

ETI Data System Library for C/Microsoft SQL Server 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.

Table of Contents

1	PREREQUISITES	3
2	DSL INSTALLATION	3
	IF YOU HAVE ETISOLUTION INSTALLED	3
	LOADING THE DSL	3
3	IMPORTANT NOTICES	4
3.1	DATE LENGTH AND RETRIEVE SCHEMA	4
3.2	READ FILTER MODIFICATIONS	4
3.3	DATE ARITHMETIC FUNCTIONALITY WAS CHANGED	4
3.4	METHOD FOR DYNAMIC MEMORY ALLOCATION WAS CHANGED	4
3.5	SINGLE STEP IS DEFAULT	5
3.6	IMPACT TO EXISTING 4.3 DA DSL INSTALLATIONS	5
3.6.1	<i>Considering User Customizations</i>	5
3.7	IMPACT TO EXISTING 4.2 OR EARLIER DA DSL INSTALLATIONS	5
3.7.1	<i>Cleanse Filters and VarChars</i>	5
3.7.2	<i>Large Integer and Floating Point Number Support and Temporary Variables</i>	5
3.7.3	<i>Changes to the Shell/Bat Templates between 4.2 (and prior) DSLs and 4.3</i>	6
4	RELEASE 4.3.3	7
4.1	PURPOSE	7
4.2	FEATURES IN RELEASE 4.3.3	7
4.2.1	<i>Single Step Processing Supported</i>	7
4.2.2	<i>Enhanced Dialogs for Entering Arithmetic Expressions in Business Rules</i>	7
4.2.3	<i>Access to Pre and Post Transformed Values</i>	7
4.2.4	<i>C Intermediate Actions</i>	7
4.2.5	<i>In-Memory Lookup</i>	7
4.2.6	<i>Select Distinct Option for Selecting Records Conditionally</i>	7
4.2.7	<i>New Option for Validating or Reformatting a String that Contains a Date</i>	8
4.2.8	<i>Support for SQL Server 2005 Integration Services</i>	8
4.2.9	<i>SQL Server Live Connection for Retrieve Schema</i>	9
	RELEASES 4.3.0 – 4.3.2	11

1 PREREQUISITES

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

- **ETI Solution® Version 5.2.2p3.** 5.2.2 is required to ensure that the DSL will generate the expected code, while 5.2.2p3 is required to support the SQL Server Live Connect feature (refer to section 4.2.9 for information on the Live Connect feature).
- **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.
- **Executive 4.3.1.** 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 4.3.3.** 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 ETISolution Installed

To install ETI Data System Libraries, you must 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 IMPORTANT NOTICES

3.1 Date Length and Retrieve Schema

To meet ETI Data System Library ifile standards and to correctly process the SQL Server DateTime data type, the 4.3 Microsoft C/SQL Server retrieve process now sets the part extract length to 26 for all DateTime data types. Previous releases of the Microsoft C/SQL Server DSL retrieve schema processes incorrectly set the extract length to 23.

Attempting to generate a conversion with a schema that has been retrieved with a previous release of the C/SQL Server DSL will now result in a generation error and generation will terminate with an error resembling the following:

```
<<Fatal Error: Datetime part <part name> requires an extract length of 26 >>  
Compilation terminated. Cleaning internal State ...Complete
```

To resolve this error you must re-retrieve the schema and regenerate the conversion.

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 Date Arithmetic Functionality Was Changed

The date arithmetic functionality in the C language-based DSLs was changed beginning with the 4.2 release. 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.

Note that this functionality only works with Date data types and not DateTime data types.

3.4 Method for Dynamic Memory Allocation Was Changed

The method of dynamic memory allocation in the C language-based DSLs has changed beginning with the 4.2 release¹. 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,

¹ As there was no 4.2 release of the DSL for Microsoft C/SQL Server, these features are now available within the DSL for Microsoft C/SQL Server 4.3.3 release.

flushed, and potentially increased per iteration of the process flow. This new method may provide a dramatic increase in performance.

