SQL Schema Tool®

Version 1.0.2.1

User Manual

Overview

The SQL Schema Tool or SST as it is now known started as a command-line developer's tool. As such, the tool was designed to assist developers working against a SQL server database, so as to migrate any database changes from a development environment into a test or nightly build environment.

The SST is an application for generating XML schema documents from SQL server databases, comparing those databases or the XML schema documents, and finally generating a DiffGram XML document from the compare results.

Simply put, the purpose is to make a snapshot of the current latest database schema and merge those changes into a build of the previous database schema (live or snapshot).

The tool can make snapshots of databases, DTS packages, or even the data itself, to perform offline compare(s) and create update script(s). The tool can also be used to generate schema create scripts for archival or source control purposes.

As developers, testers, or deployment people, we want to take database B, and make it look like database A.

Table of Contents

Table of Figures

Design

Architecture of the application has been designed to deal with the database at the object level. This allows individual groups of objects to be selected for the compare or generate schema operations. It also allows the user of the application to narrow their focus to the exact objects of a database that are in flux, or changing often.

When you examine the XML snapshot you can see this approach, by noting the XML nodes or elements are grouped by object type, and sorted by object name.

All principal output from the application is in the form of XML, that is in turn transformed into the appropriate output, by the use of XSLT and XML transformations.

Future design will revolve around the expansion of that paradigm so as to allow easy customization of reports, or adding additional user designed outputs; i.e. changing scripts from one database manufacturer specific language to another.

Flexibility is the key to a changing code universe…

The SQL Schema Tool is designed to consider a master or source database, which is the current or latest database schema for use in a project. The destination or target database is considered by SST to be the older schema, or the schema that needs updating.

A major design consideration is to keep intact any data that exists in the target database. A second major design consideration is the ability to make schema comparisons in a fully or partially disconnected mode by the use of saving a XML schema snapshot.

Starting Team Database Development

You are a database developer who is using a development project to help create a database. Over time, your team builds successive versions of the database and deploys them to the testing servers. Other members of your team also contribute to the design of the database.

You have just finished making a round of changes to the database schema and tested the changes on your own computer. Because you found no errors, you now want to publish the updates to a testing server so that your team's quality-assurance staff can test them.

You start the **SQL Schema Tool**. You establish a connection to the testing server. You compare your project (the source) to the deployed database (the target) and determine that many objects differ between the two entities. But you also see that many changes are to database objects where the modifications are not your responsibility. Other people have made those changes, and they do not want them overwritten.

Other development scenarios might include the following:

Typically, large team development efforts use a single development database with one person responsibly to make those development schema changes to that database necessary for the coding requirements. That person may be a full time DBA, a developer performing dual duties, or the team lead. At a point the development database becomes fairly stable after the initial development phase of the middle tier objects domain.

Once that development database becomes stable, the process often moves to testing of code under development in a phased approach to minimize QA turns later on. At this juncture the development database needs to have the minor schema tweaks and sometimes even meta-data moved or merged into a testing database.

SST calculates the changes or difference between two databases to be made by using a source and a target database. In the SST environment, the source database is the master copy and the target the one to be updated. By careful selection of which database is the source you can merge changes from one database to another and back again.

SST can be automated using the command-line versions of the tools. These command-line versions offer almost all the same functionality of the GUI version.

Limitations

The tool does not support the SQL "WITH NOCHECK" or extended properties. Additionally, the tool only supports adding calculated columns using CREATE TABLE or ALTER TABLE ADD 'column name' SQL statements. Also, calculated columns are added last in order, so that the columns they may be dependant on will already be in place. Column Default Constraints are dropped using sp_unbindefault and they **are not** re-added later in the diffgram SQL script script using sp_bindefault.

Instead the tool assigns the default value to the column using the ALTER TABLE ADD DEFAULT 'default value' FOR column_name method. Doing the diffgram schema this way does cause a DB updated with the diffgram to not compare exactly using the Redgate® SQL Compare tool. However, the equivalent database functionality is still there.

The order of columns may also be different between the source and destination databases as the calculated columns are added to the table last. This by design so as to avoid any dependency issues with other columns that the calculation may have some dependency on.

Speaking of dependant objects, especially stored procedures or views, these are output in the order indicated in the sysdepends table. If the sysdepends table gets out of sync with the database, then the stored procedures may also be out of sync.

This tool uses embedded XSLT 'style sheets' to transform the generated XML documents into the SQL script files.

Check out the following URL for great tips on error handling in the SQL Language: SQL Error Handling (http://www.sommarskog.se/error-handling-II.html#whyerrorcheck)

UDDT Issue

SQL Commands for the following:

Alter Table

Add column

Can contain base column types along with collation type, if they are not null and have no default value then the tool requires an additional pass using ALTER Column to change to not null. If the column has a UDDT column type, it requires an additional SQL command using ALTER Column to change type using the tool.

ALTER OR ADD COLUMNS COLLATE clause cannot be used on user-defined data types.

Attempted -
ADD test3 varchar(10) COLLATE SQL_Latin1_General_CP1_CI_AS not null
or
ADD test3 varchar(10) not null
RESULT message:
ALTER TABLE only allows columns to be added that can contain nulls or have a DEFAULT definition specified.
Column 'test3' cannot be added to table 'ADMIN' because it does not allow nulls and does not specify a DEFAULT definition.

This works:

```
alter table ADMIN
      ADD test3 varchar(10) COLLATE SQL_Latin1_General_CP1_CI_AS null
alter table ADMIN
      ALTER COLUMN test3 varchar(10) not null
```

OR

```
alter table ADMIN
      ADD test3 varchar(10) COLLATE SQL_Latin1_General_CP1_CI_AS null
alter table ADMIN
      ALTER COLUMN test3 [uddt_name] not null
```

Current Feature List

1. Schema XML Snapshots, for portable SQL schema generation/compare without a database.
2. Schema generation – with selectable object types.
3. Schema compare – with selectable object types.
4. DTS package XML Snapshots, for creating portable DTS packages.
5. 2 Command-line tools for the above listed functions. This should allow ant build or other automation tools to compare or generate schema updates automatically.
6. Data compare – currently of individual tables only.
7. Data export – CVS and XLS, plan to add text file output for SQL insert statements.
8. Reports for data compare, Schema (tables only, though I hope to get this updated with more objects before testing is finished)
9. XML tree browser with XPath search of XML snapshots generated by the tools.
10. Run generated SQL against selected DB without switching to yet another tool.
11. Logging of all events, actions and errors - both SQL related and application type errors.

Installation

Run the MSI file by double-clicking it in file Explorer, or by right-clicking the MSI file and selecting the Install menu item. Choose an appropriate installation program directory. There will be three subdirectories created under that parent directory. Each of the three child directories has one of the tools that make of the SST application. There is the command-line tool for SQL schema, the command-line tool for DTS packages, and last but not least the SST GUI application that supports both of the command-line tool operations and much more.

Setup

The setup for SST is straightforward. Start the application, open the options dialog by selecting the Window\Options menu item. Once the options dialog is open there are a few check boxes regarding the application operations, dealing mostly with saving of windows position/size settings, and the reloading of open documents from the last closing of the application.

The other options settings are check boxes that allow the user to select which groups/types of database objects are acted on by the actions menu items.

The current list of selectable database objects is as follows (see above command line options for a more complete description):

- Tables
- Views

- Stored Procedures (Sprocs)
- User Defined Functions (Funcs)
- Triggers
- Rules
- Defaults
- User Defined Data Types (UDDTs)

The NLog settings are important to properly configure information, debug, warning, and error output. NLog settings allow output to be sent to multiple locations or separated by type. These settings are located in the NLog.config file that is in the same directory as the application. Each of the command-line tools also have their own copy of a NLog.config file.

Below is a sample NLog.config:

```
<?xml version="1.0" encoding="utf-8" ?>
<nlog xmlns="http://www.nlog-project.org/schemas/NLog.xsd"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

    <targets>
        <target name="file" xsi:type="File" fileName="log.txt" />
    </targets>

    <rules>
        <logger name="*" minlevel="Trace" writeTo="file" />
    </rules>
</nlog>
```


Using the Tools

SST From the command line:

For SQL Schema operations, use the SSTCL.EXE to start a schema compare of two databases, or generate a new schema from a single database.

