

# DTS xChange Miscellaneous and Troubleshooting

Last Modified on 30 September 2021

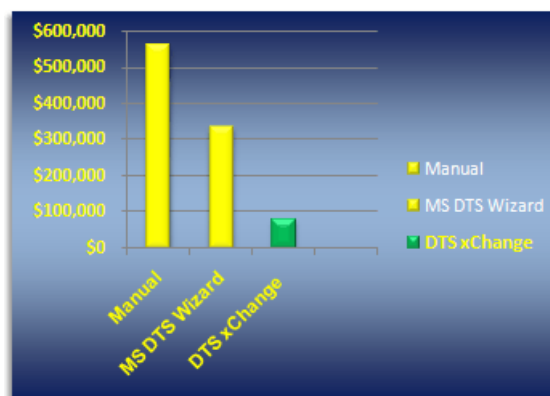
**EOL:** DTSxChange reached its end of life date on October 1, 2021. See the [Solarwinds End of Life Policy](#) for more information.

## Licensing and Product Questions

### How can DTS xChange save time and money?

DTS xChange is developed by a team of experienced developers and DBAs who have a lot of experience with DTS to SSIS migration. With thousands of packages migrated, and having already faced the challenges you're likely to face during migration, our developers understand the pain of the migration process. We packaged all the necessary functionality for DTS migration in an easy to use and solid framework: DTS xChange.

DTS xChange uses a proprietary rule based engine to migrate DTS Packages to SSIS by applying SSIS best practices. DTS xChange helps you to get a higher migration success rate, and new features supported by SSIS (Event Handlers, Logging, XML Configuration Files etc). If you're planning a manual migration without using any tools, consider the following features of DTS xChange which will save you significant time and money:



### Top Ten Features you get with DTS xChange

1. Automatic Variable migration.
2. Automatic Connection migration.
3. Automatic Task migration.
4. Automatic child package migration.
5. Improved support for Flat Files. Spend less time troubleshooting mapping and datatype issues for flat files.
6. Migrates Dynamic Properties Task.
7. Migrates UDL connections.
8. Supports Event Handler Logging. This feature saves you significant amounts of time if you're planning to add auditing for package/task execution.

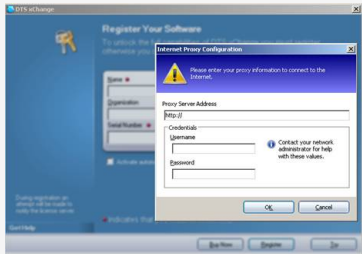
9. Consolidates connections automatically so you don't end up with several connections pointing to the same datasource.
10. Makes your connection dynamic using config files so you can easily switch packages from Dev to Prod environments without modifying several connections of hundreds of packages.

## Is there any 3rd party tool available in the market other than DTS xChange to help with DTS to SSIS migration?

No - DTS xChange is the only available third party tool. Microsoft has an inbuilt DTS Migration Wizard, but studies suggest that the Microsoft Migration Wizard has a very low success rate. DTS xChange has several enhancements and new features compared to the Microsoft DTS Migration Wizard that save you significant time and money.

## Product registration issues

The following section lists some common product registration issues you may encounter with DTS xChange:

Error	Reason and Solution	Image
<p><b>ERROR: Could not activate license at server. The underlying connection was closed: Could not establish trust relationship for the SSL/TLS secure channel.</b></p>		
<p><b>ERROR : The remote server returned an error: (407) Proxy Authentication</b></p>		
<p><b>ERROR: remote server returned an error: (400) Bad Request. --&gt; The remote server returned an error: (400) Bad Request.</b></p>		
<p><b>ERROR: remote server returned an error: (403) Forbidden. --&gt; The remote server returned an error: (403) Forbidden.</b></p>	<p>Solution: Press <b>Ctrl + F7</b> on the Registration screen when you enter the serial number and you get the following proxy setting dialog box. Enter proxy credentials and select OK and then Register.</p>	

## DTS and SSIS Questions

### How to pass parameters to DTS or SSIS package

A Parameterized package is when the package executed from the command line with parameters passed to it.

Most common parameters are global variable initial value. This is an issue related to command line, and has nothing to do with Package migration. When you migrate your SQL Server 2000 Jobs to SQL Server 2005 you might want to review command line changes.

### Parameterized DTS Package - Passing parameters to DTS package from command line

```
DTSRun /S "(local)" /N "PkgTest" /A "varFile":"8"="File_001.xls" /E
```

Note: "8" is DataType ID

### Parameterized SSIS Package - Passing parameters to SSIS package from command line

```
DTSExec /SQL "\PkgTest" /SERVER "(local)" /SET "\Package.Variables[User::varFile].Value";"File_001.xls"
```

[Additional Information](#): For more information about DTExec command line options, see the [dtexec Utility \(SSIS Tool\) MSDN article](#).

## What's new in SSIS 2008?


The following section lists the major changes you can see in SSIS 2008:

SSIS 2008 Change	Description
New Script Environment	The Script Task now supports VB.net and both C# languages. In the previous version of SSIS only VB.net was supported.
Enhanced SQL Statements	Transact-SQL supports the use of a MERGE operation in a SQL statement. The MERGE operation enables you to express multiple INSERT, UPDATE, and DELETE operations in a single statement against a specified target table
Change Data Capture (CDC)	Integration Services includes a new technology called change data capture. This new Database Engine feature captures insert, update, and delete activity that's applied to SQL Server tables. Change data capture also makes the details of these changes available in an easily consumed relational format.
DataFlow task Improvements	The new DataFlow engine has smarter thread allocation and processing of execution trees. Long chains of synchronous transforms are optimized, and most packages see a performance increase up to 80% on multi-processor machines.
	The Data Profiling task is a new task in the Integration Services toolbox. You can use this task inside an Integration Services package to profile data that is stored in SQL Server. The information provided by the profile helps you identify potential problems with data quality. The Data Profiling task provides profiles that help identify data quality problems within individual columns and with column relationships:

<p><b>SSIS 2008 Change</b></p> <p>New Data Profiling Task and Data Profile Viewer</p>	<p><b>Profiles that help identify problems within individual columns</b></p> <ul style="list-style-type: none"> <li>• The distribution of lengths in the column values</li> <li>• The percentage of null values</li> <li>• The distribution of values in the column</li> <li>• Column statistics for numeric columns</li> <li>• Regular expressions that match string columns</li> </ul>	<p><b>Profiles that help identify problems within column relationships</b></p> <ul style="list-style-type: none"> <li>• Candidate key columns</li> <li>• Functional dependencies between columns</li> <li>• The inclusion of the set of values in one column in the set of values in another column</li> </ul>
<p>Enhanced Performance and Caching for the Lookup Transformation</p>	<p>Performance enhancements to the Lookup transformation include faster cache loading and more efficient lookup operations. These enhancements are possible because of the following features:</p> <ul style="list-style-type: none"> <li>• The ability to take rows that do not have matching entries in the reference dataset and load those rows into the cache.</li> <li>• The ability to use separate data flows to load the reference dataset into the cache and to perform lookups on the reference dataset.</li> </ul>	
<p>New ADO.NET Components</p>	<p>Integration Services now includes the following ADO.NET components:</p> <ul style="list-style-type: none"> <li>• An ADO.net source component that consumes data from a .net Framework provider and makes the data available to the data flow. An ADO.net destination component that loads data into a variety of ADO.net-compliant databases that use a database table or view.</li> </ul>	
<p>New Date and Time Data Types</p>	<p>The following new date and time data types are available in Integration Services:</p> <ul style="list-style-type: none"> <li>• DT_DBTIME2</li> <li>• DT_DBTIMESTAMP2</li> <li>• DT_DBTIMESTAMP2OFFSET</li> </ul> <p>These new Integration Services data types provide the following benefits:</p> <ul style="list-style-type: none"> <li>• Support for a larger scale for fractional seconds.</li> <li>• Support of user-defined precision.</li> <li>• Support for a time zone offset.</li> </ul>	
	<p>You can create debug dump files (.mdmp and .tmp) that provide information</p>	

New Debug Dump Files  
**SSIS 2008 Change**

about what happens when a package runs. This information can help you in  
**Description** troubleshooting issues that occur when you run the package.

 **Additional Information:** For more information about the updates to SSIS 2008, see the following sources:

- [What's new in SQL Server 2008 for SSIS - Part one](#)
- [What's New \(Integration Services 2008\)](#)

## How to loop through files in a specified folder, load one by one and move to archive folder using SSIS

See the blog post [How to loop through files in a specified folder, load one by one and move to archive folder using SSIS](#).

## How to send HTML email in SSIS

SSIS improved the Send Mail Task by adding support of SMTP. The Send Mail Task still lacks the most common email requirement, sending emails in the HTML format. The Send Mail Task in SSIS only supports the text format, and you'll need to use the script task and write some VB.net code to send emails in the HTML format. You can use the following reusable routine to send HTML formatted emails:

```

// Example :
// SendMail("someone@mycompany.com, _
//     "support@dtsxchange.com", _
//     "HTML Test Email!!!", _
//     "Hello How are you?", _
//     True)
Private Sub SendMail(_
    ByVal SendTo As String, _
    ByVal From As String, _
    ByVal Subject As String, _
    ByVal Body As String, _
    Optional ByVal IsBodyHtml As Boolean = True, _
    Optional ByVal SMTPServer As String = "localhost", _
    Optional ByVal UserName As String = "", _
    Optional ByVal Password As String = "", _
    Optional ByVal Domain As String = "", _
    Optional ByVal Attachments As String = "")    Dim oMessage As System.Net.Mail.MailMessage
Dim mySmtpClient As System.Net.Mail.SmtpClient    oMessage = New System.Net.Mail.MailMessage(From, Sen
dTo, Subject, Body)
oMessage.IsBodyHtml = IsBodyHtml
//Attachments
If Not String.IsNullOrEmpty(Attachments) Then
    Dim sFiles() As String
    Dim sFile As String
    sFiles = Split(Attachments, ";")
    For Each sFile In sFiles
        If Not String.IsNullOrEmpty(sFile) Then
            oMessage.Attachments.Add(New Net.Mail.Attachment(sFile))
        End If
    Next
End If
mySmtpClient = New System.Net.Mail.SmtpClient(SMTPServer, 25)
If UserName = "" Then
    mySmtpClient.Credentials = System.Net.CredentialCache.DefaultNetworkCredentials
Else
    mySmtpClient.Credentials = New System.Net.NetworkCredential(UserName, Password, Domain)
End If
mySmtpClient.Send(oMessage)
End Sub

```

## Can I migrate OLAP Task from DTS to SSIS?

No, you can't migrate the OLAP task from DTS to SSIS. In DTS 2000 (SQL Server Enterprise Edition) Microsoft provided the OLAP Task to process Cubes stored on Analysis Services 2000. There is no support for Analysis Services 2000 Cube Processing in SSIS, however Microsoft does provide support for Analysis Services 2005 Cube Processing using Analysis Services Processing Task.

To migrate your AS2000 Cube Processing task to SSIS, you can use the embedded DTS package option that can be executed using Execute DTS 2000 Package Task. DTS xChange migrates all your OLAP Task to Execute DTS 2000 Package Task (Embedded DTS Package).

## How to make my SSIS connections dynamic so I can easily switch to different environment (e.g. Dev, QA, Prod ...)

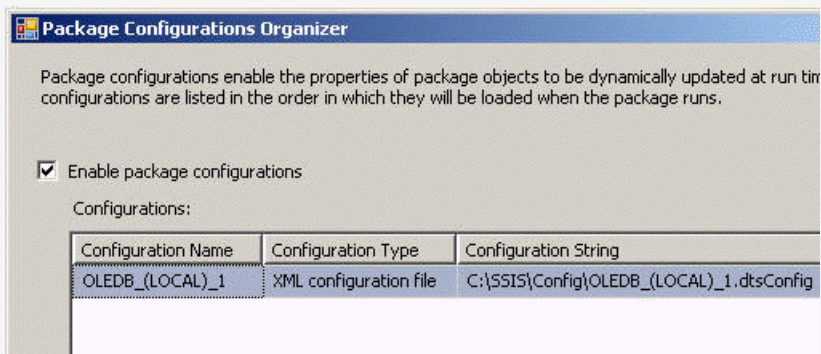
Using UDL files or the Dynamic Properties task in DTS could make your connection attributes dynamic (Server, UserName, Password etc) but these methods may be ineffective and have their own drawbacks.

