

Ascential DataStage™

Complex Flat File Stage Guide

Version 2.0



This document, and the software described or referenced in it, are confidential and proprietary to Ascential Software Corporation ("Ascential"). They are provided under, and are subject to, the terms and conditions of a license agreement between Ascential and the licensee, and may not be transferred, disclosed, or otherwise provided to third parties, unless otherwise permitted by that agreement. No portion of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Ascential. The specifications and other information contained in this document for some purposes may not be complete, current, or correct, and are subject to change without notice. NO REPRESENTATION OR OTHER AFFIRMATION OF FACT CONTAINED IN THIS DOCUMENT, INCLUDING WITHOUT LIMITATION STATEMENTS REGARDING CAPACITY, PERFORMANCE, OR SUITABILITY FOR USE OF PRODUCTS OR SOFTWARE DESCRIBED HEREIN, SHALL BE DEEMED TO BE A WARRANTY BY ASCENTIAL FOR ANY PURPOSE OR GIVE RISE TO ANY LIABILITY OF ASCENTIAL WHATSOEVER. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL ASCENTIAL BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. If you are acquiring this software on behalf of the U.S. government, the Government shall have only "Restricted Rights" in the software and related documentation as defined in the Federal Acquisition Regulations (FARs) in Clause 52.227.19 (c) (2). If you are acquiring the software on behalf of the Department of Defense, the software shall be classified as "Commercial Computer Software" and the Government shall have only "Restricted Rights" as defined in Clause 252.227-7013 (c) (1) of DFARs.

This product or the use thereof may be covered by or is licensed under one or more of the following issued patents: US6604110, US5727158, US5909681, US5995980, US6272449, US6289474, US6311265, US6330008, US6347310, US6415286; Australian Patent No. 704678; Canadian Patent No. 2205660; European Patent No. 799450; Japanese Patent No. 11500247.

© 2005, 2000-2004 Ascential Software Corporation. All rights reserved. DataStage®, EasyLogic®, EasyPath®, Enterprise Data Quality Management®, Iterations®, Matchware®, Mercator®, MetaBroker®, Application Integration, Simplified®, Ascential™, Ascential AuditStage™, Ascential DataStage™, Ascential ProfileStage™, Ascential QualityStage™, Ascential Enterprise Integration Suite™, Ascential Real-time Integration Services™, Ascential MetaStage™, and Ascential RTI™ are trademarks of Ascential Software Corporation or its affiliates and may be registered in the United States or other jurisdictions.

Adobe Systems, Inc. Microsoft, Windows, Windows NT, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company, Ltd. Other marks mentioned are the property of the owners of those marks.

The software delivered to Licensee may contain third-party software code. See *Legal Notices* (**LegalNotices.pdf**) for more information.

How to Use This Guide

The DataStage Complex Flat File stage converts data extracted from complex flat files that are generated on an IBM mainframe. Complex flat files typically have hierarchical structure or include legacy data types. Version 2.0 of DataStage Complex Flat File is compatible with Ascential DataStage Release 7.5.1.

Audience

This guide is intended for DataStage designers who create or modify jobs that use the Complex Flat File stage.

How This Book is Organized

The following table lists topics that may be of interest to you and it provides links to these topics.

To learn about	Read...
Functionality	"Functionality" on page 2
Terminology	"Terminology" on page 3
Installing the stage	"Installing the Plug-In" on page 4
Defining the Complex Flat File stage	"Defining the Complex Flat File Stage" on page 4
Defining an output link	"Defining an Output Link" on page 6
Date considerations	"Date Considerations" on page 17

Related Documentation

To learn more about documentation from other Ascential products and third-party documentation as they relate to the Complex Flat File stage, refer to the following section/table.

Ascential Software Documentation

Guide	Description
<i>Ascential DataStage Server Job Developer's Guide</i>	Instructions for using a stage in a DataStage job
<i>Ascential DataStage Designer Guide</i>	General principles for designing jobs
<i>Ascential MetaStage User's Guide</i>	Information about Ascential MetaStage™
<i>Ascential DataStage NLS Guide</i>	Information about NLS and techniques for character-set mapping
<i>Ascential DataStage Plug-In Installation and Configuration Guide</i>	Information required to configure your system and install this stage

Conventions

Convention	Used for...
bold	Field names, button names, menu items, and keystrokes. Also used to indicate filenames, and window and dialog box names.
user input	Information that you need to enter as is.
code	Code examples
<i>variable</i> or <variable>	Placeholders for information that you need to enter. Do not type the greater-/less-than brackets as part of the variable.
>	Indicators used to separate menu options, such as: Start >Programs >Ascential DataStage
[A]	Options in command syntax. Do not type the brackets as part of the option.
B...	Elements that can repeat.

Convention	Used for...
A B	Indicator used to separate mutually-exclusive elements.
{ }	Indicator used to identify sets of choices.

