

ETI Data System Library for C/DB2 DA

Release Notes

Revision 4.2.1A

August, 2002

Copyright © 2002 by Evolutionary Technologies International, Inc. All rights reserved.

Evolutionary Technologies International, ETI, ETI Solution, ETI•EXTRACT, the ETI logo, and MetaStore are trademarks or registered trademarks of Evolutionary Technologies International, Inc.

All other product and company names mentioned are included for identification only and may be trademarks and/or registered trademarks of their respective companies or institutions.

Table of Contents

1	PREREQUISITES	3
2	DSL INSTALLATION	3
	IF YOU HAVE ETI•EXTRACT INSTALLED.....	3
	INSTALLING THE DSL	3
3	IMPORTANT NOTICES	4
3.1	SINGLE STEP IS DEFAULT	4
3.2	READ FILTER MODIFICATIONS	4
3.3	METHOD FOR DYNAMIC MEMORY ALLOCATION WAS CHANGED	4
3.4	DATE ARITHMETIC FUNCTIONALITY WAS CHANGED	4
3.5	DOCUMENTATION ADDENDUM	4
4	RELEASE 4.2.1	5
4.1	PURPOSE.....	5
4.2	FEATURES IN RELEASE 4.2.1	5
4.2.1	<i>Conversion Processing</i>	5
4.2.2	<i>Single-Step Conversions</i>	5
4.2.3	<i>Multi-Step Conversions</i>	6
4.2.4	<i>Controlling the Type of Conversion Generated</i>	6
4.2.5	<i>Potential Differences in Conversion Results with Single-Step Versus Multi-Step Conversions</i>	7
4.2.6	<i>In-Memory Lookup</i>	8
4.2.7	<i>Select Distinct Option for Selecting Records Conditionally</i>	8
4.2.8	<i>New Option for Validating or Reformatting a String that Contains a Date</i>	8
4.2.9	<i>New Option for Connecting as a Different User</i>	9
4.2.10	<i>New Default Value for shell_flags_compiler</i>	9
4.2.11	<i>Obsolete Property shell_flags_precompiler_cdb2</i>	9
4.2.12	<i>Support Added for DB2 6.x Retrieve Schema</i>	9
5	RELEASE 4.2.0	9

1 PREREQUISITES

The ETI Data System Library (DSL) for C/DB2 DA requires the following products to be at the indicated release number *or later*:

- **ETI•EXTRACT® Tool Suite Release 4.2.2** or **ETI Solution® Version 5.0.0**. Required to ensure that the DSL will generate the expected code.
- **Shared Objects 4.2.2**. This component will be loaded automatically during the installation of the DSL when you select the option to auto-load the prerequisites.
- **TCL Functions 4.2.1**. This component will be loaded automatically during the installation of the DSL when you select the option to auto-load the prerequisites.
- **C Intermediate Actions 4.2.1**. This component will be loaded automatically during the installation of the DSL when you select the option to auto-load the prerequisites.

2 DSL INSTALLATION

If You Have ETI•EXTRACT Installed

To install ETI Data System Libraries, follow the directions in the manuals listed below, depending on the version of the software you are using:

For ETI•EXTRACT Tool Suite Release 4.2.2:

- Windows: *ETI•EXTRACT Tool Suite: Installation Guide for Windows*, Chapter 10, “Installing Data System Libraries”
- UNIX: *ETI•EXTRACT Tool Suite: Installation Guide for UNIX*, Chapter 6, “Installing Data System Libraries”

For ETI Solution Version 5.0.0:

- *ETI Solution Administration Guide*, Chapter 3, “Installing and Loading ETI Data System Libraries”

Warning: If you do not follow the DSL installation procedures described in the manuals listed above, but instead manually copy files from the CD-ROM, then you will not get the updated version of the DSL install script, which will cause the DSL installation to fail.

Installing the DSL

With the new DSL patch process, a DSL can be shipped with any available patches. To install the DSL simply follow the normal procedure for installing a DSL. If patches are available you will be prompted to load the patches, answer “yes”.

