

ETI Data System Library™ DA Release Notes

Revision 4.2.0A

Regarding ETI Data System Library DA for:

DSL for COBOL/File System DA 4.2

February, 2002

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

Evolutionary Technologies International, ETI, ETI•EXTRACT, the ETI logo, Data System Library, Starter Kit, AnswerLine, AnswerFlash, AnswerLink, Success First, Metadata Exchange Library (MDX), MetaStore, ETI•Accelerator, ETI•Accelerators, ETI•Meta Scheduler, ETI•Meta Schedulers, ETI•Impact Analyst, ETI•Bookshelf and Workset Browser 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.

Documentation regarding the DSL for C/Microsoft SQL Server was jointly developed by Evolutionary Technologies International, Inc. and DAMAN Consulting, Inc.

Table of Contents

1.0	PREREQUISITES	3
2.0	PURPOSE OF THIS RELEASE	3
3.0	DSL INSTALLATION	3
	IF YOU HAVE ETI•EXTRACT INSTALLED.....	3
	INSTALLING THE DSL FOR COBOL/FS DA.....	3
4.0	IMPORTANT NOTICES	4
	DSL FOR COBOL/FS DA	4
	COBOL INTERMEDIATE ACTIONS	4
	<i>Differences in Merge Processing in COBOL Intermediate Actions DA 4.2</i>	4
	<i>Using Pre-DA 4.2 Releases of COBOL Intermediate Actions DA for Merge Processing</i>	4
	DOCUMENTATION UPDATE — INCORRECT PROPERTY DRIVER USED ON A JOIN	5
	NEW PROPERTY SHELL_COMPILER	6
	NEW PROPERTY MERGE_MATCH	6
	MIXING DIFFERENT VALUES FOR MERGE PROPERTIES	6
	SETTING THE DRIVER UNIT.....	6
5.0	NEW FEATURES OF THIS RELEASE	7
	CONVERSION PROCESSING.....	7
	SINGLE-STEP CONVERSIONS	7
	<i>Requirements for Single-Step</i>	8
	<i>Message Program Information</i>	8
	<i>Overview of Differences between Pre-DA Single-Step and DA Single-Step</i>	8
	MULTI-STEP CONVERSIONS	8
	CONTROLLING THE TYPE OF CONVERSION GENERATED	9
	<i>Generating Single-Step Conversions</i>	9
	<i>Generating Separate Query and Populate Programs with Ifiles</i>	9
	<i>Generating Multi-Step Conversions (pre-DA 4.2 Style)</i>	10
	POTENTIAL DIFFERENCES IN CONVERSION RESULTS WITH SINGLE-STEP VERSUS MULTI-STEP CONVERSIONS	10
	<i>Sort Order Output Differences with Single-Step Versus Multi-Step</i>	10
	<i>Multiple Source Databases—Stopping Query Program On Error</i>	10
	<i>Spaces In Source Data Fields</i>	11
	WINDOWS BATCH SCRIPTS	11

1.0 PREREQUISITES

The 4.2 release of the DA Data System Libraries (DSLs) requires ETI•EXTRACT® Tool Suite Release 4.2.2 to ensure that the DSL will generate expected code.

2.0 PURPOSE OF THIS RELEASE

The 4.2 Data System Libraries will be released in phases. The following Data System Library Release 4.2 is now available, and the first in the series of 4.2 DSLs to be released:

- DSL for COBOL/File System DA

This DSL now supports single-step conversion processing. See the section “NEW FEATURES OF THIS RELEASE” on page 7 for information about single-step.

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.

Note: The relational DSLs for Release 4.2 are not yet available. They will be released in a later phase.

3.0 DSL INSTALLATION

If You Have ETI•EXTRACT Installed

To install ETI Data System Libraries, follow the installation instructions included in the installation manual for the platform on which you will install:

- 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”

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 for COBOL/FS DA

With the new patch process, some DSLs (including the DSL for COBOL/FS DA) will be shipped with patches. To install the DSL for COBOL/FS DA simply follow the normal procedure for installing a DSL and when you are prompted to load the patches, answer “yes”.

Shared Objects 4.2 is a prerequisite for using the DSL for COBOL/FS DA. This component will be loaded automatically during the installation of the DSL for COBOL/FS when you select the option to auto-load the prerequisites.

4.0 IMPORTANT NOTICES

Release 4.2 includes DSL objects named with the “42” suffix that contain fixes and support the COBOL/FS single-step feature.

DSL for COBOL/FS DA

The DSL for COBOL/FS DA provides the ability to query and populate COBOL/FS sources and targets. The DSL for COBOL/FS DA is designed to generate single-step conversion programs if possible (or multi-step conversion programs otherwise) to execute on any Unix, MVS, Windows NT, Windows 2000, or Windows XP computer running a supported COBOL compiler. The DSL for COBOL/FS DA generates the COBOL programs and Unix shell scripts, Windows batch files, or MVS JCL.

