

Data Import
Fourth Shift Release 8.00

Fourth Shift Help
Release 8.00

Important Notices

Copyright © 2016 Infor. All rights reserved.

The material contained in this publication (including any supplementary information) constitutes and contains confidential and proprietary information of Infor.

By gaining access to the attached, you acknowledge and agree that the material (including any modification, translation or adaptation of the material) and all copyright, trade secrets and all other right, title and interest therein, are the sole property of Infor and that you shall not gain right, title or interest in the material (including any modification, translation or adaptation of the material) by virtue of your review thereof other than the non-exclusive right to use the material solely in connection with and the furtherance of your license and use of software made available to your company from Infor pursuant to a separate agreement, the terms of which separate agreement shall govern your use of this material and all supplemental related materials ("Purpose").

In addition, by accessing the enclosed material, you acknowledge and agree that you are required to maintain such material in strict confidence and that your use of such material is limited to the Purpose described above. Although Infor has taken due care to ensure that the material included in this publication is accurate and complete, Infor cannot warrant that the information contained in this publication is complete, does not contain typographical or other errors, or will meet your specific requirements. As such, Infor does not assume and hereby disclaims all liability, consequential or otherwise, for any loss or damage to any person or entity which is caused by or relates to errors or omissions in this publication (including any supplementary information), whether such errors or omissions result from negligence, accident or any other cause.

Without limitation, U.S. export control laws and other applicable export and import laws govern your use of this material and you will neither export or re-export, directly or indirectly, this material nor any related materials or supplemental information in violation of such laws, or use such materials for any purpose prohibited by such laws.

Trademark Acknowledgements

The word and design marks set forth herein are trademarks and/or registered trademarks of Infor and/or related affiliates and subsidiaries. All rights reserved. All other company, product, trade or service names referenced may be registered trademarks or trademarks of their respective owners.

Contents

Data Import Module 4

Using the Data Import Features 5

IMPT – Data Import 32

RCVR – System Recovery 39

TMPT – Template Maintenance 45

Template Browse 55

Data Conversion Table 57

Data Import Module

The data import features in the System Control Module make it easy to input information into the database in a batch mode just as if you entered the data online from a keyboard. Using these features, you can easily define an input data format. The system uses this input format to translate incoming data and update the database.

The data import features facilitate conversion from one system to Fourth Shift during the installation process. It also allows you to perform global maintenance, batch updates, and transaction recovery on the database. Fourth Shift can now be interfaced to other systems such as AutoCAD, along with bar code readers, for example, for automatic updates.

Using the Data Import Features

Several other manuals are available for data import process reference information.

- **Data Import Macro Basics** is a reference to the commands and techniques you can use to create custom macros for specialized transactions. A macro is a set of simple commands which rapidly replays keystrokes.
- **Data Import Dictionary** is a guide to the information necessary to use the predefined transactions or macros, currently provided for Fourth Shift. A predefined transaction uses the format of the archive file, ARCHIVE.FIL, and its corresponding screen(s) to import data into the system automatically.

The Data Import Process

Macro Identification

You can use predefined macros or your own custom macros.

For predefined macros, use the **Data Import Dictionary** to determine the type of transaction to be performed and its associated macro requirements. Each transaction has a six-character identification: the Function Code followed by two digits. For example, the macro id for adding locations on the LMMT screen is LMMT00.

For custom macros, print out the macro file for reference.

Incoming Data File Preparation

Preparing an incoming data file involves the following steps:

Creating Templates

Your incoming data file may or may not be formatted in the field layout expected by the Data Import process. When the field layout does not exactly match the order or format expected by the Data Import process, a relationship must be defined between the incoming data and the format expected by the Data Import process. The relationship definition is called a **template**. Templates are not required unless you need to define this relationship.

The TMPT (Template Maintenance) screen is used to create a template, which relates your incoming data fields to the format of required macro fields. A template contains a definition for each field required by a macro. See "TMPT (Template Maintenance)" for detailed information on the TMPT screen.

The **Data Import Dictionary** is a companion to the TMPT screen. The **Data Import Dictionary** is a guide to the information necessary to use the predefined transactions, or macros, currently provided for Fourth Shift. A predefined transaction uses the format of the archive file, ARCHIVE.FIL, and its corresponding screen(s) to import data into the system automatically.

For an example of how to use the dictionary, see "A Dictionary Example" in the Data Import Dictionary.

Analyze Required Macro Fields

Predefined Macros. To use predefined macros, incoming data records must be defined in terms of archive records. Use the **Required Transaction Fields** table in the **Data Import Dictionary** for the transaction you have identified. For each required field note the field number, type and maximum length. Also note whether a field must have a value for proper system identification (indicated by an asterisk by the **Field#**).

Custom Macros. To use custom macros, incoming data records must be defined in terms of the fields required by the macro. Using the macro printout, highlight all field references. Identify the field number, type and maximum length. Also note whether a field must have a value for proper system identification.

Analyze Incoming Data Records

An incoming data record must contain the fields identified as necessary for proper system identification. These fields are marked with asterisks in the **Required Transaction Fields** table for the transaction in the **Data Import Dictionary**.

1. Identify any required macro fields missing in the incoming data record. Missing fields can be:
 - initialized using a literal
 - identified as null and tabbed through to remain unchanged

See "Using Incoming Data Files" for more information on using literals and null fields.

2. Determine the field validation performed by the screen(s) used.

Incoming data fields should match Fourth Shift validation rules. Use the online help for each screen used by the macro to help determine validations. Compare the incoming data field length to the Fourth Shift field length for possible truncation. See "TMPT (Template Maintenance)" for truncation rules.

Assign any conversion values necessary to meet Fourth Shift field validations. Conversion values are specified in the Data Conversion Table window. See "Data Conversion" in this section for more information on conversion techniques.

3. Identify the record type and layout of incoming data. For CDF records, identify the field number for each field in the record. For SDF records, identify the starting column position and length of each field in the record.

Incoming Data File/Macro Relationship

A relationship must be established between the incoming data file and the macro used to import the data. The ways in which you can establish this relationship depend on the format of your incoming data file.

CDF File

A macro name for a CDF file can be identified in one of two ways:

- added as the first field in the data file
- referenced as a literal using the TMPT screen

For example, the BILLREV.DAT data file looks like this:

```
"18-LM", "105376", "ASSM9", "010", "A"
"18-LM", "21841", "ASSM9", "020", "A"
"18-LM", "21842", "ASSM9", "030", "A"
"18-LM", "51-50", "ASSM9", "040", "A"
```

The modified BILLREV.DAT data file referencing the macro {BILLRV} looks like this:

```
"BILLRV", "18-LM", "105376", "ASSM9", "010", "A"
"BILLRV", "18-LM", "21841", "ASSM9", "020", "A"
"BILLRV", "18-LM", "21842", "ASSM9", "030", "A"
"BILLRV", "18-LM", "51-50", "ASSM9", "040", "A"
```

Using the TMPT screen, the same end can be achieved by creating a template with the following field definitions:

Fld#/Offset	Reqd Fld#	Reqd Lgth	Fld Type	Literal/Table Name
	01	06	L	BILLRV
01	02	15	A	
02	03	15	A	
03	04	05	A	
04	05	03	A	

SDF File

A macro name for an SDF file is referenced as a literal using the TMPT screen. For example, the BILLREV.DAT data file looks like this:

```
DA1800      S200 FAB  01002
DA1800      T150 FAB  01002
DA1800      WC50 FAB  01002
DA1800      T150 FAB  01102
```

Using the TMPT screen, a template with the following field definitions is created:

Fld#/Offset	Length	Reqd Fld#	Reqd Lgth	Fld Type	Literal/Table Name
		01	06	L	BILLRV
00	15	02	15	A	
15	05	03	15	A	
20	05	04	05	A	
25	03	05	03	A	
28	02	06	02	A	

Data Import Processing

Use the BSET (Batch Process Setup) screen to set up a batch process containing the IMPT task. Use default parameters as much as possible to utilize the space available in the **Parameters** field.

Parameter validation using the F4 key provides detailed information on invalid file parameters. It is important to validate parameters before running the IMPT task. See "Data Import Error Messages" for an explanation of error messages for the IMPT task.

Single-User and Multi-User Mode

You can run an import task using either single-user or multi-user mode. Your choice of operational mode depends on how much data you are importing at one time. For example, single-user mode is recommended when initializing your database.

If you use multi-user mode, it is recommended that you use the Loop Back to Retry parameter to the IMPT task. The IMPT task retry parameter is designed to work with the message:

DATABASE IN USE

When the database is in use as the IMPT task tries to perform a transaction, the IMPT task tries again based on the retry parameter. If the retry parameter has not been specified, the IMPT task tries to complete the transaction three times. If unsuccessful, the transaction is not performed and the incoming data record is written to the error log file.

Managing the Import Process

When a data import task (such as the IMPT or RCVR task) is running, the Data Import processing dialog box is displayed. This dialog box reports processing status messages, indicates the progress of the current import task, and displays the following transaction processing statistics:

Type	Description
Completed transactions	The number of transactions successfully imported by this task
Rejected transactions	The number of transactions which were not able to be imported by this task
Cycle	The number of times the task has processed the current incoming data file (for continuous import processes using the Loop Back to Retry parameter)
Retry	The number of transactions that were not processed because the database was in use (transactions will be retried in accordance with the Loop Back to Retry parameter)

If rejected transactions are reported, check the incoming data file for errors such as incorrectly formatted data. When the import task is complete, the dialog box closes. The next task in the batch process, if any, starts automatically.

If you want to monitor an import task while it is running, or troubleshoot an import macro, you have several options to choose from.

View Transaction Processing

Normally, transaction processing is hidden during import. However, you can watch transactions as they are imported through data entry screens by selecting the **Show transaction processing on screen** checkbox. This option allows you to monitor incoming data as it is imported.

Pause the Import Process

If an import macro is not executing as you would expect it to, you may want to investigate the problem. Press P at any time to suspend import processing, then select the **Show transaction processing on screen** checkbox to watch transactions being processed.

While the import process is paused, you can:

- press SPACE BAR to execute the macro one keystroke at a time
- press C to continue the import process where it left off
- press S to stop the import process entirely

Step Through the Import Process

Stepping through the import process can help you isolate and correct a macro error. Each time you press SPACE BAR, the next keystroke in the macro is executed. You can monitor each keystroke if the **Show transaction processing on screen** checkbox is selected.

Note: You can start an import task in step mode by using the S parameter to the IMPT task.

Stop the Import Process

Press S at any time to discontinue import processing. The import task is stopped and the BEXE screen is refreshed.

Troubleshooting

See "Troubleshooting" for more information on tracking down import problems.

Note: If you are using the Screen Customizer (SCRM) module, you cannot step through an import macro or view transaction processing on screen. For more information, see "Using Data Import with Screen Customizer" in the Fourth Shift Basics manual.

Error Log File Analysis

Two log files are produced in the MFGSYS directory during batch process execution:

- batch process log file (LOG.xxx) where xxx is the **Process Id**
- IMPT error log file (IMPORT.ERR)

Check both log files after executing the batch process. See "Data Import Error Messages" for an explanation of error messages for the IMPT task. Also see "Using Error Log Files" for information on how to use the IMPORT.ERR file as an incoming data file.

System Recovery

The daily operation of the system and system recovery using the data import features is different based on your system. See the "Recovery" topic in the SQL Server Administration and Utilities manual for more information.

Using Incoming Data Files

Incoming data files are located in the MFGSYS directory and have a DAT file extension. Data Import macros, predefined or custom, work directly with CDF (comma delimited format) data files. The TMPT (Template Maintenance) screen must be used when the incoming data file is in any other file format.

Note: Incoming data files cannot be named ARCHIVE.FIL. The ARCHIVE.FIL file name is reserved for the audit trail file used to record each Fourth Shift transaction.

Literals

To initialize a field using a literal, use the TMPT screen to specify the characters in the **Literal/ Table Name** field with a **Field Type** = L (literal). Once the data is imported, the field can be edited on a per-record basis.

When a literal is used for an incoming data field, the field does not have to be added to the incoming data record. The **Required Macro Length** must be equal to the length of the literal.

Note: If you need a literal more than eight characters in length, specify a table for the field and use the Data Conversion Table window to identify a **Converted Value** for an **Incoming Value** = * (wildcard = all fields).

Blank Lines

Blank lines in an input record is considered a single line comment record. It is counted as an input line, but is skipped over and not processed.

Null Fields

A null field is an empty field in a data record. A null field in an incoming data record signals the system to tab through the field and leave the field unchanged.

Null fields are used to complete a template when there is no existing relationship between an incoming **Fld#**/**Offset** and a required macro **Field#**. One null field added to the end of incoming records can relate to any number of required macro fields. See "A Dictionary Example" in the Data Import Dictionary for more information.

