

# ETI Data System Library for C/XML DA

## Release Notes

### Revision 4.3.3A

**December, 2005**

Copyright © 2005 by Evolutionary Technologies International, Inc. All rights reserved.

Evolutionary Technologies International, ETI, ETI Solution, ETI•EXTRACT, the ETI logo, Dialogue Coach, AnswerLink, Success First, 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.

The DSL for C/XML DA includes software developed by the Apache Software Foundation (<http://www.apache.org>).

# Table of Contents

<b>1</b>	<b>PREREQUISITES</b>	<b>3</b>
<b>2</b>	<b>DSL INSTALLATION AND LOADING</b>	<b>3</b>
	IF YOU HAVE ETISOLUTION INSTALLED	3
	LOADING THE DSL	3
<b>3</b>	<b>LIBXML AVAILABILITY AND CONFIGURATION</b>	<b>4</b>
3.1	LIBXML DISTRIBUTIONS	4
3.1.1	<i>Libxml Distribution for UNIX</i>	4
3.1.2	<i>Libxml Distribution for Windows</i>	4
3.1.3	<i>Libxml Source Distributions</i>	4
3.2	LIBXML COMPILE/LINK OPTIONS	4
3.3	DEPLOYING C/XML PROGRAMS WITH LIBXML	4
<b>4</b>	<b>IMPORTANT NOTICES</b>	<b>5</b>
4.1	IMPACT TO EXISTING 4.3 DA DSL INSTALLATIONS	5
4.2	IMPACT TO EXISTING 4.2 OR EARLIER DA DSL INSTALLATIONS	5
4.2.1	<i>Cleanse Filters and VarChars</i>	5
4.2.2	<i>Large Integer and Floating Point Number Support and Temporary Variables</i>	5
4.2.3	<i>Changes to the Shell/Bat Templates between 4.2 and 4.3</i>	6
<b>5</b>	<b>RELEASE 4.3.3</b>	<b>7</b>
5.1	PURPOSE	7
5.2	FEATURES IN RELEASE 4.3.3	7
5.2.1	<i>Retrieve Schema Now Supports all XSD Data Types</i>	7
5.2.2	<i>Element Names and Data Types Can be Same Name</i>	7
5.2.3	<i>Extended Namespace Support</i>	7
5.2.4	<i>Complex Types with Attributes</i>	7
5.3	LIMITATIONS OF THIS RELEASE	7
<b>6</b>	<b>RELEASE 4.3.2</b>	<b>8</b>
6.1	PURPOSE	8
6.2	FEATURES IN RELEASE 4.3.2	8
6.2.1	<i>Support for XSD-based XML files</i>	8
6.3	LIMITATIONS OF THIS RELEASE	8
6.3.1	<i>XML and XSD Data Files Not Validated</i>	8
6.3.2	<i>Retrieving DTDs with External Entity References</i>	8
6.3.3	<i>Deprecated Grammar Options in Business Rules</i>	8
6.3.4	<i>N-Way Merge Not Yet Supported</i>	8
6.3.5	<i>Lookups</i>	8
<b>7</b>	<b>RELEASE 4.3.1</b>	<b>9</b>
7.1	PURPOSE	9
7.2	FEATURES IN RELEASE 4.3.1	9
7.2.1	<i>Support for DTD-based XML files</i>	9
7.2.2	<i>Character Encodings</i>	9
7.2.3	<i>Processing Hierarchies</i>	9
7.2.4	<i>Single-Step Support</i>	9
7.2.5	<i>Enhanced Dialogs for Entering Arithmetic Expressions in Business Rules</i>	9
7.2.6	<i>Access to Pre and Post Transformed Values</i>	9
7.2.7	<i>C Intermediate Actions</i>	9

7.3	LIMITATIONS OF THIS RELEASE.....	10
7.3.1	XML Schema Not Supported.....	10
7.3.2	XML Data Files Not Validated.....	10
7.3.3	Retrieving DTDs with External Entity References.....	10
7.3.4	Deprecated Grammar Options in Business Rules.....	10
7.3.5	N-Way Merge Not Yet Supported.....	10
7.3.6	Lookups.....	10
8	RELEASE 4.3.0 .....	11

## 1 PREREQUISITES

---

The DSL for C/XML DA requires the following products to be at the indicated release number *or later*:

- **ETI Solution® Version 5.2.** Required to ensure that the DSL will generate the expected code.  
**Note** You must use the Integration Client for the C/XML DA retrieve schema process.
- **Shared Objects 4.3.2.** This component will be loaded automatically during the installation of the DSL when you select the option to auto-load the prerequisites.
- **TCL Functions 4.3.3.** 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 4.3.** Required for supporting enhanced merge processing. This component will be loaded automatically during the installation of the DSL when you select the option to auto-load the prerequisites.

