

ETI Data System Library

DA 4.3 DSL Patch Notice

Regarding Patches for ETI Data System Libraries (DSLs), Release 4.3

October 04, 2005

Copyright © 2003 by Evolutionary Technologies International, Inc. All rights reserved.

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

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

Table of Contents

1. INTRODUCTION	3
2. OVERVIEW	3
3. INSTALLING AND LOADING DSL PATCHES	4
INSTALLING DSL PATCHES INTO THE DSL-INSTALL DIRECTORY	4
<i>Installing from CD-ROM</i>	4
<i>Installing from download or e-mail</i>	4
LOADING PATCHES INTO YOUR METASTORE.....	4
4. PATCH IMPACT ON CUSTOMIZATIONS	4
LIBRARIES, MODULES, AND FUNCTIONS.....	4
GRAMMAR EXTENSION MODULES (GEMS)	4
5. PATCHES—DA-COMPLIANT DSLS	5
C/XML DA 4.3.2	5
<i>Patch 01</i>	5
6. PATCHES—ETI ACCELERATORS	6
7. PATCHES—DA-COMPLIANT 4.3 CORE COMPONENTS	6
CORE DA 4.3.....	6
<i>Patch 02</i>	6
<i>Patch 01</i>	7
APPENDIX A –DA 4.3 PATCH NAMING AND IDENTIFICATION	9
APPENDIX B – INSTALLING PATCHES OBTAINED VIA DOWNLOAD OR E-MAIL	10
<i>Solaris, AIX, HPUX 10.20 or HPUX 11.0</i>	10
<i>Windows</i>	10
APPENDIX C – THE STRUCTURE OF DA DSLS	12
<i>DA 4.3 Component Descriptions</i>	12
<i>DA Core Components Matrix</i>	14

1. Introduction

ETI™ has developed a process for patching an ETI Data System Library (DSL) to address problems that are identified and need to be fixed between modification releases of a DSL. The objective is to provide critical bug fixes with the minimum impact on existing MetaStores, associated conversions, and generated programs.

Note: ETI will clearly identify any patch that modifies program generation or requires a revalidation of existing conversions.

- Since patches are implemented for a specific version of a DSL, the DSL must be loaded before you apply any patches.
- The most current patch includes any fixes from previous patches. Therefore, it is not necessary to load all patches—only the latest patch for a specific DSL release.
- The next release of the DSL will include all fixes from prior patches.
- DSL products ship with the latest patches, if any exist.

This patch document specifically addresses patches applied to DA DSL products at the release level of 4.3. To view patch documentation for earlier releases of all DSLs, refer to the document titled *ETI Data System Library Patch Notice* dated May 19, 2003.

2. Overview

All DA 4.3 DSL template libraries contain a module named “patch”. The patch module will be the first module in each template library as originally shipped by ETI, and will be empty for each base product release (for example DA 4.3). When a patch is required, ETI ships a package of “patch” modules for each product or component that is affected. These patches, when loaded by the customer, are automatically applied to the correct Template Libraries. Simply by loading this package through the normal DSL load process within the ETI Administration tool, the patch will be installed and ready to use.

An updated patch module replaces any existing patch module. The updated patch module is cumulative and thereby includes all fixes shipped as a part of the prior patch. Any function found in the patch module will override any function by the same name in the library using the standard override method for all templates. This approach greatly minimizes the possibility of manual error in installing the patch.

By distributing patches within patch modules, ETI can now directly patch the core DSL libraries, and apply fixes to multiple DA DSLs. For example, should ETI find a bug in the COBOL/DB2 code, a single patch for the core_db2 library will fix both the COBOL/DB2 and C/DB2 products.

Note that when loading patches into an existing MetaStore™, some reconciliation may be needed to ensure that user customizations do not override fixes in the patch or vice-versa. Please refer to the section “Libraries, Modules, and Functions” on page 4 for details.

3. Installing and Loading DSL Patches

Installing DSL Patches into the DSL-Install directory

DSL Patches are installed in the ETI Solution DSL-Install directory. After the installation process, the DSL-Install directory contains patch directories and files containing the contents of the patches, which can then be loaded into your MetaStore through the ETI Solution Administration tool. See Appendix A for information on identifying the patches within the DSL-Install directory.