Clearing Fields

The F1 key is used to clear a field. **When an @ appears as a single character input field, the associated field on the screen is cleared as if the F1 key was pressed on the keyboard.**

To delete existing information in an alphanumeric (or date) field, use "@". For example:

```
"BILLRV", "DA1800", "S200", "FAB", 010, "@", "02"
```

To delete existing information in a numeric field, use @. For example:

```
"BILLRV", "DA1800", "S200", "FAB", 010, @, "02"
```

If you use the @ symbol as part of a multiple character string, the field information will not be deleted.

Using Error Log Files

An error log file is a reject suspense file for transactions that were unable to be processed by the IMPT or RCVR task. Error log files are created in the MFGSYS directory. The default error log file created by the IMPT task is IMPORT.ERR. The error log file created by the RCVR task is RCVR.ERR.

An error log file contains rejected records and associated error messages. Error messages are identified as comments—starting with /* and ending with */. Any lines between the /* and */ pair are ignored by the IMPT and RCVR tasks. In this way, an error log file can be renamed and used as the incoming data file once all errors have been analyzed.

For example, an IMPORT.ERR file could look similar to this:

```
/*
User Id: AAA Task: IMPT Job history log.
START DATE: 7/27/1999
START TIME: 020:41:49
STARTING RECORD #:      1
*/
/*
LAST FIELD # ENTERED:      3
RECORD #:      1
COMPARISON FIELDS DO NOT MATCH
*/
DA1800          S200 FAB  010 02
/*
LAST FIELD # ENTERED:      3
RECORD #:      2
COMPARISON FIELDS DO NOT MATCH
```

```
*/
DA1800          T150 FAB  010 02
```

Using this IMPORT.ERR file as an incoming data file, the IMPT task actually recognizes two lines as incoming data records:

```
DA1800          S200 FAB    010 02
DA1800          T150 FAB    010 02
```

All other lines in the IMPORT.ERR file are ignored. Remember to rename the error log files before using them as incoming data files.

Data Conversion

Use the Data Conversion Table window, opened from the TMPT screen, to list specific field values to be converted during the import process. One field format definition can have one or more field value conversions. As the incoming data records are processed, the table is referenced for each field having a table identified as part of its field format definition.

When a matching value is found for a field, the field is converted as specified. When a matching value is not found for a field, the incoming data record is written to the error log file (MFGSYS\IMPORT.ERR).

and the record is not processed. Conversion techniques include the use of specific characters, wildcards, ranges and field reconfiguration.

Note: Date conversions are performed automatically based on the **Field Type** entered on the TMPT screen. See "Date Conversions" in the TMPT (Template Maintenance) screen documentation for more information on date formats handled by the data import features.

Performing Character Conversions

Specific characters can be imported as different characters such as:

Incoming Value	Converted Value	Import Result
A	B	Incoming A fields become Bs; does not convert As within a field
123-456-789	321-456-789	Converts account number to new number

Using Wildcards

"Wildcard" characters (* and ?), similar to those in your operating system commands, allow you to convert information based on a specified group of values. For example:

Incoming Value	Converted Value	Import Result
123*	123A	Substitute 123A for Incoming Value beginning with 123.

123?5	123A5	If the first three characters of Incoming Value are 123 and the last character is 5, convert whatever value is in the fourth position of Incoming Value to A.
AA???	BB???	If the first two characters of Incoming Value are AA followed by three characters, change the first two characters to BB and import the Incoming Values for the remaining three characters as is.
WC?R?ABCD	WC?S?123	If the data contains a reserved character like a brace ([]), use a wildcard in place of the reserved character.
*	AA	Use default value AA for all Incoming Values .
*	*	If no table match is found, import Incoming Value as is.

Note: If the * wildcard is not used to end a table to cover the "no match found" situation, the incoming data record is rejected when a match is not found.

Using Wildcards for Reserved Characters

Several characters reserved for specific uses in data conversion syntax could also be used in your Fourth Shift data. For example, workcenter identifiers are defined using the square brace characters such as WC[R]123. These brace characters are used in data conversion to identify characters groups to be reconfigured. See Reconfiguring a Field for more information on characters groups.

In order to use the data conversion features when one of these reserved characters are part of your data, use wildcards in place of the reserved character.

For example, to change a work center identifier from one value to another, such as:

Starting Data	Desired Conversion
WC[R]ABCD	WC[S]123

Use the following data conversion values:

Incoming Value	Converted Value
WC?R?ABCD	WC?S?123

Specifying a Range

A range of **Incoming Values** can be converted to a specific **Converted Value** using two consecutive field value definitions as follows:

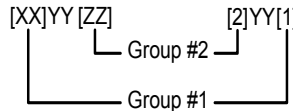
	Incoming Value	Converted Value
line 1:	beginning value	{RANGE}
line 2:	ending value	value for range

The use of **{RANGE}** signals a range and the second line defines the conversion value. A range is determined by an ASCII character sort. Be certain to zero-fill numbers, where necessary, to specify the actual range desired. For example:

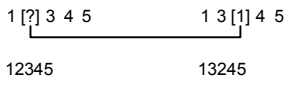
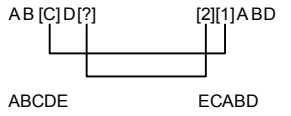
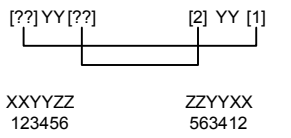
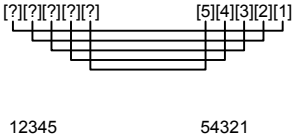
Incoming Value	Converted Value	Import Result
001	{RANGE}	Convert any value between 1 and 500 to 250
500	250	

Reconfiguring a Field

Up to nine groups of characters within a field can be identified to change the character order during import. Brackets are used to identify the character groups as follows:

<u>INCOMING VALUE</u>	<u>CONVERTED VALUE</u>	<u>IMPORT RESULT</u>
[XX]YY [ZZ]	[2]YY[1]	ZZYYXX
		

The brackets in the **Incoming Value** field are placed around the characters to be moved. The brackets in the **Converted Value** field are placed in the new position desired. The number within the brackets in the **Converted Value** field identifies the group order in the **Incoming Value**: [1] = the first character group, [2] = the second character group and so on. Conversion possibilities are endless, especially with the use of wildcards. For example:

<u>INCOMING VALUE</u>	<u>CONVERTED VALUE</u>	<u>IMPORT RESULT</u>
 1 [?] 3 4 5 12345 1 3 [1] 4 5 13245		Move whatever is in the second position in INCOMING VALUE to the third position of CONVERTED VALUE.
 AB [C] D [?] ABCDE [2] [1] A B D ECABD		If the character in the third position is a C, move to second position of CONVERTED VALUE. Move whatever is in the fifth position of INCOMING VALUE to the first position of CONVERTED VALUE.
 [?] YY [?] XYYZZ 123456 [2] YY [1] ZZYYXX 563412		Move the characters in the first group to the end of the CONVERTED VALUE. Move the characters in the second group to beginning of the CONVERTED VALUE.
 [?][?][?][?] 12345 [5][4][3][2][1] 54321		Change position of all characters in a five position character string.
[???]* 12345	ZZ[1] ZZ123	Move first three characters to end of CONVERTED VALUE ZZ.
[444]* 44445	ZZ[1] ZZ444	If the first three characters are 444, move 444 to end of CONVERTED VALUE ZZ.
[???][???] 123456	[2][1] 456123	For a six-character INCOMING VALUE, transpose first three characters with second three characters.

The bracket characters ([and]) cannot be used as literal values in data conversion tables.

Initializing a Database

You can use the data import features to initialize your database. If you do this, it is recommended that you use single-user mode.

Special attention must be paid to the use of default values in the system. Default values can be established for:

- Default-Item
- Default-Vendor
- Default-Customer
- Default-Order (manufacturing, purchase and customer)
- Default-Invoice
- Default-CR-Memo

- Default-Walk-In

These values can be used when entering groups of similar data. This default information is stored in the system if not changed during entry. Archive records reflect the values stored.

For example, when entering items, ITEM MASTER default values from the following screens are all stored for the item:

- ITMB
- Item Master Detail
- Item Master Planning Detail
- Item Lot Trace and Serialization Detail
- ITMC

Be certain to refer to the **Setting Up the Module** section in the related manuals before entering data in a batch mode.

Continuous Processing

The IMPT and RCVR tasks can be executed in a continuous mode so that information from other systems, such as bar code readers, can continually update the database. This continuous mode of processing is specified using the Continuous Read parameter. Using the Continuous Read (R) parameter, you indicate how often the IMPT task checks the input data file to see if additional transactions have been added after an end of file has been encountered.

Continuous read processing is expressed in terms of seconds to wait at the end of file before checking for additional transactions.

The IMPT/RCVR task stays on the current screen, waits the specified number of seconds and then checks for a new transaction. If new transactions have not been added to the end of the input data file, the IMPT/RCVR task returns to the Fourth Shift Master Menu and continues to check for new transactions every specified number of seconds. In this way, other users are not locked out of entry screens.

Additional Parameters Available

Continuous processing can also be used with other features available for the IMPT and RCVR tasks including the following.

Processing Status Field Update (P)

You may specify a field in each input record to be updated during the IMPT/RCVR task processing. This field, called the **Processing Status** field, indicates whether an input transaction was processed successfully or encountered processing problems. This parameter allows you to reprocess an input data file without determining the specific starting record number and preventing the duplication of transactions. If you are using the Loop Back to Retry Transactions parameter, the processing status field specification is included with the O parameter and the P parameter is ignored.

The Processing Status field defaults to Required Macro Field 5 when using the RCVR task. This field must be initialized to "0" (zero enclosed in quotes) before processing begins.

If you are using a template, a field in the incoming data file must be defined in the template for the Processing Status field so that the import process has a place to update.

Required Macro Field 5 (although not normally needed) is recommended because it is the same field used in archive records formatted for use by the RCVR (System Recovery) task. However, any Required Macro Field can be specified as long as that field is not being used to import data.

For example, the template definition on the TMPT screen would look similar to the following when using Required Macro Field 5 to update the value in field 9 of the incoming data file.

Fld#/Offset	Length	Reqd Fld#	Reqd Length	Fld Type
9		5	1	A

Multi-User Retry (U)

You may specify the number of times the system will try to complete a transaction when a database in use (locked) condition is encountered.

- If the transactions must be imported in chronological order, the recommended setting for the U parameter is a high number of retries (up to the maximum 999). Also set the Error Level parameter to E2 to stop processing at the first error.
- If the order of transaction processing is not relevant and transactions should be processed as rapidly as possible, the recommended setting for the U parameter is 1 with the Loop Back to Retry parameter specified.

Loop Back to Retry (O)

The Loop Back to Retry parameter causes the input data file to be reprocessed when an end of file indicator is encountered and one or more records were left unprocessed in the input file due to database-in-use conditions. This allows you to run a continuous import process while other users are accessing the system and could be using the same data elements included in the import process. The input data file will be reprocessed beginning with the first record that failed because of the database-in-use condition during the previous attempt.

A field number must be included with this parameter to indicate which field value should be overwritten with the Processing Status during processing. The processing status field must be defined in templates used during with the IMPT task.

The Process Status Indicator is updated when the O parameter is specified, even in the P parameter (Processing Status Field Update) is not specified.

Processing Status Indicator

You may specify a field in each input record which will be updated during the IMPT/RCVR task processing. This field, called the **Processing Status** field, indicates whether an input transaction was successfully processed or encountered a processing problem.

Note: Processing Status is only updated if the P (Processing Status Field Update) or O (Loop Back to Retry) status is specified.

The **Processing Status** field contains one of the following statuses:

Status	Description
0	Initial value; transaction needs to be processed and values imported using the IMPT or RCVR task.

1	Transaction did not process because data needed was in use; transaction will be reprocessed on the next pass of the input data file.
2	This transaction failed because a "business rule" was not passed. A prior record in the input data file has been marked with Processing Status = 1. This transaction will be reprocessed on the next pass of the input data file.
3-7	Reserved for future use.
8	This transaction failed because a "business rule" was not passed and there were no prior records marked with Processing Status = 1. This transaction will be skipped on all subsequent passes of the input data file and not reprocessed.
9	Transaction processed successfully. Transaction is skipped on all subsequent passes of the input data file.

Stopping Continuous Processing

Continuous processing mode waits for more transactions to be entered after reaching the end of an incoming data file. This waiting will continue until you stop the processing mode.

There are several ways to stop continuous processing.

- Press **S**. The task stops file processing. The next task in the batch process is then executed.
- Add a transaction to the incoming data file which contains the string **TermIMPT** (when running the IMPT task) or **TermRCVR** (when running the RCVR task).

