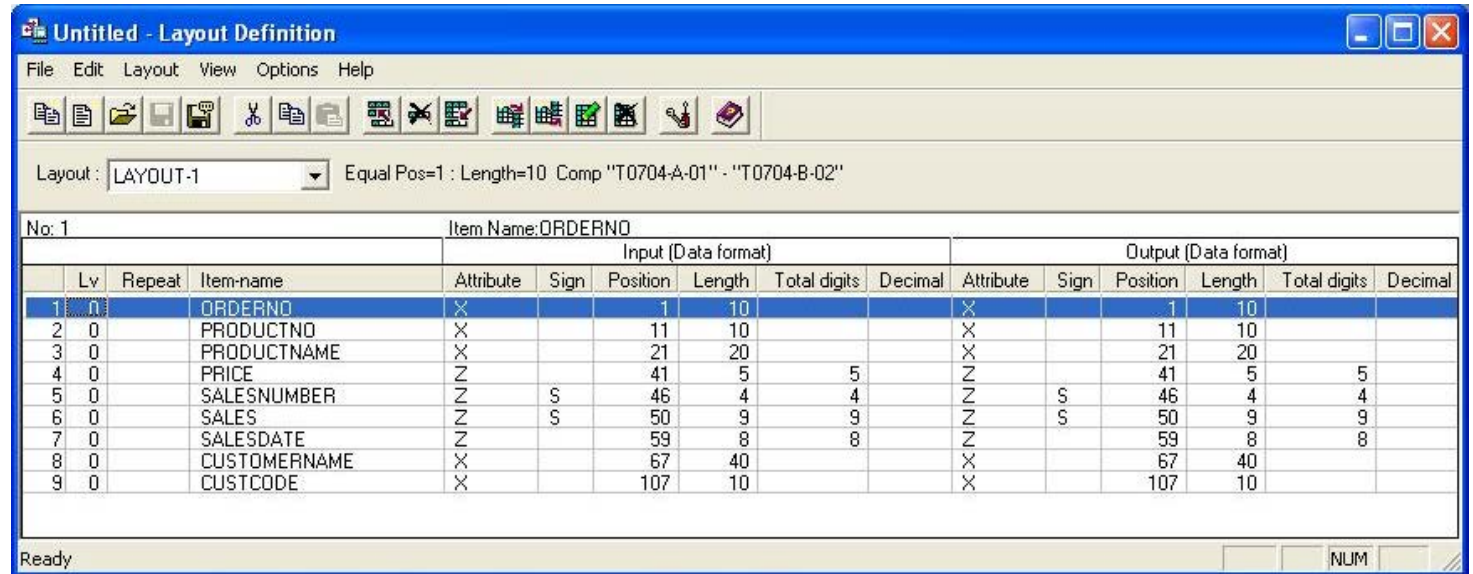
Select **Save as** from the file menu to save the multi layout definition file.



## Layout Definition Dialog Boxes

### 1. Layout Definition Dialog Box

Click on the **Layout Definition** icon to start Layout Definition from the **Start** menu or Program Manager.



### 2. Specifying Layout Definition Dialog Bar

#### 1. List of layout names

The list of layout names in the multi layout definition file is displayed. Select a name from the list to display the layout definition.

#### 2. Control strings of selected layout

The control strings of the selected layout are displayed.

- Format op pos:len comp min-max

op : The operator EQ/NOT/OTHER/NONE is displayed.

pos: The starting position of comparison data is displayed.

len: The length of comparison data is displayed.

min: The minimum value of comparison data is displayed.

max: The maximum value of comparison data is displayed.

Layout : LAYOUT-1 Equal Pos=1 : Length=10 Comp "T0704-A-01" - "T0704-B-02"

### 3. Layout Definition Template Dialog Box

#### 1. Layout Name

Specify the layout name, up to 90 bytes. This layout name must be different from the existing layout name that is specified in the **Basic Information** dialog box.

#### 2. Condition

Specify the conditional expression. This field is required. EQ means Equal, NOT means Not Equal, OTHER means Other, and NONE means None.

#### 3. Offset

Specify the offset in records from which comparison of data starts. This field is required. The minimum is 1 and the maximum is 32767.

#### 4. Length

Specify the comparison data length in bytes. This field is required. The minimum is 1 and the maximum is "32767 - base position".

#### 5. Minimum

Specify the minimum value that is to be compared to the record data. Specify a hexadecimal value or a character string enclosed in double quotations. When the specified

value length is shorter than the comparison data length specified above, Data Converter inserts NULL in the part of the comparison data that exceeds the minimum value length. When the specified value length is longer than the comparison data length, Data Converter truncates the part of the minimum data that exceeds the comparison data length.

If you specify a minimum value and you do not specify a maximum value, Data Converter compares only the minimum value with the input file records.

#### 6. Maximum

Specify the maximum value that is to be compared to the record data. Specify a hexadecimal value or a character string enclosed in double quotations. When the specified value length is shorter than the comparison data length specified above, Data Converter inserts NULL in the part of comparison data that exceeds the minimum value length. When the specified value length is longer than the comparison data length, Data Converter truncates the part of the maximum data that exceeds the comparison data length. If you specify a maximum value and you do not specify a minimum value, Data Converter compares only the maximum value with the input file records.

#### 7. Previous

Click on the **Previous** button to save the specified conditions and display the previous layout conditions. If the first layout is specified, a warning dialog box will be displayed.

#### 8. Next

Click on the **Next** button to save the specified conditions and display the next layout conditions. If the final layout is specified, a new layout will be created and displayed.

#### 9. OK

Click **OK** to check the specified conditions. If the conditions are valid, they are saved and the window closes. The relationship between Base position/Length and the actual record length is checked when the layout file is saved.

#### 10. Cancel

Click on the **Cancel** button to clear the editing conditions and close the window.

#### 11. Help

Click on the **Help** button to display the online help window.

**Layout Definition Template**

Layout name: LAYOUT-1

Condition: EQ

Base point: 1

Length: 10

Minimum: "T0704-A-01"

Maximum: "T0704-B-02"

Move layout

Previous

Next

OK Cancel Help

You can run Data Converter in batch mode. In batch mode, the conversion process is executed without user intervention. If specified, it is possible to execute the process in the background. Data Converter stores the execution results in a log file and sets the command return code. The execution result can be confirmed according to the return value of the command and the content of the log file event log.

To prepare for executing a conversion in batch mode, create a conversion information file.

- [Running Data Converter in Batch Mode](#)
- [Examining the Log File](#)

Conversion error information is output to the conversion error file. The conversion error display function uses an external viewer program to view the output conversion error file (specified by an option: default is Notepad).

→ [Conversion Information File](#)

→ [Displaying Conversion Errors](#)

→ [Preferences](#)

## Running Data Converter in Batch Mode

Use the following command to execute conversion in batch mode. A message box will indicate an error if an option is not correctly specified.

**Command Line Format** (items enclosed in brackets [ ] are optional)

```
Dataconv.exe      /x|b Conversion-Information-File-name(*.cif)
                  [/c Default Folder name]
                  [/i Input File name]
                  [/o Output File name]
                  [/e Conversion Error Output File]
                  [/f]
                  [/v]
```

### Explanation of Parameters

- x** Specify a conversion information file. A progress indicator is displayed during execution. Click the **Stop** button on the progress indicator to stop the processing.
- b** Specify a conversion information file name. If specified, no progress indicator window is displayed.
- c** Specify a default folder name.
- i** Specify an input file name.
- o** Specify an output file name.
- e** Specify a conversion error output file name.
- f** Select forced overwrite mode
- v** Specify when outputting to an event log instead of the log file.