Installing from CD-ROM

Refer to Installing Software in Chapter 2 of the ETI Solution Administration Guide for instructions on installing DSL products and patches from CD-ROM.

Installing from download or e-mail

See Appendix B for information on installing patches obtained by downloading from the ETI web site, or via email.

Loading Patches into your MetaStore

See *Load DSL Updates* in Chapter 3 of the ETI Solution Administration Guide for instructions on loading DSL patch products into your MetaStore.

4. Patch Impact on Customizations

Libraries, Modules, and Functions

If you have made template customizations, you can use the ETI Solution Template Work Area to analyze the patch modifications provided by ETI with your own customizations to determine the impact of the changes. With the utilities and functions provided in the Template Work Area, you can:

- Determine if the patch overrides any of your customized functions
- Determine if any of your customized functions override the patch
- Compare functions for detailed change analysis

If there is conflict between a patch and a customization, then you need to determine whether your customization is still needed. If the customization is still needed and is not provided by the patch, then you will need to integrate the patch functionality into the code for your customization.

See *Template Function Overrides* in Chapter 7 of the ETI Solution User's Guide for information on using the utilities and functions within the ETI Solution Template Work Area to determine the impact of Template function changes distributed with a patch.

Grammar Extension Modules (GEMS)

If you have made customizations to the GEMs being patched, your modifications will be overwritten when you load a patch containing updates to the same GEMs. You may export the customized GEMs **prior** to loading the patch to ensure the capture of the customizations. Once you have loaded the patch, you must apply the GEM customizations manually.

5. Patches—DA-compliant DSLs

C/XML DA 4.3.2

Patch 01

The 4.3.2 Patch 1 release of DSL for C/XML DA requires ETI Solution Version 5.2.1 or greater to ensure that the DSL will generate expected code. This patch fixes errors identified in the 4.3.2 release of the DSL for C/XML DA. Installing this patch does not require program re-generation and a validation of existing conversions.

The C/XML DA 4.3 components and associated DSL DA objects patched are listed below. Fixes implemented for this release include:

XML XSD Namespace Handling

- XML namespaces are now enforced for all conversions. XML namespaces can be specified in an XSD schema and are now utilized in the following way:
 - Query** - The name of an element and its namespace is verified when reading element data from an XML file. It is important that any namespace declarations used in the XML document schema be applied to the source XML document to ensure a seamless conversion.
 - Populate** - The specified target namespace will be applied as a qualified prefix of XML elements using the standard syntax, 'myNs:elementName' (*where myNs is the namespace and elementName is the element name*). XML attributes are written without a namespace prefix.

XML Handling of Repeating Parts

- The parser for the retrieve schema process will automatically sense and set the proper values for a repeating part. A repeating part in an XML file is an element with one or more child elements that can repeat more than one time. Due to type derivation and inheritance issues in XML schema languages, the retrieve schema process will use the largest possible value for a repeating part when that value is defined multiple times.

XML XSD Data Types

- With this patch, all XSD-defined data types are supported except the following:
 - any**: This data type is used to define a rule that any element with any name and data type can be used. Since ETI products are designed to move data, it is necessary to know the element name and its data type to allow mapping to or from that element.
 - anyType**: This data type is used to define a named element which can be of any XSD type or have any type for its child elements. Since ETI products are designed to move data, it is necessary to know the element name and its data type to allow mapping to or from that element.
 - anyAttribute**: This data type is used to define any attribute name or data type for a named element. Since ETI products are designed to move data, it is necessary to know the element name, its attribute names, and the related data types to allow mapping to or from an attribute.
- Note:** DTD schemas do not define data types and are therefore unaffected by this change.

cxml_43_patch.CXML43P_4_3_102_1
Template Module: das_cxml_43

cxml_da_patch.CXMLDAP_4_3_201_1
Data Access System: cxml_da
Retrieve Schema:
classes.jar
xmlRetrieveSchema.tcl

Note: This patch includes updates to the cxml_da DAS. If you have made any changes to the cxml_da DAS you should identify those changes and reapply after installation.