Parameter Examples

Example 1: Continuous using Multi-User Retry

The following examples shows the RCVR task parameters:

- input data file SAMPLE.TXT
- run in continuous mode checking every 300 seconds after reaching the end of the file
- multi-user retry (database in use) 60 times

```
01 RCVR ISAMPLE.TXT R300 U60
```

Example 2: Continuous with Processing Status Enabled

The following examples shows the IMPT task parameters:

- macro filename INVM.MAC
- input data file SAMPLE.TXT
- template name ITMBPLAN
- run in continuous mode checking every 300 seconds after reaching the end of the file
- multi-user retry (database in use) 60 times
- updating the Processing Status field (field 5) during processing

```
01 IMPT MINVM.MAC ISAMPLE.TXT TITMBPLAN R300 U60 P5
```

Example 3: Continuous using Loop Back and Retry

The following examples shows the IMPT task parameters:

- macro filename INVM
- input data file SAMPLE.TXT
- template name ITMBPLAN
- run in continuous mode checking every 300 seconds after reaching the end of the file
- multi-user retry (database in use) 999 times
- loop back if needed and update the Processing Status field (field 5) during processing

```
01  IMPT  MINVM  ISAMPLE.TXT  TITMBPLAN  R300  U999  05
```

Data Import Troubleshooting

Macros are extremely detailed and any kind of typing, logic or field error can make the difference between a successful and an unsuccessful result. Here are some things to consider if you run into a problem.

The Log Files

1. Print out and analyze the LOG.XXX file for the batch process you just ran.
2. Print out and analyze the IMPORT.ERR file produced by the IMPT task.

See "Data Import Error Messages" for an explanation of the error messages recorded in the log files.

The Incoming Data File

Check your incoming data file for errors. Errors can include:

- **Ln#** (line number) and **Seqn** (sequence number) fields must be zero-filled in order for a comparison to be successful.

See "Troubleshooting" in the Data Import Macro Basics manual for more information on locating problems in macros.

Import Setup

1. Check the batch process set up to run the IMPT task.
Make sure you specify a template if you are using one.
2. Check the template and table(s) created using the TMPT (Template Maintenance) screen and the Data Conversion Table window, if necessary.

Import Execution

Rerun the IMPT task. See "Managing the Import Process" for information on options including:

- stopping the process
- controlling the process from the keyboard a single keystroke at a time

Data Import Error Messages

Comparison fields do not match

A cut and compare was executed using the **{REJECT}** command as the comparison result. Comparison values are case sensitive. Verify that the comparison value is listed in uppercase format. Review the incoming data file for record layout or value problems before trying again.

Configuration Error: RCVR.IDX and RCVR.MAC files do not match, Index = nnnnn

The .MAC file does not match its associated .IDX file. This error is caused by an entry in the .IDX file that is not pointing to a '## GROUP START ##' record in the .MAC file. This error usually occurs when you copy a new .IDX file without its associated .MAC file or copy a new .MAC file without its associated .IDX file.

Ending record parameter must not be less than starting record number parameter

You have entered a range of records to process and have entered an ending value that would be located before the starting record in the incoming data file. Review your range values and correct the range before trying again.

Error message from Screen 'xxxx'

This is an information message which is followed by the message displayed from the screen when the data import process was running. Normally, this message indicates a problem which prevented the data from processing successfully. Review any other messages following this one.

Error reading incoming data file, error = 999

The incoming data file does not exist or you may not have rights to open the file. Verify the file exists in the location specified and that you have rights to open the file.

Error reading macro index file

The RCVR.IDX file needed for the RCVR task cannot be accessed. Verify that the file exists in the MFGSYS\I directory (or the location specified by the **ImportFiles** configuration variable) and that you can access the file.

Error reading Table file 'xxxxxxx', error = 999

The table (*.TBL) file does not exist or you may not have rights to open the file. Verify the file exists in the location specified and that you have rights to open the file.

Error reading Template file, error = 999

The template (*.TPL) file does not exist or you may not have rights to open the file. Verify the file exists in the location specified and that you have rights to open the file.

Extended text page number is invalid

You have attempted an extended text transaction, but the page number specified in the transaction does not exist for an insert, change, or delete for this message.

Field Entry Error - Illegal Character at Screen Coordinates 'row','column'

You have made an invalid entry in a field. This error is probably caused by one of the following actions:

- An invalid character was entered. For example, you can enter only an M, B or S in the **Make/Buy** field on the ITMB screen. Entering a character other than M, B or S results in this error.
- A field length restriction was violated. For example, you may have entered a nine character value in a field restricted to eight characters.
- A macro tried to access an invalid field. For example a macro might try to use a tab when already in the last open field on a screen.

Verify the entry rules for the field and try again.

Found <command> Macro Command in {REJECT} message

A macro command has been included in the custom {REJECT} message text. Macro commands are not permitted in the {REJECT} message text. Remove the text between the {REJECT} and {ENDDDEF} commands.

Import End of Job due to TERMRCVR Incoming Data command

The reserved word TERMRCVR was found in your incoming data file. This reserved word indicates that the import process should stop. No error is indicated by this message. No further action is needed for this message.

Incoming Data field format error, line = 999

An error exists in the Incoming Data file. A number of situations can cause this error. Here are two examples:

- The line is missing one or more closing quotes. Quotes must be entered in pairs or the import process cannot locate the end of a field. Update the line and retry the import process.
- One or more of the fields in the incoming data file line contain unexpected data. Special characters such as EOF (End of File) markers are placed in ASCII files by some editors. Review the Incoming Data file with an ASCII editor like NotePad to detect and remove special characters.

Incoming Data Record is missing Transaction User Id

Several fields are required in an incoming data record. These include a user identifier, transaction date and transaction time. The user identifier is missing or not in the field where expected. Review the incoming data file, make the necessary corrections and try again.

Incoming Data Record contains invalid Transaction Date

Several fields are required in an incoming data record. These include a user identifier, transaction date and transaction time. The transaction date is missing or not in the field where expected. Review the incoming data file, make the necessary corrections and try again.

Incoming Data Record contains invalid Transaction Time

Several fields are required in an incoming data record. These include a user identifier, transaction date and transaction time. The transaction time is missing or not in the field where expected. Review the incoming data file, make the necessary corrections and try again.

Incoming data file name not specified as a parameter

You have failed to identify the incoming data file to be used. An incoming data file name is a required entry in the **Parameters** field. Identify an incoming data file before trying again. The

default path for an incoming data file is MFGSYS. The default file extension for an incoming data file is DAT.

Invalid alternate path name parameter

You have entered the A parameter which indicates an alternate location to use for locating the template and table files but have not specified the alternate location. Both the A parameter and the alternate location must be specified in order to use the parameter. Enter the alternate location before trying again.

Invalid continuous read wait seconds parameter

You have entered a wait seconds value using the R parameter. The R parameter requires a value in the range 1-999. Correct the wait value before trying again.

Invalid ending record number parameter

You have specified an ending record number for the import process but the value uses an incorrect format. Valid values are 1-9,000,000. Correct your record number value before trying again.

Invalid error level parameter

You have entered an error level option that is not valid. Valid error levels are 1, 2 and 3. Correct the error level option before trying again.

Invalid incoming data file name parameter

You have failed to identify the incoming data file to be used for the IMPT task. An incoming data file name is a required entry in the **Parameters** field. Identify an incoming data file before trying again. The default path for an incoming data file is MFGSYS. The default file extension for an incoming data file is DAT.

Invalid log file name parameter

You have identified a log file which cannot be located. Check the path, file name and file extension for entry errors before trying again. The default path for a log file is MFGSYS (or the location specified by the **AuxDataFiles** configuration variable).

Invalid macro file name parameter

You have failed to identify the macro file to be used. A macro file name is a required entry in the **Parameters** field. Identify a macro before trying again. The default path for a macro file is MFGSYS\I (or the location specified by the **ImportFiles** configuration variable). The default file extension for a macro is MAC.

Invalid multi-user retry count parameter

You have entered the multi-user retry option using the U parameter but have entered an invalid retry count value. The U parameter requires a value in the range 1-999. Correct the retry value before trying again.

Invalid or missing Incoming Data format Template record

A problem exists in the template file. The required internal file structure is incorrect. Recreate the template file using the TMPT screen and retry the data import process.

Invalid process stamp field number in loop back parameter

You have selected to update the processing field when the transaction is processed. This field identifies the result of the import process for this transaction. The valid values for this field are 1-250. Correct the processing field value before trying again.

Invalid process stamp field number in parameter

You have selected to update the processing field when the transaction is processed. This field identifies the result of the import process for this transaction. The valid values for this field are 1-250. Correct the processing field value before trying again.

Invalid starting record number parameter

You have specified a starting record number for the import process but the value uses an incorrect format. Valid values are 1-65535. Correct your record number value before trying again.

Invalid template file name parameter

You have identified a template file which cannot be located. Check the path, file name and file extension for entry errors before trying again. The default path for a template file is MFGSYS\I (or the location specified by the **ImportFiles** configuration variable). The default file extension for a template is TPL.

Invalid TERMIMPT Incoming Data command

The data import process is running and the Incoming Data file contains a TERMIMPT command. This error is caused by one of the following:

- The RCVR task is being used for the continuous data import process. The TERMRCVR command must be used to stop the continuous RCVR process.
- The IMPT task is being used but the task is not running in continuous import mode. Press S to stop the running data import process.

Invalid TERMRCVR Incoming Data command

The data import process is running and the Incoming Data file contains a TERMRCVR command. This error is caused by one of the following:

- The IMPT task is being used for the continuous data import process. The TERMIMPT command must be used to stop the continuous IMPT process.
- The RCVR task is being used but the task is not running in continuous import mode. Press S to stop the running data import process.

Length of incoming data file record exceeds maximum allowed

The incoming data record (line in the data file) cannot exceed 10,240 characters. If you are using a custom macro, consider using several macros and templates to accomplish your import for all the fields listed.

Length of Template file record exceeds maximum allowed

The template file cannot exceed 2048 characters. Review the template and make the necessary changes.

Macro character chosen is reserved for a command

The custom macro is assigned a name that is the same as a reserved macro command. You may have forgotten to insert a macro name before a screen cut or other macro operation. Verify that your macro begins with a name that is not a reserved character or name.

Macro file name not specified as a parameter

You have failed to identify the macro file to be used. A macro file name is a required entry in the **Parameters** field. Identify a macro file before trying again. The default path for an incoming macro file is MFGSYS\I (or the location specified by the **ImportFiles** configuration variable). The default file extension for an incoming data file is MAC.

Macro Logic Error: {BACKUP} in {IF} statement would terminate macro playback

The custom macro contains an error. The {BACKUP} command is used within an {IF} statement. Completing this sequence of steps would cause the macro to end unexpectedly. Review and update the screen cut coordinates and retry the data import process.

Macro Syntax Error: {DISONL} end cut position less than start position, line = 999

The custom macro contains a syntax error. The coordinates for the screen cut are not valid.

The {DISONL} command consists of the following parts:

- {BEGDEF} and {ENDDEF} commands at the beginning and end of the screen cut
- variable identifier for the screen cut, such as {cut}
- the NO RETURNS clause to indicate that no carriage return is added at the end of the cut
- search mode (SEARCH FOR or ABSOLUTE)
- optional mode criteria such as PACK or TRIM
- screen coordinates in the range 01, 01 through 25, 80

For example, a complete command sequence would look similar to the following:

```
{BEGDEF}{DISONL}{cut}NO RETURNS,ABSOLUTE & PACK, 01, 11, 01, 22,{ENDDEF}
```

Review and update the screen cut coordinates and retry the data import process.

Macro Syntax Error: {DISONL} invalid numeric character in cut position, line = 999

The custom macro contains a syntax error. The coordinates for the screen cut are not valid.

The {DISONL} command consists of the following parts:

- {BEGDEF} and {ENDDEF} commands at the beginning and end of the screen cut
- variable identifier for the screen cut, such as {cut}
- the NO RETURNS clause to indicate that no carriage return is added at the end of the cut
- search mode (SEARCH FOR or ABSOLUTE)
- optional mode criteria such as PACK or TRIM
- screen coordinates in the range 01, 01 through 25, 80

For example, a complete command sequence would look similar to the following:

```
{BEGDEF}{DISONL}{cut}NO RETURNS,ABSOLUTE & PACK, 01, 11, 01, 22,{ENDDEF}
```

Review and update the screen cut coordinates and retry the data import process.

Macro Syntax Error: {ENDDEF} before {BEGDEF}, line = 999

The custom macro contains a syntax error. There is an unmatched set of beginning and ending macro commands. Verify that each macro section contains a {BEGDEF} and {ENDDEF} command.

Macro Syntax Error: Command used as macro identifier

The custom macro contains a syntax error. The {BEGDEF} command is immediately followed by another reserved command such as {IF}. A macro must have a name following the {BEGDEF} command, such as {MACRO1}.