### Notes:

- Either x or b must be specified, but not both.
- Specifying c, i, o, or e will override any corresponding folder/file name specified in the conversion information file.
- Any of the c, i, o, and e parameters can be omitted. In this case, the contents of the conversion information file will be used instead.
- If a wildcard is specified for the i parameter, the character string specified for the o parameter is assumed to be an extension. The file name whose extension is replaced with the above extension is set as the output file name.
- A wildcard can be specified for the i parameter.
- If the file specified by the o parameter already exists, and the f parameter is specified, any existing output file will be overwritten. If the f parameter is not specified, processing will stop.

Note: File names that include blank characters must be enclosed in double quotation marks.

### Return Values

Execution ends with one of the following return values.

Note: A correct return value cannot be guaranteed when a process is abnormally terminated.

- 0** Normal termination without conversion errors
- 1** Normal termination with conversion errors
- 2** Abnormal termination due to log file access errors
- 3** Abnormal termination with description or errors output to log files

## 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 **Preferences**.

→ [Preferences](#)

If a log file of the specified name already exists, then 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)

```
2001/08/12 17/17/02[data1.cif] Starts the conversion process.
2001/08/12 17/17/04[data1.cif] Conversion successful. Number of input records: 14 Number of output records: 14 Number of errors:
2
2001/08/21 22/06/30[xdata1.cif] Starts the conversion process.
2001/08/21 22/06/30[xdata1.cif] File not found. File:'Data2.dat' Call:CreateFile,Error: 2
```

## Linking from Data Editor

You can invoke Data Converter from Data Editor.

### Actions within Data Editor

- 1 Open the data file.
  - 2 Apply any required edits. Note that deleted items are also 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 **Data Converter**.
- >> Data Converter creates the necessary files, such as the converter information file and input file, as temporary files →[Conversion Information File](#)  
in the Data Editor work file folder. →[Layout Definition File](#)

### Actions within Data Converter

- 1 Confirm the output file name and code system. →[Conversion Wizard Step 4](#)
- 2 Select the desired output file format, and enter or adjust other information as appropriate.
- 3 Click **Next** to 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 re-start the process in Data Editor.



## Command Button

Button Name	Explanation	
Conversion Wizard	Sets or updates conversion information via the Conversion Wizard.	→ <a href="#">Understanding Data Converter</a>
Preview	Confirms the conversion result on the screen beforehand. The preview function cannot be used on an output file unless it is a general file.	→ <a href="#">Previewing Conversion Results</a>
Convert	Starts the conversion processing.	
Display Errors	Displays the content of the error files generated by the conversion processing, if a conversion error file exists.	→ <a href="#">Displaying Conversion Results</a>

## Preferences

### Basic Information

Field	Explanation
Work Folder	Specify the name and location of 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. The default is to create the file DATACONV.LOG in the Work Folder.
Program for Displaying Conversion Errors	The external viewer program which displays the conversion error is specified by a full path name. Notepad or other text editors can also be specified. Notepad is used by default.

### COBOL

Option	Explanation
Allow COBOL files to exceed 1GB	Specify this item when you treat a COBOL file (sequential or line sequential files) that exceeds 1 GB.
Lengths specify number of Bytes	Select when using binary byte type in binary items.
Use COBOL specifications on binary files	Specify whether to use COBOL specifications on binary files when the code system is Unicode. <ul style="list-style-type: none"><li>- When this box is checked, alphanumeric and national items will be treated as UCS2.</li><li>- When this box is unchecked, alphanumeric items will be treated as UTF-8, and national items will be treated as UCS2.</li></ul> When this check box is selected, data compatible with V3L40 and earlier versions will be output. This box is checked as the default.

## Conversion Wizard Step 1 - Common Information

### Menu Function

### Explanation

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

→ [Data File Conversion](#)

→ [Text Conversion](#)

Default Folder

If you specify a default folder, you can set up 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.

Menu Function	Explanation
Specify Layout Definition File	Specify a layout definition file name to be used when converting a data file.
Update Layout Definition File	Activate the Layout Definition Utility.
Create new Layout Definition File from library	Specify an existing COBOL library to create a layout definition file.
<a href="#">Select Input Record</a>	Use the <b>Data Condition Setting</b> dialog to select the input records.

**A Layout Definition File can be created in any of the following three ways:**

- Activate the Layout Definition Utility to create a new layout definition file from an existing COBOL library.
- Click the **Analyze** button of the Conversion Wizard to create a new layout definition file from an existing COBOL library.
- Activate the Layout Definition Utility to manually enter all item definitions.

---

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

## Input Record Selection Conditions Dialog

Use this screen to specify data conditions for the records to be extracted from the input file. Only the records that satisfy the specified data conditions will be output.

For record extraction specifications, see [Record Extraction](#) in Extraction Specifications.

For notes, see [Notes on Record Extraction Function](#).

Item	Explanation												
Add	To add an extraction condition, specify Item Name, Operator (hexadecimal form), and Condition value within the conditions group (Condition Connector must also be specified for the second and subsequent conditions), and then click the Add button. The extraction condition is added to the Specified Conditions List. If an error occurs, correct the condition according to the error message.												
Update	To modify an extraction condition, select the condition you want to modify from the Specified Conditions List, specify Item Name, Operator (hexadecimal form), and Condition value within the conditions group (Condition Connector must also be specified for the second and subsequent conditions), and then click the Update button. The extraction condition is updated on the Specified Conditions List. If an error occurs, correct the condition according to the error message.												
Delete	To delete an extraction condition, select the condition you want to delete from the Specified Conditions List, and then click the Delete button. The extraction condition is deleted from the Condition List.												
Condition List	The specified data conditions are listed. Record extraction will be performed based on the conditions listed here. <table border="1" style="margin-left: 20px;"> <tr> <td>No.</td> <td>Indicates the sequence number of the data condition. The specified data conditions are applied in ascending order of the sequence number, from 1 to 10.</td> </tr> <tr> <td>Connected Condition</td> <td>Indicates the concatenator AND or OR for the data condition, for the second and subsequent conditions.</td> </tr> <tr> <td>Item Name</td> <td>Indicates the item name for the data condition.</td> </tr> <tr> <td>Operator</td> <td>Indicates the operator for the data condition.</td> </tr> <tr> <td>Condition Value</td> <td>Indicates the condition value for the data condition.</td> </tr> <tr> <td>Hexadecimal (e.g. F0F5)</td> <td>YES is displayed if Hexadecimal form is selected.</td> </tr> </table>	No.	Indicates the sequence number of the data condition. The specified data conditions are applied in ascending order of the sequence number, from 1 to 10.	Connected Condition	Indicates the concatenator AND or OR for the data condition, for the second and subsequent conditions.	Item Name	Indicates the item name for the data condition.	Operator	Indicates the operator for the data condition.	Condition Value	Indicates the condition value for the data condition.	Hexadecimal (e.g. F0F5)	YES is displayed if Hexadecimal form is selected.
No.	Indicates the sequence number of the data condition. The specified data conditions are applied in ascending order of the sequence number, from 1 to 10.												
Connected Condition	Indicates the concatenator AND or OR for the data condition, for the second and subsequent conditions.												
Item Name	Indicates the item name for the data condition.												
Operator	Indicates the operator for the data condition.												
Condition Value	Indicates the condition value for the data condition.												
Hexadecimal (e.g. F0F5)	YES is displayed if Hexadecimal form is selected.												
Condition Connector	If you specify two or more data conditions, select AND or OR as the concatenator between the first condition and the subsequent condition. No initial value is set.												
Item Name	Specify the item name in the layout definition file that you have specified in Step 2 of the Conversion Wizard. You cannot specify group items or variable-occurrence items.												
Operator	Specify the operator to be used to compare the data in the input file with the condition value. The specifiable operators vary depending on the data attribute: <table border="1" style="margin-left: 20px;"> <tr> <td>Character type attribute</td> <td>String Equals, Prefix Match, !=</td> </tr> <tr> <td>Numeric type attribute</td> <td>=, !=, &gt;, &gt;=, &lt;, &lt;=</td> </tr> </table>	Character type attribute	String Equals, Prefix Match, !=	Numeric type attribute	=, !=, >, >=, <, <=								
Character type attribute	String Equals, Prefix Match, !=												
Numeric type attribute	=, !=, >, >=, <, <=												
Hexadecimal (e.g. F0F5)	Check this check box if you want to specify the condition value in hexadecimal. This checkbox is available only for the character type attribute.												
Condition Value	Specify the condition value to be used to extract data records from the input file.  The maximum length of condition value specifications is 150 bytes. For the numeric type attribute, a number consisting of up to 18 digits (+, -, and . (decimal point) are not included) can be specified. If no condition value is entered, Data Converter considers that NULL data is specified. You can only omit the condition value when the data attribute is the character string item (M). *Null condition value can be used for NULL data extraction for CSV/XML-formatted character string items.  To specify a condition value when Hexadecimal form is selected: Available characters are single-byte alphanumeric characters 0 to 9, A to F, and a to f, and semicolons (;). Semicolons must be used after the end of a byte. If a semicolon is inserted between bits of a byte, Data Converter omits 0 (hexadecimal) for the low-order bit.												
Output Unmatched Records	Check this checkbox if you want to output records that do not satisfy the data conditions.												
Unmatched Record File	Specify the name of the file to output unextracted records. If there is any input record that does not satisfy the record extraction condition, it is output to this file. Since the unextracted records file contains the data that is yet to be converted by Data Converter, it can be used as the input file for record re-extraction.												