Refer to the section “Core DA 4.3” on page 6 for information about the related patch to DA core components.

6. Patches—ETI Accelerators

There are currently no patches to 4.3 accelerator products.

7. Patches—DA-Compliant 4.3 Core Components

This section describes patches available to DA 4.3 core components. DA 4.3 core component patches may be applicable to multiple DA 4.3 products (Data System Libraries, ETI Accelerators, and Intermediate Actions). Refer to the DA 4.3 Core Components **Matrix** on page 15, to identify the DA 4.3 products that can be affected by patches to specific DA 4.3 core components.

Note: The Load DSL Updates function within the ETI Solution Administration tool identifies the DA product patch updates available to products already loaded within your MetaStore. Updating a DA 4.3 product with patch updates applies all available core DA 4.3 patches to that product.

Core DA 4.3

Patch 02

This patch contains fixes available to all hierarchical DA 4.3 DSLs and all C-Based DSLs. The components and associated patched DA DSL objects are shown below.

Affect All Hierarchical DA 4.3 DSLs

The following patched component includes fixes identified for this component in Patch 01, plus:

- A problem identified in hierarchical Single Step conversions where the generated program would loop while processing the data has been addressed.

hier_43_patch.HIER43P_4_3_102_1
Template Modules:
pflow_43
pflow_hier_pop_43
pflow_hier_qry_43

Affect All C-Based DA 4.3 DSLs

The following patched component includes fixes identified for this component in Patch 01, plus:

- An issue with a C/XML date part.

core_c_43_patch.CORE43P_4_3_102_1
TemplateModules:
filt_c_43
ifile_c_43
layout_c_43
layout_c_static_43
layout_c_msg_log_43
trans_c_43

Patch 01

This patch contains fixes available to all DA 4.3 DSLs, and all C-based DA 4.3 DSLs and Accelerators. The components and associated DSL DA objects patched are shown below.

Affect All DA 4.3 DSLs

The following patched component includes fixes for:

- Added check to determine if processing merge instruction

core_cmp_ex_43_patch.CORECMP43P_4_3_101_2
TemplateModule: sh_common_43

The following patched component includes fixes for:

- Automatically set property max_units_per_populate when the write_method = FASTLOAD. This is required when reading multiple units.
- Fix unbalanced quote in the Grammar phrase SUBQUERY_PART.

core_filters_43_patch.COREFLTP43P_4_3_101_1
GramExtModule: core_filter_fcns_43
Grammar: common_value_43

The following patched component includes fixes for:

- Support for selecting or rejecting at the unit or record level.

core_foundation_43_patch.COREFOUN43P_4_3_101_1
TemplateModule: filt_43

The following patched component includes fixes for:

- Add DAS specific check-gen for SQL/Teradata as it follows different guidelines for generating shell scripts than other DSLs.

shell_43_patch.SHELL43P_4_3_101_2
TemplateModule: shell_43

Affect All Hierarchical DA 4.3 DSLs

The following patched component includes fixes for:

- Support for selecting or rejecting at the unit or record level.

hier_43_patch.HIER43P_4_3_101_1
Template Modules:
pflow_43
pflow_hier_pop_43
pflow_hier_qry_43

Affect All C-Based DA 4.3 DSLs

The following patched component includes fixes for:

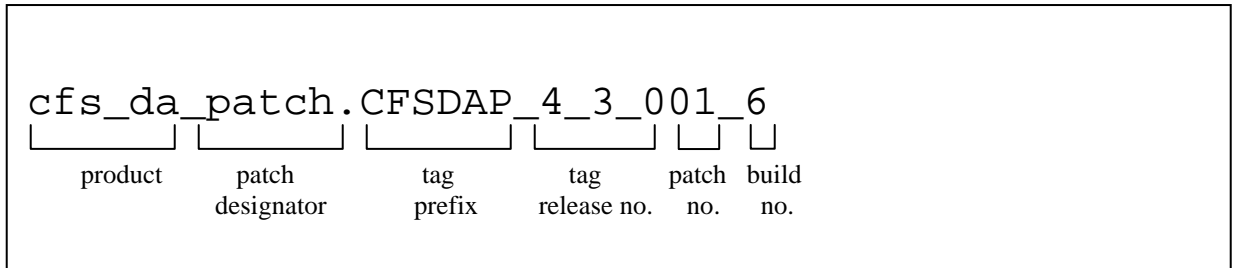
- Support for selecting or rejecting at the unit or record level.
- A memory error with the BTREE date parts.
- Issue with null part when active but not mapped

core_c_43_patch.CORE43P_4_3_101_2
TemplateModules:
filt_c_43
ifile_c_43
layout_c_43
layout_c_msg_log_43
trans_c_43