Macro Syntax Error: Identifier used as operand in {IF} statement has not been defined

The custom macro contains a syntax error. A conditional statement contains an undefined identifier. This can be caused by a number of situations such as:

- the identifier does not exist in the macro file
- the identifier is spelled differently from the name used previously

Review and update the macro before trying again.

Macro Syntax Error: Invalid {BEGDEF} placement, line = 999

The custom macro contains a syntax error. A second {BEGDEF} command is within an existing macro before the {ENDDEF}. Only one set of {BEGDEF}/{ENDDEF} commands can be listed together. Review and update the macro.

Macro Syntax Error: Invalid {ENDDEF} placement, line = 999

The custom macro contains a syntax error. A second {ENDDEF} command is within an existing macro. Only one set of {BEGDEF}/{ENDDEF} commands can be listed together. Review and update the macro.

Macro Syntax Error: Invalid conditional operator in {IF} statement, line = 999

The custom macro contains a syntax error. A conditional operator used within the macro is not valid. See the Data Import Macro Basics manual for a list of the conditional operators available. Review and update the macro.

Macro Syntax Error: Invalid flow control directive in {IF} statement, line = 999

The custom macro contains a syntax error. The conditional statement contains a series of commands that are not valid. Verify that the **call**, **backup** or **jump** directive is used correctly.

Macro Syntax Error: Invalid macro id in {DISONL} statement, line = 999

The custom macro contains a syntax error. The DISONL command must be followed by a valid macro name which will be used later in the macro as a variable to the screen cut contents. You may be using a reserved word or a macro name longer than six characters. Correct the macro name before trying again.

Macro Syntax Error: Invalid macro identifier

The custom macro contains a syntax error. A macro name must be enclosed in braces. You may be using a reserved word or a macro name longer than six characters. Update the macro name and try again.

Macro Syntax Error: Invalid nested macro definition, line = 999

The custom macro contains a syntax error. There is an unmatched set of beginning and ending macro commands. Verify that each macro section contains a {BEGDEF} and {ENDDEF} command. Another macro cannot be started before an {ENDDEF} unless you are defining screen cut coordinates using the {DISONL} command.

Macro Syntax Error: Invalid operand in {IF} statement, line = 999

The custom macro contains a syntax error. A conditional statement contains an undefined identifier or invalid field reference. This can be caused by a number of situations such as:

- the identifier does not exist in the macro file
- the identifier is spelled differently from the name used previously

Review and update the macro before trying again.

Macro Syntax Error: Invalid screen cut mode in {DISONL} statement, line = 999

The custom macro contains a syntax error. The information following the NO RETURNS statement is not a valid mode (ABSOLUTE or SEARCH FOR) for the screen cut or it contains invalid clauses for these statements.

The {DISONL} command consists of the following parts:

- {BEGDEF} and {ENDDEF} commands at the beginning and end of the screen cut
- variable identifier for the screen cut, such as {cut}
- the NO RETURNS clause to indicate that no carriage return is added at the end of the cut
- search mode (SEARCH FOR or ABSOLUTE)
- optional mode criteria such as PACK or TRIM
- screen coordinates in the range 01, 01 through 25, 80

For example, a complete command sequence would look similar to the following:

```
{BEGDEF}{DISONL}{cut}NO RETURNS,ABSOLUTE & PACK, 01, 11, 01, 22,{ENDDEF}
```

Review and update the screen cut coordinates and retry the data import process.

Macro Syntax Error: Invalid screen cut position in {DISONL} statement, line = 999

The custom macro contains a syntax error. The coordinates for the screen cut are not valid.

The {DISONL} command consists of the following parts:

- {BEGDEF} and {ENDDEF} commands at the beginning and end of the screen cut
- variable identifier for the screen cut, such as {cut}
- the NO RETURNS clause to indicate that no carriage return is added at the end of the cut
- search mode (SEARCH FOR or ABSOLUTE)

- optional mode criteria such as PACK or TRIM
- screen coordinates in the range 01, 01 through 25, 80

For example, a complete command sequence would look similar to the following:

```
{BEGDEF}{DISONL}{cut}NO RETURNS,ABSOLUTE & PACK, 01, 11, 01, 22,{ENDDEF}
```

Review and update the screen cut coordinates and retry the data import process.

Macro Syntax Error: Invalid symbol expression, line = 999

The custom macro contains a syntax error. A symbol used within the macro is not valid. See the Data Import Macro Basics manual for a list of the symbols available. Review and update the macro.

Macro Syntax Error: Name exceeds maximum length, symbol = x, line = 999

The custom macro contains a syntax error. The macro name is greater than 6 characters. Reduce the number of characters in the macro name.

Macro Syntax Error: Nested macro definition must be {DISONL}, line = 999

The custom macro contains a syntax error. There is an unmatched set of beginning and ending macro commands. Verify that each macro section contains a {BEGDEF} and {ENDDEF} command. Another macro cannot be started before an {ENDDEF} unless the macro is for a screen cut definition.

Macro Syntax Error: Reached end of macro file before {ENDDEF}, line = 999

The custom macro contains a syntax error. There is an unmatched set of beginning and ending macro commands. Verify that each macro section contains a {BEGDEF} and {ENDDEF} command.

Macro Syntax Error: Required field missing in {DISONL} statement, line = 999

The custom macro contains a syntax error. The coordinates for the screen cut are not valid, you are missing the search mode (SEARCH FOR or ABSOLUTE) or an invalid number of clauses is entered.

The {DISONL} command consists of the following parts:

- {BEGDEF} and {ENDDEF} commands at the beginning and end of the screen cut
- variable identifier for the screen cut, such as {cut}
- the NO RETURNS clause to indicate that no carriage return is added at the end of the cut
- search mode (SEARCH FOR or ABSOLUTE)
- optional mode criteria such as PACK or TRIM
- screen coordinates in the range 01, 01 through 25, 80

For example, a complete command sequence would look similar to the following:

```
{BEGDEF}{DISONL}{cut}NO RETURNS,ABSOLUTE & PACK, 01, 11, 01, 22,{ENDDEF}
```

Review and update the screen cut coordinates and retry the data import process.

Macro Syntax Error: Too many fields in {DISONL} statement, line = 999

The custom macro contains a syntax error. The coordinates for the screen cut are not valid, you are missing the mode (SEARCH FOR or ABSOLUTE) or an invalid number of clauses is entered.

The {DISONL} command consists of the following parts:

- {BEGDEF} and {ENDDEF} commands at the beginning and end of the screen cut
- variable identifier for the screen cut, such as {cut}
- the NO RETURNS clause to indicate that no carriage return is added at the end of the cut
- search mode (SEARCH FOR or ABSOLUTE)
- optional mode criteria such as PACK or TRIM
- screen coordinates in the range 01, 01 through 25, 80

For example, a complete command sequence would look similar to the following:

```
{BEGDEF}{DISONL}{cut}NO RETURNS,ABSOLUTE & PACK, 01, 11, 01, 22,{ENDDEF}
```

Review and update the screen cut coordinates and retry the data import process.

Macro Syntax Warning: Should be 'NO RETURNS' clause in {DISONL} statement, line = 999

The custom macro contains a syntax error. The screen cut is not valid and is missing the No Returns clause.

The {DISONL} command consists of the following parts:

- {BEGDEF} and {ENDDEF} commands at the beginning and end of the screen cut
- variable identifier for the screen cut, such as {cut}
- the NO RETURNS clause to indicate that no carriage return is added at the end of the cut
- search mode (SEARCH FOR or ABSOLUTE)
- optional mode criteria such as PACK or TRIM
- screen coordinates in the range 01, 01 through 25, 80

For example, a complete command sequence would look similar to the following:

```
{BEGDEF}{DISONL}{cut}NO RETURNS,ABSOLUTE & PACK, 01, 11, 01, 22,{ENDDEF}
```

Review and update the screen cut coordinates and retry the data import process.

Maximum nested macro reference limit exceeded, check for circular reference

The custom macro contains a logic error. References to other macros and CALL commands used to call other macros exceeds 25. Review the macro layout and rework the structure. Also verify you have not defined macro calls back into the same macro.

Maximum number of 99 pages of extended text allowed

You have attempted to insert a new page of extended text, but there are already 99 pages of extended text for this message.

Nested macro reference in {REJECT} message

You have combined several macro references together to form a "nested macro." Nested macro references are not allowed in the custom {REJECT} message text. Rewrite the text in the macro between the {REJECT} and {ENDDEF} commands.

No corresponding macro for this Incoming Data record

The macro id is not found in the macro file being used. Verify the macro file name. If a template is being used, verify the first field definition for the template identifies the correct macro.

No matching conversion mask in Table for incoming data field, xxxxxx, line = 999

The conversion table specified for this field does not contain a relationship for the field data in this line. A conversion is required for each data format unless the wildcard * is used in the table. Update the table before trying again.

Non-printable data found in {REJECT} message

There are non-printing characters in the custom {REJECT} message text. Only printable ASCII characters, certain ASCII formatting commands, {FLDnnn} references, and screen CUT references are permitted. Rewrite the text in the macro between the {REJECT} and {ENDDEF} commands.

Number of Template field specification records exceeds maximum allowed

The template specifies greater than 255 fields. No more than 255 fields can be specified. If your custom macro requires more than 255 fields, consider using several macros and templates to accomplish your import.

Numeric Incoming Data field too large for Template descriptor output field, line = 999

The numeric value in the incoming data is larger than the template definition. Either update the template or modify the incoming data.

Parameter x is listed more than once for this task

You have entered a parameter multiple times. If you are using CONT lines to continue parameters to another line, review all lines for duplicate parameters. Review and update your parameter entries.

{REJECT} message exceeds maximum length

The text in the custom {REJECT} message exceeds the maximum length of 256 characters. Rewrite the text in the macro between the {REJECT} and {ENDDEF} commands using abbreviations or other wording.

Size of macro body exceeds maximum allowed

The custom macro is larger than the maximum allowed. Break the macro into smaller sub-macros before trying again.

Starting record '999' does not exist in Incoming Data file

You have specified a starting record number greater than the last record number in the incoming data file. Check the parameters for entry errors before trying again.

Table file 'xxxxxxx': Contains consecutive {RANGE} converters, line = 999

The data conversion table contains more than one {RANGE} command. A range is specified by entering the starting value and {RANGE} on the first line, and the ending value of the range and

the conversion value on the second line. Only one {RANGE} command is used. Review your data conversion table and remove the extra {RANGE} command.

Table file 'xxxxxxx': Converter contains invalid sub-field end delimiter, line = 999, sub-field = xxxxxx

The data conversion table for the **Converted Value** contains invalid formatting for identifying resequencing character groups. A resequencing character group must use a set of bracket characters for each group and the groups cannot be nested. The bracket characters ([and]) cannot be used as literal values in data conversion tables. Review and update your conversion table before trying again.

Table file 'xxxxxxx': Converter contains invalid sub-field identifier, line = 999, sub-field = xxxxxx

The data conversion table for the **Converted Value** contains invalid formatting for identifying resequencing character groups. A resequencing character group must use a set of bracket characters for each group and the groups cannot be nested. The bracket characters ([and]) cannot be used as literal values in data conversion tables. Review and update your conversion table before trying again.

Table file 'xxxxxxx': Converter contains nested sub-field, line = 999, sub-field = xxxxxx

The data conversion table for the **Converted Value** contains invalid formatting for identifying resequencing character groups. A resequencing character group must use a set of bracket characters for each group and the groups cannot be nested. The bracket characters ([and]) cannot be used as literal values in data conversion tables. Review and update your conversion table before trying again.

Table file 'xxxxxxx': Converter missing ending sub-field delimiter, line = 999

The data conversion table for the **Converted Value** contains invalid formatting for identifying resequencing character groups. A resequencing character group must use a set of bracket characters for each group and the groups cannot be nested. The bracket characters ([and]) cannot be used as literal values in data conversion tables. Review and update your conversion table before trying again.

Table file 'xxxxxxx': Invalid record format, line = 999, 'xxxx'

The data conversion table file (*.TBL) is not in the expected format. The data conversion tables can only be modified using the Data Conversion Table window from the TMPT screen. Recreate the conversion table file and try again.

Table file 'xxxxxxx': Mask contains nested sub-field, line = 999, sub-field = xxxxxx

The data conversion table for the **Incoming Value** contains invalid formatting for identifying resequencing character groups. A resequencing character group must use a set of bracket characters for each group and the groups cannot be nested. The bracket characters ([and]) cannot be used as literal values in data conversion tables. Review and update your conversion table before trying again.

Table file 'xxxxxxx': Mask contains unbalanced sub-field delimiters, line = 999

The data conversion table for the **Incoming Value** contains invalid formatting for identifying resequencing character groups. A resequencing character group must use a set of bracket characters for each group and the groups cannot be nested. The bracket characters ([and]) cannot be used as literal values in data conversion tables. Review and update your conversion table before trying again.