Generated C/XML programs require an XML parser so that the Query program can validate, parse, and read XML data. The DSL for C/XML DA currently requires the publicly available libxml parser from [www.xmlsoft.org](http://www.xmlsoft.org). Libxml conforms to the XML standard specified at [www.w3.org/TR/REC-xml](http://www.w3.org/TR/REC-xml), and is available on a wide variety of platforms. Libxml libraries are available for the Sun Solaris, HP HP/UX, IBM AIX, and Windows platforms, and can be downloaded from the **Software Downloads** area of the ETI AnswerLink™ website at [www.eti.com/answerlink](http://www.eti.com/answerlink) (located under the Libraries heading). Source distribution of libxml is also available from the download area.

## 2 DSL INSTALLATION AND LOADING

---

### If You Have ETISolution Installed

To install ETI Data System Libraries, follow the directions listed in *ETI Solution Administration Guide*, Chapter 2, “Getting Started”.

**Warning:** If you do not follow the DSL installation procedures described in the manual 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.

### Loading the DSL

To load the DSL into a MetaStore, follow the procedure listed in *ETI Solution Administration Guide*, Chapter 3, “Populating MetaStores” . Any available patches will be loaded automatically with the DSL.

## 3 LIBXML AVAILABILITY AND CONFIGURATION

---

### 3.1 Libxml Distributions

Libxml distributions contain include files for compiling C/XML generated source, and pre-built libxml libraries (for the most common platforms) for linking the generated programs. Source distributions are also available.

#### 3.1.1 Libxml Distribution for UNIX

Libxml distributions are available for the Sun Solaris, HP HPUX, and IBM AIX platforms. After downloading for the appropriate platform from the **Software Downloads** area of the ETI AnswerLink website, you must install on the machine where you will be compiling and linking the C/XML-generated programs. It is recommended that you install to /usr/local/libxml.

#### 3.1.2 Libxml Distribution for Windows

Libxml distributions are also available for the Windows platform from the **Software Downloads** area. It is recommended that you install to c:\libxml.

#### 3.1.3 Libxml Source Distributions

Complete source distribution of the libxml libraries is also available from the **Software Downloads** area, allowing you to build libxml for any platform where you intend to run generated C/XML DA programs.

### 3.2 Libxml Compile/Link Options

The cxml\_da DAS object for the DSL for C/XML DA provides default compiler and link option properties for Solaris, HPUX, AIX, and Windows platforms, based on the recommended installation locations as detailed in Section 3.1. If you install libxml in a different location, you should override the compile and link option properties on your Installation object.

### 3.3 Deploying C/XML programs with Libxml

The default compile and link option properties have been configured to link with the libxml library statically (as opposed to linking with the libxml shared library/dll). While making the executable a little larger, this allows you to deploy generated C/XML DA programs to other machines more easily. That is, C/XML DA program executables generated for a Sun Solaris system can be deployed to other Sun Solaris systems without the libxml libraries.

If you change the compile/link properties to link dynamically with the libxml shared library or dll, then you will need to ensure that the libxml shared library/dll is installed on each machine where you will be executing the generated code.

## 4 IMPORTANT NOTICES

---

### 4.1 Impact to Existing 4.3 DA DSL Installations

The DSL for C/XML DA 4.3.2 ships with patches to some of the core DA components. Loading C/XML DA 4.3.2 into a MetaStore™ automatically loads and applies these patches. If your MetaStore contains other 4.3-based DSLs, the patches may affect those DSLs.

Patches contain fixes to address problems or issues in the DSL products. After loading C/XML DA 4.3.2 into a MetaStore that contains other 4.3 DSLs, it may be possible that functions in a custom library would override fixes supplied in the patch modules. You should review any custom functions and reconcile with the new patches to ensure that custom functions do not override the fixes supplied by the patches.