COBOL Intermediate Actions

Differences in Merge Processing in COBOL Intermediate Actions DA 4.2

Releases 4.2 of ETI•EXTRACT and COBOL Intermediate Actions DA provide the user much better control on how data is merged than in previous releases of the DSL. (For details please refer to the sections beginning with “Merge Processing” in Chapter 16 of the *ETI Data System Library DA: Procedures* manual.)

Given the enhanced merge control now provided, the merge processing now differs from the merge in prior releases of COBOL Intermediate Actions DA, which means that existing conversions may exhibit different behavior when the DSL for COBOL/FS 4.2 DA is installed:

Prior to 4.2, users controlled the order of processing joins through the definition of a driver unit. The driver was specified for the target unit that was populated from a merge. The driver unit determined both the sequence in which the merges were performed, and which join unit would be used to drive each individual merge operation.

The rule of thumb in COBOL Intermediate Actions DA 4.2 is that the “**From**” (child/many) unit is the **driver in each merge operation**. When multiple merge operations are required, the driver unit can change from merge to merge. Therefore, ETI recommends that you *not select a driver unit* with ETI•EXTRACT 4.2 or later and that you **use the join sequence numbers and the child unit in the relationship as the driver instead** (see “Driver Unit Specified for the Target Unit (Not Recommended)” in Chapter 16 of the *ETI Data System Library DA: Procedures* manual.). You have more control over the order of processing joins using these features.

Using Pre-DA 4.2 Releases of COBOL Intermediate Actions DA for Merge Processing

Customers that want to continue using the pre-DA 4.2 merge processing can still do so. You might want to do this if you have existing conversions that produce merges and you do not want to introduce the new merge processing yet.

Note: ETI recommends using the 4.2 merge processing unless you have existing conversions which will require the 4.1 processing.

- ETI will provide new definitions for 4.1 merge processing code in the following Template Definitions:

```
cobol_merge41_da
cobol_sh_m_cmp41_da
cobol_sh_m_exe41_da
cobol_sh_m_std41_da
jcl_mrg_cmp41_da
jcl_mrg_exe41_da
jcl_mrg_std41_da.
```

These definitions will be loaded automatically when you load the DA 4.2 DSL components. If you load DA 4.2 into a MetaStore that does not already contain a COBOL-based DA 4.1 DSL, the template definitions will be empty. If you later load a COBOL-based DA 4.1 DSL into the MetaStore, or if you load 4.2 into an existing MetaStore with a COBOL-based DA 4.1 DSL, the template definition will contain the 4.1 merge libraries.

- To use the 4.1 merge template definitions, you will need to specify these new templates as your merge intermediate action (see “Attaching Intermediate Action Templates” in Chapter 1 of the *ETI Data System Library DA: Procedures* manual), which may require you to set up additional installations.

Documentation Update — Incorrect Property `driver` Used on a Join

In setting up joins within the Conversion Editor, the parent database should always be set as the **From** Database (see discussion on “Value Joins” in Chapter 16 “Data Relationships and Intermediate Actions” of the *ETI Data System Library DA: Procedures* manual).

The default driver in 4.2 is the many side (the child, or “**To**” side of the join) of a one to many relationship specified on a join. The section titled “Driver Unit Specified for a Relationship (Rarely Necessary)” in Chapter 16 “Data Relationships and Intermediate Actions” of the *ETI Data System Library DA: Procedures* manual specifies that you can set the parent side as the driver for an individual join by setting the property `driver` to `true` on the “**From**” property of a join:

- The name of the property should be `relationship_driver` rather than `driver`.
- This functionality is not yet available in the COBOL/FS DA 4.2 release.

If you need to specify the parent to be the driver in the join, swap the parent/child relationship within the join, placing the child on the “**From**” side of the join and the parent on the “**To**” side.

New Property `shell_compiler`

A new property `shell_compiler` has been added with this release to specify the name of the COBOL compiler:

shell_compiler

Purpose: Identifies the path name of the COBOL compiler (for compiling COBOL programs).

Available in: DSL for COBOL/FS DA

Output: Shell scripts executing a compile step

Values: `cob` (for MicroFocus)

`cob2` (for IBM)

<CVAR `shell_compiler_ibm`>

How to Set: Set as a property on either the DAS, host, or installation object

Note: The `shell_compiler` property is set on the default `cobolfs_da` DAS object and contains either the shell command to invoke the compiler, or the CVAR that resolves to the desired command for a particular platform.

Related Properties: `shell_flags_compiler`

New Property `merge_match`

A new property `merge_match` has been added with this release to provide greater control in merges. Refer to Chapter 16 “Data Relationships and Intermediate Actions” in the *ETI Data System Library DA: Procedures* manual for details on this new property.

Mixing Different Values for Merge Properties

Setting the property `merge_match` to different values for the **From** and **To** sides of a join produces unpredictable results.

Setting the Driver Unit

Setting the driver unit is typically no longer necessary. For details please refer to the sections “Driver Unit Specified for the Target Unit (Not Recommended)” and “Driver Unit Specified for a Relationship (Rarely Necessary)” in Chapter 16 “Data Relationships and Intermediate Actions” of the *ETI Data System Library DA: Procedures* manual.