Table file 'xxxxxxx': Mask missing ending sub-field delimiter, line = 999, sub-field = xxxxxx

The data conversion table for the **Incoming Value** contains invalid formatting for identifying resequencing character groups. A resequencing character group must use a set of bracket characters for each group and the groups cannot be nested. The bracket characters ([and]) cannot be used as literal values in data conversion tables. Review and update your conversion table before trying again.

Table file 'xxxxxxx': Missing end {RANGE} descriptor record

The data conversion table contains a {RANGE} command but no end of the range. A range is specified by entering the starting value and {RANGE} on the first line, and the ending value of the range and the conversion value on the second line. Both the start and end of the range must be specified. Review your data conversion table and add the ending range line.

Table file 'xxxxxxx': Record length exceeds maximum allowed

The data conversion table contains too many entries. The maximum length is 2048. Review the data conversion table to determine if you can use {RANGE} commands or wildcards to reduce the number of lines. You could also break the incoming data file into several parts and create different tables for each based on the data in each file.

Template record missing required fields, line = 999

You have defined a template for use by data import. The template must contain the complete set of fields required for the macro to function. See the Data Import Dictionary for more information about the required fields for this macro.

Terminated Due to {GETOUT} macro playback

The macro command GETOUT was encountered in a macro. This command when encountered automatically terminates the macro. Review the macro processing and data file if this is an unexpected result.

Terminated Due to 'TERMIMPT' Incoming Data Command

The reserved word TERMIMPT was found in your incoming data file. This reserved word indicates that the import process should stop. No error is indicated by this message. No further action is needed for this message.

Terminated Due to 'TERMRVCVR' Incoming Data Command

The reserved word TERMRVCVR was found in your incoming data file. This reserved word indicates that the import process should stop. No error is indicated by this message. No further action is needed for this message.

Terminated on Error

The task was unable to run to completion. For example, a file might not have been able to be opened. Check both the batch process log file and the import error log file to determine the cause of the error.

The parameter 'x' is not valid

You have entered a parameter that is not available for this task. Review and update your parameter entries.

Unable to format Incoming date field via Template descriptor, line = 999

The incoming data file field contains a date which cannot be interpreted by the template. Review the date format used and update the template before trying again.

Unable to open file = xxxxx, error = x

The file does not exist or you may not have rights to open the file. Verify the file exists in the location specified and that you have rights to open the file.

Unable to read macro file, error = 999

The macro file does not exist, you may not have rights to open the file or the file is not in ASCII format. Verify the file exists in the location specified, that you have rights to open the file and that the file can be opened with an ASCII editor such as Notepad.

Unable to start a macro or load group for this Incoming Data record

The macro for the incoming data file record id is not found in the macro file being used. Verify the macro file name and the data file. If a template is being used, verify the first field definition for the template identifies the correct macro.

Warning: Delete Transaction may have completed successfully, check screen to confirm

This transaction encountered a "data in use" condition on either the data included in the delete transaction or the previous item in the list being accessed after the delete is complete. Manually verify that the delete completed successfully before retrying the data import process.

Warning: Partial record detected at end of Incoming Data file

The incoming data file does not end with a carriage return/line feed character. This is generally caused by the editor used. Open the file with Notepad, save the file and try again.

IMPT – Data Import

The IMPT (Data Import) task inputs data into the system in a batch mode just as if you entered the data from a keyboard. The IMPT task can use the predefined macros provided or custom macros you have designed for specialized transactions.

The IMPT task and the RCVR (System Recovery) task perform the same type of data import processing but the IMPT task has several other options.

- **Macro File Specification.** Predefined and custom macros are defined in specific files. Each file contains a set of macros. See "Macro Filenames by Screen" under "Transaction Cross Reference Lists" in the Data Import Dictionary manual to identify the macro file needed. The set of macros you wish to use with a data import process can be identified when you are using the IMPT task.
- **Template Definition Use.** Your incoming data file may or may not be formatted in the field layout expected by the Data Import process. When the field layout does not exactly match the order or format expected by the Data Import process, a relationship or **template** must be defined between the incoming data and the format expected by the Data Import process. Templates are defined on the TMPT (Template Maintenance) screen.

With the IMPT task, you can specify the template relationship you wish to use. The RCVR task requires the same transaction record format that is used in the ARCHIVE.FIL file.

The template defines the relationship between the fields in your incoming data file and the "Required Macro Field" entry when necessary. This required field indicates the field number from the file layout expected by the Data Import process. The template will make the necessary substitution to the field position in your incoming data file.

Task Reference

Monitoring the Import Process

While the IMPT task is processing, the Data Import Processing dialog box is displayed. Use this dialog box to monitor the processing status of the task.

For information on using the options available on the dialog box, see "Managing the Import Process" in the System Control manual.

Data Import Errors

Two log files are produced in the MFGSYS directory during batch process execution:

- batch process log file (LOG.xxx) where xxx is the **Process Id**
- IMPT error log file (IMPORT.ERR)

Check both log files after executing the batch process. See the "Data Import Error Messages" in the **Using the Data Import Features** section of the System Control manual for more information.

Caution: If you need to run the IMPT task concurrently in more than one batch process, be sure to identify separate error log files for each process. Log files are specified using the L parameter. Log files are opened in exclusive mode so the content cannot be written to by more than one process at the same time.

Parameters

To request Data Import, enter the IMPT task as one of the sequenced tasks in a batch process.

Caution: If you need to run the IMPT task concurrently in more than one batch process, be sure to identify separate error log files for each process. The log files are opened in exclusive mode so the content cannot be written to by more than one process at the same time.

The IMPT task has the following parameters:

Task Parameter	Format	Entry Is...
1: Macro File & Incoming Data File	MX..X IX..X	Required
2: Record Number Range; Loop Back to Retry Transactions	F9..9 Z9..9 O999	Optional
3: Template File & Alternate Path	TX..X AX..X	Optional
4: Error Level; Log File; Update Process Status Field	E9 LX..X P999	Optional
5: Continuous Read; MU Retry; Step; View	R999 U999 S VX	Optional

File defaults are as follows:

Import File Type	File Extension	Configuration Variable	Standard Location
Converter	.OVL	ProgramFiles	MFGSYS\E
Error Log	.ERR	AuxDataFiles	MFGSYS
Incoming Data	.DAT	AuxDataFiles	MFGSYS
Macro	.MAC	ImportFiles	MFGSYS\I
Data Conversion Table	.TBL	ImportFiles	MFGSYS\I
Template	.TPL	ImportFiles	MFGSYS\I

Parameter 1: Macro and Incoming Data Files

Macro File. Identify the macro to be used for importing data. The macro can be a predefined transaction or a custom macro. The required macro file extension of MAC is used as a default. The required path of MFGSYS\I is used as a default. See "Transaction Cross Reference Lists" in the Data Import Dictionary manual to locate the necessary macro filename.

Incoming Data File. Identify the incoming data file name. If no file extension is specified, a DAT file extension is assumed. If no path is specified, a path of MFGSYS is assumed.

Parameter 2: Record Number Range; Loop Back to Retry Transactions

Record Number Range. Specify the beginning and ending record number of the input data file that you want to process.

- If no beginning record number is specified, processing starts with the first record in the input data file.
- If no ending record number is specified, the input data file is processed until an end of file indicator is encountered. The ending record number is ignored if you have also specified the Loop Back (O) or Continuous Read (R) parameter.

Loop Back to Retry Transactions. The O parameter causes the input data file to be reprocessed when an end of file indicator is encountered and one or more records were left unprocessed in the input file due to database in use conditions. The input data file is reprocessed beginning at the first record which was left unprocessed. If no unprocessed records exist, the IMPT task completes.

A Required Macro Field number must be included with this parameter to indicate which field value will be overwritten with the Processing Status value when the record is processed. See the **Data Import Dictionary** manual for details about Required Macro Fields for each transaction.

Format for the processing status field is in the range 1-250. Preceding zeros should not be used. Use "5" not "05" for referencing field 5. See "Processing Status Indicator" in the **Using the Data Import Features** section of the System Control manual for more information on the reserved values.

Consider the following when using this parameter:

- The **Continuous Read (R) parameter** allows you to have a continuous import process available for all incoming transactions from another source. If you do not use the R parameter to specify a wait time, the system waits 10 seconds when an end of file indicator is encountered before looping back to reprocess the input data file.
- The **Loop Back (O) parameter** requires that you have an incoming data file field defined for the Processing Status field. This field must be initialized to "0" (zero enclosed in quotes) before processing begins. If you are using a template, a field in the incoming data file must be defined in the template for the Processing Status field so that the import process has a place to update. Required Macro Field 5 (although not normally needed) is recommended because it is the same field used in archive records formatted for use by the RCVR (System Recovery) task. However, any Required Macro Field can be specified as long as that field is not being used to import data. For example, the template definition on the TMPT screen would look similar to the following when using Required Macro Field 5 to update the value in field 9 of the incoming data file.

Fld#/Offset	Length	Reqd Fld#	Reqd Length	Fld Type
9		5	1	A

- For the **Retry (U) parameter**, you may specify the number of times the system will try to complete a transaction when a database in use (locked) condition is encountered.
 - If the transactions must be imported in chronological order, the recommended setting for the U parameter is a high number of retries (up to the maximum 999 at 5 seconds for each try or approximately 83 minutes). Also set the Error Level parameter to E2 to stop processing at the first error.

Using a high number of retries with Error Level E2 parameter addresses this situation: an input transaction changes a quantity on a purchase order. The transaction is not completed because of data-in-use conflict. Later in the input data file, another transaction changes the quantity again and this transaction is successful. When the first transaction is reprocessed on a subsequent loop of the input data file, the order quantity is changed again but this may now be in error.

- If the order of transaction processing is not relevant and transactions should be processed as rapidly as possible, the recommended setting for the U parameter is 1 with the Loop Back to Retry parameter specified.

Parameter 3: Template File and Alternate Path

Template File. Specify a template file when using an incoming data file that does not match the required macro format. The template file must be previously defined on the TMPT (Template Maintenance) screen. The system-assigned TPL file extension is used as a default. The required MFGSYS\I path is used as a default.

Alternate Path. Identify an alternate path for macro, template and table files. If no alternate path is identified, MFGSYS\I is assumed.

Parameter 4: Error Level; Error Log File; Processing Status Field Update

Error Level. Specify the error level in effect during import. The error levels are:

- **E1.** Do not stop at each error. Record the error in the error log file (IMPORT.ERR) and process the next transaction.
- **E2.** Stop at first error. Record the error in the error log file and abort the IMPT task.
- **E3.** Do not stop at each error. Record additional macro debugging information in the error log file.

If no error level is specified, E1 is used as a default.

If the E3 error level is specified, any transactions that were suspended because the task attempted to access locked records are recorded in the error log file. Transactions continue to be reprocessed as normal in this case. Additionally, the error log file traces any {REJECT} commands by recording the line number of the {IF} statement preceding the {REJECT} command. This error level is most effective while debugging custom macros. For more information on additional task parameters available to assist in debugging custom macros, see "Troubleshooting" in the Data Import Macro Basics manual.

Error Log File. Identify the error log file for rejected records and error messages. If no file name is specified, the IMPORT.ERR file is created in the MFGSYS directory. When an IMPORT.ERR file already exists, rejected records and error messages are appended to the end of the IMPORT.ERR file.

Processing Status Field Update. Specify whether the identified processing status field in the incoming data file should be updated to indicate that processing of this record has been completed. Format for the processing field is in the range 1-250. The P parameter is not used if you are also using the O parameter for looping import processing.

If this field is updated, you can reprocess a file without determining the specific starting record. If **Parameter 4** is not specified, the processing field is not updated. Any data in the field specified is overwritten by the processing flag.

The Processing Status Field Update (P) parameter requires that you have an incoming data file field defined for the Processing Status field. This field must be initialized to "0" (zero enclosed in quotes) before processing begins. If you are using a template, a field within the incoming data file must be defined in the template for the Processing Status field so that the import process has a place to update. Field 5 (although not normally required) is recommended because it is the same field used in archive records formatted for use by the RCVR (System Recovery) task. However, any Required Macro Field can be specified as long as that field is not being used to import data.

For example, the template definition on the TMPT screen would look similar to the following when using Required Macro Field 5 to update the value in field 9 of the incoming data file.

Fld#/Offset	Length	Reqd Fld#	Reqd Length	Fld Type
9		5	1	A

Parameter 5: Continuous Read; Multi-User Retry; Step Mode; View Mode

Continuous Read. Use the **R** parameter to indicate whether the IMPT task should wait for more transactions to enter after reaching the end of an incoming data file. The advantage of this parameter is that information from other systems can continually update the Fourth Shift database.

Note: This parameter is available only when you are using multi-user mode

The continuous-read parameter specifies how often the IMPT task checks the incoming data file for a new transaction. The parameter is expressed in terms of seconds. For example, enter **R300** to indicate that the system should wait 300 seconds before checking to see if additional transactions are available.