Contacting Support

To reach Customer Care, please refer to the information below:

Call toll-free: 1-866-INFONOW (1-866-463-6669)

Email: support@ascentialsoftware.com

Ascential Developer Net: <http://developernet.ascential.com>

Ascential eService: <http://www.ascential.com/eservice>

Please consult your support agreement for the location and availability of customer support personnel.

To find the location and telephone number of the nearest Ascential Software office outside of North America, please visit the Ascential Software Corporation website at <http://www.ascentialsoftware.com>.

Contents

Audience	iii
How This Book is Organized	iii
Related Documentation	iv
Ascential Software Documentation	iv
Conventions	iv
Contacting Support	v
Introduction	1
Existing Jobs Built with Version 1 of the Complex Flat File Stage	2
Functionality	2
Terminology	3
Installing the Plug-In	4
Defining the Complex Flat File Stage	4
NLS Tab	5
Defining an Output Link	6
About the Output Page	6
General Tab	7
Source Columns Tab	9
Edit Column Meta Data Dialog Box	10
Select Columns Tab	14
Selection Criteria Tab	15
Destination Columns Tab	17
Date Considerations	17

Introduction

The Ascential DataStage Complex Flat File stage lets you convert data extracted from complex flat files that are generated on an IBM mainframe. A complex flat file has hierarchical structure in its arrangement of columns. It is physically flat (that is, it has no pointers or other complicated infrastructure,) but logically represents parent-child relationships. You can use multiple record types to achieve this hierarchical structure.

Recognizing a Hierarchical Structure

For example, use records with various structures for different types of information, such as an 'E' record for employee static information, and a 'S' record for employee monthly payroll information, or for repeating groups of information (twelve months of revenue). You can also combine these record groupings, and in the case of repeating data, you can flatten nested OCCURS groups.

Managing Repeating Groups and Internal Structures

You can easily load, manage, and use repeating groups and internal record structures such as GROUP fields and OCCURS. You can ignore GROUP data columns that are displayed as raw data and have no logical use for most applications. The meta data can be flattened into a normalized set of columns at load time, so that no arrays exist at run time.

Selecting subsets of columns

You can select subset of columns from a large COBOL File Description (CFD). This filtering process results in performance gains since the stage no longer parses and processes hundreds of columns if you only need a few.

Complex flat files can also include legacy data types.

Output Links

The Complex Flat File stage is a source stage that supports multiple outputs. An output link specifies the data you are extracting, which is a stream of rows to be read.

When using the Complex Flat File stage to process a large number of columns, for example, more than 300, use only one output link in your job. This dramatically improves the performance of the GUI when loading, saving, or building these columns. Having more than one output link causes a save or load sequence each time you change tabs.

The Complex Flat File stage does not support reference lookup capability or input links. For information about jobs built with Version 1 of this stage, see ["Existing Jobs Built with Version 1 of the Complex Flat File Stage" on page 2.](#)

Existing Jobs Built with Version 1 of the Complex Flat File Stage

When you first open the Complex Flat File stage, the grid for the **Source Columns** tab is empty for existing jobs until you click **OK** and save the job. The GUI generates the source column values using the output columns list, and you can upgrade existing jobs by selecting **Yes** to the following query:

This stage was developed with a previous version of the CFF plugin. Would you like to upgrade this stage?

The source columns are populated, and you can select and modify columns.

You can stop the upgrade by selecting **No** to the query. You cannot use the **Source Columns** tab, and grid remains empty. Existing jobs run unaltered as long as you do not save the job.

Functionality

The Complex Flat File stage supports the following functionality:

- EBCDIC and ASCII raw data.
- The following COBOL data types:
 - Packed (COMP-3).
 - Zoned (signed DISPLAY).
 - Character (DISPLAY or PIC X).
 - Integer (COMP. Integer also supports non-standard 1- and 3-byte implementations.)
 - Binary (same as COMP, but unsigned.)
 - Decimal (COMP-2).
 - Float (COMP-1).
 - Fixed format structures. It does not support columns that are separated by delimiters.
- One REDEFINES clause.
- Parallel OCCURS using separate output links.
- Nested OCCURS in the same COBOL File Description (CFD) format. They must be flattened at load time (not at run time).
- Fixed OCCURS with no DEPENDING ON clause.
- Complex flat files generated on an IBM mainframe.

- NLS (National Language Support). See *Ascential DataStage NLS Guide*.
- The ability to select subsets of columns from a CFD.

The following functionality is not supported:

- Non 8-bit byte file formats.
- Complex flat files generated on non-IBM mainframes (for example, Windows NT).