## Conversion Wizard Steps 3 and 4 - Input/Output File Information Setup

Menu Function	Explanation	
Encoding	Select the code system of the file being converted.	→ <a href="#">Conversion Rules - Character Code</a>
File Name	Specify an input or output file name using a full pathname. If you have set a default folder, you can also use a relative pathname.	
File Type (Data Format)	(for Data File Conversion) Specify a <b>Binary or Text file</b> or a COBOL File. Specify <b>Binary or Text file</b> for anything other than a COBOL File on Windows.	→ <a href="#">Binary or Text file</a> → <a href="#">COBOL File</a>
Key Information (Data Format)	(to Output a COBOL Index File) Specify an index key.	→ <a href="#">Index Key Information</a>
Setup (CSV Format)	(for a CSV-format File) Specify a CSV format.	→ <a href="#">CSV Information Setup</a>
XML Information Setup	(for an XML-format File) Specify an XML format.	→ <a href="#">XML Information Setup</a>
Specify Unicode Attributes	(for Unicode) Specify information on Unicode.	→ <a href="#">Endian</a> → <a href="#">Byte Order Marker</a>

Input										
Item	Value	file type								
		Text file	Binary or Text file *1	Binary or Text file *2	COBOL Sequential file	COBOL Line Sequential file *3	COBOL Relative file	COBOL Index file	CSV	XML *5
Encoding	UCS2	Y	N	Y	N	N	N	N	Y	N
	UTF8									Y
Endianness	Big	Y	Y	Y	Y	N	Y	Y	Y	N
	Little									
Output										
Item	Value	file type								
		Text file	Binary or Text file *1	Binary or Text file *2	COBOL Sequential file	COBOL Line Sequential file *4	COBOL Relative file	COBOL Index file	CSV	XML *6
Encoding	UCS2	Y	N	Y	N	Y	N	N	Y	N
	UTF8									Y
Endianness	Big	Y	Y	Y	Y	N	Y	Y	Y	N
	Little									
Omit Signature		Y	N	N	N	N	N	N	Y	Y
Y : usable N : disabled										
*1 This specification is handled as COBOL specification. (**) *2 This specification depends on the selected code system. (**) *3 When the input is a Line Sequential file, it follows the Signature. *4 When the output is Line Sequential file, endian cannot be specified, and the output is set to little endian. (specification of COBOL) *5 The encoding form is given priority when the input is specified as XML. *6 Specify whether to omit the output of the Signature. (**)Refer to "Use COBOL specifications on binary files" in <a href="#">Options</a> for details.										

Omit Unicode Byte Order Marker	Specify whether to omit the output of the Unicode byte order marker (BOM). Check this box only if you wish to suppress output of the Unicode byte order marker. (This option is only enabled when either the input or output file specification is Unicode.)	→ <a href="#">Unicode Byte Order Marker</a>
--------------------------------	--	---

Specify Record Attributes Specify information on a record. In general, specify information as follows:

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)
XML format	Variable length	New line code (CR+LF)

### Wildcard Specification

If a wildcard is used to specify an input file, the wildcard is assumed to apply to the filename only. The input file name extension will be replaced by the extension set for the output file name. (See exception for XML format below.)

#### Example

```
Input file name item   : *.dat
Output file name item : .out

Retrieved file names : aaa.dat, bbb.dat ...
Output file name     : aaa.out, bbb.out ...
```

#### About wildcard options when editing an XML format

##### When the input file is in XML format

→ [Precautions - Conversion of XML Files](#)

```
Input file name item   : *.xml
Output file name item  : aaa.dat

Retrieved file names  : aaa.xml, bbb.xml ...
Output file name      : aaa.dat
```

##### When the output side is in XML format

```
Input file name item   : aaa.dat
Output file name item  : *.xml

Input file name       : aaa.dat
Output file name      : aaa00000.xml, aaa00001.xml, aaa00002.xml ...
```

#### Error Files

- Specifying a wildcard as an input file name sets the default file name as Dataconv.err.
- All retrieved input file errors are output to an error file in the addition mode.
  - If an error file already exists, the existing file is deleted and a new file is created.
- The same layout definition file is used.
- If an output file already exists, a message box appears asking if the output file is to be overwritten.
  - In batch mode, processing is terminated if an output file already exists.
  - Specifying the /f option causes an existing output file to be overwritten.

## Conversion Wizard Step 5 - Other Information (Error, User Defined Conversion Table)

### Convert rule

Item	Explanation
Don't convert if input and output encodings are the same	If the input encoding is the same as the output encoding, the conversion is not done and the input data is output directly.

### Specify User Defined Conversion Table

Item	Explanation
File Name	If required, enter the name of a user defined conversion table file that specifies the code conversion to be used. → <a href="#">Creating User Defined Conversion Table</a>

### Specify Error File Information

Item	Explanation
File Name	Enter the full path name of a conversion error file. Relative path names can be used if a default folder is set. The error file extension defaults to .ERR. → <a href="#">Conversion Error File</a>
Conversion Error Limit	Specify the maximum number of error messages to be written to the conversion error file, from 1 to 3000. The default is 100.
Continue when # of errors exceeds limit	When this box is checked, Data Converter runs the conversion to the end, regardless of the number of errors. → <a href="#">Overview of Batch Mode</a> When this box is not checked, Data Converter displays a dialog box in batch mode or when the limit is reached. Data Converter aborts the conversion without displaying a message.
Continue Rule	If the maximum number of errors that can be output is exceeded during the conversion process, select to continue the conversion process with or without issuing a conversion error.
1 byte error substitution 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 hexadecimal digits. If no character is specified in this field, "_" (underscore) is output.