Once any transactions in the file are processed, waits the specified number of seconds and then checks for a new transaction. If a new transaction is not detected, the IMPT task returns to the Fourth Shift Master Menu and continues to wait for new transactions. In this way, other users are not locked out of entry screens. If you also select Parameter 2 to reprocess the input file each time the end of the file is reached, the continuous import process retries any unsuccessful transactions and then checks for new transactions.

When you press **S**, the IMPT task stops file processing. The next task in the batch process is then executed. If the last record is the string **TermIMPT**, the continuous process terminates without interaction.

If continuous read is not indicated, the end of the incoming data file signals the IMPT task to stop file processing. The next task in the batch process is then executed.

Multi-User Retry. Use the **U** parameter to indicate the number of times the IMPT task should try to complete a transaction when a *database-in-use* condition is encountered. The IMPT task waits five seconds between tries. The **U** parameter is available only when you are using multi-user mode. If retry is not specified, the IMPT task tries to complete a transaction three times before writing the transaction to the error log file. For example:

- Enter **U5** to indicate that the IMPT task should retry the transaction 5 times.
- Enter **U0** to indicate that no multi-user retry should be attempted.

Step Mode. Use the **S** parameter to start the data import process in single step mode. This allows you to review each transaction for troubleshooting purposes. Usually it is sufficient to limit the number of records you process in single step mode. For example, to validate the process with a single record, use parameters **F1 Z1**.

Note: The **S** parameter is not available if you are using the Screen Customizer (SCRM) Module. Using the parameter task does not cause an error but you cannot access single step mode. You can temporarily disable screen customizations by renaming the *.DAT files in the MFGSYS\SCREENS directory.

View Mode. Use the **V** parameter to control how the user interface is displayed during import processing. The options include:

Parameter	Description
VH	Hide Screens. Shows the Data Import Processing dialog when processing starts and hides the transaction processing. This is the default mode if the V parameter is not specified.
VM	Minimize. Once processing starts, the entire Fourth Shift session is minimized. This mode provides the fastest transaction processing. The percent complete indicator is displayed as the title of the minimized session.

Parameter	Description
VS	Show Screens. Show the Data Import Processing dialog when processing starts and also show the transaction processing through the screen. This is useful for troubleshooting but will cause the data import process to run more slowly than the other options. It is recommended that you use this option in combination with Single Step mode.

Parameter Validation

Parameter validation provides detailed information on invalid file parameters. It is important to validate parameters before running the IMPT task using the BEXE screen. See the "Data Import Error Messages" in the **Using the Data Import Features** section of the System Control manual for more information.

Example

For example, to request Data Import as the first task in the process using the:

- MFGSYS and MFGSYS\I directories (default)
- INVM.MAC macro file (*.MAC default)
- ITMB0389.DAT incoming data file (*.DAT default)
- E2 error level, stop at first error
- IMPORT.ERR error log file
- ITMBPLAN template file
- starting record number 25
- no continuous-read parameter
- no loop back parameter
- no process status field update parameter
- no retry parameter

the **Seq Num**, **Task** and **Parameters** fields in the task line are entered like this:

```
01  IMPT  MINVM  IITMB0389  E2  TITMBPLAN  F25
```

Entry Field	Example Value	Description
Seq Num	01	First task in process
Task	IMPT	Data Import
Parameter 1	MINVM IITMB0389	Macro File and Incoming Data File
Parameter 2	E2	Error Level
Parameter 3	TITMBPLAN	Template File
Parameter 4	F25	Starting Record Number

To run Data Import, use the BEXE (Batch Process Execution) screen to execute the process in which the IMPT task is entered. Two log files are produced in the MFGSYS directory during batch process execution:

- batch process log file (LOG.xxx) where xxx is the **Process Id**
- IMPT error log file (IMPORT.ERR)

Check both log files after executing the batch process.

RCVR – System Recovery

The RCVR (System Recovery) task assists in performing a system recovery. After restoring the last system backup, the RCVR task is used to read the Archive file and reenter screen transactions automatically. Batch tasks are rerun at the appropriate points during the system recovery process. The ARCREORG and ARCBEXE utilities aid in determining which screen transactions can be reentered as a group.

The RCVR task is used for system recovery only if you are not able to use the SQL Server logging feature. See "Recovery" in the SQL Server Administration and Utilities manual for more information. The RCVR task re-enters screen transactions. The following do not have corresponding archive records and are not entered by the RCVR task:

- TMPT (Template Maintenance)
- Shop Calendar

Important: Important: RCVR will not recover SOP Archive transactions.

The following screens and tasks, for which there are corresponding archive records, are an exception and are not entered by the RCVR task:

- APCF (A/P Configuration)
- APCW (A/P Payment Processing)
- APTP (A/P 1099 Processing)
- ARCF (A/R Configuration)
- CNFG (System Installation Parameters)
- GLCF (G/L Configuration)
- GLCO (G/L Consolidation Process)
- GLHP (G/L History Process)
- LEXP (Lot Expiration/Retest Evaluation)
- PASS (Password Maintenance)

These screen transactions, along with batch processes executed using the BEXE screen (BEXE00), must be reentered at the appropriate points in the system recovery process. The RCVR task stops file processing when a non-recovered archive record is reached. For your convenience, archive records for non-recovered transactions are offset by dashed lines in the Archive printout for easy identification.

Note: Batch process transactions numbered higher than BEXE00 are a result of CIM Customizer processing and do not represent database updates. Transactions numbered higher than BEXE00 are skipped by the RCVR task.

The RCVR task sets the date to match transactions being reentered. The time defaults to the system time of the workstation used to reenter the transactions. **After the RCVR task has completed processing, the date is reset to the operating system date.** This is important to remember if you continue the recovery process with manual entries or date-sensitive batch processes.

Note: System recovery works best if a system backup is done twice a day—once before batch runs and once before daily activity.

Task Reference

Preparation

The RCVR task is used for system recovery only if you are not able to use the SQL Server logging feature. See "Recovery" in the SQL Server Administration and Utilities manual for more information.

System recovery preparation involves:

- reorganizing the Archive file
- restoring the last system backup
- setting up batch process(es) to reenter screen transactions

Note: Contact Fourth Shift Support Services to obtain an additional utility ARCBEXE and instructions for system recovery. These instructions include more detailed information on using the ARCREORG utility.

1. Reorganize the Archive File

Some Fourth Shift transactions produce multiple interdependent archive records for one transaction. The G/L Distribution after entering a cash receipt on the ARCD (A/R Cash Deposit) screen is one example. In a multi-user configuration, the transactions are not necessarily recorded contiguously in the ARCHIVE.FIL file.

The ARCHIVE.FIL file, therefore, needs to be reorganized so that interdependent archive records are contiguous. These records include a flag identifying where the transaction belongs in a sequence. The ARCREORG utility uses this flag to reorganize the ARCHIVE.FIL file.

The ARCREORG utility copies the ARCHIVE.FIL file to the ARCHIVE.SAV file and then reorganizes the ARCHIVE.FIL file.

See ARCREORG in the Utilities and Tools manual for more information.

2. Run ARCBEXE

Run ARCBEXE according to the instruction document obtained from Fourth Shift Support Services.

3. Restore Last System Backup

1. Delete any MODCOM*. files and BATCHUPD.* files in the MFGSYS directory. These files are recreated during system recovery. In this way, double posting is avoided.
2. Restore the last system backup, including database and non-database files. **Do not restore the ARCHIVE file.** A new ARCHIVE file is created as transactions are reentered.

4. Perform RCVR Tasks

Perform RCVR tasks to recover selected data. See the instructions provided with the ARCBEXE utility.

Monitoring the Import Process

While the RCVR task is processing, the Data Import Processing dialog box is displayed. Use this dialog box to monitor the processing status of the task.

For information on using the options available on the dialog box, see "Managing the Import Process" in the System Control manual.

System Recovery Errors

Three log files can be produced in the MFGSYS directory during batch process execution:

- batch process log file (xxx.LOG) where xxx is the **Process Id**
- RCVR error log file (RCVR.ERR)
- RCVR.SOP log file contains skipped SOP Archive Transactions (only if SOPM is installed)

Check the log files after executing the batch process. See the "Data Import Error Messages" in the **Using the Data Import Features** section of the System Control manual for more information.

Caution: If you need to run the RCVR task concurrently in more than one batch process, be sure to identify separate error log files for each process. The log files are opened in exclusive mode so the content cannot be written to by more than one process at the same time.

Parameters

To request System Recovery, enter the RCVR task as one of the sequenced tasks in a batch process

Caution: If you need to run the RCVR task concurrently in more than one batch process, be sure to identify separate error log files for each process. The log files are opened in exclusive mode so the content cannot be written to by more than one process at the same time.

The RCVR task has the following parameters:

Task Parameter	Format	Entry Is...
1: Incoming Data File; Use Workstation Date/Time; Use Original User Id	IX..X D X	Optional
2: Record Number Range; Loop Back to Retry Transactions	F9..9 Z9..9; O	Optional
3: Error Level; Log File; Update Process Status Field	E9 LX..X P999	Optional
4: Continuous Read; MU Retry; Step	R999 U999 S	Optional
5: View Mode	VX	Optional

Parameter 1: Incoming Data File; Use Workstation Date/Time; Use Original User Id
Incoming Data File. Identify the renamed ARCHIVE file. If no file is specified, the RCVR.DAT file is assumed. If no path is specified, the MFGSYS path is assumed.

Use Workstation Date/Time. Indicates which date/time information will be assigned when the import transaction is completed. If this parameter is not specified, the date/time information from the incoming data file is assigned. Enter **D** to use the workstation date/time when the incoming transaction is processed.

Use Original User Id. Use the **X** parameter to indicate that the user identification defined in the incoming transaction should be defined in the resulting ARCHIVE.FIL record after completing the transaction. This parameter allows both the source ARCHIVE.FIL and resulting ARCHIVE.FIL files to contain the same user audit trail data.

Parameter 2: Record Number Range; Loop Back to Retry Transactions

Record Number Range. Specify the beginning and ending record number of the archive file that you want to process.

- If no beginning record number is specified, processing starts with the first record in the archive file.
- If no ending record number is specified, the archive file is processed until an end of file indicator is encountered. The ending record number is ignored if you have also specified the Loop Back (O) or Continuous Read (R) parameter.

Loop Back to Retry Transactions. The O parameter causes the archive file to be reprocessed when an end of file indicator is encountered and one or more records were left unprocessed in the file due to database in use conditions. The input data file is reprocessed beginning at the first record which was left unprocessed.

Field 5 is overwritten with the Processing Status value when the record is processed. See Processing Status Indicator in the **Using the Data Import Features** section of the System Control manual for more information on the reserved values.

Consider the following when using this parameter:

- The Continuous Read (R) parameter allows you to have a continuous import process available for all incoming transactions from another source. If you do not use the R parameter to specify a wait time, the system waits 10 seconds when an end of file indicator is encountered before looping back to reprocess the input data file.
- If the transactions must be imported in chronological order, the recommended setting for the U parameter is a high number of retries (up to the maximum 999 at 5 seconds for each try or approximately 83 minutes). Also set the Error Level parameter to E2 to stop processing at the first error.

Using a high number of retries with Error Level E2 parameter addresses this situation: an input transaction changes a quantity on a purchase order. The transaction is not completed because of data-in-use conflict. Later in the input data file, another transaction changes the quantity again and this transaction is successful. When the first transaction is reprocessed on a subsequent loop of the input data file, the order quantity is changed again but this may now be in error.

- If the order of transaction processing is not relevant and transactions should be processed as rapidly as possible, the recommended setting for the U parameter is 1 with the Loop Back to Retry parameter specified.

Parameter 3: Error Level; Processing Status Field Update

Error Level. Specify the error level in effect during import. The error levels are:

- **E1.** Do not stop at each error. Record the error in the error log file (RCVR.ERR) and process the next transaction.
- **E2.** Stop at first error. Record the error in the error log file and abort the RCVR task.
- **E3.** Do not stop at each error. Record additional macro debugging information in the error log file.

If no error level is specified, E1 is used as a default.

If the E3 error level is specified, any transactions that were suspended because the task attempted to access locked records are recorded in the error log file. Transactions continue to be reprocessed as normal in this case. Additionally, the error log file traces any {REJECT} commands by recording the line number of the {IF} statement preceding the {REJECT} command. This error level is most effective while debugging custom macros. For more information on additional task

parameters available to assist in debugging custom macros, see "Troubleshooting" in the Data Import Macro Basics manual.

Error Log File. Identify the error log file for rejected records and error messages. If no file name is specified, the RCVR.ERR file is created in the MFGSYS directory. When an RCVR.ERR file already exists, rejected records and error messages are appended to the end of the RCVR.ERR file.