See the following SST command line parameters:

```
/? or /Help for this help message

PARAMETERS: (SQL Connections or Files to use)
========================================================================
/Src_Server=[Source sql server]
/Src_Database=[Source sql catalog or database] OR
/Src_Catalog=[Source sql catalog or database]
/Src_Trusted - use trusted connection, user and pwd not required
/Src_User=[Source sql server user login] OR
/Src_UID=[Source sql server user login]
/Src_Password=[Source sql server user password] OR
/Src_PWD=[Source sql server user password]
/Src_File=[output filename for source SQL server schema create script]
/Src_Schema=[filename for existing source SQL server schema xml file]
```

When connecting to a SQL server:
   When wanting to just get the SQL create schema XML/SQL file(s)
   for a single SQL DB, only use the Src_?????? parameters or the
   /CreateXMLFile parameter.

   Only use the Dest_?????? parameters when doing a compare.

/Dest_Server=[Destination sql server]
/Dest_Database=[Destination sql catalog or database] OR
/Dest_Catalog=[Destination sql catalog or database]
/Dest_Trusted - use trusted connection, user and pwd not required
/Dest_User=[Destination sql server user login] OR
/Dest_UID=[Destination sql server user login]
/Dest_Password=[Destination sql server user password] OR
/Dest_PWD=[Destination sql server user password]
/Dest_File=[output filename for destination SQL server schema create
script]
/Dest_Schema=[filename for existing destination SQL server schema xml file]

When providing just a single schema or diff XML file,
   Use the following parameters:

/DiffXMLFile=[XML file name] - used by itself, this will apply the SQL diff
   transform against specified XML file. Turns on the transform flag.
   Autogenerates SQL file name from XML input file name if none is
   provided via the next parameter '/DiffSQLFile'.

/DiffSQLFile=[SQL file name] - used with the '/DiffXMLFile' paramter, this
   will assign the SQL file name used for the transformation output.

/CreateXMLFile=[XML file name] - used by itself, this will apply the SQL
   Create transform against specified XML file. Turns on the transform
   flag. Autogenerates SQL file name from XML input file name if none
   is provided via the next parameter '/CreateSQLFile'.

/CreateSQLFile=[SQL file name] - used with the '/CreateXMLFile' paramter,
   this will assign the SQL file name used for the transformation output.

/SqlObjectMask=[bitmask] – controls which SQL objects are compared or
   generated as XML and/or SQL. The Selection of ALL defined objects are
   the default.

   The option takes a HEX string such as 0xFF (for all defined objects).
   You combine the values below using an OR to produce a select list of
   object types to work with, such as combining tables or views would
   require a passed in value of 0x82.

```
        /// <summary>
        /// Indicates sproc object types.
        /// </summary>
        SPROC = 0x01,
        /// <summary>
        /// Indicates view object types.
        /// </summary>
        VIEW = 0x02,
```

```
            /// <summary>
            /// Indicates function object types.
            /// </summary>
            FUNCTION = 0x04,
            /// <summary>
            /// Indicates trigger object types.
            /// </summary>
            TRIGGER = 0x08,
            /// <summary>
            /// Indicates default object types.
            /// </summary>
            DEFAULT = 0x10,
            /// <summary>
            /// Indicates rule object types.
            /// </summary>
            RULE = 0x20,
            /// <summary>
            /// Indicates UDDT object types.
            /// </summary>
            UDDT = 0x40,
            /// <summary>
            /// Indicates table object types.
            /// </summary>
            TABLE = 0x80
```

OPTIONS: (changes operations of compare or output)

==========

/CompareSprocText - add this option to command line to compare source and dest stored procedure texts line by line. Otherwise if both source and dest sprocs exists, the source sproc is always forced to 'alter procedure'.

/Transform - add this option to command line to perform translation of the XML file into a SQL file for creating or updating the SQL Destination database. This requires some combination of Source and Destination SQL connections and/or XML Schema files. /Translate may be used interchangeably with /Transform.

/Primary - add this option to force all tables to use the Primary FileGroup for CREATE TABLE or ALTER TABLE when serializing the schema. This option will also cause the compare to ignore FileGroups, since The output all will be set to PRIMARY.

/NoThreads - add this option for operations to run sequentially, rather than threaded.

DTS From the command line:

For DTS Package operations, use the DTSCL.EXE to start a DTS serialization of a database DTS package into a XML DTS snapshot, or generate a new DTS Package from a XML DTS snapshot.

See the following DTS command line parameters:

```
Command Line Parameters:
[/?] or [help]         - display this window.
[/d]                   - display debug messages, [/h] overrides this.
[/f logfile name]      - set the package logging file name.
[/e password]          - set the package password.
[/h]                   - hide all debug and error message windows.
[/i {packageID}]       - sets the package ID to use.
                         ...use with [/n] option.
[/l DTS/XML file name] - load package from DTS/XML file.
                         ...use with [/n] option.
                         ...automatically adds the .XML extension.
[/n DTSPackageName]    - sets the DTS Package Name.
                         ...this is a REQUIRED parameter.
[/p password]          - change the default admin password.
[/r]                   - remove the existing DTS package by name.
                         ...use with [/n] option.
[/pi]                  - return package ID and version information.
                         ...use with [/n] option.
[/s sqlservername]     - change the default server.
                         ...this defaults to the local machine name.
[/t]                   - tests the SQL server connection, then exits.
[/u username]          - change the default admin user.
[/v {versionID}]       - sets the version ID to use.
                         ...use with [/n] option.
[/w]                   - use Windows authentication for SQL server.
                         ...overrides SQL mixed mode settings.
                         ...use with load, remove and save operations.
[/x XML file name]     - output package to XML file.
                         ...automatically adds the .XML extension.
                         ...use with [/n] option.
```

Actions

Actions are those methods or functions that provide the basis of the SST operations. The three main actions are Compare, Generate, and DTS package serialization. There are several additional actions that relate to the main actions; such as setting/changing the SQL server login security, or getting the current list of DTS packages from a server.

There are also some actions that may only appear as right mouse clicks that display context menu items, such as synchronization of the XML document with the XML Node Explorer, selecting objects for comparison, expanding the list of data tables in a database, or exporting the data from two database tables as files.

The compare action creates XML snapshots of both a source and a target database. These snapshots can be used again later on for offline or disconnected schema compare or generate operations.

Additionally a difference XML snapshot is created and can be used at any time in an offline fashion to create a SQL update script for any given database of a know target schema.

In the world of the SST application, the source database is the database to compare against, or to create schema from. It is the 'Gold' standard. The target database is exactly that, a moving target or the database that always needs its schema updated.

Automating the operations:

A batch file can be constructed to generate schema, compare schema of both live databases and XML schema snapshots that were previously generated.

The following is an example of the text required for such a batch file:

```
C:\Program Files\Sql Schema Tool\SST CommandLine\SSTCL.exe /src_server=llewis-lt\llewis2000 /src_catalog=aus4 /src_uid=sa /src_pwd=somepassword /dest_server=llewis-lt\llewis2000 /dest_catalog=uhc /dest_uid=sa /dest_pwd=somepassword /translate
```

… (Functionality needs to be added to generate necessary bat files)

Previewing the XML Database Snapshot

Generating or Comparing database schemas will automatically generate a XML database snapshot of the source or the target database(s) and display the snapshot(s) in the XML document window. The XML database snapshot can then be transformed into a SQL create script at some later point in time. A HTML and a Standard XML report can also be viewed from the database snapshot. This report shows the schema of the source or target database.

Previewing the XML Diff Snapshot

Comparing database schemas will automatically generate a XML difference snapshot and display it in the XML document window. This XML diff snapshot

can then be transformed into a SQL update script at some later point in time. A HTML and a Standard XML report can also be viewed from the diff snapshot. This report shows any changes to the schema of the target database.

Previewing the SQL Update Script

Comparing database schemas will automatically generate a synchronization script and display it in the SQL Schema update script document window. This update script can then be run (or edited and re-saved first) against a designated SQL server and a selected target database.

Previewing the XML DTS Package Snapshot

Generating a DTS package snapshot will automatically generate a XML DTS snapshot and display it in the XML document window. This XML snapshot can then be transformed into a DTS Package at some later point in time.

Reviewing the Table Data Compare Report

Comparing two database tables will produce an HTML report of the table data differences using a color coded key.

Exporting Data

Data from any table can be exported as CSV or XLS (MS Excel) file formats. As a CVS file, it would be possible to compare the data using a third party text file compare tool, similar to WinDiff or others.