## Layout Definition Utility - Understanding Item Definition Information

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

→ [Advanced Settings](#)

→ [View Menu](#)

A number of the labels for the information columns are truncated in the display. 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.

Field	Meaning	
Lv	Displays the repetition levels of 0-7. Note that the repetition levels are different from the COBOL data description level numbers. Refer to <i>Defining repeating items</i> for a full description of repetition levels. Directly edits the sheet. Level: From 0 To 7	→ <a href="#">Defining repeating items</a>
Repeat (OCCURS)	Displays the number of repetitions. Lines with repetition counts only have the repetition level and repetition count. Refer to <i>Defining repeating items</i> for a full description. Directly edits the sheet. OCCURS: From 2 To 32767	
Item-name	An arbitrary name is assigned if no entry is made.	
Attributes	Displays the attributes of the item. Attributes are divided into character and numeric types. You cannot convert between character and numeric attributes.	→ <a href="#">Attribute Description</a>
Sign	(For numeric items) <b>S</b> is displayed when the sign format of an external decimal item is shown. Directly edits the sheet With a Sign: <b>S</b> A specified value is different in case of an external decimal, depending on the attribute. Please refer to <i>External Decimal Sign Type</i> for details.	→ <a href="#">External Decimal Sign Type</a>
Position	For data file format. Displays the offset of the item within the record. The position numbers start at 1.	
Length	Displays the length of the item in bytes. This is always an even number for National characters. Not needed for CSV format files.	
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 in the decimal part.	
Decimal	For numeric attribute items in data file format. Displays the number of digits after the decimal point.	
Quotes(*)	Specifies that Input CSV-formatted items are separated by quotation marks. Specifies that Output CSV-formatted items are to be created using quotation marks as separators. <b>Y</b> is displayed for items that have quotes. By default, character type items are enclosed in quotes, and numeric type items are not in quotes. Directly edits the sheet Enclose in quotation marks: <b>Y</b>	
Input Check(*)	Performs an input check.  <b>Y</b> is displayed for input data that will be checked for a match with the item attributes. This does not affect the output file - it only affects the messages that are written to the error file. Directly edits the sheet Perform input check: <b>Y</b>	
No Conversion(*)	Output data without converting it.  <b>Y</b> is displayed next to items to be output as-is, with no conversion. Directly edits the sheet Output data without converting: <b>Y</b>	
Fixed Output(*)	Output fixed length data.  <b>Y</b> is displayed for items to be output according to the length specified. The length is treated as 0 when no output area length is specified. The item is the output side variable-length (When the output is CSV format, include). When the	

output has a numeric character string attribute (V), the number of whole digits and number of fraction part digits specifications are applied.

If this specification is applied to all items, the output file becomes the length of a fixed record.

Directly edits the Output fixed length data: **Y**  
sheet Item length: Specifies the length in bytes of each output data item.

Zero Suppress(\*) Valid for numeric character strings with fixed output.

Enable suppression of zeros.

In the numeric character string attribute (V) of the fixed output specification, neither leading zeros nor the fraction part subordinate to position 0 are suppressed. When **Y** is displayed, zeros are suppressed and blanks are output.

Directly edits the Enable suppression of zeros: **Y**  
sheet

Null(\*) No conversion error generated when input data is null.

When the data file format has an alphanumeric character item attribute and the input data is all null code (0x00), the null code usually causes a conversion error and excludes the character. When **I** is specified, no conversion error is generated, and the initial value of the output side item attribute is output.

Directly edits the No error if null data: **I**  
sheet

Tag(\*) (for XML format)

Nothing is displayed in the Tag column when the item name is output as a start/end tag. **A** is displayed when it is to be output as an attribute. **E** is displayed when the item name is to be output as an EMPTY tag. **N** is displayed when the item name is not allowed to be output as a tag/attribute.

Directly edits the sheet Start/End Tag: *(blank)*  
Attribute: **A**  
EMPTY Tag: **E**  
None: **N**

Items are divided into character and numeric types. CSV format and XML format files can contain only **M** and **V** attributes. An asterisk "\*" in the meaning column marks the attributes in which the internal format differs according to the Endian of the supporting system.

→ [Endian](#)

The sign column shows a set value when the edit is input directly.

Type	Display char	Common Description	Meaning
Character	<b>X</b>	Alphanumeric character	String of 1 byte characters.
	<b>N</b>	National character	String of 2 bytes characters.
	<b>M</b>	Character string Mixed char	String of character.
	<b>C</b>	char type	Character string terminated by a null byte.
	<b>R</b>	Variable length string	Character string with date length in the first 2 bytes.(*). The support code system is ASCII, Unicode.
	numeric	<b>Z</b>	External decimal
<b>P</b>		Internal decimal	COBOL packed decimal format
<b>B</b>		Binary	COBOL binary format
<b>5</b>		COMP-5	COBOL binary format(*)
<b>S</b>		short type	16bit binary(*)
<b>L</b>		long type	32bit binary(*)
<b>V</b>		Numeric string	Numeric in character form (signs and the decimal point are included as characters).
Other	-	No character	Used for addition or deletion of items.

You can define the information for each item using the **Item Definition**.

→ [Understanding Item Definition Information](#)

→ [Advanced Settings](#)

Refer to *Understanding Item Definition Information* for the meaning of each item.

### Changing the line to be edited

The layout definition line to be edited can be changed using the buttons **Top**, **Previous**, **Next**, and **Bottom**.

Clicking the **Next** button when the final line is selected will create a new line.

### Defining Group Item

To clear a relative position, the area length, the number of whole digits, and the number of fraction part digits 0, select the blank line at the end of the Input Information/ Output Information **Attribute** combo box, and click **OK**. When defining a group item, it is recommended to not correct an existing item, but instead create a new line using line insertion and line addition.

### Automatic Update of Positions

When you change the length of an item, you affect the positions of all 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 the positions are not updated automatically, you must edit them manually. The automatic update function does not update repetition levels and numbers of repetitions. Also, it cannot be used when an item whose length has been changed has a repetition level greater than 1. The automatic change function cannot be used to change the hierarchy or the number of repetitions.

### Effective range of Cancel button

Only the item currently being edited is affected by the **Cancel** button. Changes made using the **Top**, **Previous**, **Next**, or **Bottom** button cannot be cancelled.

You can specify the conditions of the multi format file layout with the Layout Definition.

For more information about each condition, please refer to the Layout Definition Information dialog box.

### **Changing the layout definition to be edited**

The layout definition to be edited can be changed by clicking on **Next** or **Previous**. Clicking the **Next** button when the final layout is selected will create a new layout.

### **Effect of Cancel button**

Only the layout information currently being edited is affected by the **Cancel** button. If layout information was changed using the **Next** or **Previous** button, the changes cannot be cancelled.

[Layout Definition Information](#)

Specify advanced settings for the item. Display this dialog box by clicking the **Advanced...** button in the **Item Definition** dialog box.

→ [Item Definition](#)

→ [Understanding Item Definition Information](#)

Special file encoding information can be specified in the layout definition file.

Item	Explanation
Layout	Specify the layout name. (Treated as a comment.)
Record	Specify the record name. The record name is used for the root tag of XML (DTD) files, or the 01 level item of a COPY library.

The following items select a set of objects on the input side and each object on the output side.