Terminology

The following table lists the Complex Flat File stage terms used in this document:

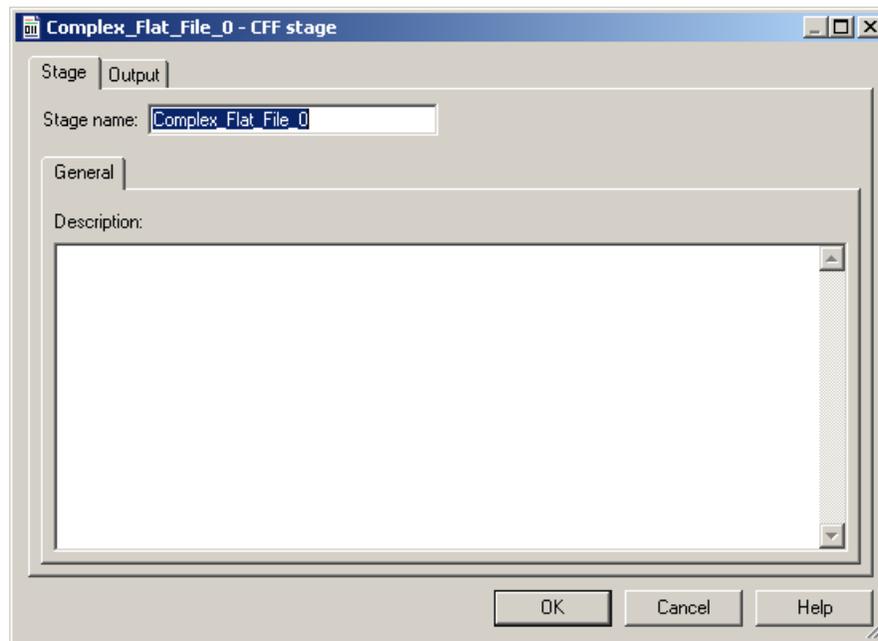
Term	Description
CFD	COBOL File Description format.
Complex flat file	A superset of the simple flat file, but it contains complex data structures. The complex data structures supported by the Complex Flat File stage are groups, arrays, and redefines, which are found in VSAM or QSAM files. (Load-ready or delimited files do not contain these complex structures.) A complex flat file has hierarchical structure implied in its arrangement of columns. Complex flat files can also include legacy data types.
DCLGen Import	The Ascential DataStage Manager component to import a table definition in a DCLGen file that was previously exported from VS/DB2.
Fixed-width flat file	A file characterized by delimited or fixed length (binary) files.
Flattening	The conversion of files containing complex data structures, such as arrays, groups, and redefines, into data files that contain records with no structured relationships.
Normalization	The conversion of records in NF ² (nonfirstnormal form) format, containing multivalued data, into one or more 1NF (first normal form) rows.
QSAM	Queued Sequential Access Method.
VSAM	Virtual Storage Access Method. This method is a file management system for the IBM MVS mainframe operating system.

Installing the Plug-In

For instructions and information supporting the installation, see *Ascential DataStage Plug-In Installation and Configuration Guide*.

Defining the Complex Flat File Stage

When you use the custom GUI to edit a Complex Flat File stage, the **CFF Stage** dialog box appears:



This dialog box has the **Stage** and **Output** pages:

- **Stage.** This page displays the name of the stage you are editing in the **Stage name** field. The **General** tab describes the purpose of the stage in the **Description** field. (The **NLS** tab appears only if you have installed NLS. For details, see "[NLS Tab](#)" on page 5.)

Note You cannot change the name of the stage from this dialog box. For details on changing stage names, see *Ascential DataStage Designer Guide*.

- **Output.** This page specifies the data sources to use and the associated column definitions for each output link.

NLS Tab

You can define a character set map that interprets the input file for a stage. Do this on the **NLS** tab on the **Stage** page, which appears only if you have installed NLS.

If NLS is installed, the **NLS** option is selected in the **Data Format** list box on the **General** tab of the **Output** page, and the **NLS** tab on the **Stage** page is enabled. 1

Specify information using the following button and fields:

- **Map name to use with stage.** The default character set map is defined for the project or the job. You can change the map by selecting a map name from the list.
- **Use Job Parameter...** . Specifies parameter values for the job. Use the format *#Param#*, where *Param* is the name of the job parameter. The string *#Param#* is replaced by the job parameter when the job is run.
- **Show all maps.** Lists all the maps that are shipped with Ascential DataStage.
- **Loaded maps only.** Lists only the maps that are currently loaded.