### 4.2 Impact to Existing 4.2 or Earlier DA DSL Installations

The DSL for C/XML DA includes enhancements and fixes that are applicable for all C-based DSLs. These changes impact the generated `exfcns.h` and `exfcns.c` files, which are not specific to a version of the DSL. **This can result in a change in behavior to existing 4.2 C-based DSLs, if the DSL for C/XML DA is loaded into a MetaStore which already contains 4.2 C-based DSLs.**

You should review the changes listed below before loading the DSL for C/XML DA into an existing MetaStore.

#### 4.2.1 Cleanse Filters and VarChars

The DSL for C/XML DA includes fixes for an error that could occur in prior releases of the C-based DSLs when using cleanse filters and VARCHAR data. Application of a cleanse filter (compress multiple spaces, reformat, etc.) that can remove white space within a varchar may provide a result that contains no (character) data; that is, its length is zero. Previous releases of the C-based DSLs would incorrectly interpret the result as an empty string (“”), while the applied fixes now correctly interpret the result as null.

#### 4.2.2 Large Integer and Floating Point Number Support and Temporary Variables

The DSL for C/XML DA includes enhancements for Large Integer and Floating Point Number support and improvements in the handling of temporary variable allocation which can impact existing 4.2 C-based DSLs.

A set of patches for 4.2-based DSLs are provided with the DSL for C/XML DA 4.3 release to provide backward compatibility with the changes. **When loading C/XML DA 4.3 into an existing MetaStore containing 4.2 C-based DSLs, the following patches must be loaded to ensure correct behavior of the existing 4.2 DSLs:**

- `rel_c_42_patch.RELC42P_4_2_102_1`
- `core_c_42_patch.COREC42P_4_2_102_1`

As these patches are for the 4.2 release, they will not be automatically loaded with C/XML DA 4.3. **Refer to the *ETI Solution Administration Guide*, Chapter 3, “Load DSL Updates”** for instructions on loading these patches into your MetaStore.

Note that when loading into an existing development environment in which you are using a previously-generated `exfcns` object file, you will need to re-create `exfcns.o`. If you normally generate programs with

the `ex_generate_static_h` and `ex_generate_static_c` properties set to false, then you must temporarily set these properties to true in order to re-generate the `exfcns` files after loading the DSL for C/XML DA 4.3. Once the `exfcns.h` and `exfcns.c` files have been re-generated, you may reset these properties to false to avoid regenerating the `exfcns` files each time you generate programs.

### 4.2.3 Changes to the Shell/Bat Templates between 4.2 and 4.3

DA 4.3 DSLs contain changes to the structure of the Shell/Bat templates. These changes provide a common set of Template Definitions that can be extended by attaching DSL-specific Template Libraries.

DA 4.2 DSLs provided DSL-specific Template Definitions for the shell/bat scripts, each of which contained one or two DSL-specific Template Libraries and a large number of DSL-independent (core) libraries. For example, C/DB2 DA 4.2 contained the following Template Definitions:

```
cdb2_sh_qry_cmp_da, cdb2_sh_qry_exe_da, cdb2_sh_qry_std_da
cdb2_sh_pop_cmp_da, cdb2_sh_pop_exe_da, cdb2_sh_pop_std_da
cdb2_bat_qry_cmp_da, cdb2_bat_qry_exe_da, cdb2_bat_qry_std_da
cdb2_bat_pop_cmp_da, cdb2_bat_pop_exe_da, cdb2_bat_pop_std_da
```

The DA 4.3 DSLs provide a common set of shell/bat Template Definitions for all DSLs:

```
sh_qry_cmp_da, sh_qry_exe_da, sh_qry_std_da
sh_pop_cmp_da, sh_pop_exe_da, sh_pop_std_da
bat_qry_cmp_da, bat_qry_exe_da, bat_qry_std_da
bat_pop_cmp_da, bat_pop_exe_da, bat_pop_std_da
```

By default, the 4.3 Template Definitions contain common DSL-independent (core) libraries. However, at DSL load time, DSL-specific Template Libraries are attached to the Template Definitions, providing the functionality specific to that DSL.

For example, when loading a MetaStore with the C/XML DSL, the `sh_cxml_43` Template Library will be added to the `sh_pop_cmp_da` Template Definition to provide functionality specific to C/XML DA. Subsequent loading of the C/DB2 DA DSL into the same MetaStore will add `sh_cdb2_43` to the `sh_pop_cmp_da` Template Definition and provide the C/DB2 DA functionality.