Item	Explanation	See Also
Add NULL display area	<p>If selected, inserts a 2 byte null display area in front of each item. The actual item becomes 2 bytes longer than the length in <b>Item Information</b>.</p> <p>When you change this selection, you must also change the positions of each item. The <b>All Items</b> function can be used to update the positions.</p>	<p>→ <a href="#">Null display Area</a></p> <p>→ <a href="#">Item Definition Information</a></p> <p>→ <a href="#">Operations on All Items</a></p>
Treat binary as big endian.	<p>Specifies the endian of binary numeric on data. Selecting this item affects the following:</p> <ul style="list-style-type: none"><li>- Internal display of COMP-5, short, and long items.</li><li>- The data length added to variable length items (attributes R and Y)</li></ul> <p>It is recommended that you check this when you are targeting SPARC machines.</p>	→ <a href="#">Endian</a>
Treat NULL as input string terminator. (Used when the input is a data file format.)	Enable this option when a null in the input data is to be treated as a character string end.	

You need to select which of three data formats apply to the input and output files.

The three formats are:

- [Data File](#)
- [CSV File](#)
- [XML File](#)

### **Precaution on Changing the Data Format**

When you change the data format of an input or output file associated with an existing layout definition file, the information in the layout file is no longer valid. Moreover, when you change from CSV or XML format to data file format, it is necessary to define a relative position and area length for all items. When you change the data format, it is recommended that you reconfirm the definition of the file from the COPY library.



## Layout Definition Utility - Operations on All Items

Settings can be made for all items at the same time.

---

<b>Item</b>	<b>Explanation</b>
Operations menu	Select a function from the list. See the descriptions in the explanation box for the details of each function.
Target	Select whether the target of the function is the input definition, the output definition, or both. Input/Output selections are disabled according to what is appropriate for the function.
Explanation	Displays each function of a setup menu.

## Layout Definition Utility - COBOL Library Analysis option

Specify the required information to analyze a COBOL library.

Items with (\*) are default settings at first-time startup.

### Menu Function Explanation

Analysis Method Specify the analysis method.

Item	Explanation
COBOL analysis library(*)	Analyzes files using COBOL analysis library.
Fujitsu NetCOBOL compiler	Analyzes files using FUJITSU NetCOBOL compiler. Fujitsu NetCOBOL must be installed to use this option.

Analysis Options Specify other analyzing information.

Item		Explanation
Reference Format	Fixed	Specify the format of the COBOL library.
	Variable(*)	
	Free	
BINARY Option	WORD(*)	Indicate how binary items are to be handled. If this option is checked, binary items will be handled at a byte level. If this option is not checked, binary items will be handled at the word level. For unsigned binary items, the handling of the leftmost bit can also be specified: BINARY(WORD,MLBON): The leftmost bit is the sign bit. BINARY(WORD,MLBOFF): The leftmost bit is a number.  Fujitsu NetCOBOL must be installed to use this option.
	BYTE	
Additional COPY sentence clause		If the copy sentence does not specify the REPLACING phrase when the COBOL source contains a temporary original (parentheses etc.), the analysis may fail. To analyze it correctly, add a COPY sentence clause here, using COBOL grammar. The specification of the JOINING/DISJOINING phrase is also permitted. Example:  REPLACING ==() == BY ==AA ==  This option is only available when <b>COBOL Library</b> is specified for a file type.
Group Item Option	Show(*)	Displays group items on the layout definition as the result of analysis. The item name of the group item and the repetition definition is displayed.
	Hide	Does not display group items as the result of analysis on the layout definition. This option specifies the same display as the layout definition of versions before V3L31.

## Conversion Rules - Character Code

The following table shows the input and output Character Code combinations supported by Data Converter.

Input/Output	ASCII	EBCDIC	Unicode	Latin1	Code Page
ASCII	Y	Y	Y	N	N
EBCDIC	Y	Y	Y	N	N
Unicode	Y	Y	N	Y	Y
Latin1	N	N	Y	Y	Y
Code Page	N	N	Y	Y	Y

Y : usable

N : disabled

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 regardless of whether 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.

The following table shows the input and output file format combinations supported by Data Converter.

Input/Output	Binary file	COBOL file	CSV file	XML file	Text file
Binary file	Y	Y	Y	Y	N
COBOL file	Y	Y	Y	Y	N
CSV file	Y	Y	N	Y	N
XML file	Y	Y	Y	Y	N
Text file	N	N	N	N	Y

Y : usable

N : disabled

The following table shows the code systems that are supported for each file format in Data Converter.

format/code	ASCII	EBCDIC	UCS2	UTF8	Latin1	Code Page
Binary file	Y	Y	Y	Y	Y	Y
COBOL file	Y	Y	Z	Z	Y	Y
CSV file	Y	N	Y	Y	Y	Y
XML file	Y	N	N	Y	Y	N
Text file	Y	Y	Y	Y	Y	Y

Y : usable

Z : stores by X item (UTF8 form) and N item (UCS2 form).

N : disabled

**Size Errors when Converting COPY Sentence Clauses**

---

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

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

**Treatment of NULL (0x00) and extended ASCII code**

Low-value is a conversion error because Data Converter does not consider 0x00 and extended ASCII code (0x80-0xFF) to be a character string. For low-value data, use the User Defined Conversion Tables (.CNV) as follows.

Example of User Defined Conversion Tables:

```
#ASCI ASCI  
00:00  
#ASCI EBCD  
00:00
```

### Signs in CSV Format, XML Format

---

#### CSV Output Files and XML Output Files

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

#### CSV Input Files and XML 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 "+".

#### Values Output when Numeric Errors are Detected

Numeric errors occur when input numeric items contain invalid 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). For example, if the input field contains **A** (0x41), the output is the digit 1; if the input is **L** (0x4c), the output digit is 2 (0x4c divided by ten leaves a remainder of 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 keep the digits from the input number.

## Extraction Specifications - Record Extraction

Specifications for record extraction are explained below.

- Conversion of conditional value's character code (for character type attribute data)
- Numeric value conversion for comparison against conditional value (for numeric type attribute data)
- Supplement: Rules applied when multiple data conditions are specified

For notes on extraction specifications, see [Notes on Record Extraction Function](#).

### Conversion of conditional value's character code (for character type attribute data)

When records are extracted on the condition represented by data with a character type attribute, Data Converter first converts the character code of the conditional value specified on the Set Data Conditions dialog box to the input file's character code system, and then compares the data in the input file against the conditional value. Code conversion of the conditional value is performed based on the following rules. If the input file's code system is Shift JIS, or if Hexadecimal Specification is selected, code conversion will not be performed.

\* Code conversion from Shift JIS to the input file's code system is performed.

Input file's code system	Conversion
ASCII	Code conversion not performed. The data specified for the conditional value will be used.
EBCDIC	ASCII to EBCDIC
Unicode (UCS2/UTF8)	ASCII to Unicode

After character code conversion, extraction rules are applied based on the operator specification. Extraction rules are explained below.

#### Alphanumeric character or character string (CSV/XML) items

- String Equals**      The data item in the input file will be evaluated as a "string equals" when its area length and value match the area length and value of the conditional value.
- Prefix Match**      The data item in the input file will be evaluated as a "prefix match" when the part of the conditional value that fits within the conditional value area length matches the value of the data item in the input file.

#### CHAR-type items

- String Equals**      The data item in the input file (not including the NULL character that indicates the end of the character string) will be evaluated as a "string equals" when its area length and value match the area length and value of the conditional value.
- Prefix Match**      The data item in the input file will be evaluated as a "prefix match" when the part of the conditional value that fits within the conditional value area length matches the value of the data item in the input file.

#### Variable-length character string items

- String Equals**      The data item in the input file (including only the part that fits within the valid area length in RDW) will be evaluated as a "string equals" when its area length and value match the area length and value of the conditional value.
- Prefix Match**      The data item in the input file will be evaluated as a "prefix match" when the part of the conditional value that fits within the conditional value area length matches the value of the data item in the input file.