In SSIS Microsoft added SSIS Configurations. You can read configuration settings from any of the following

sources:

- Read from XML File
- Read from SQL Server Table
- Read from Registry Key
- Read from Environment Variable
- Read from Parent

**Note:** The most common way of reading configurations is XML Config File.



DTS xChange automatically creates the configuration files for the selected connection types during migration. DTS xChange detects all of your common connections based on connection attributes (server, username, password, etc) and generates one config file that can be referenced in multiple packages. You can modify the config files for a new connection, to point all of the packages using that config file to the new server or database.

## How to read/write variable value from child/parent package in SSIS

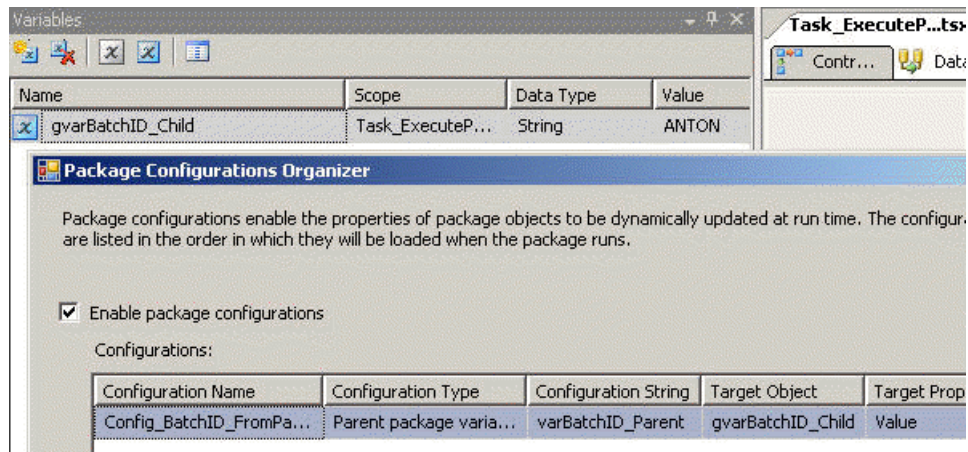
Changing the value of the child package variable was possible in DTS using the inner/outer variables Execute DTS Package Task settings. This functionality is not directly supported in SSIS if you using the Execute SSIS Package Task.

You can use VB.net code inside the script task to read/write the child package variable. You can also read the parent package variable in SSIS using Configurations. The following two sections illustrate both techniques.

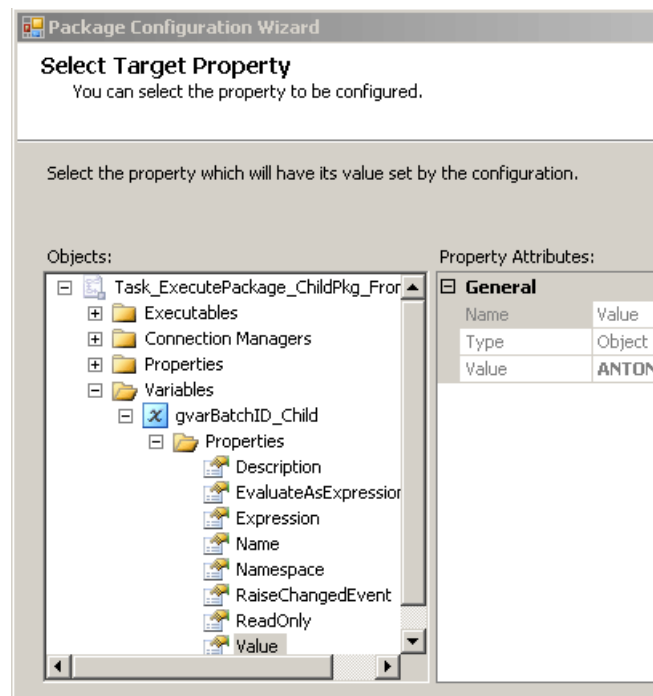
### Reading the Parent Package variable from the Child Package using Parent Variable Configuration

The Parent Variable Configuration feature in SSIS allows variables from a parent package to be inherited by a child package. This is different than the DTS where variables were pushed down from the parent package to the child package. In SSIS the child package requests the variable by name from the parent package, allowing the variable to be inherited from any calling parent package that uses the Execute Package task to call the child package.

In the following screen value of parent variable varBatchID\_Parent is being assigned to child package variable gvarBatchID\_Child:



Select any available property as the configuration target property including variables, connections, task or package property. In the following example, the target is the value property of the variable called gvarBatchID\_Child:



### Read/Write child package variable from the Parent Package using the Script Task

Many times you have to pass variables or Static Values at run time to the child package. The following code example displays how you can load any SSIS package at run time, set package variables, and execute the package:



```

Option Strict Off
Imports System
Imports System.Data
Imports System.Math
Imports Microsoft.SqlServer.Dts.Runtime
Public Class ScriptMain
    Public Sub Main()
        Dim oApp As Microsoft.SqlServer.Dts.Runtime.Application = New _
            Microsoft.SqlServer.Dts.Runtime.Application()
        Dim oPkg As Microsoft.SqlServer.Dts.Runtime.Package //Load Child Package from SQL Server
        oPkg = oApp.LoadFromSqlServer("\DW_Packages\Task_ExecSQL", "(local)", "", "", Nothing) Dim vars As Variable
s //If Child Package Variable Doesn't exist at runtime
//then create a new variable and pass the value from Parent package variable.
If oPkg.Variables.Contains("gvChild1") Then
    oPkg.VariableDispenser.LockOneForWrite("gvChild1", vars)
    Try
        vars("gvChild1").Value = "Hello!!! Set From Parent..."
    Catch ex As Exception
        Throw ex
    Finally
        vars.Unlock()
    End Try
Else
    oPkg.Variables.Add("gvChild1", False, "", "Hello!!! Set From Parent...")
End If oPkg.Execute() //Execute the Child Package Dts.TaskResult = Dts.Results.Success
End Sub
End Class

```

**Note:** Due to the complexity of logic you might want to use scripting instead of the Configuration technique explained in the previous section.