Appendix A –DA 4.3 Patch Naming and Identification

DA 4.3 patches will be named and distributed according to conventions that allow easy identification.

For example, the name for a C/FS DA 4.3 patch 1 is might resemble the following:



This name specifies the DSL name (`cfs_da`) and release (`4_3_0`) being patched. The name indicates that this represents a patch by the use of the patch designator (`patch`). It also specifies the number of the patch (01) and the actual build number of the patch (6).

Appendix B – Installing Patches obtained via Download or E-mail

Patches may be downloaded from the ETI Answerlink web page, or even provided via E-mail. In either case, the patch will be distributed as a compressed tar file (.gz). After obtaining the patch, it must be transferred to the DSL-Install directory on the MetaStore Server, uncompressed, and untarred before it can be loaded into a MetaStore.

Solaris, AIX, HPUX 10.20 or HPUX 11.0

1. Copy the compressed tar file to the DSL-Install directory.
2. Log in as the ETI Administrator.
3. Change directories (cd) to your ETI installation directory.
4. Set the ETI environment variables.
 - For Korn shell users (the leading period and space are required):
 - `. ex_env.ksh`
 - For C shell users:
 - `source ex_env.csh`
5. Change directories (cd) to `$EXTRACT_ROOTDIR/DSL-Install`.
6. Unzip the patch file.
 - `gunzip -c ./<patch>.tar.gz | tar xvf -`
7. Delete the tar file.

Windows

You can use a WinZip program to uncompress and untar the DSL patch, or you can uncompress and untar the DSL patch using the gunzip and gtar utilities which ETI provides with the ETI installation. To use WinZip, you must already have it installed on your Windows system.

Instructions for WinZip

Note: Some WinZip programs are unable to handle long file names correctly. If your WinZip program cannot uncompress and untar the DSL patch, follow the steps for using gunzip and gtar on the patch files.

1. Log in as the ETI Administrator.
 - Note:** The ETI Administrator must have write permission on the DSL-Install directory.
2. Copy the compressed tar file to the DSL-Install directory.
3. Using Windows Explorer, double-click on the patch file you copied to the DSL-Install directory.
 - Note:** Depending on your Windows settings, the .tgz extension may not be visible when you view the file from Windows Explorer.
4. A dialog appears, prompting you to decompress the file to a temporary folder and open it. Select Yes.

5. Click the Extract button.
6. A dialog appears, prompting you for a path to extract the patch files. Select the DSL-Install directory. Keep the default settings for the other options and click the Extract button.
7. The patch files are extracted to the DSL-Install directory.

Instructions for gunzip and gtar

1. Log in as the ETI Administrator.
Note: The ETI Administrator must have write permission on the DSL-Install directory.
2. Transfer the patch attachment to the DSL-Install directory.
Note: Depending on your Windows settings, the .tgz extension may not be visible when viewing the file from Windows Explorer.
3. Use the ETI Command Window by selecting **Start->Programs->ETI->Integration Design Studio 5.x->ETI Command Window**.
4. Change to the drive on which ETI software has been installed.
5. Change directories (cd) to %EXTRACT_ROOTDIR%\DSL-Install.
6. Unzip the patch file.
➤ `gunzip -f .\<<patch>.tgz`
7. Untar the patch file.
➤ `gtar xvf .\<<patch>.tar`
8. Delete the tar file.