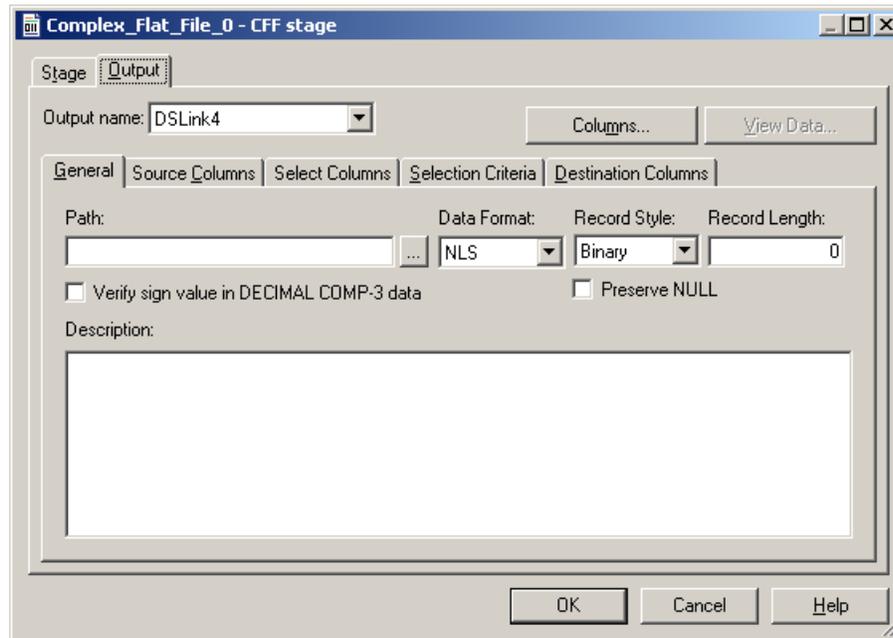
If NLS is not installed, the **NLS** option is unavailable in the **Data Format** list box and the **NLS** tab on the **Output** page is disabled. The input file is interpreted according to the option that is selected in the **Data Format** list box on the **General** tab of the **Output** page.

For more information about NLS, see *Ascential DataStage NLS Guide*, *Ascential DataStage Designer Guide*, or *Ascential DataStage Administrator Guide*.

Defining an Output Link

When you read data from a data source, the Complex Flat File stage has an output link.

Define the properties of this link and the column definitions of the data on the **Output** page in the **CFF Stage** dialog box.



About the Output Page

The **Output** page has an **Output name** field, the **General**, **Source Columns**, **Select Columns**, **Selection Criteria**, and **Destination Columns** tabs, and the **Columns...** and **View Data...** buttons. (The **NLS** tab appears only if you have installed NLS. For details, see “[NLS Tab](#)” on [page 5](#).)

- **Output name.** The name of the output link. Select the link you want to edit from the **Output name** list box. This list displays all the output links from the Complex Flat File stage.
- **Columns... button.** Click the **Columns...** button to display a brief list of the columns in the data source file. As you enter detailed meta data on the **Source Columns** tab, you can leave this list displayed.
- **View Data... button.** Click the **View Data...** button to start the Data Browser. This lets you look at the data associated with the output link. For a description of the Data Browser, see *Ascential DataStage Designer Guide*.

General Tab

This tab is displayed by default. You can optionally enter text to describe the purpose of the output link in the **Description** field. Enter the appropriate information for the following fields:

- **Path.** The input pathname of the data source to retrieve data from. You can also click the ... button at the right of the field to browse the directories on the server for the data source.
- **Data Format.** The format of the input file: EBCDIC or ASCII, or NLS. If NLS is enabled, the Data Format is set to NLS and is uneditable. (NLS maps support EBCDIC data.)
- **Record Style.** The end-of-line treatment for records according to the following table:

Data Format	Record Style	Record Length	Comments
ASCII or EBCDIC	Binary	Greater than zero	The record length is determined by the Record Length field. If the meta data defines a record that is longer than the Record Length field, the columns that have start positions greater than Record Length are set to Null.
ASCII or EBCDIC	Binary	Zero	The record length is determined by the meta data.
ASCII or EBCDIC	CR/LF	Any value	The record length is determined by the CR/LF delimiter. If the meta data defines a record that is longer than Record Length as determined by the CR/LF, the columns that have start positions greater than the position of the CR/LF are set to NULL. If the meta data defines a record that is less than the position of the CR/LF, the data after the end of the record and before the CR/LF (as determined by the meta data) is discarded. If the data being read is in EBCDIC format, coincidental CR/LF characters can occur. These also delimit a record.
NLS	Binary	> 0	The record length is determined by the Record Length field. If the meta data defines a record that is longer than the Record Length field, the columns that have start positions greater than Record Length are set to spaces.

Data Format	Record Style	Record Length	Comments
NLS	CR/LF	Any value	<p>The record length is determined by the CR/LF delimiter.</p> <p>If the meta data defines a record that is longer than Record Length as determined by the CR/LF, the columns that have start positions greater than the position of the CR/LF are set to spaces</p> <p>If the meta data defines a record that is less than the position of the CR/LF, the data after the end of the record and before the CR/LF (as determined by the meta data) is discarded.</p> <p>If the data being read is in EBCDIC format, coincidental CR/LF characters can occur. These also delimit a record.</p>
NLS	Binary	0	The record length is determined by the meta data.