## What are the advantages and disadvantages of storing SSIS packages to msdb vs File System?

There are two places you can store your packages:

- on the file system
- in the msdb database

Each storage option has its own pros and cons, and your selection should depend on the settings that are more applicable to your environment's needs. The following table gives you a high-level view of each storage option, and their functionality:

Functionality	Best in File System	Best in MSDB
Security		X
Backup and Recovery	X	
Deployment	X	

TroubleShooting Functionality	X Best in File System	Best in MSDB
Execution Speed	X	X
Availability	X	

**Note:** The options that are marked with an X in the table above reflect what is the most optimized for the functionality.

## Security

If security concerns you greatly, you may want to consider placing your packages in the msdb database. To secure your packages on the file system, you could have multiple layers of security by using the Windows Active Directory security on the folder in the file system where the packages are located. You could also place a password on the packages to keep users that may have administrator rights to your machine from executing the package. This does add extra complexity to your package deployments in some cases. If you store your packages in the msdb database, you can assign package roles to each package to designate who can see or execute the package. The packages can also be encrypted in the msdb database, which strengthens your security even more.

## Backup and Recovery

Backup and recovery is simpler with storing your packages in the msdb database. If you were to store your packages in the msdb database, then you must only wrap the msdb database into your regular maintenance plan to backup all the packages. As packages are added, they are wrapped into the maintenance plan. The problem with this is that you can restore a single package using this mechanism. You'd have to restore all the packages to a point in time, and that would also restore the jobs and history. The other option is a file system backup, which would just use your favorite backup software to backup the folders and files. If you did this, you must rely on your Backup Operator to do this for you, which makes some uneasy. You could though at that point, restore individual packages to a point in time. In reality, you may just go ahead and redeploy the packages from Source Safe if you couldn't retrieve a backup file.

## File System Deployments

File system deployments are much simpler in SSIS. To deploy packages onto the file system, you must only copy them into the directory for the package store. You can create subdirectories under the parent directory to subdivide it easily. You can also copy a single package in case you need to make a package change. To import a package into the package store using the msdb database, you must use Management Studio (or a command-line tool called dtutil.exe) and import them package by package. To do a bulk migration, you could use the deployment utility.

## Troubleshooting

If you run into any troubleshooting issues, and you want to see if the packages in production were the same release as the packages you thought you had deployed, you only need to copy the files down to your machine and perform a comparison using Source Safe or another similar tool. If the files were stored in the msdb database, you would have to right-click on each package in Management Studio and select Export. If the same packages were stored in the file system, you must only copy the files to your machine.

## Package Availability

Availability of your packages is always on the top of the list for DBAs. If you were to store the packages in the msdb database and the database engine were to go down, the packages are unavailable. If they were stored in the file system, then your packages would be available for execution. Of course, if the database engine is down, then probably one of your key data sources would also be down at the time.

## Why should I migrate DTS packages to SSIS (advantages of SSIS over DTS)?

Consider the following reasons why you should migrate your DTS packages to SSIS:

- Microsoft has discontinued support for SQL Server 2000 and DTS.
- SSIS has new many built in tasks that can save you significant time compared to manual coding in DTS (Looping through files in a folder is inbuilt in SSIS).
- Making packages dynamic at runtime is easier using expression and configurations.
- SSIS DataFlow engine is significantly faster than DTS DataPump.
- In memory transformations in DataFlow (JOIN, Split, Sort, Aggregate, Union).
- DataViewer support to debug DataFlow.
- Event Handlers support with separate control flow for each event.
- ScriptTask in SSIS uses VB.net language with rich .net framework support. Script Task code is compiled so run much faster than DTS ActiveX script.
- Expression based control flow along with Success, Failure and Complete.
- Reusable connection architecture.

[Additional Information:](#) For more information about the benefits of SSIS, download the [What's New in SSIS MSDN white paper](#).

## Which data providers are supported in SSIS?

The following is a list of known data providers:

**Note:** This is not a definitive list of all data providers. Not all of the providers listed have been tested by Microsoft with SSIS. Information about third-party products was provided by the product vendors and could not be independently verified.

### SQL Server

Data Source	Released by	Data access API	Supported Database Versions	Vendor Support with SSIS?	64-bit Support?
-------------	-------------	-----------------	-----------------------------	---------------------------	-----------------

Data Source	Released by	Data access API	Supported Database Versions	Vendor Support with SSIS?	64-bit x64, IA64 Support?
SQL Server	Microsoft	ADO.NET	2000 and later versions	Yes	x64, IA64
SQL Server	Microsoft	ODBC	2000 and later versions	Yes	x64, IA64
SQL Server	Microsoft	OLE DB	2000 and later versions	Yes	N/A
SQL Server	Microsoft	OLE DB	2000 and later versions	Yes	N/A

## DB2

Data Source	Released by	Data access API	Supported Database Versions	Vendor Support with SSIS?	64-bit Support?
DB2	Microsoft	OLE DB	All DRDA-compliant versions	Yes	x64, IA64
DB2	IBM	OLE DB	z/OS and UDB 7.0 and later, AIX 5.0 and later, AS400 4.5 and later versions	Yes	N/A
DB2	IBM	ADO.NET	DB2 UDB 9.0	No	N/A
DB2	IBM	ODBC	z/OS and UDB 7.0 and later versions AIX 5.0 and later versions, AS400 4.5 and later versions	No	No
DB2	Attunity	OLE DB	6.1, 7.x, 8.0 on z/OS 7.x, 8.0 on UNIX 7.x, 8.0 on Windows	Yes	N/A

<b>Data Source</b> DB2/400	<b>Released by</b> Attunity	<b>Data access</b> OLE DB	<b>Supported Database Versions</b> On AS/400 5.1 and later versions	<b>Vendor Support with SSIS?</b> Yes	<b>64-bit Support?</b> N/A
DB2	DataDirect	OLE DB	z/OS and UDB 7.0 and later versions AIX 5.x AS400 4.5 and later versions	No	N/A
DB2	HIT	OLE DB0	z/OS and UDB 8.0 and later versions AIX 5.x, AS400 4.5 and later versions	No	N/A
DB2	DataDirect	ADO.NET	z/OS and UDB 7.0 and later AIX 5.x, AS400 4.5 and later versions	Yes	N/A
DB2	ETI	Bulk Load	8.0 and later versions	Yes	N/A
DB2 <sup>1</sup>	Persistent	Data Flow Component Bulk Write, Bulk Read	N/A	No	N/A