3.5 Single Step is Default

Beginning with Release 4.3.3, the DSL for Microsoft C/SQL Server is designed to generate single-step conversion programs by default if at all possible.

3.6 Impact to Existing 4.3 DA DSL Installations

The DSL for Microsoft C/SQL Server DA 4.3 ships with updates to the core DA components. Loading Microsoft C/SQL Server DA 4.3 into a MetaStore™ automatically loads these components. If your MetaStore contains other 4.3-based DSLs, the components may affect those DSLs.

3.6.1 Considering User Customizations

New product releases may have patches from prior releases rolled into the base components, as well as new functions. After loading Microsoft C/SQL Server DA 4.3.3 into a MetaStore that contains other 4.3 DSLs, it may be possible that functions in a custom library would override changes made within the base core components. You should review any custom functions and reconcile with the new components to ensure that custom functions do not improperly override core function.

3.7 Impact to Existing 4.2 or Earlier DA DSL Installations

The DSL for Microsoft C/SQL Server DA includes enhancements and fixes that are applicable for all C-based DSLs. These changes impact the generated exfns.h and exfns.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 Microsoft C/SQL Server 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 Microsoft C/SQL Server DA into an existing MetaStore.

3.7.1 Cleanse Filters and VarChars

The DSL for Microsoft C/SQL Server DA includes fixes for an error that could occur in prior releases of the 4.2 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 4.2 DSLs would incorrectly interpret the result as an empty string (“”), while the applied fixes now correctly interpret the result as null.

3.7.2 Large Integer and Floating Point Number Support and Temporary Variables

The DSL for Microsoft C/SQL Server 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 Microsoft C/SQL Server DA 4.3 release to provide backward compatibility with the changes. **When loading Microsoft C/SQL Server 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 Microsoft C/SQL Server 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 `exfns` object file, you will need to re-create `exfns.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 `exfns` files after loading the DSL for Microsoft C/SQL Server DA 4.3. Once the `exfns.h` and `exfns.c` files have been re-generated, you may reset these properties to false, to avoid regenerating the `exfns` files each time you generate programs.