Processing Status Field Update. Specify whether the processing field (Field #5) in the Archive layout should be updated to indicate that processing of this record has been completed. If this field is updated, you can reprocess a file without determining the specific starting record. If Parameter 3 is not specified, the processing field is not updated. The P parameter is not used if you are also using the O parameter for looping import processing.

The Processing Status Field Update (P) parameter requires that you have Field #5 defined as the Processing Status field if you are using an Incoming Data file other than the ARCHIVE.FIL file. This field must be initialized to 0 (zero) before processing begins.

Parameter 4: Continuous Read; Multi-User Retry; Step Mode

Continuous Read. Indicate whether the RCVR task should wait for more transactions to enter after reaching the end of an incoming data file. The advantage of this parameter is that information from other systems can continually update the database.

Note: This parameter is available only when you are using multi-user mode

The continuous-read parameter specifies how often the RCVR task checks the incoming data file for a new transaction. The parameter is expressed in terms of seconds.

For example, enter R300 to indicate that the system should wait 300 seconds before checking to see if additional transactions are available.

Once any transactions in the file are processed, waits the specified number of seconds and then checks for a new transaction. If a new transaction is not detected, the RCVR task returns to the Fourth Shift Master Menu and continues to wait for new transactions. In this way, other users are not locked out of entry screens. If you also select Parameter 2 to reprocess the input file each time the end of the file is reached, the continuous import process retries any unsuccessful transactions and then checks for new transactions.

When you press S, the RCVR task stops file processing. The next task in the batch process is then executed. If the last transaction is the string **TermRCVR** (followed by a carriage return), the continuous process terminates without interaction.

If continuous read is not indicated, the end of the incoming data file signals the RCVR task to stop file processing. The next task in the batch process is then executed.

Retry. Indicate the number of times the RCVR task should try to complete a transaction when a "database in use" condition is encountered. The RCVR task waits five seconds between tries. If retry is not specified, the RCVR task tries to complete a transaction three times before writing the transaction to the error log file. For example:

- Enter **U5** to indicate that the system should retry the transaction 5 times.
- Enter **U0** to indicate that no multi-user retry should be attempted.

Step. Use the **S** parameter to start the data import process in single step mode. This allows you to review each transaction for troubleshooting purposes.

Note: The **S** parameter is not available if you are using the Screen Customizer (SCRM) Module. Using the parameter task does not cause an error but you cannot access single step mode.

Parameter 5: View Mode

Use the **V** parameter to control how the user interface is displayed during import processing. The options include:

Parameter	Description
VH	Hide Screens. Shows the Data Import Processing dialog when processing starts and hides the transaction processing. This is the default mode if the V parameter is not specified.
VM	Minimize. Once processing starts, the entire Fourth Shift session is minimized. This mode provides the fastest transaction processing. The percent complete indicator is displayed as the title of the minimized session.
VS	Show Screens. Show the Data Import Processing dialog when processing starts and also show the transaction processing through the screen. This is useful for troubleshooting but will cause the data import process to run more slowly than the other options.

Parameter Validation

Parameter validation provides detailed information on invalid file parameters. It is important to validate parameters before running the RCVR task. See "Data Import Error Messages" in the **Using the Data Import Features** section of the System Control manual for an explanation of error messages for the RCVR task.

Example

For example, to request System Recovery as the first task in the process using the:

- MFGSYS\RCVR.DAT incoming data file (default)
- starting record number 25
- ending record number 50
- E2 error level, stop at first error

the **Seq Num**, **Task** and **Parameters** fields in the task line are entered like this:

```
01  RCVR  F25  Z50  E2
```

Entry Field	Example Value	Description
Seq Num	01	First task in process
Task	RCVR	System Recovery
Parameter 2	F25 Z50	Starting and Ending Record Numbers
Parameter 3	E2	Error Level

Note: You might want to include the QUIT task after the RCVR task so that you are forced to sign back onto the system using the correct date for the non-recovered transactions.

To run System Recovery, sign onto the system in single-user or exclusive mode and use the BEXE (Batch Process Execution) screen to execute the process in which the RCVR task is entered.

TMPT – Template Maintenance

Use this screen to identify the record format of incoming data as it relates to the format required by a Data Import macro. Once the record format relationship is defined as a template, the incoming data can be imported using the IMPT (Data Import) task. Value substitution using wildcards, ranges and field reconfiguration is specified in the Data Conversion Table window.

Features

Transportation Shortcuts

You can use shortcut keys or transport buttons to go to the following related screens.

Destination	Shortcut Key(s)
BEXE (Batch Process Execution)	F8
FCMT (Function Code Maintenance)	F9
IMPT (Data Import)	F10

Additional Information

Window	Available From	Shortcut Key(s)
Data Conversion Table	Field Format Definition section	ALT+F4

Web Links

If you use Web UI, you can link to other screens by clicking tabs or hyperlinks.

There are no tabs or hyperlinks available for this screen.

Screen Reference

Format

The TMPT screen has two sections: **Template Identification** and **Field Format Definition**.

Use the **Template Identification** section to specify the template file that defines the relationship between an incoming data record and its corresponding macro requirements. The template is identified as CDF (comma delimited format) or SDF (standard delimited format) based on the incoming data record. Use the **Template Identification** section to add or delete a template. Templates are ASCII files created with a TPL file extension.

Use the **Field Format Definition** section to define the relationship between fields in the incoming data record and corresponding fields required by the identified macro. A template must contain a field format definition for each field required by the macro. You can add, change or delete field

format definitions and access the Data Conversion Table window. Data conversion tables are ASCII files created with a TBL file extension.

Before You Create a Template

Identify the Macro

You can use predefined macros or your own custom macros.

Predefined Macros. To use predefined macros, incoming data records must be defined in terms of archive records. Use the **Data Import Dictionary** to determine the type of transaction to be performed and its associated macro requirements. Each transaction has a six-character identification: the Main screen Function Code followed by two digits. For example, the transaction id for adding locations is LMMT00.

Custom Macros. To use custom macros, incoming data records must be defined in terms of the fields required by the macro. Print a copy of the custom macro to use as a reference.

Analyze Required Macro Fields

Predefined Macros. Use the **Required Transaction Fields** table in the **Data Import Dictionary** for the transaction you have identified. For each required field, note the field number, type and maximum length. Also note whether a field must have a value for proper system identification (indicated by an asterisk by the **Field#**).

Custom Macros. Using the printed macro, highlight all field references. Identify the field number, type and maximum length. Also note whether a field must have a value for proper system identification.

Analyze Incoming Data Record

1. Identify any required macro fields missing in the incoming data record. Missing fields can be:

- initialized by inserting a literal
- identified as null and tabbed through to remain unchanged

Literal Insertion. To initialize a field by inserting a literal, the specific characters to be entered are specified in the **Literal/Table Name** field and are identified as a **Field Type = L** (literal). Once the data is imported, the field can be edited on a per-record basis.

When a literal is inserted for an incoming field, the field is not added to the incoming data record. The **Required Macro Length** must be equal to the length of the literal.

Note: If you need a literal that is more than 8 characters in length, specify a table for the field and use the Data Conversion Table window to identify a **Converted Value** for an **Incoming Value = *** (wildcard = all fields).

Null Fields. A null field is an empty field in a data record. A null field in an incoming data record signals to tab through the field and leave the field unchanged.

Null fields are used to complete a template when there is no existing relationship between an incoming **Fld#/Offset** and a required macro **Field#**. One null field added to the end of incoming records can relate to any number of required macro fields.

Null Field: CDF File. A null field in a CDF file can be represented as alphanumeric (two consecutive quotation marks) or numeric (one comma) as shown below:

	Alphanumeric	Numeric
Null Field at End of Record	"01","01",""	"01","02",
Null Field Within Record	"01","","01"	"01",,"02"

Null Field: SDF File. A null field in an SDF file is identified on the TMPT screen as having a specific **Offset** and **Incoming Record Length** = 0.

2. Determine the field validation performed by the screen(s) used.

Incoming data fields should match Fourth Shift validation rules. Use the on-line help for each screen used by the macro to help determine validations. Compare the incoming data field length to the Fourth Shift field length for possible truncation. Truncation rules are:

- Alphanumeric: Truncated from the right.
- Number: Truncated first by deleting non-significant zeros on the left, then by truncating decimal places on the right.

Assign any conversion values necessary to meet Fourth Shift field validations. Conversion values are specified in the Data Conversion Table window.

3. Identify the record type and layout of incoming data. For CDF records, identify the field number for each field in the record. For SDF records, identify the starting column position and length of each field in the record.

Date Conversions

Date fields in an incoming data record can be converted to the six-digit system date based on the **Date and Number Format** field on the CNFG (System Installation Setup) screen.

Date conversion is specified on the TMPT (Template Maintenance) screen in the **Field Type** field. The **Field Type** field identifies the character format of the incoming data field. Date formats are:

- U = U.S. (MMDDYY)
- B = British (DDMMYY)
- E = European (DDMMYY)
- C = Chinese and Canadian (YYMMDD)
- S = Swedish (YYMMDD)
- D = Date not converted; changed into a six-digit format

Date Delimiters

The following date delimiters are valid for incoming date fields.

Separator	Description	Example
	space	5 15 99
-	hyphen	5-15-99
.	period	5.15.99
,	comma	5,15,99
/	forward slash	5/15/99
\	backslash	5\15\99

U.S. Date Format

The following incoming date formats can currently be converted to the appropriate system format. These examples use the forward slash delimiter or no delimiter but any of the delimiters are valid:

Incoming Date	Field Type	System Date	System Format
mmddyy	U	mmddyy	U
yyyymmdd	U	mmddyy	U
mm/dd/yyyy	U	mmddyy	U
mm/dd/yy	U	mmddyy	U
m/dd/yy	U	mmddyy	U
m/d/yy	U	mmddyy	U
mm/d/yyyy	U	mmddyy	U
m/dd/yyyy	U	mmddyy	U
m/d/yyyy	U	mmddyy	U
mm/d/yy	U	mmddyy	U

Note: **yyyymmdd** is the date format extracted from Fourth Shift using the data export features. In order to import extracted dates back into the system, specify a **Field Type** = U.

Incoming Date	Field Type	System Date	System Format
mmddyy	U	ddmmyy	E or B
yyyymmdd	U	ddmmyy	E or B
mm/dd/yyyy	U	ddmmyy	E or B
mm/dd/yy	U	ddmmyy	E or B
m/dd/yy	U	ddmmyy	E or B
m/d/yy	U	ddmmyy	E or B
mm/d/yyyy	U	ddmmyy	E or B
m/dd/yyyy	U	ddmmyy	E or B
m/d/yyyy	U	ddmmyy	E or B
mm/d/yy	U	ddmmyy	E or B

Incoming Date	Field Type	System Date	System Format
mmddyy	U	yyymmdd	C or S
mm/dd/yyyy	U	yyymmdd	C or S
mm/dd/yy	U	yyymmdd	C or S
m/dd/yy	U	yyymmdd	C or S
m/d/yy	U	yyymmdd	C or S

Incoming Date	Field Type	System Date	System Format
mm/d/yyyy	U	yymmdd	C or S
m/dd/yyyy	U	yymmdd	C or S
m/d/yyyy	U	yymmdd	C or S
mm/d/yy	U	yymmdd	C or S
yyyymmdd	U	yymmdd	C or S

British or European Date Format

The following incoming date formats can currently be converted to the appropriate system format. These examples use the forward slash delimiter or no delimiter but any of the delimiters are valid:

Incoming Date	Field Type	System Date	System Format
ddmmyy	E or B	ddmmyy	E or B
dd/mm/yyyy	E or B	ddmmyy	E or B
dd/mm/yy	E or B	ddmmyy	E or B
d/mm/yy	E or B	ddmmyy	E or B
d/m/yy	E or B	ddmmyy	E or B
dd/m/yyyy	E or B	ddmmyy	E or B
d/mm/yyyy	E or B	ddmmyy	E or B
d/m/yyyy	E or B	ddmmyy	E or B
dd/m/yy	E or B	ddmmyy	E or B

Incoming Date	Field Type	System Date	System Format
ddmmyy	E or B	mmddyy	U
dd/mm/yyyy	E or B	mmddyy	U
dd/mm/yy	E or B	mmddyy	U
d/mm/yy	E or B	mmddyy	U
d/m/yy	E or B	mmddyy	U
dd/m/yyyy	E or B	mmddyy	U
d/mm/yyyy	E or B	mmddyy	U
d/m/yyyy	E or B	mmddyy	U
dd/m/yy	E or B	mmddyy	U

Incoming Date	Field Type	System Date	System Format
ddmmyy	E or B	yymmdd	C or S
dd/mm/yyyy	E or B	yymmdd	C or S