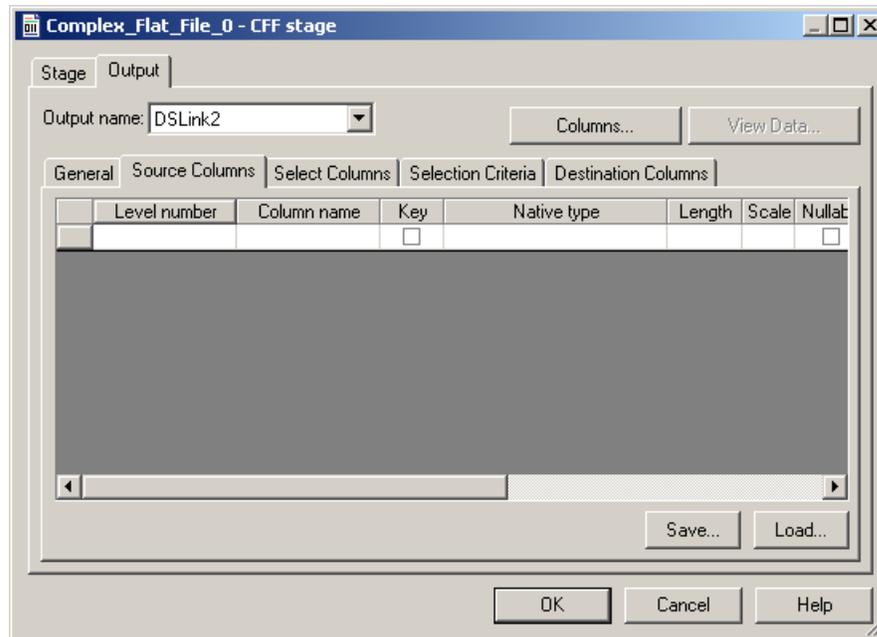
Note If **Record Style** is set to CR/LF, the last record in the data set should not be terminated by CR/LF. If one or more CR/LFs are at the end of the data set, empty records are generated for each CR/LF.

- **Verify sign value in DECIMAL COMP-3 data.** If selected, the stage checks for a valid sign value in data defined as COMP-3. If the value is other than a hexadecimal "C," "F," or "D," the Plug-in writes an error message to the DataStage log.
- **Preserve NULL.** If selected, the stage interprets the value of any column containing binary null values as a SQL_NULL. If not selected, the stage interprets the value as zero. The default is **Preserve NULL** not selected.
- **Description.** Optional text describing the purpose of the output link.

Source Columns Tab

The **Source Columns** tab contains the full list of columns. Using the **Select Columns** tab, you can select columns from this list to output on the CFF output link. (You can also use the Transformer stage to do this, but slower performance results because of the unused column data.)

Use the **Source Columns** tab to enter the column data that make up the flat file manually, or you can use **Load**.



Note Do not use the **Source Columns** tab to omit columns from the destination file. **Source Columns** must accurately and completely describe the source file, or the file will not be parsed correctly. Rather, use the **Select Columns** tab, described on [page 14](#), to omit columns.

Load Button

Load automatically flattens the arrays by generating column names with monotonically increasing numbers if you answer **Yes** to the following query:

Do you want to flatten the occurs in the columns being loaded?

If you enter data manually, you should flatten any arrays manually.

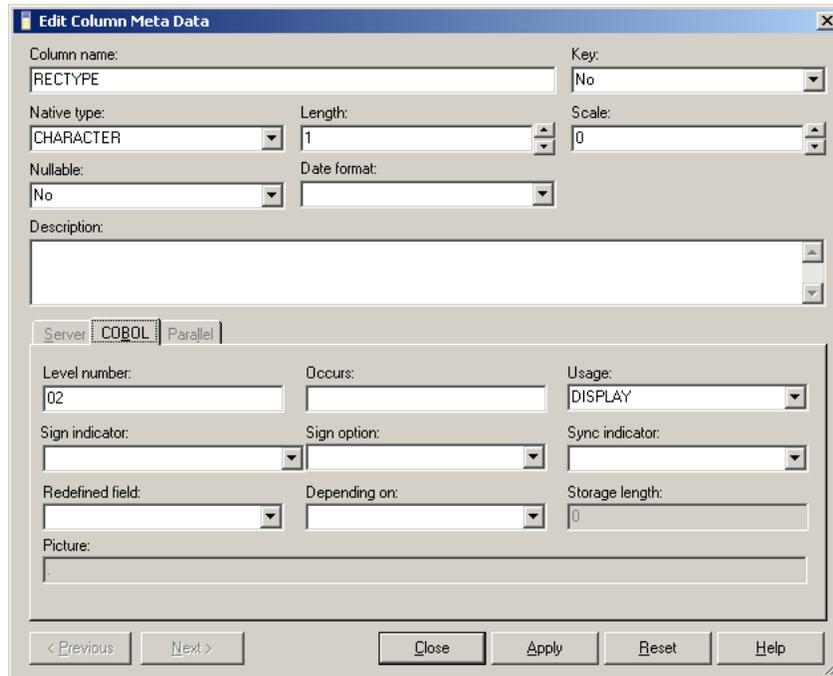
The following columns have special meaning for the Complex Flat File stage:

- **Level number.** Represents the level number of the column within a COBOL file description.
- **Native type.** Represents the COBOL data type.