Appendix C – The Structure of DA DSLs

DA DSLs are built from multiple components which are assembled as building blocks to create the specific DSL. The construction of these components allow sharing at various levels; i.e., by all DSLs (these are core components), by language, by Data Access System, by release. The ETI naming standards make it easy to identify how that component is shared, as the name is comprised of elements that provide an indication of the level of sharing – going from the most general to the most specific. Names are constructed as follows:

<function>_<optional subfunction, language or database>_<release specific or release independent designator>

Where:

function indicates the general function being provided

subfunction identifies more specific function

language identifies functions required for targeted language

database identifies functions required for targeted database system

release specific indicates the component applies to a specific release number

release independent indicates the component is not release specific

Below are examples and description of some of the component building blocks:

- core_filters_43 - core filter functions for (shared by) all 4.3 DA DSLs.
- core_filters_da – core filter functions for (shared by) all DA DSLs.
- core_c_43 – core functions for all c-based 4.3 DA DSLs.
- core_db2_da - core functions for all db2-based DA DSLs
- hier_43 – functions for all hierarchical-based 4.3 DA DSLs.

DA 4.3 Component Descriptions

See the tables below for a brief description of the current shipping 4.3 DA DSL components. Using the two tables (identifying release-dependent and release-specific components) it is easy to determine the impact of a patch release for any of the components.

Note that a patch shipped for a *release-independent* component (for example, a patch to **core_ia_da** required to fix a problem with Shell Sort Template Definitions), indicates that the patch could potentially affect *all DA* product releases.

However, a patch shipped for a *release-specific* component (for example, a patch to **core_cmp_ex_43** required to fix a problem with Shell Template Libraries), indicates that the patch would potentially affect only products with a *specific product release* (4.3).

DA Release-Independent Components	
cia_da	C-language Intermediate Action (Merge) Template Libraries
core_c_da	Template Definitions for exfcns.c and exfcns.h
core_cbl_da	COBOL Intermediate Actions Template Definitions and MSGPROG Template Definitions
core_db2_da	DA DB2 Bulk Load Templates
core_filters_da	Virtual Part Grammar
core_ia_da	Batch Sort Template Definitions
	Shell Sort Template Definitions
core_oracle_da	DA Oracle Bulk Load Templates
jcl_da	Template Definitions for generating JCL scripts
shared_objects	Languages, Operating Systems, Sample Hosts, Sample Users, Plan Template Libraries, Plan Template Definitions, Report Template Libraries
shell_da	Template Definitions for generating shell scripts and batch files
tcl_fcns	Non-DA Retrieve Schema Grammar Extension Modules (GEMs) Non-DA Create GEMs Non-DA Filter GEMs Program Generator Extension Modules (PGEMs)

DA 4.3 Release-Specific Components	
core_c_43	C-specific Template Libraries for generating C programs
core_cbl_43	COBOL-specific Template Libraries for generating COBOL programs
core_cmp_ex_43	Template Libraries for generating compile and execute code in the shell scripts and batch files
core_db2_43	DB2-specific Template Libraries for generating DB2 programs
core_filters_43	Contains filters and common GEMs
core_foundation_43	Generic Template Libraries that support generating code in all languages
core_ia_43	Template Libraries that generate shell scripts and batch files for IA
core_mqs_43	Common Template Library and GEM for MQSeries
core_oracle_43	Template Libraries that generate Oracle code
core_sql_43	Template Libraries that generate SQL code
core_teradata_43	Template Libraries that generate Teradata code
hier_43	Template Libraries that generate non-language specific code for navigating hierarchies
hier_c_43	Template Libraries that generate C code for navigating hierarchies
hier_cbl_43	Template Libraries that generate COBOL code for navigating hierarchies
jcl_43	Template Libraries for generating JCL scripts
rel_c_43	Template Libraries that generate C code for accessing relational databases
rel_cbl_43	Template Libraries that generate COBOL code for accessing relational databases
shell_43	Template Library that generate common code for shell scripts and batch files

DA Core Components Matrix

Use the Matrices on the following pages to answer the following questions:

- Which 4.3 products may be affected by a patch to a 4.3 component?

Within the *DA 4.3 Core Components Matrix*, find each component that was patched in the list of components, and then read across the row to determine which products may be affected by the patch (they are marked with the symbol ●)

- Which components are included within a particular product?

Read down the column for a particular product to determine its components (they are marked with the symbol ●).