3 IMPORTANT NOTICES

3.1 Single Step is Default

Beginning with Release 4.2.1, the DSL for C/DB2 DA is designed to generate single-step conversion programs by default if at all possible.

3.2 Read Filter Modifications

The read_part filter has been modified to correct a problem whereby a read grammar involving conditional modification of the part value would use an incorrect type for the part to modify. A modification to the read_part filter to correct the problem changes the order of the prompts in the read_part filter. The filter translation and generated code is not affected, therefore existing read_part filters have a valid translation and generate valid code but will no longer replay successfully.

ETI™ has developed a migration utility **ex_fixFilters.pl** that will rearrange the saved responses from existing read_part filters into the order now required by the read_part filter. The migration utility **ex_fixFilters.pl** can be located in the MetaStore's™ dsl/bin directory. Refer to the documentation for the **ex_fixFilters.pl** utility in the “Miscellaneous Documentation” section of the *ETI Data System Library DA: Documentation Addendum* for description and usage of the **ex_fixFilters.pl** utility.

3.3 Method for Dynamic Memory Allocation Was Changed

The method of dynamic memory allocation in the C language-based DSLs has changed dramatically in the 4.2 release. The new method does not make memory allocation system calls for each variable needing storage, for example temporary variables, intermediate filter variables, and so forth. Instead, a single system call is made during the initialization of the program to create a large buffer that can be used, flushed, and potentially increased per iteration of the process flow. This new method may provide a dramatic increase in performance.

3.4 Date Arithmetic Functionality Was Changed

The date arithmetic functionality has changed in the C language-based DSLs. This change offers a wider range of supported dates. The new methods will process any date after October 15, 1582, which is the start of the Gregorian calendar, and will use messages ETI546 and/or ETI547 to warn the user if the resulting date is *before* October 15, 1582.

3.5 Documentation Addendum

The *ETI Data System Library DA: Documentation Addendum* contains documentation errata and updates since the release of the *ETI Data System Library DA: Procedures* and *ETI Data System Library DA: Reference* manuals. You should review the addendum for this information.

4 RELEASE 4.2.1

4.1 Purpose

This DSL now supports single-step conversion processing and in-memory lookup. See the section “Single-Step Conversions” on page 5 for information about single-step. Refer to the section “In-Memory Lookup” on page 8 for information about in-memory lookup.

A series of bug fixes and enhancements to the architecture have been implemented. A complete list of CRs (software problems) fixed in this release can be provided by the AnswerLine upon request.

4.2 Features in Release 4.2.1

4.2.1 Conversion Processing

By default, ETI•EXTRACT attempts to generate conversions with the fewest number of execution steps (conversion programs) possible. Minimizing the number of execution steps simplifies the work of the user, as there are fewer steps to execute, monitor, and verify.

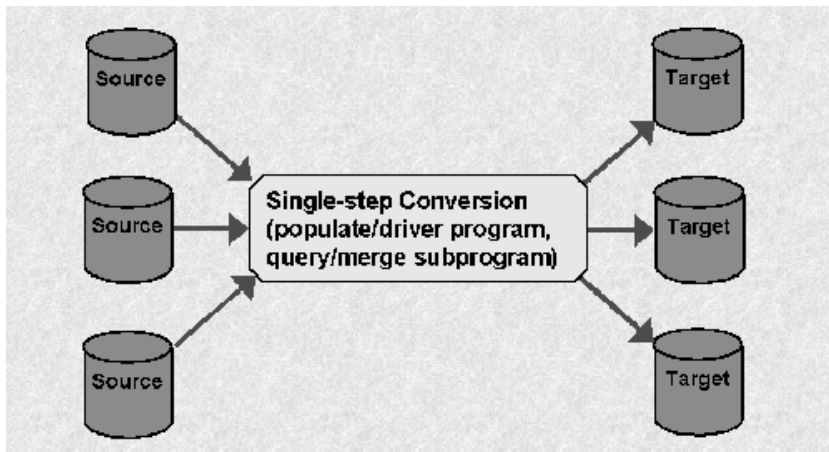
A single-step conversion is usually more efficient than a multi-step conversion. However there are constraints that must be met for ETI•EXTRACT to be able to generate a single-step conversion. There may also be cases when you will get the most efficient solution by forcing ETI•EXTRACT to generate a multi-step conversion.