5.0 NEW FEATURES OF THIS RELEASE

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.

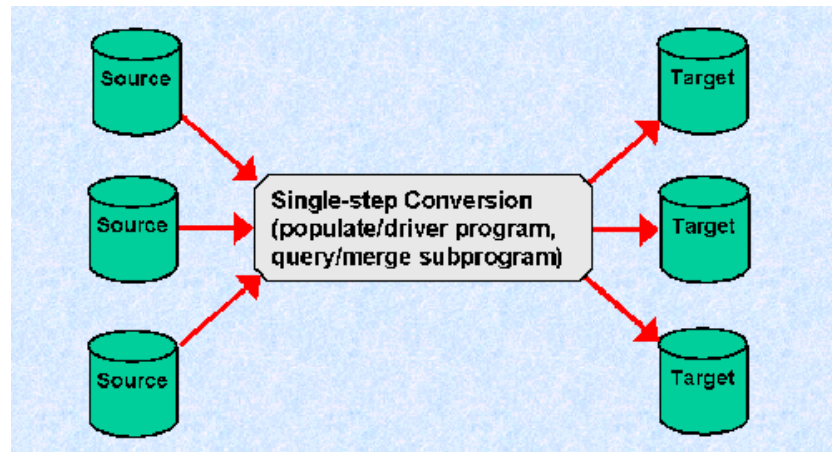
A single-step conversion is usually more efficient than a multi-step conversion. However there are constraints that must be met for ETI•EXTRACT to be able to generate a single-step conversion. There may also be cases when you will get the most efficient solution by forcing ETI•EXTRACT to generate a multi-step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

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.

Requirements for Single-Step

A single-step conversion can be generated when:

- All source and target database objects were created using the same installation object.
An installation object represents a Data Access System (DAS), (cobolfs_da, for example) running on a particular host computer. Therefore, all source and target databases must use the same DAS and be running on the same host.
- Files to be joined must already be sorted on the join key (hierarchical DSLs only).
For relational DSLs, the query subroutine may also contain SQL ORDER BY clauses that cause data to be retrieved in a sorted order, thus eliminating the need for a separate sort instruction.
- 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.

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 SSSCONV would be called:

- SSSCONVpopulate1.log
- SSSCONVpopulate1.wrn
- SSSCONVpopulate1.err

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”.

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.

Controlling the Type of Conversion Generated

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

Generating Single-Step Conversions

When using the default value of 4 for the `optimization_level` property, ETI•EXTRACT will automatically generate a single-step conversion when possible.

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.

Generating Separate Query and Populate Programs with Ifiles

Set the value of `optimization_level` to 4 and `ifile_io` to `true` to generate a multi-step with ifiles and separate query and populate programs with the merge included within the query instruction.

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).

Generating Multi-Step Conversions (pre-DA 4.2 Style)

Set the value of `optimization_level` to 3 to generate a multi-step conversion with separate intermediate actions.

Reasons to Choose a Multi-Step Conversion (pre-DA 4.2 Style)

You might want to do this when:

- You want ETI•EXTRACT to produce the same steps and generate a multi-step conversion in the same fashion as with DA 4.1 and earlier.
- You want to run different steps in parallel using a massively parallel system.

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

Sort Order Output Differences with Single-Step Versus Multi-Step

In some circumstances, a target database containing multiple record types will be populated in a different sequence in single-step than in multi-step, with the result that the records in the target database are in a different sequence. In single-step the records will be grouped together by key in the target database. In multi-step the records will be grouped together by record type in the target db. This situation is illustrated in the following table.

Order of records in the target database using single-step versus multi-step:

Single-Step (records grouped by key)	Multi-Step (records grouped by record type)
A1	A1
B1	A2
A2	B1
B2	B2

Multiple Source Databases—Stopping Query Program On Error

In the event of an abnormal termination (where the user has chosen the option to error and quit in a filter or by setting a property), it is *always* extremely important to inspect the results. This section presents an example of why this is the case, and why abnormal termination can produce different results in a single-step versus a multi-step conversion.

When using multiple source databases, there is a difference between single-step and multi-step when stopping the query program on an error. This is because when using optimization level 4 with a single-step conversion, operations will be combined to produce a larger run unit. For example, with single-step, two source databases could be combined and processed within one query instruction, whereas with multi-step, each source database would be processed separately.

As a result, in the multi-step case, an error that occurs in processing one source database would not affect the processing of the other source database, which could continue normally to completion. In the single-step case, on the other hand, where the processing of both source databases has been combined, an error

in processing one of the source databases will also cause an abnormal termination in the processing of the other source database.

The use of the larger run unit in the single-step conversion has significantly increased the scope of the impact of an abnormal termination on the conversion results, when compared to the multi-step conversion.

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.

Windows Batch Scripts

The DSL for COBOL/FS DA now generates Windows batch scripts that you can use to compile and execute source code on any Windows NT, Windows 2000, or Windows XP computer running a supported COBOL compiler.