## 5 RELEASE 4.3.3

---

### 5.1 Purpose

The primary purpose of this release is to add to minor bug fixes and remove some limitations in earlier releases in the retrieve schema support for XML Schema Definition (XSD) files.

### 5.2 Features in Release 4.3.3

#### 5.2.1 Retrieve Schema Now Supports all XSD Data Types

In this release, the Retrieve Schema process now supports all XSD 1.0 data types.

#### 5.2.2 Element Names and Data Types can have the same Name

The Retrieve Schema process now understands the concept of the XSD Symbol spaces. This concept partitions element types based on a set of rules, and allows for schemas which have element names and corresponding types of the same name.

#### 5.2.3 Extended Namespace Support

Retrieve Schema has had a number of changes to the namespace support. Changes were made due to certain cases not being validated by all XML/XSD parsers which are not part of the W3C specification. Changes ensure that certain namespace declarations are properly recognized even when a corresponding namespace prefix is not properly applied to an XSD document.

Example: `elementFormQualified` is true, but there is no namespace prefix on any element.

#### 5.2.4 Complex Types with Attributes

Some XSDs define `complexType`s that contain attributes that are defined through a deeply nested set of derivations and extensions that ultimately resolve to `simpleType` definitions.

The W3C specification appears to state that this should not happen - attributes of derive or extended `complexType` should be defined with `ATTRIBUTE` elements and not through the restriction of an extended `simpleType` element.

The Retrieve Schema process now allows attributes to be defined by `simpleTypes` regardless of the extension or derivation of the parent `complexType`.

### 5.3 Limitations of This Release

The 4.3.3 Release of C/XML DA DSL contains the same limitations as contained in the 4.3.2 release.

## 6 RELEASE 4.3.2

---

### 6.1 Purpose

The primary purpose of this release is to add support for XML Schema Definition (XSD) files to a DA DSL that can read/transform/write hierarchical XML data using the C programming language.

### 6.2 Features in Release 4.3.2

#### 6.2.1 Support for XSD-based XML files

The DSL for C/XML DA 4.3.2 provides the ability to read and write XML files compliant to an XML Schema Definition (XSD).

See *ETI Data System Library DA: Procedures*, Chapter 2, “Representing Data as an ETI•EXTRACT Schema” for more information about retrieving XSD files. Also see Appendix C, “XML Retrieve Schema Reference” for reference information about using XSD files.

### 6.3 Limitations of This Release

The following limitations apply to this release of the DSL for C/XML DA.

#### 6.3.1 XML and XSD Data Files Not Validated

XML data files read by the generated programs are expected to be valid XML, compliant with its DTD or XSD. The generated query program does not attempt to validate the XML against its DTD or XSD.

#### 6.3.2 Retrieving DTDs with External Entity References

DTD files that contain references to external entities require special handling for the retrieve schema process. In order to collect all data from a given DTD input source which has external entities, the DTD must first be "flattened". The flattening process simply reads the DTD source (called the "master"), collects all external entity files referenced, and writes a compiled version to a name similar to the master (i.e., the master dtd file with an "\_new" postfix).

See *ETI Data System Library DA: Procedures*, Chapter 2, “Representing Data as an ETI•EXTRACT Schema” for more information.

#### 6.3.3 Deprecated Grammar Options in Business Rules

Release 4.3 of all DA DSLs includes updates to grammars to provide for enhanced functionality (see Sections 7.2.5 and 7.2.6 below). As a part of this update, some grammar options are no longer useful or needed. These grammar options are identified in the options pane of the Business Rule Editor as deprecated and will be removed in a future release.

Deprecated options should not be chosen when building new business rules.

#### 6.3.4 N-Way Merge Not Yet Supported

The C Intermediate Actions does not yet support the ability to perform n-way merges. An n-way merge can be disabled by setting the property **disable\_nway\_merge** to **true** on the conversion or installation objects.

#### 6.3.5 Lookups

Lookup operations are not yet supported.

## 7 RELEASE 4.3.1

---

### 7.1 Purpose

The primary purpose of this release is to provide a DA DSL that can read/transform/write hierarchical XML data using the C programming language.

### 7.2 Features in Release 4.3.1

#### 7.2.1 Support for DTD-based XML files

The DSL for C/XML DA 4.3 provides the ability to read and write XML files compliant to an XML Document Type Definition (DTD).