## Oracle

<b>Data Source</b>	<b>Released by</b>	<b>Data access API</b>	<b>Supported database versions</b>	<b>Vendor Support with SSIS?</b>	<b>64-bit Support?</b>
Oracle	Microsoft	OLE DB	7.3.4 and later versions <sup>2</sup>	Yes	No
Oracle	Microsoft	ADO.NET	8.0 and later versions	Yes	x64, IA64
Oracle	Oracle Corp	OLE DB	8i and later versions	Yes	x64, IA64
Oracle	Oracle Corp	ADO.NET	8i and later versions	Yes	x64, IA64
			8i and later		

Oracle	Oracle Corp	ODBC	versions	No	N/A
<b>Data Source</b>	<b>Released by</b>	<b>Data access API</b>	<b>Supported database versions</b>	<b>Vendor Support with SSIS?</b>	<b>64-bit Support?</b>
Oracle	Microsoft	ODBC	8i and later versions		N/A
Oracle	Attunity	OLE DB	9i and later versions	Yes	N/A
Oracle	DataDirect	OLE DB	8i and later versions	Yes	N/A
Oracle	DataDirect	ADO.NET	8i and later versions	Yes	N/A
Oracle	ETI	Bulk Load	9.0 and later versions	Yes	N/A
Oracle	Persistent	Data Flow Component Bulk Write	8i and later versions	No	N/A

### Sybase

<b>Data Source</b>	<b>Released by</b>	<b>Data access API</b>	<b>Supported database versions</b>	<b>Vendor Support with SSIS?</b>	<b>64-bit Support?</b>
Sybase	Sybase	OLE DB	11.5 and later versions	No	N/A
Sybase	Sybase	ADO.NET	11.5 and later versions	No	N/A
Sybase	Attunity	OLE DB	12.0 and later versions	No	N/A
Sybase	DataDirect	OLE DB	11.5 and later versions	Yes	N/A
Sybase	DataDirect	ADO.NET	11.5 and later versions	Yes	N/A

### Informix

<b>Data Source</b>	<b>Released by</b>	<b>Data access API</b>	<b>Supported Database Version</b>	<b>Vendor Support with SSIS?</b>	<b>64-bit Support?</b>
			7.3 and later		

Informix	IBM	OLE DB	versions <b>Supported</b>	No <b>Vendor</b>	N/A
<b>Data Source</b> Informix	<b>Released by</b> Attunity	<b>Data access</b> OLE DB	<b>Database</b> Version 7, 9, 10	<b>Support</b> with SSIS? Yes	<b>64-bit</b> <b>Support?</b> N/A
Informix <sup>1</sup>	Persistent	Data Flow Component Bulk Write & Bulk Read	N/A	No	N/A


## Teradata

<b>Data Source</b>	<b>Released by</b>	<b>Data access</b> <b>API</b>	<b>Supported</b> <b>Database</b> <b>Version</b>	<b>Vendor</b> <b>Support</b> <b>with SSIS?</b>	<b>64-bit</b> <b>Support?</b>
Teradata	Teradata	OLE DB	2.6 and later versions	No	N/A
Teradata	Teradata	ADO.NET	2.6 and later versions	No	N/A
Teradata	ETI	Bulk Load	2.5 and later versions	Yes	N/A
Teradata	ETI	Bulk Extract	2.5 and later versions	Yes	N/A

## Other

<b>Data Source</b>	<b>Released by</b>	<b>Data access</b> <b>API</b>	<b>Supported</b> <b>Database</b> <b>Version</b>	<b>Vendor</b> <b>Support</b> <b>with SSIS?</b>	<b>64-bit</b> <b>Support?</b>
SAP <sup>1</sup>	Microsoft	ADO.NET	R/3 4.6C and later versions	Yes	N/A
SAP	Theobald	OLE DB	R/3	Yes	N/A
Office Access	Microsoft	OLE DB	2003 and earlier versions	Yes	N/A
Office Excel	Microsoft	OLE DB	2003 and earlier versions	Yes	N/A
Office 2007	Microsoft	OLE DB	2007	No	N/A
FoxPro	Microsoft	OLE DB	8.0 and later versions	No	N/A
			Any Jet 4.0–		

File DBs Data Source	Microsoft Released by	OLE DB Data access API	compatible Supported Database Version	No Vendor Support with SSIS?	N/A 64-bit Support?
Adabase	Attunity	OLE DB	Version 6.2.2 to 7.4.x on z/OS 3.3 to 5.1 on Open Systems (UNIX, Windows, OpenVMS)	Yes	N/A
CISAM	Attunity	OLE DB	On UNIX	Yes	N/A
DISAM	Attunity	OLE DB	On UNIX, Linux, and Windows	Yes	N/A
Ingres II	Attunity	OLE DB	2 to 2.56	Yes	N/A
Oracle Rdb	Attunity	OLE DB	7.1.x, on OpenVMS Alpha and Integrity (Itanium)	Yes	Yes (HP Integrity)
RMS	Attunity	OLE DB	On OpenVMS Alpha and Integrity (Itanium)	Yes	Yes (HP Integrity)
Enscribe	Attunity	OLE DB	On HP NonStop G-Series and H-Series	Yes	Yes (HP Integrity)
SQL/MP	Attunity	OLE DB	On HP NonStop G-Series and H-Series	Yes	Yes (HP Integrity)
IMS/DB	Attunity	OLE DB	6.1 and later versions	Yes	N/A
VSAM	Attunity	OLE DB	On z/OS 1.1 and later versions	Yes	N/A
LDAP	Persistent	ODBC	All LDAP-compliant servers	No	Yes

 **Additional Information:** For more information about the above lists, see the [Connectivity and SQL](#)



[Server 2005 Integration Services MSDN article.](#)

## How to enable/disable task at runtime In SSIS

In DTS you could disable a Task using the ActiveX script and setting DisableStep property to True or False. In SSIS, you can enable/disable certain task in SSIS using one of the following methods:

- Use expression on Disable Property of Task.
- Use expression on precedence constraint.

**Note:** It is more effective to use the expression on precedence constraint method.

## How to run SSIS packages using 32-bit drivers on a 64-bit machine

See the blog post [How to run SSIS packages using 32-bit drivers on a 64-bit machine.](#)

## How to connect Integration Services running on a remote machine

You may need to perform additional security configurations if you are connecting to an SSIS Server running on a remote machine.