4.2.2 Single-Step Conversions

With single-step conversions, ETI•EXTRACT generates the following:

- a subroutine that queries the source database(s) and merges the data as required
- a main populate routine that calls the query subroutine and populates the target databases

The generation of a single executable program eliminates the I/O operations required to pass the data through intermediate files and programs. A simplified illustration of the flow of data is shown in the following diagram:



Since a single conversion program runs on one computer, the execution of the conversion is simple and easy to manage.

If ETI•EXTRACT determines that a single-step conversion is not possible, it generates a multi-step conversion.

4.2.2.1 Requirements for Single-Step

A single-step conversion can be generated when:

- One source and one target database object, each created using the same installation object.
An installation object represents a Data Access System (DAS), (cdb2_da, for example) running on a particular host computer. Therefore, the source and target databases must use the same DAS and be running on the same host.
- The business rules for the conversion can be processed in a single-step.
- ETI•EXTRACT has determined there is no need to write data to an ifile. If ifiles are needed, then single-step is not possible.

4.2.2.2 Message Program Information

ETI•EXTRACT writes messages to three files with the following names:

- `<conversion_name><instruction_name>.log`
- `<conversion_name><instruction_name>.wrn`
- `<conversion_name><instruction_name>.err`

where `<conversion_name>` is either the name or an abbreviation of the name of the conversion and `<instruction_name>` is the name of the instruction that wrote the message (or an abbreviation of the instruction name).

In single-step conversions there is one executable program and one set of message files. In that case the instruction is populate1. For example, the message files for the single-step conversion SSCONV would be called:

- SSCONVpopulate1.log
- SSCONVpopulate1.wrn
- SSCONVpopulate1.err

4.2.2.3 Overview of Differences between Pre-DA Single-Step and DA Single-Step

Please refer to the section “Differences with Pre-DA Single-Step” in the *ETI Data System Library DA: Procedures* manual, in Chapter 15 “Single-Step and Multi-Step Conversions”.

4.2.3 Multi-Step Conversions

With a multi-step conversion, there is at least one executable query instruction and one executable populate instruction. Data passes through at least one ifile. A multi-step conversion may or may not require separate intermediate actions (separate executable instructions for sorting and merging data) and associated ifiles.

4.2.4 Controlling the Type of Conversion Generated

This section describes how you can generate different types of conversions.

4.2.4.1 Generating Single-Step Conversions

ETI•EXTRACT will automatically generate a single-step conversion when possible.

4.2.4.1.1 Reasons to Generate a Single-Step Conversion

A single-step conversion typically offers several advantages over a multi-step conversion:

- faster throughput
- lower disk space requirements
- less CPU usage
- simpler execution plans

These advantages result from the fact that a single-step conversion program will not pass data through any intermediate files (ifiles). Multi-step conversions using individual programs create one or more ifiles, written to disk, for subsequent processing by intermediate action programs and by the final populate programs.

Using a single-step conversion may prove most advantageous when you:

- are converting very large files
- have limited disk space

A multi-step conversion may require that you perform the conversion in phases because it:

- takes too long to execute, due to the time it takes to read and write ifiles
- limits the number of records that you can convert at one time due to ifile size limitations

There are times, however, when you may want to generate a multi-step conversion instead of a single-step, as described below.

4.2.4.2 Generating Separate Query and Populate Programs with Ifiles

Set the value of `ifile_io` to `true` to generate a multi-step with ifiles and separate query and populate programs.