Incoming Date	Field Type	System Date	System Format
dd/mm/yy	E or B	yyymmdd	C or S
d/mm/yy	E or B	yyymmdd	C or S
d/m/yy	E or B	yyymmdd	C or S
dd/m/yyyy	E or B	yyymmdd	C or S
d/mm/yyyy	E or B	yyymmdd	C or S
d/m/yyyy	E or B	yyymmdd	C or S
dd/m/yy	E or B	yyymmdd	C or S

Chinese and Canadian or Swedish Date Formats

The following incoming date formats can currently be converted to the appropriate system format. These examples use the forward slash delimiter or no delimiter but any of the delimiters are valid:

Incoming Date	Field Type	System Date	System Format
yyymmdd	C or S	yyymmdd	C or S
yyyy/mm/dd	C or S	yyymmdd	C or S
yy/mm/dd	C or S	yyymmdd	C or S
yy/mm/d	C or S	yyymmdd	C or S
yy/m/d	C or S	yyymmdd	C or S
yyyy/m/dd	C or S	yyymmdd	C or S
yyyy/mm/d	C or S	yyymmdd	C or S
yyyy/m/d	C or S	yyymmdd	C or S
yy/m/dd	C or S	yyymmdd	C or S

Incoming Date	Field Type	System Date	System Format
yyymmdd	C or S	mmddyy	U
yyyy/mm/dd	C or S	mmddyy	U
yy/mm/dd	C or S	mmddyy	U
yy/mm/d	C or S	mmddyy	U
yy/m/d	C or S	mmddyy	U
yyyy/m/dd	C or S	mmddyy	U
yyyy/mm/d	C or S	mmddyy	U
yyyy/m/d	C or S	mmddyy	U
yy/m/dd	C or S	mmddyy	U

Incoming Date	Field Type	System Date	System Format
yyymmdd	C or S	ddyymm	E or B
yyyy/mm/dd	C or S	ddyymm	E or B
yy/mm/dd	C or S	ddyymm	E or B
yy/mm/d	C or S	ddyymm	E or B
yy/m/d	C or S	ddyymm	E or B
yyyy/m/dd	C or S	ddyymm	E or B
yyyy/mm/d	C or S	ddyymm	E or B
yyyy/m/d	C or S	ddyymm	E or B
yy/m/dd	C or S	ddyymm	E or B

Other Date Formats

If the incoming data field is specified as **Field Type** = D, the incoming date is changed to a six-digit version of the incoming date format and is not converted. These examples use the forward slash delimiter or no delimiter but any of the delimiters are valid:

Incoming Date (U)	Field Type	System Date	System Format
Mmddy	D	mmddy	U
mm/dd/yyyy	D	mmddy	U
mm/dd/yy	D	mmddy	U
m/dd/yy	D	mmddy	U
m/d/yy	D	mmddy	U
mm/d/yyyy	D	mmddy	U
m/dd/yyyy	D	mmddy	U
m/d/yyyy	D	mmddy	U
mm/d/yy	D	mmddy	U

Incoming Date (E or B)	Field Type	System Date	System Format
ddmmyy	D	ddmmyy	E or B
dd/mm/yyyy	D	ddmmyy	E or B
dd/mm/yy	D	ddmmyy	E or B
d/mm/yy	D	ddmmyy	E or B
d/m/yy	D	ddmmyy	E or B
dd/m/yyyy	D	ddmmyy	E or B
d/mm/yyyy	D	ddmmyy	E or B

Incoming Date (E or B)	Field Type	System Date	System Format
d/m/yyyy	D	ddmmyy	E or B
dd/m/yy	D	ddmmyy	E or B

Incoming Date	Field Type	System Date	System Format
yymmdd	D	yymmdd	C or S
yyyy/mm/dd	D	yymmdd	C or S
yy/mm/dd	D	yymmdd	C or S
yy/mm/d	D	yymmdd	C or S
yy/m/d	D	yymmdd	C or S
yyyy/m/dd	D	yymmdd	C or S
yyyy/mm/d	D	yymmdd	C or S
yyyy/m/d	D	yymmdd	C or S
yy/m/dd	D	yymmdd	C or S

Fields

Field#

Field Number identifies the required macro field that corresponds to the field in the incoming data record. Entry is 0 through 999.

Where Used: TMPT

Fld#/Offset

Field Number/Offset is the position of a field in the incoming data record. A **Field Number** is identified for **Template Type** = CDF. An **Offset**, or beginning column position, is identified for **Template Type** = SDF. An **Offset** of 0 is the first column in an SDF file. Entry is 0 through 999.

Where Used: TMPT

Field Type

Field Type identifies the field in the incoming data record as characters, numbers, dates or specific values. **Field Types** are:

A = Alphanumeric.

Values A-Z; 0-9. Truncated from the right.

N = Number.

Values 0-9. Truncated first by deleting non-significant zeros on the left, then by truncating decimal places on the right.

D = Date.

Imported as 6-digit format of incoming field.

B = British Date.

Converted to CNFG 6-digit date format.

E = European Date.

Converted to CNFG 6-digit date format.

U = United States Date.

Converted to CNFG 6-digit date format.

C = Canadian or Chinese Date.

Converted to CNFG 6-digit date format.

S = Swedish Date.

Converted to CNFG 6-digit date format.

L = Literal.

Macro identification or specific value.

Where Used: TMPT

Function

Function codes are four-character abbreviations for screen names. Each screen has a unique code used for identification and transportation. For example, ITMB identifies the Item Master screen. Entry is 4 alphanumeric characters.

Where Used: screens and reports

Incoming Record Length

Incoming Record Length is the number of characters in a field in the incoming data record. The **Incoming Record Length** is identified for **Template Type** = SDF. Entry is 0 through 999.

Where Used: TMPT

Literal/Table Name

Literal/Table Name identifies a macro in the first template line, a specific value to be used for an incoming data field or a data conversion table only. A data conversion table can be specified when the **Field Type** = A, N, D, B, E, C, S or U. Do not use the **Literal/Table Name** field as reference. Entry is any alphanumeric combination of up to 8 characters. A TBL filename extension is automatically assigned to a table file.

Where Used: Data Conversion Table; Template Browse; TMPT

Required Macro Length

Required Macro Length is the maximum number of characters in the required macro field. When the **Field Type** = L, the **Required Macro Length** must equal the length of the literal. Entry is 0 through 999.

Where Used: TMPT

Template Name

Template Name is the user-defined filename for a set of field format definitions that relate a record layout of incoming data to required macro fields. A template contains a format definition for each required field in the macro and a line for macro identification. Entry is any alphanumeric combination of up to 8 characters. A TPL filename extension is automatically assigned.

Where Used: Template Browse; TMPT

Template Type

Template Type identifies the record layout of the incoming data to be related to required macro fields. A **Template Type** must be entered for a new **Template Name**. **Template Types** are:

CDF = Comma delimited format

SDF = Standard delimited format

Where Used: Template Browse; TMPT

Template Browse

Use this window to review and select the template you wish to update on the TMPT (Template Maintenance) screen. Templates are sorted by type and name. Non-import files are not included in this list. The list is generated from the MFGSYS\I location or the location identified by the **ImportFiles** configuration variable.

Features

Transportation Shortcuts

You can use shortcut keys to go to the following related screens.

Destination	Shortcut Key(s)
Previous screen	ESC

Web Links

If you use Web UI, you can link to other screens by clicking tabs or hyperlinks.

There are no tabs or hyperlinks available for this screen.

Reports

A standard report is not generated for this screen. Use the Print Screen key or any screen capture program to create an image of the screen.

Fields

Last Modified

Date is the day, month and year by which system transactions are identified.

Where Used: APDS; APIP; APPD; APRC; ARCD; ARIP; ARPD; ARST; AUDT; BKCB; BK CJ; BKFT; BKRC; Cash Set Selection; CCAN; Customer + Credit; Fourth Shift Sign-On; GTED; History Detail; Inventory Adjustment Application; Inventory Transaction History Report; Lot Inventory Transaction History Report; Reconciliation Detail; REDI; RVED; SUND; SUNR; SUPD; SUPR; Template Browse; Transaction Detail; Transaction History; TRUD

Macro Id

Literal/Table Name identifies a macro in the first template line, a specific value to be used for an incoming data field or a data conversion table only. A data conversion table can be specified when the **Field Type** = A, N, D, B, E, C, S or U. Do not use the **Literal/Table Name** field as reference. Entry is any alphanumeric combination of up to 8 characters. A TBL filename extension is automatically assigned to a table file.

Where Used: Data Conversion Table; Template Browse; TMPT

Template

Template Name is the user-defined filename for a set of field format definitions that relate a record layout of incoming data to required macro fields. A template contains a format definition for each required field in the macro and a line for macro identification. Entry is any alphanumeric combination of up to 8 characters. A TPL filename extension is automatically assigned.

Where Used: Template Browse; TMPT

Type

Template Type identifies the record layout of the incoming data to be related to required macro fields. A **Template Type** must be entered for a new **Template Name**. **Template Types** are:

CDF = Comma delimited format

SDF = Standard delimited format

Where Used: Template Browse; TMPT

Data Conversion Table

Important: Because of a recent field expansion enhancement, any data conversion tables created prior to R7.40G will no longer function correctly. For R7.40G systems or higher, an unexpected error will occur if the system tries to load any existing conversion table files.

Before you can *re-create* any new data conversion tables, you must *delete* the original tables then add the new tables to the appropriate template in the TMPT (Template Maintenance) screen.

Use the Data Conversion Table window to list specific field values to be converted during the import process. One field format definition can have one or more field value conversions. As the incoming data records are processed, the table is referenced for each field having a table identified as part of its field format definition. Conversion techniques include the use of specific characters, wildcards, ranges and field reconfiguration.

As incoming data records are processed, the table is referenced for each field having a table identified as part of its field format definition. When a matching value is found for a field, the field is converted as specified. When a matching value is not found for a field, the incoming data record is written to the error log file (IMPORT.ERR) along with a message and the record is not processed.

Features

Transportation Shortcuts

You can use shortcut keys to go to the following related screens.

Destination	Shortcut Key(s)
Previous screen	ESC

Web Links

If you use Web UI, you can link to other screens by clicking tabs or hyperlinks.

There are no tabs or hyperlinks available for this screen.

Reports

A standard report is not generated for this screen. Use the Print Screen key or any screen capture program to create an image of the screen.

Screen Reference

Data Conversion Table – Performing Character Conversions

Specific characters can be imported as different characters such as:

Incoming Value	Converted Value	Import Result
A	B	Incoming A fields become Bs; does not convert As within a field
123-456-789	321-456-789	Converts account number to new number

See "Reconfiguring a Field" under "Data Conversion" in the Using the Data Import Feature of the System Control manual for more information on additional options.

Data Conversion Table – Using Wildcards

"Wildcard" characters (* and ?), similar to those in your operating system commands, allow you to convert information based on a specified group of values. For example:

Incoming Value	Converted Value	Import Result
123*	123A	Substitute 123A for Incoming Value beginning with 123.
123?5	123A5	If the first three characters of Incoming Value are 123 and the last character is 5, convert whatever value is in the fourth position of Incoming Value to A.
AA???	BB???	If the first two characters of Incoming Value are AA followed by three characters, change the first two characters to BB and import the Incoming Values for the remaining three characters as is.
*	AA	Use default value AA for all Incoming Values .
*	*	If no table match is found, import Incoming Value as is.

Note: If the * wildcard is not used to end a table to cover the "no match found" situation, the incoming data record is rejected when a match is not found.

See "Reconfiguring a Field" under "Data Conversion" in the Using the Data Import Feature of the *Data Import* manual for more information on additional options.

Data Conversion Table – Specifying a Range

A range of **Incoming Values** can be converted to a specific **Converted Value** using two consecutive field value definitions as follows:

	Incoming Value	Converted Value
line 1:	beginning value	{RANGE}

	Incoming Value	Converted Value
line 2:	ending value	value for range

The use of **{RANGE}** signals a range and the second line defines the conversion value. A range is determined by an ASCII character sort. Be certain to zero-fill numbers, where necessary, to specify the actual range desired. For example:

Incoming Value	Converted Value	Import Result
001	{RANGE}	Convert any value between 1 and 500 to 250
500	250	

Fields

Converted Value

Converted Value identifies the new value to be used for a field when the incoming data record is imported. Entry is any alphanumeric combination of up to 70 characters.

Where Used: Data Conversion Table

Incoming Value

Incoming Value identifies the value of the field in the incoming data record to be imported as a new value identified by **Converted Value**. Entry is any alphanumeric combination of up to 70 characters.

Where Used: Data Conversion Table

Table Name

Literal/Table Name identifies a macro in the first template line, a specific value to be used for an incoming data field or a data conversion table only. A data conversion table can be specified when the **Field Type** = A, N, D, B, E, C, S or U. Do not use the **Literal/Table Name** field as reference. Entry is any alphanumeric combination of up to 8 characters. A TBL filename extension is automatically assigned to a table file.

Where Used: Data Conversion Table; Template Browse; TMPT