**Additional Information:** For more information about the additional security configuration to connect to a remote machine, see the [Integration Services Service \(SSIS Service\) MSDN article.](#)

## Do I need multiple connections in SSIS for parallel processing?

In DTS, it was recommended that your packages contain multiple connection objects that point to the same database if you have multiple tasks using that database. This recommendation was considered more efficient for each task to have a dedicated connection to the server.

In SSIS, DTS Connection objects have been replaced with Connection Managers. An SSIS Connection Manager maintains a pool of connections to the database rather than just a single transaction meaning that you only need one Connection Manager in SSIS instead of multiple connection objects. If you need to replicate the same behavior as DTS within a SSIS package then look into using the RetainSameConnection property.

## How to replace DTS Looping code to native SSIS

DTS had no native support for looping. Many DTS users used looping by enabling and disabling certain steps within the ActiveX script. In SSIS, you can use any of the following new tasks for looping:

SSIS Task	Application
-----------	-------------

<b>SSIS Task</b>  For Each Loop Container	<b>Application</b> Use For Loop Each Container if you are looping through any of the following collection: <ul style="list-style-type: none"> <li>• Files in the folder</li> <li>• Registry keys</li> <li>• XML Nodes</li> <li>• ADO Recordset</li> </ul>
For Loop Container	Use For Loop Container if you are looping based on numeric counter value. You can specify start, increment and end conditions.

**Note:** For more information about looping in SSIS, see the How to loop through files in a specified folder, load one by one and move to archive folder using SSIS section.

## How to modify task and connection properties in SSIS at runtime

In DTS you can change task, connections, and variable properties using the ActiveX Script or the Dynamic Properties Task. In SSIS you can only change variables and connection properties using script. To change Task properties at run time in SSIS you must use expressions.

The following sample code shows what DTS approach to modify variable, connection and task property:

```

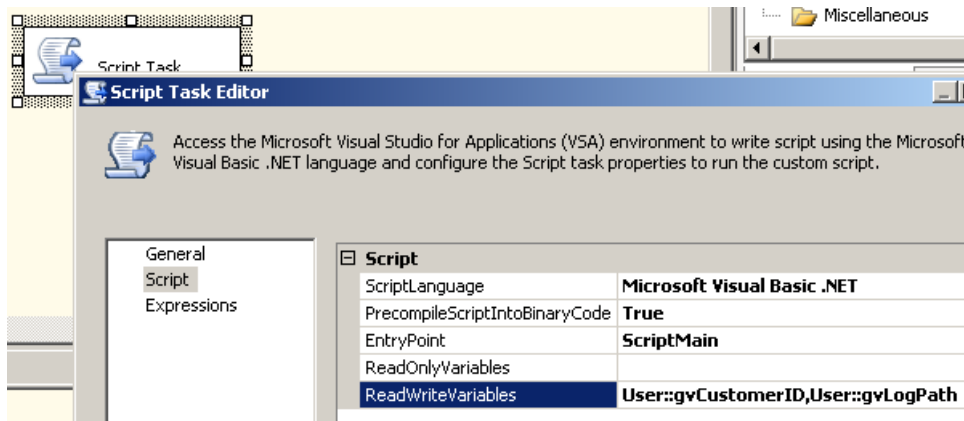
Sub Main()
  Dim oPkg, oVar, oCon, oStp, oTsk, oCust
  Dim sSql      '//Get reference to current package
  oPkg = DTSGlobalVariables.Parent      '//Change connection property
  oCon = oPkg.Connections("NorthwindConnection").DataSource = "DEVSRV001"      '//Change variable value
  oVar = DTSGlobalVariables("gvCustomerID")
  DTSGlobalVariables("gvCustomerID").Value = "ALFKI"
  '//Change task property (i.e. SourceSQLStatement propert of ExecuteSQLTask)
  oStp = oPkg.Steps("DTSSStep_DTSDDataPumpTask_5")
  oTsk = oPkg.Tasks(stp.TaskName)
  oCust = oTsk.CustomTask
  sSql = "select * from Customer where CustomerID = " & DTSGlobalVariables("gvCustomerID").Value
  'Assign the SourceSQLStatement property of the custom task
  oCust.SourceSQLStatement = sSql
  Main = DTSTaskExecResult_Success
End Sub

```

Replace the code in SSIS by completing the following steps:

### Step 1: Specify the Variable name in the ReadOnlyVariables or ReadWriteVariable Property

List your variables under **ReadOnlyVariables** or **ReadWriteVariables** property to accessing the Variable in the script.