The other columns in the grid are the standard columns for the **Columns** tab. (See *Ascential DataStage Designer Guide*.) You can edit these fields by right-clicking on a row in the grid and selecting **Edit row...** . This opens the **Edit Column Meta Data** dialog.

Edit Column Meta Data Dialog Box

Use the Edit Column Meta Data dialog box to edit the column meta data as described in this section.



The **Edit Column Meta Data** dialog box has a general area containing fields that are common to all data source types, plus two pages containing fields specific to meta data used in server jobs and information specific to COBOL data sources.

Meta Data Properties

Use the Ascential DataStage Manager to import the complex flat file meta data (CFD) into Ascential DataStage. The following table defines the properties that are captured for complex flat file meta data.

Field	Description
Column name	The name of the column.
Key	Specifies whether this column is a key.

Field	Description (Continued)
Native type	The native data type of the column. Use one of the following values: FLOAT, DECIMAL, BINARY, DISPLAY_NUMERIC, CHARACTER, or GROUP. For details about using the GROUP native type to handle dates, see " Destination Columns Tab " on page 17 .
Length	The numeric value of the precision.
Scale	The number of decimal places.
Nullable	Specifies whether the column can contain null values. If set to Yes, it is subject to a NOT NULL constraint. (Yes/No)
Date format	The format of a date column.
Description	The description of the column. See examples of Description field values at the end of this section.
Level number	The relative COBOL level number of the column.
Occurs	The number of occurrences of the column specified in the COBOL OCCURS clause. (See examples of parallel OCCURS, which are unsupported, at the end of this section.)
Usage	Specifies a COBOL usage clause. Use one of the following: COMP COMP-1 COMP-2 COMP-3 DISPLAY
Sign indicator	Specifies the sign. Set it to S if the picture character string contains the S symbol. Otherwise, set it to U.
Sign option	If you specify a sign clause and the picture character-string contains an S symbol, this attribute is set to one of the following values: L - LEADING T - TRAILING LS - LEADING SEPARATE TS - TRAILING SEPARATE
Sync indicator	Specifies whether this is a COBOL synchronized clause.
Redefined field	The name of the column being redefined.
Depending on	The name of the depending column.
Storage length	The actual storage length in bytes of the column.
Picture	Displays a generated Picture clause based on the value in Native type , Length , and Scale .

Processing the Meta Data

Use the buttons at the bottom of the **Edit Column Meta Data** dialog box to continue adding or editing columns, or to save the changes and close the dialog box.

- **Previous** and **Next**. View the meta data in the previous or next row.
- **Close**. Close the **Edit Column Meta Data** dialog box. If there are outstanding changes to the current row, you are asked whether you want to save them before closing.
- **Apply**. Save changes to the current row.
- **Reset**. Remove all changes made to the row since the last time you applied changes.

Only a subset of these properties is visible on the **Source Columns** tab. To see all the properties for a given row, right-click on a row in the grid, and select **Edit row....**

If you enter or modify meta data using the stage editor, and you want to save a copy in the DataStage Repository for use in another stage, click the **Save...** button.

To load an existing table definition into the stage, use the **Load...** button.

Handling Parallel OCCURS

The Complex Flat File stage does not support parallel OCCURS, that is, two or more OCCURS clauses in the same data definition. You need to process these parallel OCCURS clauses down separate output links.

This example uses a PHONES OCCURS clause and an ADDRESS OCCURS clause:

```
01 CLIENT.  
  03 SURNAME          PIC X(25).  
  03 FORENAME         PIC X(25).  
  03 ADDRESS          OCCURS 4.  
    05 ADDLINE        PIC X (10).  
  03 POSTCODE         PIC X (10).  
  03 PHONES           OCCURS 2.  
    05 TELNO          PIC X(10).
```

The next example uses a PHONES OCCURS clause and an ADDLINE OCCURS clause:

```
01 CLIENT.  
  03 SURNAME          PIC X(25).  
  03 FORENAME         PIC X(25).  
  03 ADDRESS.  
    05 ADDLINE        PIC X (10) OCCURS 5.  
  03 POSTCODE         PIC X (10).  
  03 PHONES           OCCURS 2.  
    05 TELNO          PIC X(10).
```

The CFF custom GUI recognizes the parallel OCCURS and displays the following error:

```
Too many occurs
```

You are not allowed to save the loaded or edited column definitions.