#### Variable-length national items

- String Equals**      The data item in the input file (including only the part that corresponds to the number of characters in RDW) will be evaluated as a "string equals" when its area length and value match the area length and value of the conditional value.  
\* For the data in the UTF8 format, RDW will be included in the area length.
- Prefix Match**      The data item in the input file will be evaluated as a "prefix match" when the part of the conditional value that fits within the conditional value area length matches the value of the data item in the input file.

### Numeric value conversion for comparison against conditional value (for numeric-type attribute data)

When records are extracted using the data with the numeric type attribute as the condition, Data Converter converts the numeric data in the input file to a format (numeric value) that allows comparison with the conditional value specified in the Set Data Conditions dialog box before comparing the data in the input file against the conditional value. Conversion to numeric values is performed based on the following rules.

After numeric value conversion, extraction evaluation is performed based on the operator specification and then the records are extracted.

Unsigned attributes	Always treated as a positive value.
Signed attributes	Treated as a negative value if a minus sign is detected or as a positive value otherwise.
Conditions making data items ineligible	External or internal decimal - Non-numeric data is contained in a numeric part. CSV/XML format - Data other than the plus/minus sign, decimal point, or numbers is contained.

for extraction

- Data having 19 or more significant digits is contained.
- A plus/minus sign is detected at a place other than the beginning of the area or more than one decimal point is detected.

**Supplement: evaluation rules used when multiple data conditions are specified**

The example below shows how evaluation is performed when two or more data conditions are specified.

Data condition setting example:

No.	Concatenation Condition	Item Name	Operator	Conditional Value	Hexadecimal Specification
1	--	Product name	String Equals	Bar code reader	
2	OR	Product name	Prefix Match	High-speed	
3	OR	Unit price	>	10000	
4	AND	Unit price	<=	90000	
5	OR	Customer name	Prefix Match	8FAC97D1	YES
6	OR	Customer name	Prefix Match	8E52;89BA	YES

Data condition evaluation order

1. No. 1 data condition is evaluated.
2. No. 2 data condition is evaluated.
3. No. 1 and No. 2 data condition evaluation results are OR'ed together.
4. No. 3 data condition is evaluated.
5. The result of step 3 above is OR'ed with the result of No. 3 data condition evaluation.
6. No. 4 data condition is evaluated.
7. The result of step 5 above is AND'ed with the result of No. 4 data condition evaluation.
8. No. 5 data condition is evaluated.
9. The result of step 7 above is OR'ed with the result of No. 5 data condition evaluation.
10. No. 6 data condition is evaluated.
11. The result of step 9 above is OR'ed with the result of No. 6 data condition evaluation.  
This is the final data condition for record extraction.

The above sequence can be represented by a formula as follows:

(((((product name = bar code reader) OR (product name » high speed)) OR (unit price > 10000)) AND (unit price £ 90000)) OR (customer name » 8FAC97D1)) OR (customer name » 8E52;89BA)

## Precautions - 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	3000	
Maximum length of character data items	32767 bytes	
Maximum number of digits in numeric items	18 digits	Includes fractional part of the number.
Maximum levels of nesting in OCCURS clause	7	
Maximum number of items in a COPY library	3000	Includes group items.
Maximum length of item name	60 bytes	
Maximum number of errors that can be output	9999	
Maximum number of XML files that can be output	100000	Differs according to the file system.
Maximum log file size	4 Gbytes	

### Maximum COBOL I/O file size

The size of the COBOL I/O file that can be processed depends on COBOL and the COBOL runtime system that uses it.

### Unsupported File Formats

- Variable length files from mainframe or mini computers (general file)
- Files with multiple record formats (multi-format)



The content described in [Understanding Data Converter](#) can be preserved as a [Conversion Information File](#) , but the following precaution exists.

- Data Converter stores full path names for the Layout Definition, Input and Output files in the Conversion Information File. For example, if the layout definition file named when the conversion instructions are saved does not exist or has been moved, an error will occur when the conversion instruction file is opened.

Take the following precautions when using [COBOL files](#) .

- Data Converter does not take exclusive control of COBOL 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 without duplicates allowed.
- Data Converter picks up the organization, record attributes and record lengths from the header records of COBOL 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 and continues with the conversion.

## Precautions - Analyzing COBOL library

- The analysis expects correct COBOL syntax. Incorrect syntax may produce unpredictable results.
- The maximum number of items in the library, including group items, is 3,000. If a library contains more than 3,000 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 could 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 01-49.
  - 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 item, Index items, Floating point items
- Numeric edited items and alphanumeric edited 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

### Direct Input of Layout definition

- Correct operation cannot be guaranteed if an item name contains trailing blank spaces.

## Precautions - 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 hexadecimal.
- Preview uses only **Courier New** font.
- Non-character codes, other than new line character, are displayed as "." (period).
- The Preview function displays up to 100 records, starting with the first record in the file.

The Precautions of the [Displaying Conversion Errors](#) function are as follows.

- 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.

## Precautions - About Quotation Marks and Delimiters in CSV Format

The following ASCII code tables show the "quotation mark" and the "delimiter" character values that can be used with the CSV format.

### ASCII code tables

	0	1	2	3	4	5	6	7
0			SP	0	@	P	`	p
1			!	1	A	Q	a	q
2			"	2	B	R	b	r
3			#	3	C	S	c	s
4			\$	4	D	T	d	t
5			%	5	E	U	e	u
6			&	6	F	V	f	v
7			'	7	G	W	g	w
8			(	8	H	X	h	x
9			)	9	I	Y	i	y
A			*	:	J	Z	j	z
B			+	;	K	[	k	{
C			,	<	L	\	l	
D			-	=	M	]	m	}
E			.	>	N	^	n	~
F			/	?	O	_	o	

usable

disabled

- Note 1) When the quotation mark character and the delimiter are specified by the same code, the conversion result cannot be guaranteed.
- Note 2) When the same character as the delimiter exists in a character item and a numeric item that does not use the quotation mark character, the conversion result cannot be guaranteed.
- Note 3) Refer to *CSV details and setting dialog box* for the method of specifying 0x09.

Precautions for converting into XML format are as follows.

- Conversion from XML format to another format is a beta function.
- In the output of [XML format](#), the XML declaration is not added by not selecting a template file in [XML Information setting](#).
- Conversion will be done for the valid XML file (which needs DTD).
- Awell formed XML file (which does not need DTD) can be converted if it has a clear structure.
- An XML file that cannot be analyzed by Fujitsu XML Library cannot be read.
- XML files that cannot express the structure of a valid XML file in the layout definition cannot be read.
- In the conversion from the XML form to another form, the error number may be returned by the errors occurring while analyzing the XML file.

The content of the **error No** error is as follows.

Error Values	Contents
-1	Failed to generate the DOM manager.
-22	The area of item information on the layout file is not opened.
-23	The area for the data acquisition of each record is not opened.
-24	Capacity shortage of the disk occurred when opening the layout file.
-25	The error of the file access prohibition occurred when opening the layout file.
-26	A file undefined error occurred when opening the layout file.
-27	An invalid path error occurred when opening the layout file.
-28	The error of the exceeding numbers of opened files occurred when opening the layout file.
-29	A file undefined error occurred when opening the layout file.
-30	A file exclusion error occurred when opening the layout file.
-31	The error occurred when opening the layout file.
-32	Failed to get the development area of item information on the layout file.
-35	Failed to get the number of areas for the record data return.
-36	Failed to get the area for the record data return.
-37	Failed to read the layout file.
-38	The input data form of the specified layout file is not XML data.
-51	Capacity shortage of the disk occurred when opening the XML file.
-52	A file access prohibition error occurred when opening the XML file.
-53	A file undefined error occurred when opening the XML file.
-54	An invalid path error occurred when opening the XML file.
-55	The error of the exceeding numbers of opened files occurred when opening the XML file.
-56	A file undefined error occurred when opening the XML file.
-57	A file exclusion error occurred when opening the XML file.
-58	The error occurred when opening the XML file.
-60	A tag other than the route tag do not exist in the specified XML file.
-61	There is a problem in the grammar in the XML file of the reading object.
-63	Corresponding tag to the record name of the specified layout file does not exist in the XML file.
-99	Failed to get the tag specified for the retrieval key.