XML Tree Explorer View

Any XML document can be loaded and nodes can searched for using standard XPath queries. The last document loaded will auto populate the XML Tree Explorer. XML Documents previously loaded can re-sync their nodes with the XML Tree Explorer View. The XML Tree Explorer also was a tabbed view using a property grid to display individual information about the selected XML node in the tree view.

SQL Server Explorer View

Any SQL Server database and tables can be viewed through this explorer. There is a refresh button/menu that will search for any viewable local or networked SQL servers. You can also manually add a SQL server to the SQL Tree view. Settings are persisted when the application closes, if the option setting is enabled (default is enabled).

Settings Options Window

Settings can be changed in the options window. Setting defaults are chosen on the initial installation of the application. Settings also include SQL compare options that allow the selection of which SQL database objects to include in the generate/compare actions.

Screen Shot Images

Fig. 1, Security Settings Dialog

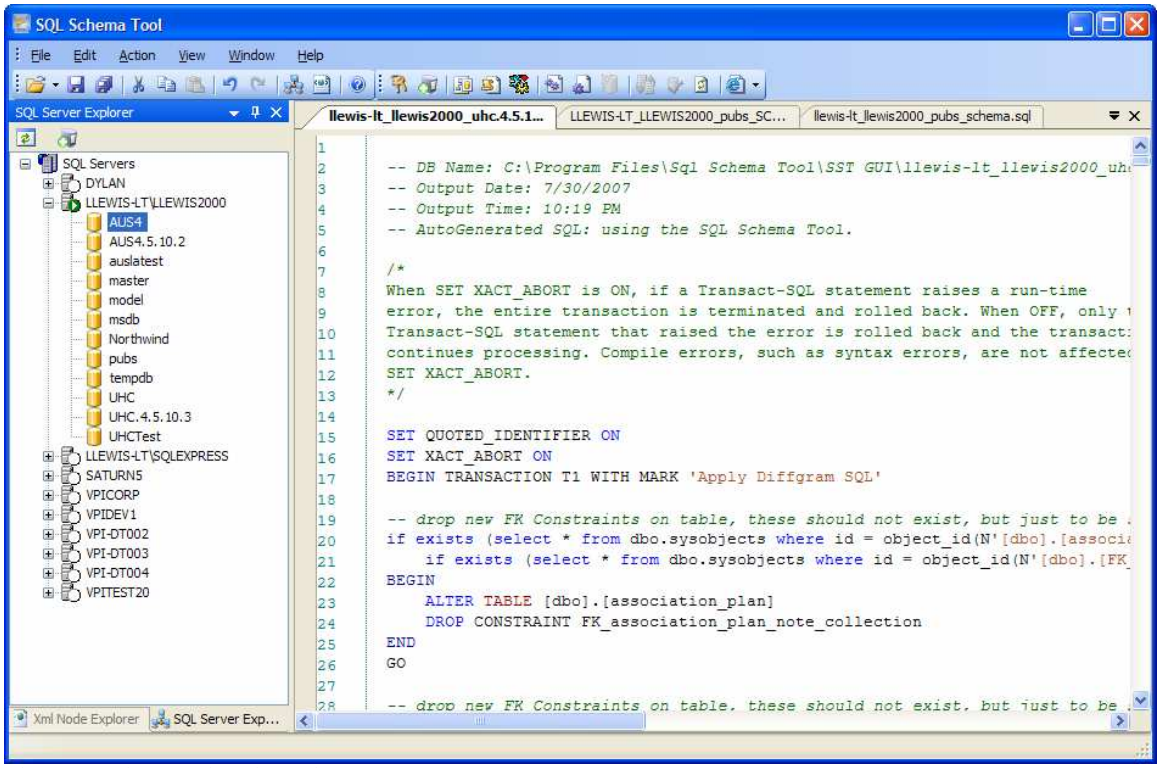


Fig. 2, Main Application Window

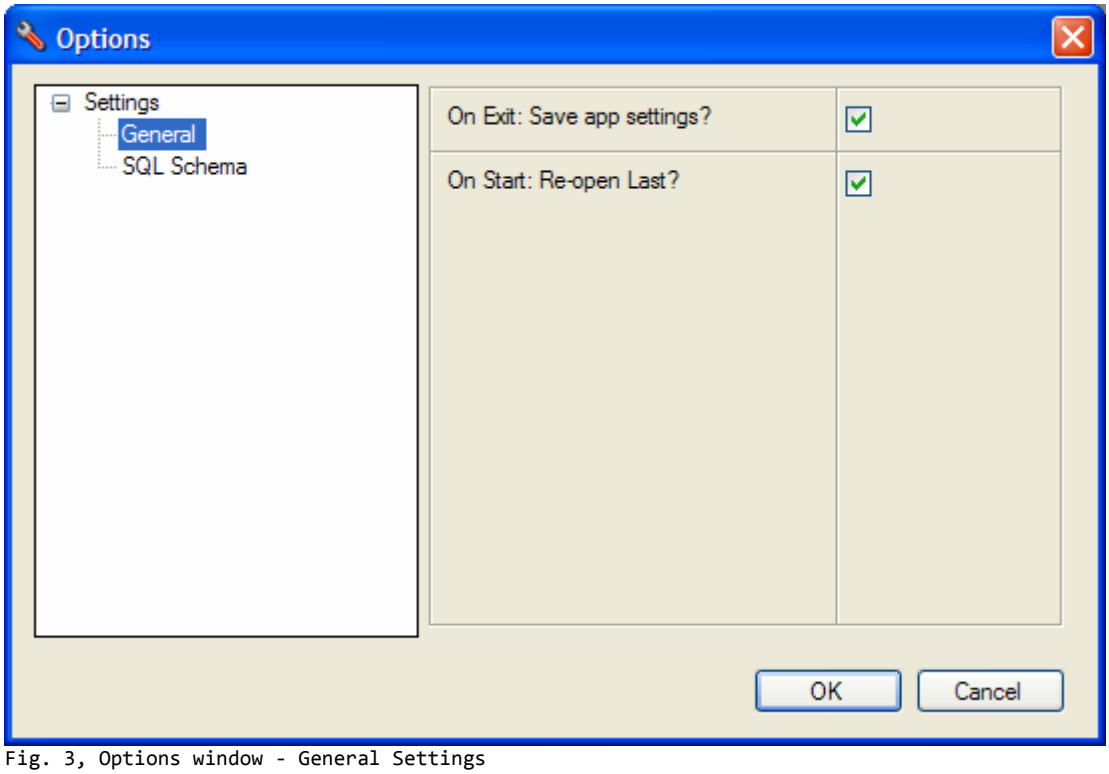


Fig. 3, Options window - General Settings

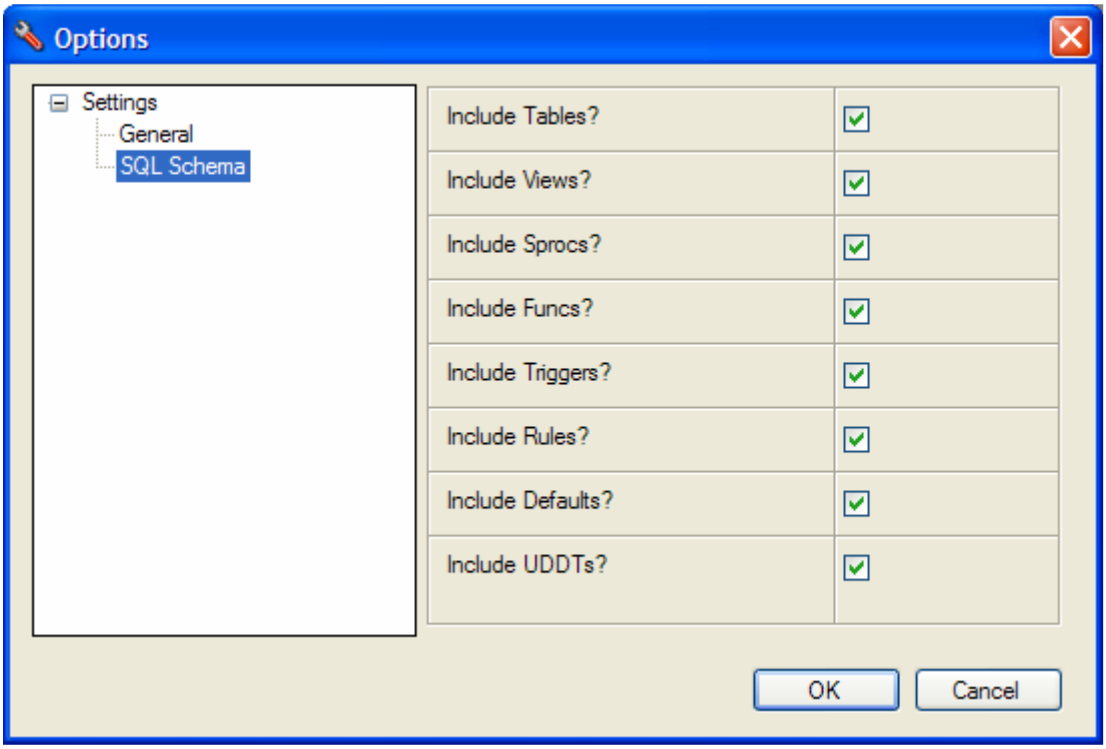


Fig. 4, Options window - Schema Settings

Fig. 5, About Window

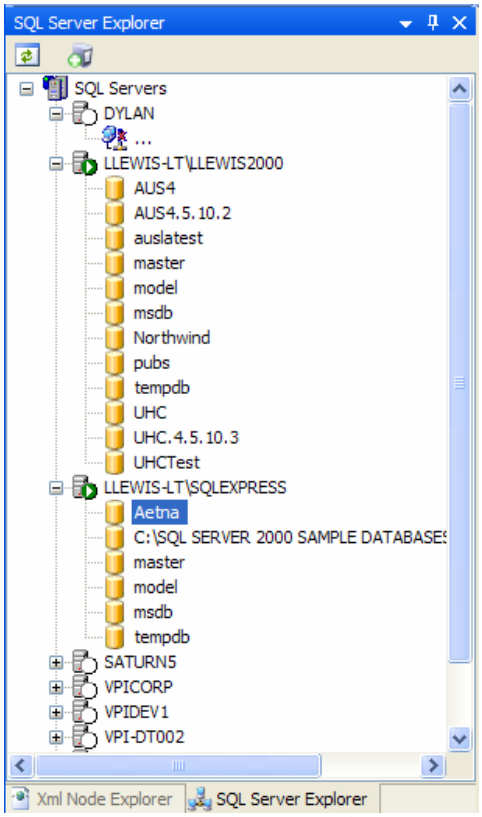


Fig. 6, SQL Server Explorer View