4.2.4.2.1 Reasons to Generate Separate Query and Populate Programs with Ifiles

You might want to do this when:

- The ifiles will be used as input to a non-ETI process, outside of the ETI•EXTRACT conversion.
You should use caution if you use ifiles for this purpose. Subtle changes in the conversion specification will cause the ifiles to change. If you are relying on the ifile data to be consistent over time, this might be a problem.
- You want to be certain that the query has worked satisfactorily before executing the populate.
- There is a customer requirement for separate access to the source and the target databases (they are not available at the same time).

4.2.5 Potential Differences in Conversion Results with Single-Step Versus Multi-Step Conversions

4.2.5.1 Spaces In Source Data Fields

In multi-step conversions, source data fields that contain all spaces are treated as null when they are written to the ifiles and when they are written to the target data files.

With single-step conversions, since they do not use the ifile system, fields that contain all spaces on the source are written to the target with all spaces.

For additional information about single- and multi-step conversions, refer to Chapter 15 “Single-Step and Multi-Step Conversions” in the *ETI Data System Library DA: Procedures* manual.

4.2.6 In-Memory Lookup

This DSL now provides an additional method of looking up data. You can now perform an in-memory lookup (loading data into program memory) as well as a database lookup.

Choose *in-memory lookup* when both:

- the volume of data is suitable to be held in memory, *and*
- the search part(s) comparison(s) are an exact match

Choose database lookup either when:

- the volume of data that must be compared is more that you would want to load into memory, *or*
- the lookup comparison(s) is not an exact match

For additional information, refer to the section “Selecting the Lookup Method” in Chapter 12 “Looking Up Data in Auxiliary Sources” in the *ETI Data System Library DA: Procedures* manual.

4.2.7 Select Distinct Option for Selecting Records Conditionally

A new **Select distinct** option has been added for selecting records conditionally for the Read operations.

The **Select distinct** option uses the DISTINCT clause to remove all duplicate values for the mapped parts read. A usage example is shown below.

Sample Results for Select Records:

Alabama
Alabama
Arkansas
California
California

Sample Results for Select **Distinct** Records:

Alabama
Arkansas
California

Notice that with the **Select distinct** option, *no* duplicate part names are returned.

4.2.8 New Option for Validating or Reformatting a String that Contains a Date

A new feature in the C language-based DSLs is the ability to validate and/or reformat a character string that contains a date of the supported date formats. This feature can be useful for those fields not defined as a date, but the data contains a valid date.

The option is available for the **Test validity by data type** and **Reformat** operations of the Cleanse stage.

4.2.9 New Option for Connecting as a Different User

A new option for connecting to a DB2 database using a particular userid and password has been added. Two new properties have been created to accomplish this:

- `shell_db2_pwd` — specifies the DB2 password to use when accessing a database
- `shell_db2_uid` — specifies the DB2 user ID to use when accessing a database

Please refer to Chapter 1 of the *ETI Data System Library DA: Reference* manual for additional information.

4.2.10 New Default Value for `shell_flags_compiler`

A new default value has been provided for the property `shell_flags_compiler`. If this property is not set, it now defaults to:

```
-I/$DB2DIR/include -ldb2
```

`$DB2DIR` is an environment variable that designates the home directory for a DB2 installation and is a new feature in this release of the DSL for C/DB2 DA. Please refer to *ETI Data System Library DA: Documentation Addendum* for additional information on this property.

4.2.11 Obsolete Property `shell_flags_precompiler_cdb2`

The property `shell_flags_precompiler_cdb2` is now *obsolete* and should not be used. Use the property `shell_flags_precompiler` instead. Please refer to *ETI Data System Library DA: Documentation Addendum* for additional information on this property.

4.2.12 Support Added for DB2 6.x Retrieve Schema

Support has now been added for DB2 6.x specific nuances for the retrieve schema process.

5 RELEASE 4.2.0

Not applicable, since 4.2.1 is the first Generally Available 4.2 release of the DSL for C/DB2 DA.