3.7.3 Changes to the Shell/Bat Templates between 4.2 (and prior) DSLs 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 and prior 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, Microsoft C/SQL Server DA 4.2 contained the following Template Definitions:

```
csqlsvr_bat_qry_cmp_da, csqlsvr_bat_qry_exe_da, csqlsvr_bat_qry_std_da  
csqlsvr_bat_pop_cmp_da, csqlsvr_bat_pop_exe_da, csqlsvr_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 Microsoft C/SQL Server DSL, the `sh_csqlsvr_43` Template Library will be added to the `bat_pop_cmp_da` Template Definition to provide functionality specific to Microsoft C/SQL Server DA. Subsequent loading of the C/FS DA DSL into the same MetaStore will add `sh_cfs_43` to the `bat_pop_cmp_da` Template Definition and provide the C/FS DA functionality.

4 RELEASE 4.3.3

4.1 Purpose

This release of the Microsoft C/SQL Server DA DSL is an enhancement release and provides significant new functionality over prior releases.

4.2 Features in Release 4.3.3

4.2.1 Single Step Processing Supported

By default, optimization level 4 with the current release of the DSL for Microsoft C/SQL Server DA will generate a single-step conversion if possible. For information about conditions that can prevent generation of single-step programs, refer to Chapter 15, “Single-Step and Multi-Step Conversions”, of the *ETI Data System Library DA: Procedures* manual.

4.2.2 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.

4.2.3 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).

4.2.4 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.

4.2.5 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.

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.

4.2.6 Select Distinct Option for Selecting Records Conditionally

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

For additional information, refer to the section “Applying Filters” in Chapter 8 “Reading Data” in the *ETI Data System Library DA: Procedures* manual.

4.2.7 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.

For additional information, refer to the section “Applying Cleanse Filters” in Chapter 9 “Cleansing Data” in the *ETI Data System Library DA: Procedures* manual.

4.2.8 Support for SQL Server 2005 Integration Services

The 4.3.3 release of the Microsoft C/SQL Server DA DSL contains a new write filter option that supports loading data directly into SQL Server 2005 Integration Services (SSIS). Using the *SSIS* write option, an SSIS populate program is generated which can be used with the *ETI High Performance Connector for Custom Source* SSIS component to quickly and efficiently load data into SSIS.

You can apply the filter at either the target database or target unit level. When you apply the filter, select **Write Operations** in the **Options** pane. Within **Select Write Method and Mode**, select **SSIS** for the **Write Method**. The SSIS write method only allows for inserting (or loading) data into SSIS. Once within SSIS the data can be further processed and/or transformed and loaded into SQL Server Analysis Services for Business Analysis needs, or into a SQL Server 2005 database.

When using the SSIS write option, ETI Solution generates one instruction per target unit. Each instruction can only reference data within the unit that it is processing. Therefore, references to a part in another unit (for example using a transformation or data selection filter or a CIP) will cause a compilation error, since the program cannot perform that cross-unit reference because the generated code for each unit is in separate instructions.

The SSIS write option is incompatible with unit-of-work processing and specifying commit intervals, since ETI Solution is processing data one unit at a time. Instead, the commit level is the table, and each table is completely processed or is not processed at all.

4.2.8.1 Using Generated SSIS Populate Programs with the *ETI High Performance Connector for Custom Source* SSIS component

When the SSIS write option is selected, ETI Solution will generate one or more Microsoft C/SQL Server populate programs (depending on the number of mapped target tables), along with one or more configuration files (one per each populate program) that describe the data to be made available to SSIS. When deployed within an SSIS environment, the populate program and configuration file can be read by the *ETI High Performance Connector for Custom Source* SSIS component to efficiently stream the data directly into SSIS. For more information refer to the *ETI High Performance Connector for Custom Source* documentation.

4.2.8.2 Deploying SSIS Populate programs for use with SSIS

Depending on the sources and targets specified within a conversion, the relationships between the sources or targets, and the complexity of the transformations within the conversion, ETI Solution can generate a single step program (a single executable program that reads, processes, and writes the data) or multiple programs that process the data (i.e., query, sort, merge, and populate programs may all be generated).

When SSIS is identified as a target, the populate program will be generated that runs on the Windows platform and interfaces with SSIS as specified in Section 4.2.8.1.

Generated SSIS populate programs can be deployed for use with the ETI High Performance Connector for Custom Source SSIS component by:

1. Compiling and linking the generated populate program to create a populate executable (<populate program name>.exe) and manifest (<populate program name>.exe.manifest).
2. Copying the executable, manifest, and a generated configuration file (<populate program name>.cfg) to a directory that is accessible from SSIS