Notes on the Record Extraction function are provided below.

### Notes on using the Set Data Conditions dialog box

If you modify the layout definition file after specifying one or more data conditions on the Set Data Conditions dialog box, you must check the data conditions. It is recommended that you delete all of the data conditions and then re-specify them.

Record extraction cannot be performed under the following conditions. If either of the following conditions exists, an error message is displayed when the dialog box is opened.

- The data in the input file contains a null display area, and the Add NULL display area checkbox (input) is checked on the Basic Information dialog box of the Layout Definition Function.
- The binary data in the input file is to be handled based on Big Endian ordering, and the Treat Binary on Big Endian Ordering checkbox (input) is checked on the Basic Information dialog box of the Layout Definition Function.

### Notes on specifications in the Set Data Conditions dialog box

- If you specify a conditional value for a non-unique item name, the record extraction is performed on the first detected item having the specified name. For correct record extraction, make item names unique within the layout definition.
- If a space or comma is used for the item name in the layout definition, record extraction cannot be performed.
- Since the area allowed for entry of the conditional value is 150 bytes, a complete match will be impossible for data items that exceed 150 bytes. Use Truncation instead.
- If you specify a non-character code in the conditional value (for example by using the Windows cut and paste function), the operation of the record extraction function is unpredictable. To detect control characters, use the Hexadecimal Specification.
- If a conditional value is specified for a variable-occurrence item, record extraction is performed on the first occurrence of the item. To perform record extraction on all occurrences of the item, take the necessary action -- e.g., expand the variable-occurrence item in the layout definition and define the conditional value for each occurrence of the item.
- Hexadecimal Specification is not applicable for numeric type items.

### Conversion Results File and Unextracted Records File

- When conversion results are to be output to multiple files (one file for one record), records ineligible for record (file) extraction will not be output. In other output formats, an empty (0 bytes in size) conversion results file will be created if there is no record eligible for extraction. However, if CSV item header output or the XML root tag or template is specified, the specified information will be output.
- If the input file format is an indexed file or XML file, the unextracted records file will not be created even if so specified.
- Output to the unextracted records file is not supported if more than one input file exists. Only unextracted records for the last processed input file will be output.

### Numeric value conversion for comparison against conditional value (for numeric-type attribute data)

- When the file format is CSV or XML, Data Converter considers NULL as 0.
- For the sign part of the numeric data of an external/internal decimal item, a strict comparison is performed only on the negative sign part. Otherwise, the sign is considered positive. (Unsigned data is always considered positive.)
- For numeric value conversion for comparison against conditional value, the number of characters will not be checked strictly. Even if numeric data within the input file is abnormal, Data Converter performs numeric value conversion assuming that the data is normal. Therefore, expected extraction may not be assured.
- Example:  
PIC 9(6) COMP3.  
When the input data (in hexadecimal) is 1234567C,  
the record extraction function considers it as a 7-digit numeric value of +1234567.



## Precautions - Multi Layout Function

---

Precautions of Multi Layout Function are as follows:

- The Multi Layout Function does not support an XML format file as an input file.
- When the file format of the input file is Binary, you cannot convert the file correctly if the file has line feed codes expressed with 0x0d0a or 0x0a, because the position of a line feed code is regarded as the end of the record.
- The COBOL runtime system must support input and output of variable length files for converting a multi format file.
- When you specify a character string enclosed in double quotations in the minimum/maximum field of the Layout Definition dialog box, the character code of the input file should be ASCII.
- When you read an existing Layout Definition File for inserting or adding a new layout file, the data format of input/output files should be the same as other layout definitions.  
When you analyze an existing COBOL library to create a new layout definition, the input/output data format that is the same as other layout definitions is applied to the new layout definition automatically.

Precautions of the Non-Conversion Output Function are as follows:

- When the input area length is longer than the output area length, the part of input data that exceeds the output area length will be truncated. In this case, Data Converter does not report digit size errors.
- When the input area length is shorter than the output area length, Data Converter inserts NULL in the part of output data that exceeds the input area length.
- For variable-length character items:
  - for input, the valid data length will be the input data length.
  - for output, if the input file format is Data, the valid data length will be the input data length.
  - for output, if the input file format is CSV, the end of the data will be the next NULL position.

**Note:**

When the input area length is longer than "the output area length - 2", the part of the input data that exceeds "the output area length - 2" will be truncated. In this case, Data Converter does not report digit size errors.

When the input area length is shorter than "the output area length - 2", Data Converter sets NULL in the part of output data that exceeds the input area length.

- Data Converter converts numeric item data normally.
- Data Converter does not support XML file input/output.

**COBOL File**  
**COPY Library**  
**CSV format**  
**XML format**  
**DTD**  
**Unicode Byte Order Marker**  
**Binary or Text file**  
**Endian**  
**Item extension information**  
**Data File Format**  
**Data file conversion**  
**Text conversion**  
**Null display region**  
**Conversion Error File**  
**Conversion Information File**  
**User definition translation table**  
**Layout Definition File**

### COBOL File

Data Converter directly supports the following COBOL file organizations:

- Sequential
- Line Sequential
- Relative
- Indexed

Any file made by a COBOL program on a general purpose machine is treated as a general COBOL file in Data Converter.

### COBOL (COPY) Library

A COBOL source file which, for Data Converter 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 allows you to configure other characters to be used as separators.

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

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

### XML Format

XML document format. The XML document is created by converting the Item name from the COBOL layout definition into XML tags and attributes.

### DTD

DTD (XML Document Type Definition) defines elements and data structures of XML documents.

### Oracle

Relational database management system ([RDBMS](#)) from Oracle Co.

### RDBMS

Abbreviation for "Relational Database Management System": software that manages a relational database. Relational databases store data as a set of rows, organizing the rows into related tables, and allowing data to be combined, extracted, or deleted based on information contained within, or derived from, a set of rows.

### Unicode Byte Order Marker

A code added to the beginning of a Unicode file to identify whether it is of type UCS2 (Little Endian), UCS2 (Big Endian), or UTF8.

### SQLServer

Relational database management system ([RDBMS](#)) from Microsoft Co.

### Symfoware

Relational database management system ([RDBMS](#)) from Fujitsu.

### Binary or Text file

Data Converter defines "Binary or Text" files as any non-COBOL file. The following are Binary or Text files.

- Binary file imported from a Windows/Intel or UNIX system.
- Binary file exported to a Windows/Intel or UNIX system.

- Text file created by Windows (DOS) or UNIX

### Endian

A 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 and other 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.

### Item Extension Information

Additional information used to guide the conversion of items: it can be displayed and edited using the Layout Definition Utility.

### Data File Format

File format in which data items are defined only by their byte position within records.

Files created by COBOL programs are in this format. An optional alternative is "CSV" format, in which data items are of variable length and separated by commas.

### Data File Conversion

A file conversion mode for all conversions other than those involving text files. Data File Conversion mode is used for conversions involving "CSV" (comma-separated values) files.

### Text Conversion

A file conversion mode for 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 these bytes contain hexadecimal '0xFFFF', the following data is ignored by Data Converter; when the bytes contain hexadecimal '0x0000', they are processed by Data Converter.