To process parallel OCCURS:

- 1 Clear the **Occurs** field using the **Edit Column Meta Data** dialog box.
- 2 Enter NONE in the **Description** field of the columns that are not being processed on this link. This lets the data for those columns flow through unchanged.
- 3 Create a separate output link using a similar procedure to process the next OCCURS.

Description Field Values

The following values for the **Description** field have special meaning at run time:

- **UNSIGNED_DECIMAL.** Use only with DECIMAL fields. You can use this value with packed decimal fields to trigger special unpacking algorithms.
- **ANYSIGN_DECIMAL.** Use only with DECIMAL fields. You can use this value with packed decimal fields to trigger special unpacking algorithms.
- **NONE.** Use with non-GROUP native types. NONE causes the data to flow through the stage unchanged, that is, no conversions are done on the data, and raw data is output. NONE is ignored if you use a date format.
- **OCCURS_COUNTER.** Behaves as a pseudo-column of the DECIMAL native type that does not expect any data in the input stream.

You must first insert a new field with the DECIMAL type into the **Columns** grid within your OCCURS clause. You must also include the OCCURS_COUNTER string in the **Description** field. At run time the stage creates its own data by automatically incrementing the counter for each occurrence that is processed. This is not currently supported for nested OCCURS.

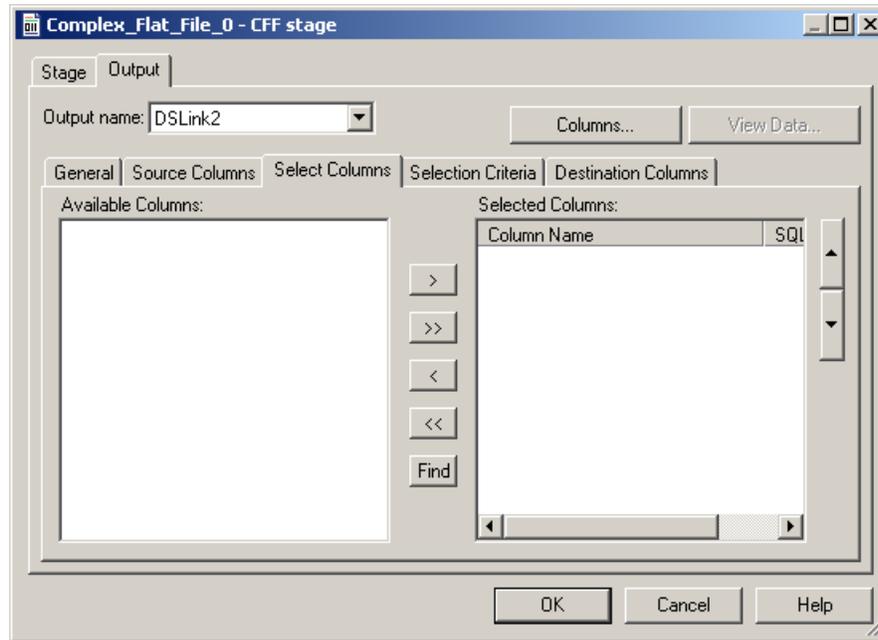
GROUP Columns and OCCURS

If a column in a GROUP that has an OCCURS is selected, and the GROUP column is not selected, incorrect results may be displayed. You should include the GROUP column with the OCCURS in your selection if any column in the GROUP is selected.

Note Only certain types of GROUP columns may be selected. See the following sections for details.

Select Columns Tab

Use the **Select Columns** tab to choose which columns to load on the output link.



The **Select Columns** tab contains a grid displaying the column definitions for the data being output on the chosen link.

Note Do not use the **Source Columns** tab for the purpose of selecting columns for the destination file. Use only the **Select Columns** tab for this purpose. The description on the **Source Columns** tab must match the source file completely and accurately (see [page 9](#)).

The **Select Columns** tab functions similarly to that of the Complex Flat File mainframe source stage, as follows:

- **Available columns** lists the source columns displayed in hierarchical format. It uses fields for non-GROUP columns and folders for GROUP columns. As you select or clear each column, a check mark appears on the column in the list.

You can only select GROUP columns if all the columns in the GROUP have the CHARACTER data type. If any column in the GROUP has a different data type, you cannot select the GROUP column and is displayed as such.

- **Selected columns** contains the list of columns you create using the arrow keys.

- Use these arrow keys to move columns back and forth between the **Available columns** list and the **Selected columns** list. Use the single arrow (>) to move highlighted columns, the double arrow (>>) to move all items.
- By default all columns are selected for loading. Click **Find** to open a dialog box which lets you search for a particular column.
- Click **OK** when your selection is complete to load the selected columns.