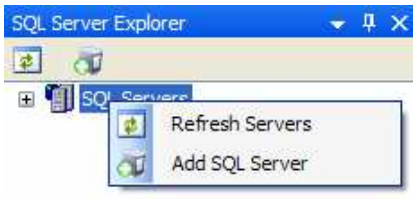


Fig. 7, Servers/Domain Context Menu

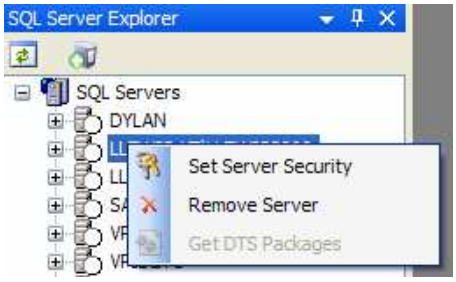


Fig. 8, Server Context Menu

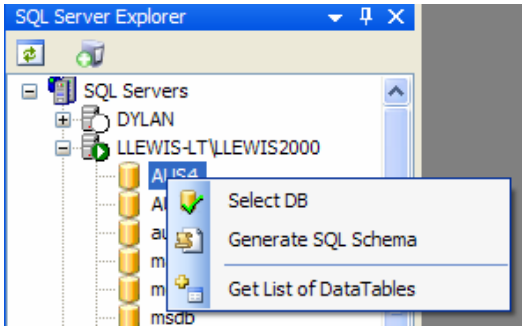


Fig. 9, Database Context Menu

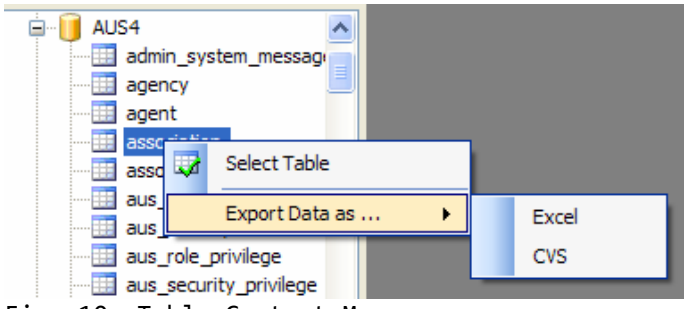


Fig. 10, Table Context Menu

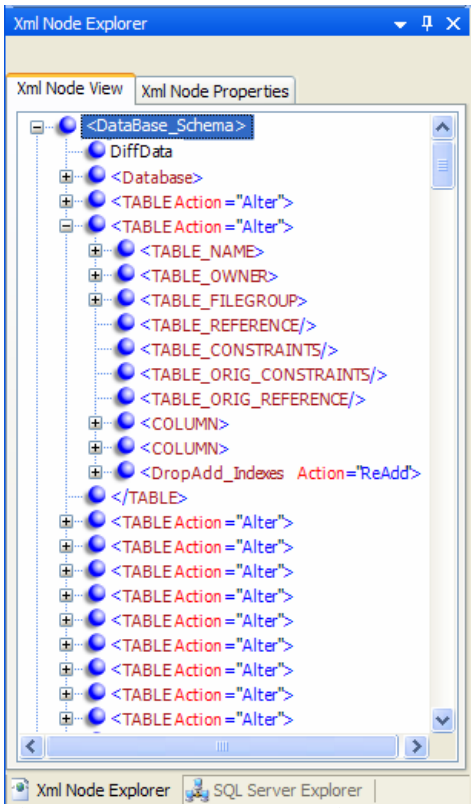


Fig. 11, Xml Node Explorer View

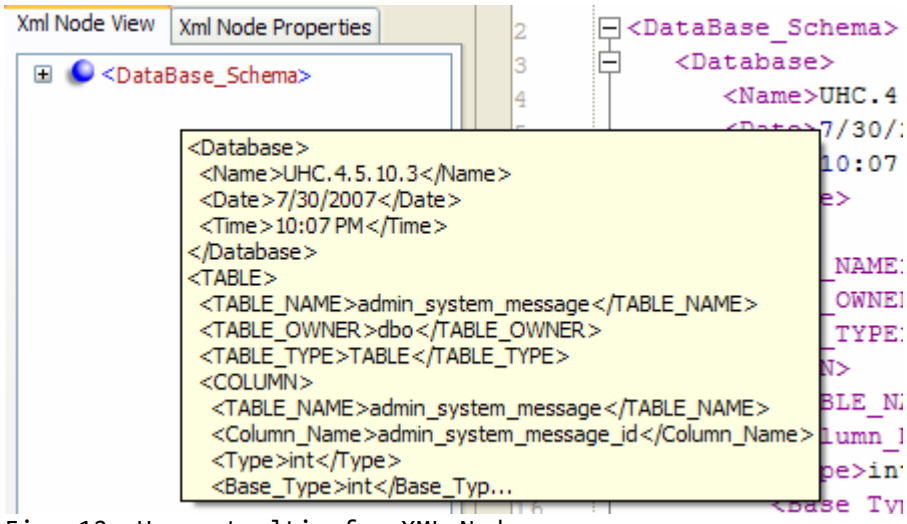


Fig. 12, Hover tooltip for XML Node

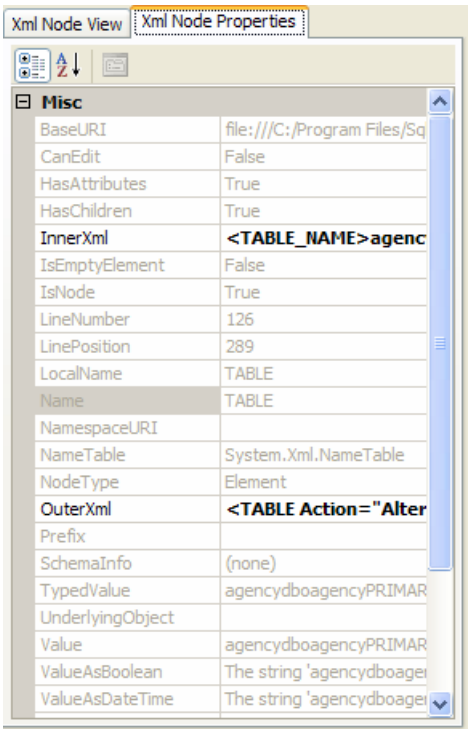


Fig. 13, Xml Properties for selected Xml Node

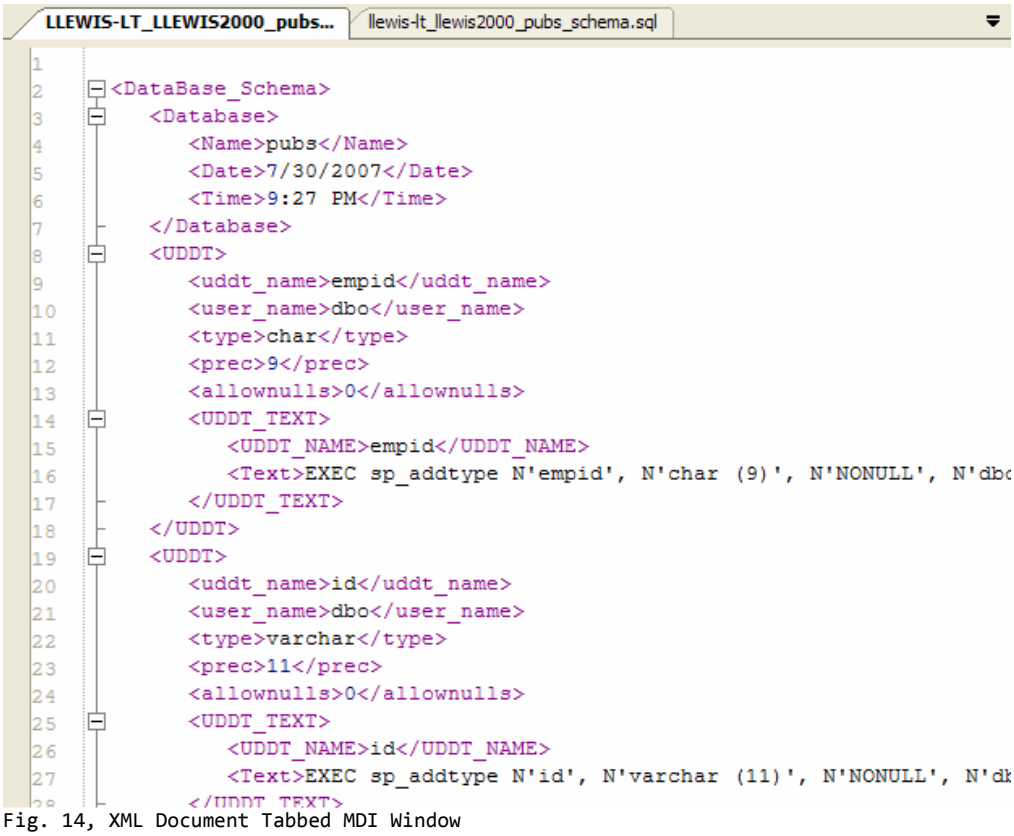


Fig. 14, XML Document Tabbed MDI Window

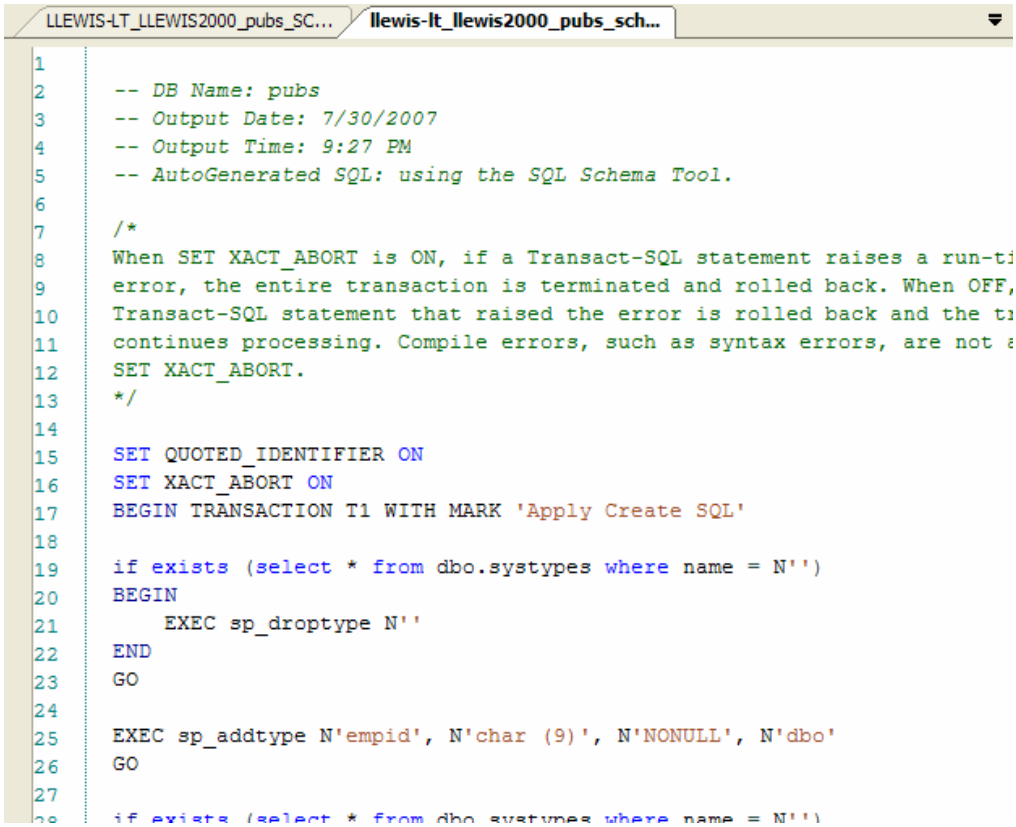


Fig. 15, SQL Document Tabbed MDI Window