### Conversion Error File

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

### Conversion Information File

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

### User Definition Translation Table

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

### Layout Definition File

A 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".

## Layout Definition Utility - Generate COBOL library

Specify whether to use input or output layout information to generate a COBOL library. The default setting is to use the output information.

---

<b>Item</b>	<b>Explanation</b>
Input	Generate COBOL library using the attributes of the input layout definition.
Output	Generate COBOL library using the attributes of the output layout definition.

## Layout Definition Function - An example of generating a COBOL library

In the layout definition, the description of the layout definition under the edit can be output as a COBOL library.

Specify whether use the input information or the output information defined in the Layout definition to [generate the COBOL Copy Book](#).

### The output of the model COBOL library

If the data format of the output object is CSV or XML, the COBOL Library is output in the form of PIC 9 (1) when an attribute is a character sequence, and PIC X (1) when an attribute is a numerical value.

#### Example)

##### Layout Definition

Lv	Repeat	Item-Name	Attribute	Sign	Length	Total digits	Decimal
0		ITEM1	X		10		
0		GROUP1					
1		ITEM2	X		3		
1		ITEM3	X		1		
0	10	GROUP2					
1		ITEM4	X		5		
1		ITEM5	X		5		
1	5	GROUP2-1					
2		ITEM6	Z			2	
1		ITEM7	X		2		
0		ITEM8	X		20		
0	5	GROUP-3					
1		ITEM9	X		8		
0		ITEM10	X		1		

##### Output COBOL Library

```
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 GROUP-3 OCCURS 5.  
 05 ITEM9 PIC X(8).  
03 ITEM10 PIC X(1).
```

## Layout Definition Function - An example of generating DTD

In the layout definition, the description of the XML form of the layout definition under the edit can be output as DTD. Since the definition of XML is not strictly expressible in the layout definition, generated DTD is not restricted. It may differ from the structure of the XML document generated from the defined layout definition. Generating to the reference level of the XML document structure is recommended.

Example)

### Layout Definition

Lv	Repeat	Item-Name	Attribute	XML Tag
0		ITEM1	X	
0		GROUP1		
1		GROUP2		N
2		ITEM2	X	
2		ITEM3	X	
0		GROUP3		
1		ITEM4	X	
1		ITEM5	X	
1		GROUP4		
2		ITEM6	Z	
2		ITEM7	X	
0		ITEM8	X	
0		GROUP5		
1		ITEM9	X	A
1		ITEM10	X	N
0	5	GROUP6		
1		ITEM11	X	
1		ITEM12	X	N

### Output DTD

```
<!ELEMENT ABC-REC (ITEM1,GROUP1,GROUP3,ITEM8,GROUP5,GROUP6)>
<!ELEMENT ITEM1 (#PCDATA)>
<!ELEMENT GROUP1 (GROUP2)>
<!-- ELEMENT GROUP2 (ITEM2,ITEM3) -->
<!ELEMENT ITEM2 (#PCDATA)>
<!ELEMENT ITEM3 (#PCDATA)>
<!ELEMENT GROUP3 (ITEM4,ITEM5,GROUP4)>
<!ELEMENT ITEM4 (#PCDATA)>
<!ELEMENT ITEM5 (#PCDATA)>
<!ELEMENT GROUP4 (ITEM6,ITEM7)>
<!ELEMENT ITEM6 (#PCDATA)>
<!ELEMENT ITEM7 (#PCDATA)>
<!ELEMENT ITEM8 (#PCDATA)>
<!ELEMENT GROUP5 (#PCDATA)>
<!ATTLIST GROUP5 ITEM9 CDATA #IMPLIED>
<!ELEMENT GROUP6 (ITEM11+)>
<!ELEMENT ITEM11 (#PCDATA)>
```



## Data Definition Utility - Preferences

Specify the edit method when double-clicking on the layout definition screen. The default setting is to edit using the Item Definition dialog.

---

Item	Explanation
Edit Using Item Definition	The double-clicked line is edited using the <b>Item Definition</b> dialog.
Directly Edit the Item Definition	The cell that is double-clicked is opened for input on the layout definition screen.

## Conversion Wizard - Index Key Information

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] ...

Offsets and lengths are numbers of bytes. Offset values start at 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
<b>1. Quotes</b>	Specifies the quote marks used for character strings.
<b>2. Delimiter</b>	Specifies the delimiter between CSV fields. The default is comma "," which is the standard CSV delimiter.
<b>3. Output quoted data using double quotes</b>	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".'
<b>4. Output null character as double quotes</b>	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.
<b>5. Output item header in first record</b>	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 import into a spreadsheet.

When you use the "Reference settings", these settings are automatically selected for the appropriate data store. The row numbers in the following table correspond to those under the heading "Item" above.

	Symfoware	Oracle	INFORMIX	SQLserver	Excel
<b>1</b>	"	"	(nothing)	(nothing)	"
<b>2</b>	,	,	,	,	,
<b>3</b>	Yes	Yes	No	No	Yes
<b>4</b>	No	Yes	No	No	No
<b>5</b>	No	No	No	No	Yes

Specifying an XML Format.

---

Menu Function	Explanation
------------------	-------------

---

Add Standard Template File

If the [DTD](#) template of the Data Converter standard is added to the head of the output file according to the code system, Data Converter checks it. One of the following two standard template files will be created.

CODE	File Name
ASCII	ASCII.xml
Unicode(UTF-8)	UTF8.xml

Template File Name

Specifies the template file name by full path. When a current folder is set, a relative path can be specified.

Browse

Opens the dialog by which the template file is selected.

\* If you do not use a template file, be sure to remove the check on **Add Data converter Standard Template File** and clear the template file name column.

External decimal items can have the following sign types.

The sign column shows a set value when the edit is input directly.

<b>Sign</b>	<b>Kind</b>	<b>Meaning</b>
<b>S</b>	Yes	Same as "T".
<b>L</b>	LEADING	LEADING sign. Stored in the high order 4 bits of the first byte.
<b>T</b>	TRAILING	TRAILING sign. Stored in the high order 4 bits of the last byte.
<b>LS</b>	LEADING SEPARATE	LEADING SEPARATE sign. Stored in the first byte.
<b>TS</b>	TRAILING SEPARATE	TRAILING SEPARATE sign. Stored in the last byte.

## Layout Definition - Layout Definition Information

The Layout Definition Information in the Multi Layout Definition is as follows:

Field	Description
Layout Name	Specify the layout name, up to 90 bytes.
Condition	Specify the conditional expression. This field is required. EQ        EQUAL NOT      NOT EQUAL OTHER    OTHER NONE     NONE
Offset	Specify the base position in records from which comparison of the data is to start, from 1 to 32767. This field is required.
Length	Specify the comparison data length in bytes, from 1 to 32767 - Offset. This field is required.
Minimum	Specify the minimum value that is to be compared to the record data. Specify a hexadecimal value or a character string enclosed in double quotations. When the specified value length is shorter than the comparison data length specified above, Data Converter inserts NULL in the part of the comparison data that exceeds the minimum value length. When the specified value length is longer than the comparison data length, Data Converter truncates the part of the minimum data that exceeds the comparison data length.
Maximum	Specify the maximum value that is to be compared to the record data. Specify a hexadecimal value or a character string enclosed in double quotations. When the specified value length is shorter than the comparison data length specified above, Data Converter inserts NULL in the part of the comparison data that exceeds the minimum value length. When the specified value length is longer than the comparison data length specified above, Data Converter truncates the part of the maximum data that exceeds the comparison data length.