Within SSIS, a High Performance Connector for Custom Source component is configured using the generated configuration file. The configuration file identifies the name of the populate executable that is used to load data into the SSIS, as well as the metadata (data mappings and data types) required for the Custom Source component (refer to the documentation for the ETI High Performance Connector For Custom Source for more information).

Note that when multiple programs are generated for a conversion, the generated query (and any sort or merge) programs must be run prior to the populate program(s), and the Intermediate File (ifile) produced by those programs must be supplied as input to the populate program.

In the case of multi-step conversions, the High Performance Connector for Custom Source can be configured to wait for the ifile data to become available before executing the populate program and loading the data into SSIS. Single Step conversions process all the data without the need for ifiles, and are therefore executed immediately.

4.2.9 SQL Server Live Connection for Retrieve Schema

The ETI Solution retrieve schema process is a two-step process² which can be characterized by:

- Reading database table information: An application communicates with a database to retrieve table metadata and produce a schema description file.
- Processing table information into an ETI Schema object: A schema description file is processed to create an ETI Schema object which can be used in a conversion.

With this release of the Microsoft C/SQL Server DSL it is possible to gather table metadata information from within ETI Solution using the new SQL Server Live Connection feature. Using Live Connection, ETI Solution can query the SQL Server database directly for the table metadata.

To run the SQL Server Live Connection interface requires:

- ETI Solution release 5.2.2 patch 3 or later.
- Installation of, and license for, the Microsoft C/SQL Server 4.3 DA DSL
- Communication between the ETI Application Server machine and the SQL Server database.
- Microsoft Data Access Components (MDAC) release 2.6 or later.
- A user id and password to access the SQL Server database.
- Privileges within the SQL Server database to access all the tables you plan to retrieve.

² See *ETI Data System Library DA Procedures*, Chapter 2, “Representing Data as an ETI*Extract Schema” for more information on the retrieve schema process.

4.2.9.1 Getting the SQL Server Table Information

To obtain the SQL Server table information, select *Tools->SQL Server Live Connect* from your ETI Solution Client. The Live Connect dialog will be displayed:

The screenshot shows the 'SQL Server Live Connect' dialog box. It features a title bar with the text 'SQL Server Live Connect' and standard window control buttons (minimize, maximize, close). The main content area is divided into several sections:

- Connection Information:** This section contains four text input fields: 'Server', 'Database', 'User name', and 'Password'.
- Tables to collect:** This section contains two radio buttons: 'All tables' (which is selected) and 'Specific tables'. Below the radio buttons are three buttons: 'Add All Tables to List', 'Add Table', and 'Delete'. There is also a large empty rectangular area below the 'Add Table' button, likely intended for a list of table names.
- Action Buttons:** At the bottom of the dialog, there are two buttons: 'Get Table Information' and 'Get Data File'.
- Close Button:** A 'Close' button is located at the bottom right corner of the dialog.

4.2.9.2 Specifying Connection Information

The *Connection Information* pane of the Live Connect dialog contains the information needed to make the connection to the proper SQL Server database. The following information is required:

1. Server - the name of the server hosting the SQL Server database.
2. Database - the name of the SQL Server database to be accessed.
3. User Name - the SQL Server user name used to access the SQL Server database³.
4. Password - the password for the user id.

4.2.9.3 Identifying the Tables

Once you have specified the connection information, you can use the options within the *Tables to collect* pane to identify the tables to query for metadata information. You have two options for entry of the table name(s):

Using the *All Tables* option, you can have Live Connection obtain all tables accessible by the given user and add each to the list box within the *Tables to collect* pane.

Using the *Specific Tables* option, you can enter a specific table name in the text entry field and then select *Add Table* to add each table you to the *Tables to collect* list box.

Tables can be removed from the list of tables by selecting one or more tables within the list, and selecting the *Delete* button.

4.2.9.4 Getting the Table Information

Once you have all the tables specified, select the *Get Table Information* button to connection to the database and obtain table information for each table within the *Tables to collect* list box. Any errors or messages will be displayed in the text area at the bottom of the dialog.

When all the table information has been collected, the button *Get DataFile* will be enabled. Select the *Get Data File* button to download the table information to your machine. You should save this file in the MetaStore files area on the MetaStore server (by default the schema files for Microsoft C/SQL Server are stored in <MetaStore Area>\files\<MetaStore Name>\das\csqlsvr_da.1 directory). The downloaded file can be identified by a subsequent ETI Solution Retrieve Schema command to create a schema object to load into ETI Solution.

RELEASES 4.3.0 – 4.3.2

Not applicable. 4.3.3 is the first Generally Available 4.3 release of the DSL for Microsoft C/SQL Server DA.

³ This release of the Live Connect uses SQL Server Authentication and requires a SQL Server login account for the user accessing the database (Windows authentication is not currently supported).