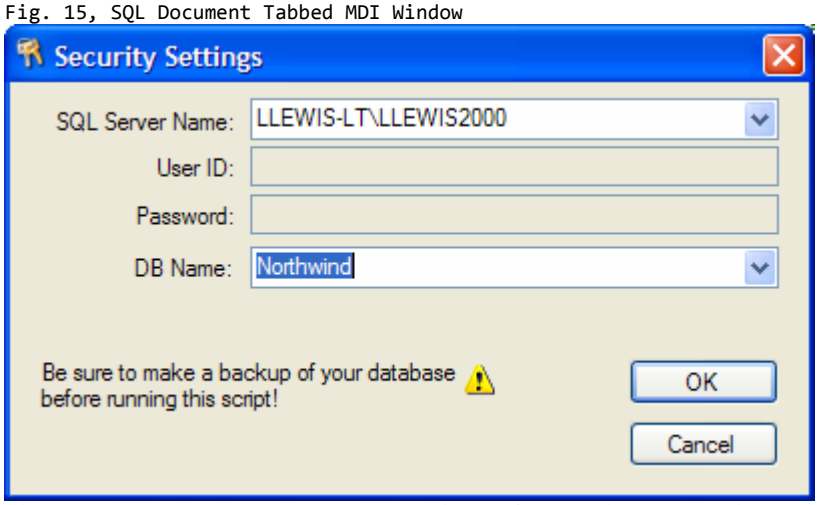


Fig. 16, Server and Database Select Window for Running SQL Script Against

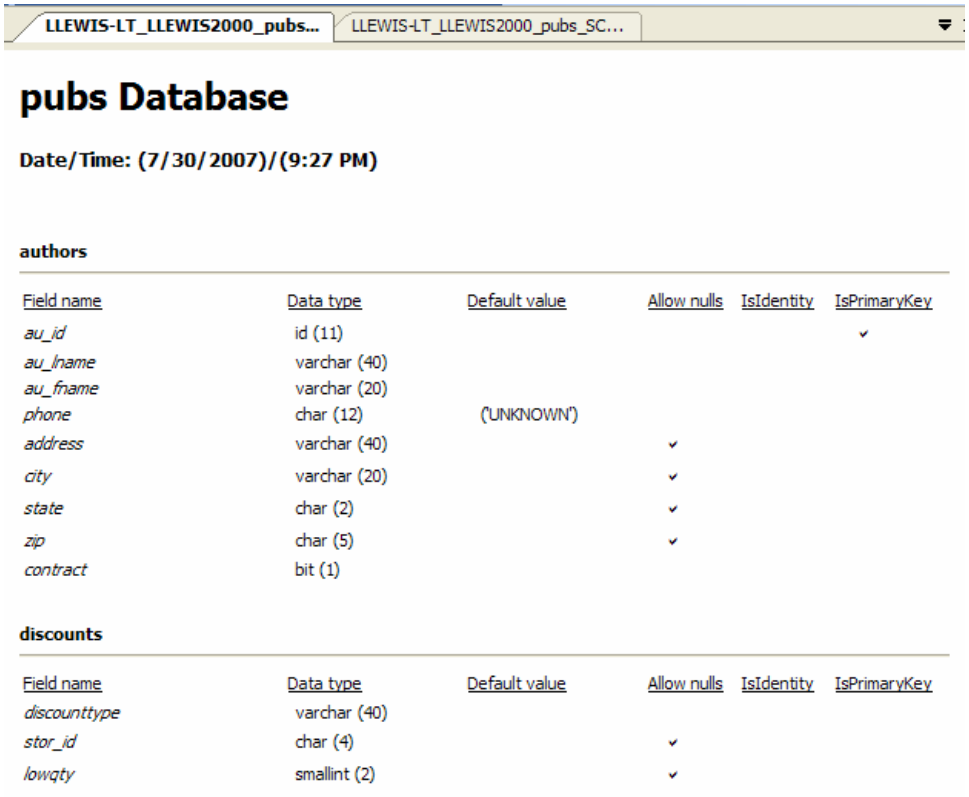


Fig. 17, HTML Report Tabbed MDI Window

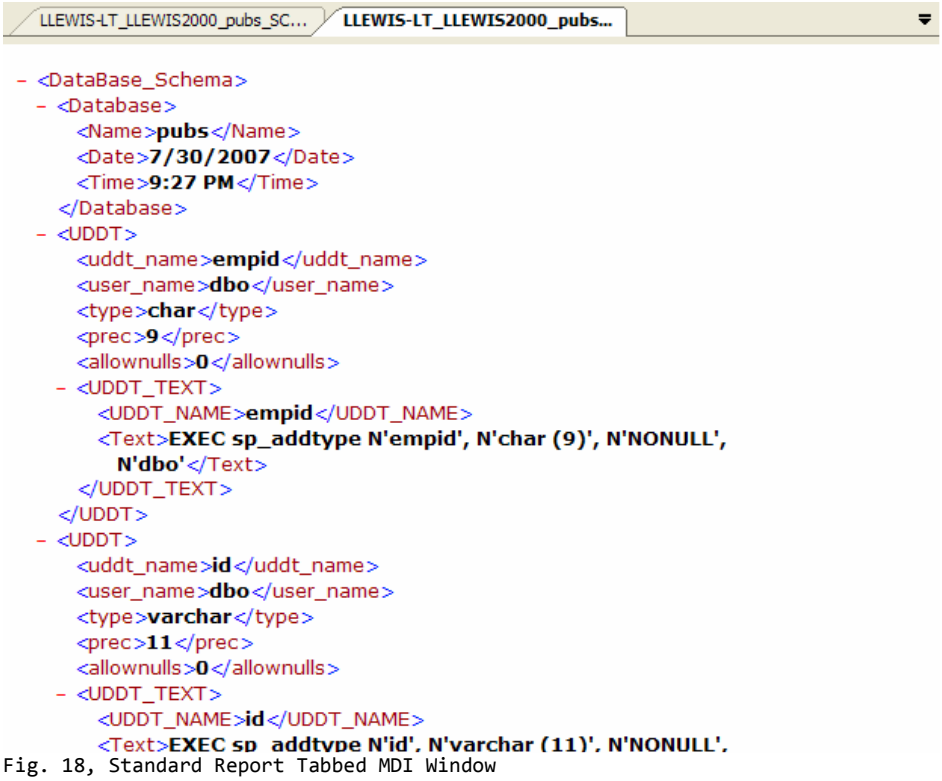


Fig. 18, Standard Report Tabbed MDI Window

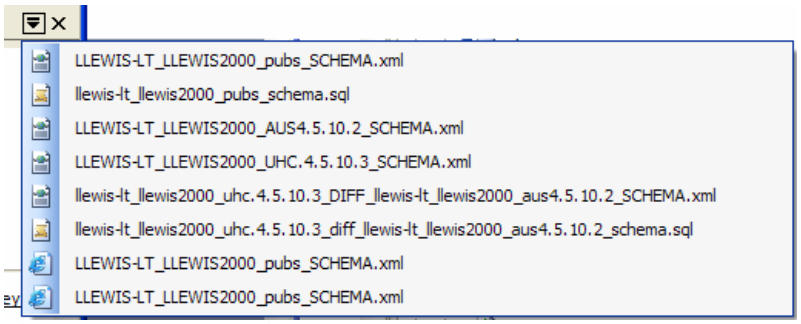


Fig. 19, Fly out MDI Window Selector

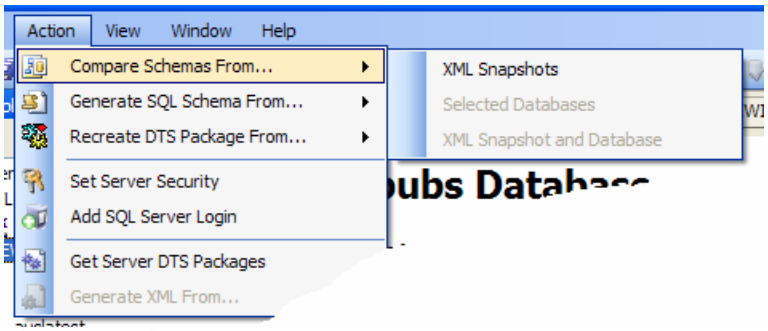


Fig. 20, Action Menu Items

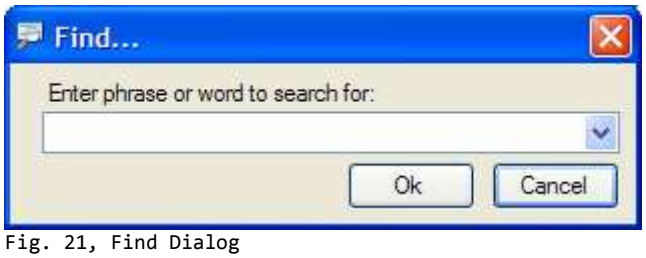


Fig. 21, Find Dialog

Fig. 22, Find and Replace (All) Dialog

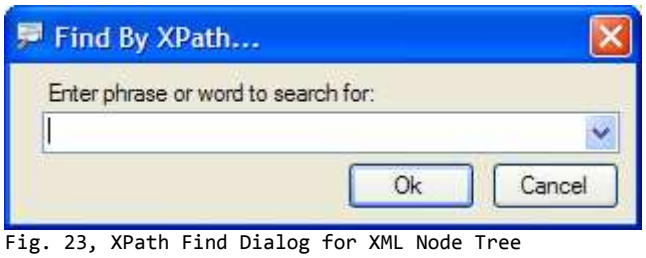


Fig. 23, XPath Find Dialog for XML Node Tree

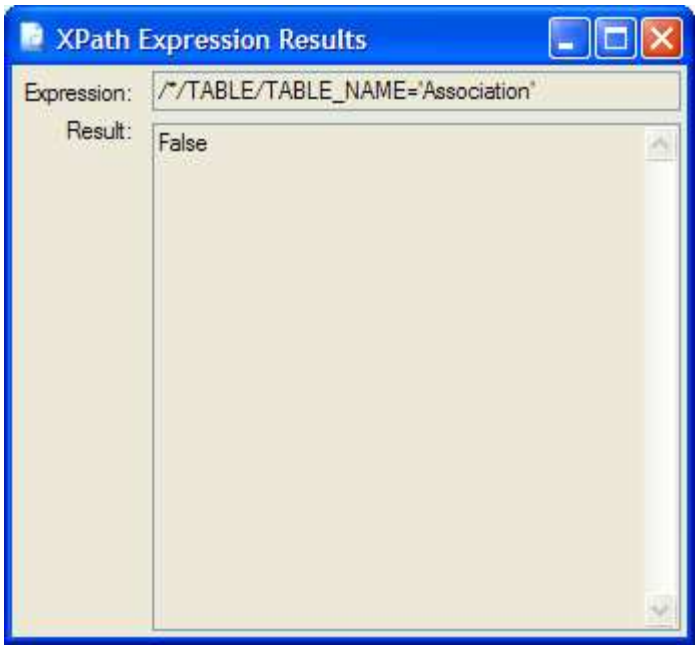


Fig. 24, XPath Results Dialog when error in XPath.