- Which 4.2 and products may be affected by a patch to a 4.3 component?

Within the *DA 4.2 Core Components Matrix*, find each component that was patched in the list of components, and then read across the row to determine which products may be affected by the patch (they are marked with the symbol ●)

Note: The *DA 4.2 Core Components Matrix* only identifies the release-independent components as only release-independent components can impact prior releases of a project. Releases of DSL products prior to 4.2 are generally not impacted by later release as the structure of those DSLs had no release-independent components except `shared_objects` and `tcl_fcns`.

Legend:

● indicates product may be affected by a patch to the component

DA 43 Core Components Matrix

Components \ Products	Products												
	COBOL/FS 4.3	COBOL/DB2 4.3	C/DB2 4.3	C/Oracle 4.3	C/File System 4.3	C/XML 4.3	SQL/Teradata 4.3	Teradata Utilities 4.3	C/SQL Server 4.3	COBOL/MQSeries	C/MQSeries 4.3	COBOL IA ¹ 4.3	C IA 4.3
cia_da			●	●	●	●			●				●
core_c_da			●	●	●	●			●		●		●
core_cbl_da	●	●								●		●	
core_db2_da		●	●										
core_filters_da	●	●	●	●	●	●	●	●	●				
core_ia_da	●	●	●	●	●	●			●	●	●	●	●
core_oracle_da				●									
jcl_da	●							●					
shared_objects	●	●	●	●	●	●	●	●	●			●	●
shell_da	●	●	●	●	●	●	●	●	●				
tcl_fens	●	●	●	●	●	●	●	●	●			●	●
core_c_43			●	●	●	●			●		●		●
core_cbl_43	●	●								●		●	
core_cmp_ex_43	●	●	●	●	●	●	●	●		●	●	●	●
core_db2_43		●	●										
core_filters_43	●	●	●	●	●	●	●	●	●				
core_foundation_43	●	●	●	●	●	●	●	●	●	●	●	●	●
core_ia_43	●	●	●	●	●	●			●	●	●	●	●
core_mqs_43										●	●		
core_oracle_43				●									
core_sql_43		●	●	●			●	●	●				
core_teradata_43							●	●					
hier_43	●				●	●							
hier_c_43					●	●							
hier_cbl_43	●												
jcl_43	●							●					
rel_c_43			●	●					●				
rel_cbl_43		●											
shell_43	●	●	●	●	●	●	●	●	●				

This table shows the relationships of core components to 4.3 products. A patch to a component will affect the the associated 4.3 product(s).

¹ IA is the abbreviation for Intermediate Actions. With the DA 4.3 release, COBOL Intermediate Actions are included with COBOL-based 4.3 DA DSLs and C Intermediate Actions are included with C-based 4.3 DA DSLs; they are not available as separate product

Legend:

● indicates product may be affected by a patch to the component

DA 4.2 Core Components Matrix

Components \ Products	Products							CIA 4.2
	COBOL/FS 4.2	COBOL/DB2 4.2	C/DB2 4.2	C/Oracle 4.2	SQL/Teradata 4.2	COBOL/MQSeries 4.2	C/MQSeries 4.2	
cia_da			●	●				●
core_c_da			●	●			●	●
core_cbl_da	●	●				●		
core_db2_da		●	●					
core_filters_da	●	●	●	●	●			
core_ia_da	●	●	●	●		●	●	●
core_oracle_da				●				
jcl_da	●	●				●		
shared_objects	●	●	●	●	●			●
tcl_fcns	●	●	●	●	●			●

This table shows the relationships of core components to 4.2 and products. A patch to any of the release-independent components will affect the associated 4.2 and prior product(s).

This situation will occur if you load a newer version of a DSL product or patch into a MetaStore that contains an older version of a DSL product which shares some release-independent component(s). For example, loading the DSL for C/FS DA 4.3 into a MetaStore containing C/Oracle 4.2 causes updates to the following components:

- cia_da
- core_c_da
- core_filters_da
- core_ia_da
- shared_objects
- tcl_fcns

In general the release-independent components are backward compatible. Any changes in behavior would be noted within the release notes for the product, or this patch notice.