Selection Criteria Tab

Use the **Selection Criteria** tab to redefine fields extracted from the input file.

The **Selection Criteria** tab appears:

Enter the appropriate information for the following fields:

- **Start Record #.** The record number at which to start processing.
- **End Record #.** The record number at which to stop processing.
- **ID Field.** Choose the field containing the record type value from the list box.
- **Value (Hex).** The value of the record type. This value is converted to the **Record Style** before comparison, for example, ASCII or EBCDIC. Only the records that contain this value are sent to the output link. Value ranges are unsupported. If the value is preceded by the ampersand (&) character, it is treated as a hexadecimal value and compared without any conversion.

Redefined Example

The Complex Flat File stage supports the redefinition of any portion of the source file. It does this by resetting the start position of a field that has its meta data redefine another field.

For example, Field-2 redefines Field-1, and so forth.

Input:

```
01 Example-Record.
  03 Field-1      Pic X(24).
  03 Field-2 redefines Field-1.
    05 Field-2a Pic X(8).
    05 Field-2b Pic X(8).
    05 Field-2c Pic X(8).
  03 Field-3      Pic X(24).
  03 Field-3 redefines Field-3.
    05 Field-4a  Pic X(8).
    05 Field-4b  Pic X(8).
    05 Field-4c  Pic X(8).
  03 Field-5 redefines Field-1.
    05 Field-5a  Pic X(8).
    05 Field-5b  Pic X(8).
    05 Field-5c  Pic X(8).
```

Input Data:

2a2a2a2a2b2b2b2b2c2c2c2c4a4a4a4a4b4b4b4b4c4c4c4c

Output Field Order:

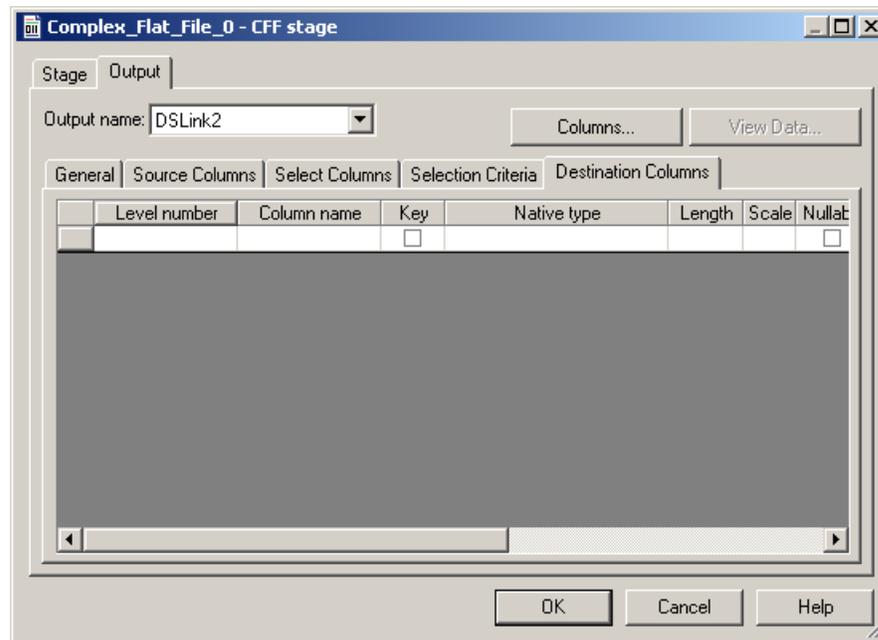
- Field-2a
- Field-2b
- Field-2c
- Field-4a
- Field-4b
- Field-4c
- Field-5a
- Field-5b
- Field-5c

Output Data:

2a2a2a2a2b2b2b2b2c2c2c2c4a4a4a4a4b4b4b4b4c4c4c4c2a2a2a2a2b2b2b2b2c2c2c2c

Destination Columns Tab

The **Destination Columns** tab on the **Output** page contains the list of columns that you created using the **Select Columns** tab.



The columns are grayed out and uneditable. You must use the **Select Columns** tab to choose which columns to load on the output link.

Date Considerations

In many cases, COBOL files define dates as a character field. For example:

```
05 Application-Date pic 99999999.
```

Click the **Source Columns** tab on the **Output** page to enter or load column definitions for your data. In this case, define the **Native type** field for Application-Date as CHARACTER. Select an appropriate format from the **Date format** field of the **Edit Column Meta Data** dialog, in this case, CCYYMMDD.

To generate a DataStage date in the column in the output link, the input data and the **Date format** field must use the same format. For example, the input data in the format "25/12/2000" must use the DD/MM/CCYY format in the **Date format** field. Otherwise, a date with a null value is generated, and a warning about a bad date conversion appears in the DataStage log.