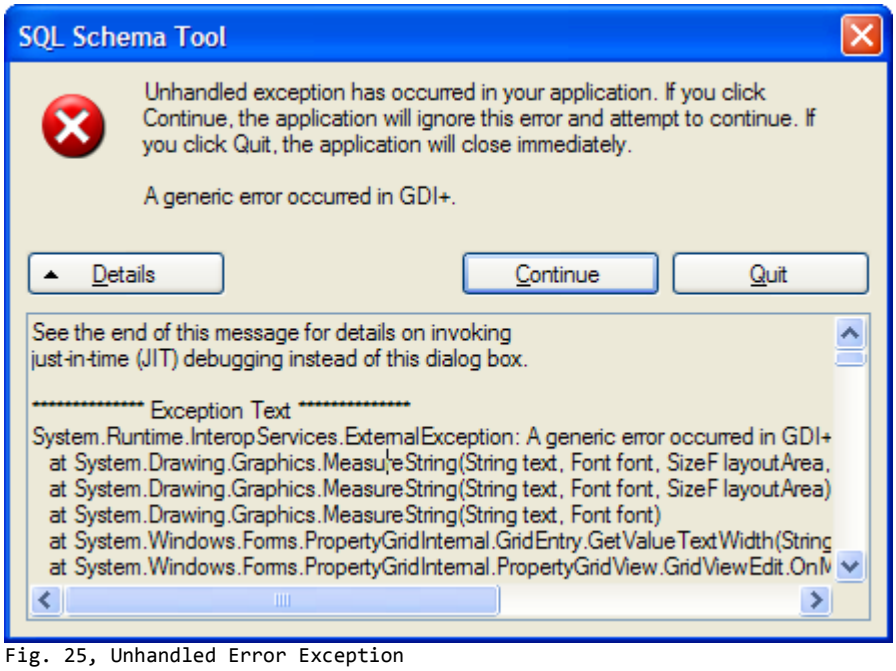


Fig. 25, Unhandled Error Exception

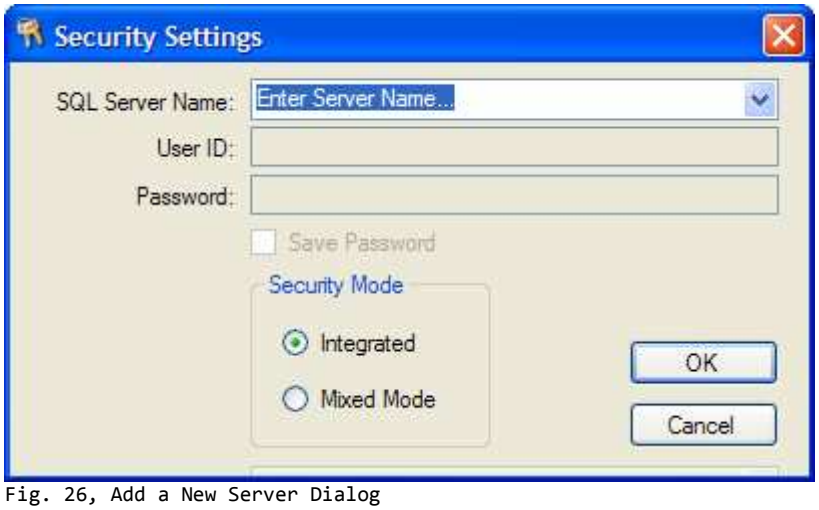


Fig. 26, Add a New Server Dialog

Fig. 27, Serialize DTS Package Dialog

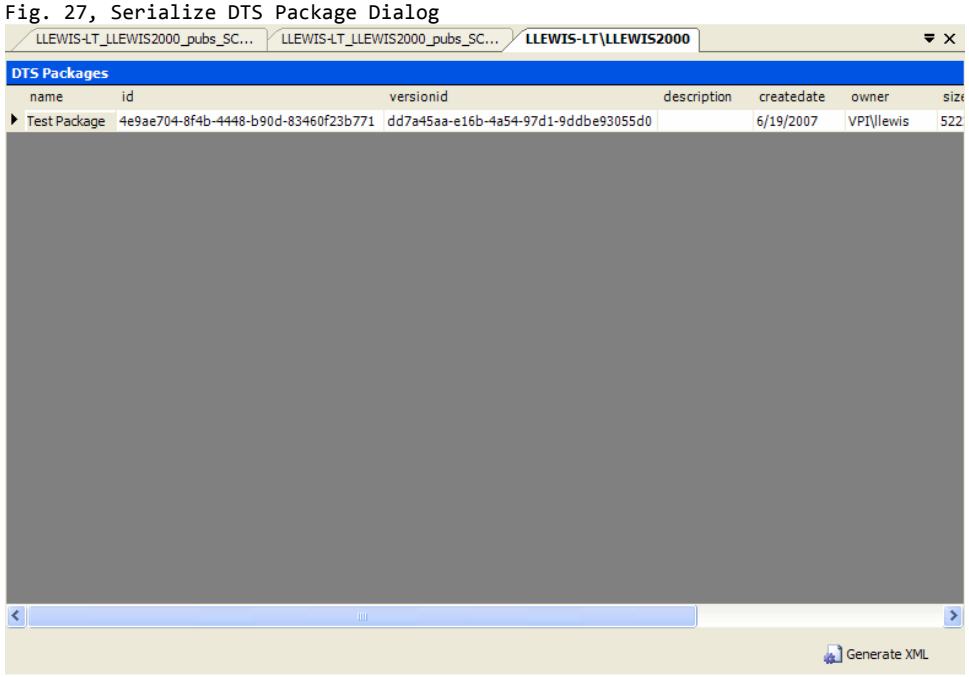


Fig. 28, DTS Package Explorer Tabbed MDI Window

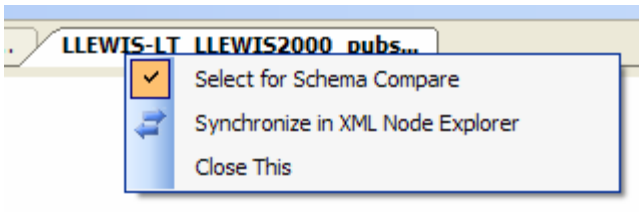


Fig. 29, Tabbed MDI XML Document Context Menu

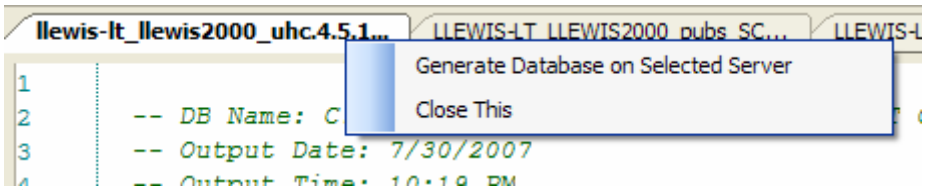


Fig. 30, Tabbed MDI SQL Document Context Menu

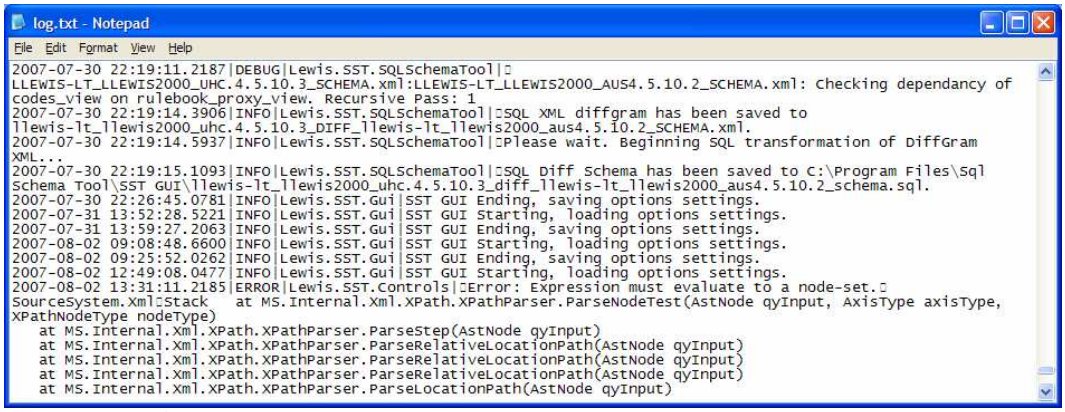
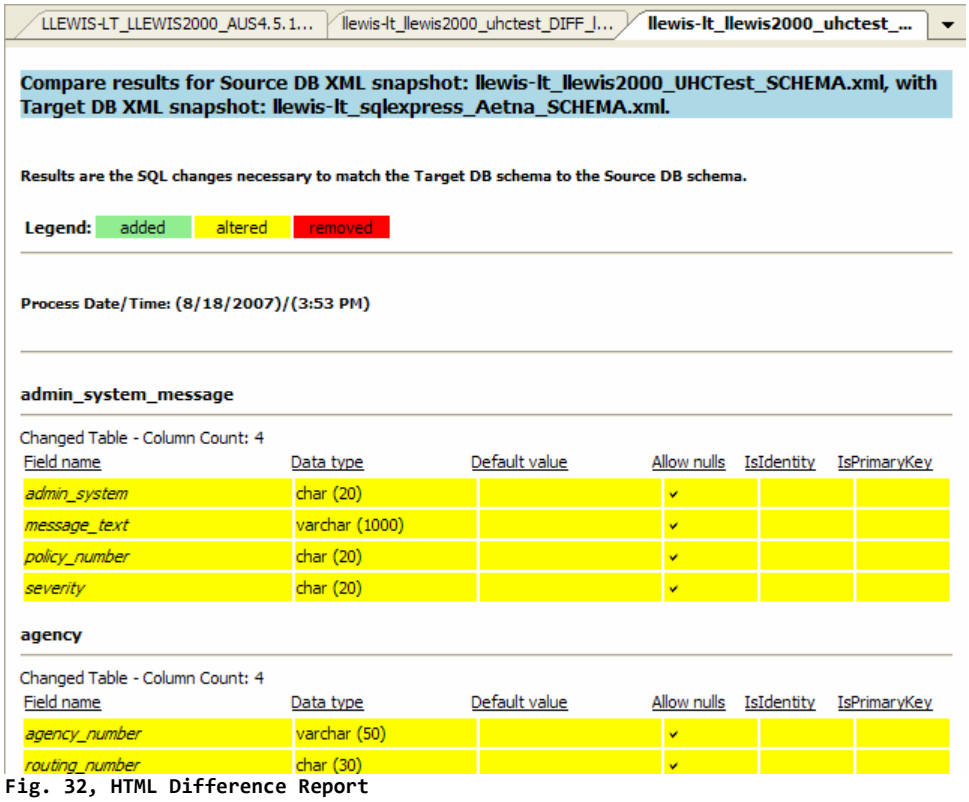


Fig. 31, Log from SST GUI Tool



LLEWIS-LT_LLEWIS2000_AUS4.5.1... | llewis-lt_llewis2000_uhctest_DIFF_l... | **llewis-lt_llewis2000_uhctest_...**

Compare results for Source DB XML snapshot: llewis-lt_llewis2000_UHCTest_SCHEMA.xml, with Target DB XML snapshot: llewis-lt_sqlexpress_Aetna_SCHEMA.xml.

Results are the SQL changes necessary to match the Target DB schema to the Source DB schema.

Legend: added | altered | removed

Process Date/Time: (8/18/2007)/(3:53 PM)

admin_system_message

Changed Table - Column Count: 4

Field name	Data type	Default value	Allow nulls	IsIdentity	IsPrimaryKey
admin_system	char (20)		✓		
message_text	varchar (1000)		✓		
policy_number	char (20)		✓		
severity	char (20)		✓		

agency

Changed Table - Column Count: 4

Field name	Data type	Default value	Allow nulls	IsIdentity	IsPrimaryKey
agency_number	varchar (50)		✓		
routing_number	char (30)		✓		

Fig. 32, HTML Difference Report

License

SQL Schema Tool ("SOFTWARE")

END-USER LICENSE AGREEMENT FOR *SQL Schema Tool* ("SOFTWARE").

PLEASE READ THE FOLLOWING LICENSE AGREEMENT. IT WILL BE NECESSARY FOR YOU TO AGREE TO BE BOUND BY THE TERMS OF THIS AGREEMENT BEFORE BEING PERMITTED TO CONTINUE TO USE THE PRODUCT.

VERY IMPORTANT-READ CAREFULLY: This LEWIS & LEWIS CONSULTING License Agreement ("LICENSE") is a legal agreement between you (either an individual or an entity) and LEWIS & LEWIS CONSULTING, for the *SQL Schema Tool* ("SOFTWARE").

By installing, copying or otherwise using the SOFTWARE, you agree to be bound by the terms of this LICENSE. If you do not agree to the terms of this LICENSE, you cannot use this SOFTWARE for any purposes.

The SOFTWARE is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The SOFTWARE is licensed, not sold.

I. LICENSE GRANT. This LICENSE grants you the following rights:
A. You may use a single of copy of this SOFTWARE identified above on a single computer or a single virtual OS session. Use on more than one computer system or virtual OS session, requires the purchase of a multiple use license. The SOFTWARE is in "use" on a computer when it is loaded into temporary memory (i.e. RAM) or installed into permanent memory (e.g. hard disk, CD-ROM, or other storage device) of that computer.