#### 7.2.2 Character Encodings

Programs generated by the C/XML DA DSL can read XML documents with the following character encodings: UTF-8, UTF-16, ISO-8859-1 (ISO-Latin-1), ASCII, and HTML.

All XML documents produced by the C/XML DA DSL are written using UTF-8 encoding.

#### 7.2.3 Processing Hierarchies

The process flow for C/XML DA conversions is governed by process hierarchies. ETI Solution identifies one or more processing hierarchies based on the relationships (joins) and mappings defined in the conversion. Processing hierarchies are the foundation that gives ETI Solution hierarchical DSLs the ability to process complex data structures in a minimum number of instructions. Refer to Chapter 5, “Defining Process Hierarchies” in the *ETI Data System Library DA: Procedures* manual for information about processing hierarchies.

#### 7.2.4 Single-Step Support

By default, the DSL for C/XML DA will attempt to generate single-step conversions. See Chapter 15, “Single-Step and Multi-Step Conversions” in the *DA Procedures* manual for a discussion on single-step processing.

#### 7.2.5 Enhanced Dialogs for Entering Arithmetic Expressions in Business Rules

Release 4.3 of all DA DSLs includes a new dialog for entering arithmetic expressions. The new dialog is more streamlined and intuitive than the previous one. The previous dialog is still available to allow replaying business rules defined with the old dialog.

#### 7.2.6 Access to Pre and Post Transformed Values

Customers have requested the ability to create a virtual part, define a business rule on the virtual part to calculate a value, and then use this virtual part as input into other business rules in the same processing stage. The problem in the past has been that references to other parts always used the value of the part *before* any business rule was applied. Beginning with DA 4.3, the business rule dialog provides the ability to reference the value of a part as it was before the current processing stage (pre-stage value) or after the current processing stage (post-stage value).

#### 7.2.7 C Intermediate Actions

Release 5.2 of ETI Solution and C Intermediate Actions DA 4.3 (C/IA DA 4.3) provide the user much better control on how data is merged than in previous releases of Intermediate Actions. (For details please refer to the sections beginning with “Merge Processing” in Chapter 16 of the *DA Procedures* manual.) C

Intermediate Actions is functionally equivalent to the COBOL Intermediate Actions except that C Intermediate Actions does not have the ability to perform n-way merges.

## 7.3 Limitations of This Release

The following limitations apply to this release of the DSL for C/XML DA.

### 7.3.1 XML Schema Not Supported

The XML Schema Definition (XSD) specification is not yet supported by the retrieve schema process, and XSD compliant XML documents cannot be read/written by the generated programs.

This release only supports the retrieval of Document Type Definitions (DTDs) and reading/writing of DTD compliant XML.

### 7.3.2 XML Data Files Not Validated

XML data files read by the generated programs are expected to be valid XML, compliant with its DTD. The generated query program does not attempt to validate the XML against its DTD.

### 7.3.3 Retrieving DTDs with External Entity References

DTD files that contain references to external entities require special handling for the retrieve schema process. In order to collect all data from a given DTD input source which has external entities, the DTD must first be "flattened". The flattening process simply reads the DTD source (called the "master"), collects all external entity files referenced, and writes a compiled version to a name similar to the master (i.e., the master dtd file with an "\_new" postfix).

See *ETI Data System Library DA: Procedures*, Chapter 2, "Representing Data as an ETI•EXTRACT Schema" for more information.

### 7.3.4 Deprecated Grammar Options in Business Rules

Release 4.3 of all DA DSLs includes updates to grammars to provide for enhanced functionality (see Sections 7.2.5 and 7.2.6 above). As a part of this update, some grammar options are no longer useful or needed. These grammar options are identified in the options pane of the Business Rule Editor as deprecated and will be removed in a future release.

Deprecated options should not be chosen when building new business rules.

### 7.3.5 N-Way Merge Not Yet Supported

The C Intermediate Actions does not yet support the ability to perform n-way merges. An n-way merge can be disabled by setting the property **disable\_nway\_merge** to **true** on the conversion or installation objects.

### 7.3.6 Lookups

Lookup operations are not yet supported.

## **8 RELEASE 4.3.0**

---

Not applicable. 4.3.1 is the first Generally Available 4.3 release of the DSL for C/XML DA.