B. You may use the SOFTWARE for the limited purposes of demonstrations and trials.

II. TITLE; COPYRIGHT. All title and copyrights in and to the SOFTWARE and any copies of the SOFTWARE are owned by LEWIS & LEWIS CONSULTING. The SOFTWARE is protected by copyright laws and international treaty provisions. Therefore, you must treat the SOFTWARE like any other copyrighted material.

III. ADDITIONAL RIGHTS AND LIMITATIONS.
A. Reverse Engineering, Decompilation, and Disassembly. You MAY NOT reverse engineer, decompile, or disassemble the SOFTWARE.
B. No Separation of Components. The SOFTWARE is licensed as a single product and the software programs comprising the SOFTWARE may not be separated for use by more than one user at a time.

B. Termination. Without prejudice to any other rights, LEWIS & LEWIS CONSULTING may terminate this LICENSE if you fail to comply with the terms and conditions of this LICENSE. In such event, you must destroy all copies of the SOFTWARE.

LIMITED WARRANTY.

EXCEPT WITH RESPECT TO REDISTRIBUTABLES, WHICH ARE PROVIDED "AS IS," WITHOUT WARRANTY OF ANY KIND, LEWIS & LEWIS CONSULTING WARRANTS THAT THE SOFTWARE WILL PERFORM SUBSTANTIALLY IN ACCORDANCE WITH THE ACCOMPANYING WRITTEN MATERIALS. NO OTHER WARRANTIES. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, LEWIS & LEWIS CONSULTING DISCLAIMS ALL OTHER WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WITH REGARD TO THE SOFTWARE. THIS LIMITED WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS. YOU MAY HAVE OTHERS, WHICH VARY FROM STATE/JURISDICTION TO STATE/JURISDICTION.

NO LIABILITIES FOR CONSEQUENTIAL DAMAGES.
TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL LEWIS & LEWIS CONSULTING BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR ANY OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE, EVEN IF LEWIS & LEWIS CONSULTING HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME STATES AND JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATION MAY NOT APPLY TO YOU.

LICENSEE ACKNOWLEDGES THAT IT HAS READ AND UNDERSTANDS THIS AGREEMENT AND AGREES TO BE BOUND BY ITS TERMS. LICENSEE FURTHER AGREES THAT THIS AGREEMENT IS THE COMPLETE AND EXCLUSIVE STATEMENT OF THE AGREEMENT BETWEEN LICENSEE AND LICENSOR, AND SUPERCEDES ANY PROPOSAL OR PRIOR AGREEMENT, ORAL OR WRITTEN, AND ANY OTHER COMMUNICATIONS RELATING TO THE SUBJECT MATER OF THIS AGREEMENT.