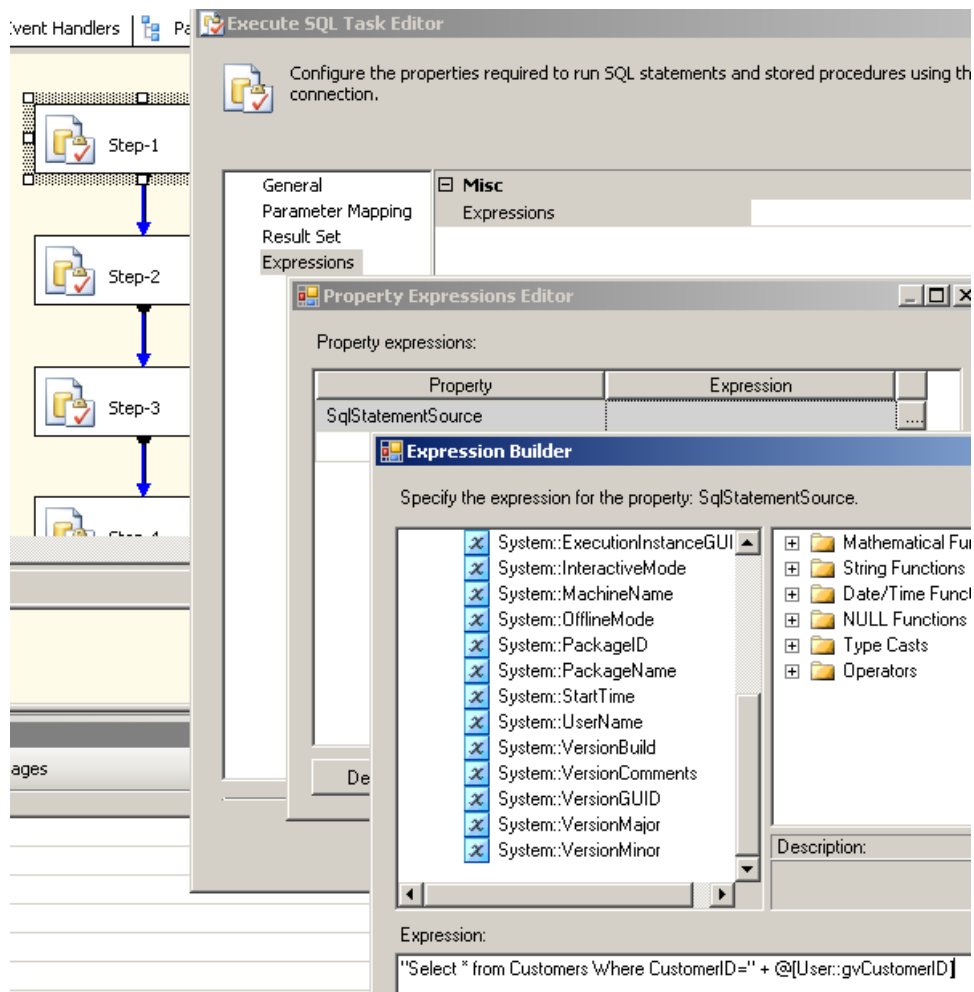


## Step 2: Specify expressions to change the SQLStatement property execute SQL task at runtime

SSIS doesn't allow you to modify task properties from code like you could in DTS. To modify the Task Property at run time, you need to use Configurations or Expressions. Complete the following steps to create a new expression on SqlStatementSource property of Execute SQL task:

1. Right click on the task for where you want to add expression and select **Edit**.
2. Select **Expressions** on the left side, highlight the row under **Misc** to display the edit expression button [..]
3. Select the **edit** button, select **SqlStatementSource Property**, and then enter the following expression in the expression editor:

```
"Select * from Customers Where CustomerID=" + @[User::gvCustomerID]
```



### Step 3: Add Code in the script task to change the variable value and connection property at runtime

1. Enter the following script:

```
Imports System
Imports System.Data
Imports System.Math
Imports Microsoft.SqlServer.Dts.RuntimePublic Class ScriptMain
    Public Sub Main()
        Dim oCon As ConnectionManager
        '//Change connection property
        oCon = Dts.Connections("NorthwindConnection")
        oCon.Properties("ServerName").SetValue(oCon, "DEVSRV001")
        '//Change variable value
        '//NOTE : Any variable you access in the script must be specified in the ScriptTask ReadOnlyVariable or ReadWriteVariable list
        '//(use comma to separate multiple variables e.g. => User::gVar1,User::gVar2,System::PackageName)
        Dts.Variables("gvCustomerID").Value = "ALFKI"
        '//Change task property (i.e. SourceSQLStatement property of ExecuteSQLTask)
        '//Use expression on SqlStatement Property of ExecuteSQL task
        '// NO similar construct available as DTS
        Dts.TaskResult = Dts.Results.Success
    End Sub
End Class
```

## How to call SSIS package from code and pass variable value

The following example displays how to call a package stored on a file system and pass variable values to it.

**Note:** If you are using this code in windows or asp.net application then make sure you add the following assembly references to your project:

- Microsoft.SqlServer.ManagedDTS.dlls

```
Option Strict Off
Imports System
Imports System.Data
Imports System.Math
Imports Microsoft.SqlServer.Dts.Runtime
Public Class ScriptMain
    Public Sub Main()
        DemoExePackage()
        Dts.TaskResult = Dts.Results.Success
    End Sub
    Public Sub DemoExePackage()
        Dim oApp As Microsoft.SqlServer.Dts.Runtime.Application = New _
            Microsoft.SqlServer.Dts.Runtime.Application()
        Dim oPkg As Microsoft.SqlServer.Dts.Runtime.Package //Load from File system
        oPkg = oApp.LoadPackage("C:\SSIS2005\Task_BulkInsert.dtsx", Nothing) '-- or -- //Load from SQL Server
        'oPkg = oApp.LoadFromSqlServer("\DW_Packages\Task_ExecSQL", "(local)", "sa", "somepassword", Nothing) '--
or -- //Load from SSIS Store //No need for uid/pwd because its always win auth,
        '//NOTE: SSIS service must be running and configured where packages stored. Fore remote connection see MSDN h
elp
        'oPkg = oApp.LoadFromDtsServer("\MSDB\DW_Packages\Task_ExecSQL", "MYDEV001", Nothing) //Only use machin
e name .. no (local) or "." //Pass variable value
        Dim vars As Variables
        If oPkg.Variables.Contains("gvCustomerID") Then
            oPkg.VariableDispenser.LockOneForWrite("gvCustomerID", vars)
            Try
                vars("gvCustomerID").Value = "ALFKI"
            Catch ex As Exception
                Throw New Exception("Variable gvCustomerID not found in child package")
            Finally
                vars.Unlock()
            End Try
        End If
        oPkg.Execute() //Execute the Child Package

    End Sub
End Class
```

## DTS designer not working on SQL 2008 x64

If you trying to edit DTS Packages in legacy mode in SSMS and getting errors, complete the following steps:

1. Download and Install DTS Designer components from [Microsoft](#).
2. Create a bat file on your desktop. Name it **Fix\_DTS\_Designer.bat**.
3. Right click on file and select Edit. Paste the following script and save the file:

```

:: THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
:: EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
:: OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
:: NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
:: HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
:: WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
:: FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
:: OTHER DEALINGS IN THE SOFTWARE.
::=====
:: Refer: http://technet.microsoft.com/en-us/library/ms143755.aspx
:: Detect correct 32 bit program files ....on 64-bit its Program Files (x86)
SET PROGRAM_32BIT=%ProgramFiles%
if not "%ProgramFiles(x86)%" == "" set PROGRAM_32BIT=%ProgramFiles(x86)%
echo PROGRAM_32BIT
:: =====For SSMS =====
COPY "%PROGRAM_32BIT%\Microsoft SQL Server\80\Tools\Binn\SEMSFC.DLL" "%PROGRAM_32BIT%\Microsoft SQL Server\
100\Tools\Binn\VSShell\Common7\IDE\SEMSFC.DLL"
COPY "%PROGRAM_32BIT%\Microsoft SQL Server\80\Tools\Binn\SQLGUI.DLL" "%PROGRAM_32BIT%\Microsoft SQL Server\
100\Tools\Binn\VSShell\Common7\IDE\SQLGUI.DLL"
COPY "%PROGRAM_32BIT%\Microsoft SQL Server\80\Tools\Binn\SQLSVC.DLL" "%PROGRAM_32BIT%\Microsoft SQL Server\
100\Tools\Binn\VSShell\Common7\IDE\SQLSVC.DLL"
COPY "%PROGRAM_32BIT%\Microsoft SQL Server\80\Tools\Binn\Resources\1033\SEMSFC.RLL" "%PROGRAM_32BIT%\Micro
soft SQL Server\100\Tools\Binn\VSShell\Common7\IDE\Resources\1033\SEMSFC.RLL"
COPY "%PROGRAM_32BIT%\Microsoft SQL Server\80\Tools\Binn\Resources\1033\SQLGUI.RLL" "%PROGRAM_32BIT%\Micro
soft SQL Server\100\Tools\Binn\VSShell\Common7\IDE\Resources\1033\SQLGUI.RLL"
COPY "%PROGRAM_32BIT%\Microsoft SQL Server\80\Tools\Binn\Resources\1033\SQLSVC.RLL" "%PROGRAM_32BIT%\Micro
soft SQL Server\100\Tools\Binn\VSShell\Common7\IDE\Resources\1033\SQLSVC.RLL"
:: =====For BIDS =====
COPY "%PROGRAM_32BIT%\Microsoft SQL Server\80\Tools\Binn\SEMSFC.DLL" "%PROGRAM_32BIT%\Microsoft Visual Studi
o 9.0\Common7\IDE\SEMSFC.DLL"
COPY "%PROGRAM_32BIT%\Microsoft SQL Server\80\Tools\Binn\SQLGUI.DLL" "%PROGRAM_32BIT%\Microsoft Visual Studi
o 9.0\Common7\IDE\SQLGUI.DLL"
COPY "%PROGRAM_32BIT%\Microsoft SQL Server\80\Tools\Binn\SQLSVC.DLL" "%PROGRAM_32BIT%\Microsoft Visual Studi
o 9.0\Common7\IDE\SQLSVC.DLL"
MKDIR "%PROGRAM_32BIT%\Microsoft Visual Studio 9.0\Common7\IDE\Resources"
MKDIR "%PROGRAM_32BIT%\Microsoft Visual Studio 9.0\Common7\IDE\Resources\1033"
COPY "%PROGRAM_32BIT%\Microsoft SQL Server\80\Tools\Binn\Resources\1033\SEMSFC.RLL" "%PROGRAM_32BIT%\Micro
soft Visual Studio 9.0\Common7\IDE\Resources\1033\SEMSFC.RLL"
COPY "%PROGRAM_32BIT%\Microsoft SQL Server\80\Tools\Binn\Resources\1033\SQLGUI.RLL" "%PROGRAM_32BIT%\Micro
soft Visual Studio 9.0\Common7\IDE\Resources\1033\SQLGUI.RLL"
COPY "%PROGRAM_32BIT%\Microsoft SQL Server\80\Tools\Binn\Resources\1033\SQLSVC.RLL" "%PROGRAM_32BIT%\Micro
soft Visual Studio 9.0\Common7\IDE\Resources\1033\SQLSVC.RLL"

```

Double click on the file to run the script. You should see the files copied to the correct directories after that. Now you can try to open DTS package in designer (SSMS Legacy Mode).

## Errors

### Cannot Acquire Connection From Connection Manager

E0001 is a high level error that is commonly seen after migration. The following is a sample of the error you may encounter in the validation log:

**Description: SSIS Error Code DTS\_E\_CANNOTACQUIRECONNECTIONFROMCONNECTIONMANAGER. The AcquireConnection method call to the connection manager "OLEDB\_NORTHWIND\_1" fail.**

This error can occur for several reasons, such as:

- One or more attribute of your connection string is invalid.
- Server name/file path specified in the connection string is invalid/offline.
- Server name/file path specified in the connection string is inaccessible due to network or firewall issue.

Resolve this error by completing the following actions:

- Open the SSIS package in Visual Studio. Right click on the connection throwing the error. Select **Edit > Test Connection**. If all connection attributes are valid then test connection should succeed.
- If you are using windows authentication then make sure you are added in to SQL Logins to access using windows authentication.
- Make sure remote machine allows SQL Connection on default port 1433.
- Make sure SQL Server is running on remote machine.
- Repeat the above steps for all columns with similar error. Check Error List for any possible Warnings/Errors in the visual studio.

## Invalid class string

E0002 is a high level error that occurs when one or more DTS components are not registered on the machine where you are running DTS xChange. This error prevents the package from loading, and you will not be able to profile or migrate the package until you install the required DLLs.

Resolve this error by completing the following actions:

- Open the DTS package causing the issue and try to find any Custom Task. OLAP Task is the most common Custom task that is not installed by default and may cause you to receive this error.
- Make sure you have DTS Runtime installed properly. If you dont have DTS Runtime then please use the following Links to download appropriate version of DTS Runtime.

**Note:** For more information about DTS xChange installation requirements, see the [Getting started with DTS xChange](#) article.

**Important:** If the above test failed and you are getting a components missing error, run the SQL Server 2005/2008 setup and install Backward Compatibility components or download from the following URL (For 32 bit OS use x86 link).

Microsoft SQL Server 2005 Backward Compatibility Components The SQL Server Backward Compatibility package includes the latest versions of Data Transformation Services 2000 runtime (DTS), SQL Distributed Management Objects (SQL-DMO), Decision Support Objects (DSO), and SQL Virtual Device Interface (SQLVDI). These versions have been updated for compatibility with both SQL Server 2005 and SQL Server 2008 and include all fixes shipped through SQL Server 2000 SP4 and SQL Server 2005 SP2.

[X86 Package](#) (SQLServer2005\_BC.msi) - 11273 KB [X64 Package](#) (SQLServer2005\_BC.msi) - 18569 KB [IA64](#)

## Warnings

### Truncation may occur

W0001 is a low level warning that occurs when you have different column sizes in the source and target. The following is an example warning you may see in the validation log:

**Truncation may occur due to inserting data from data flow column "SrcOrderDate" with a length of 35 to database column "DestOrderDate" with a length of 30.**

**Note:** You can ignore this warning for Flat File Source converted by DTS xChange. When column size metadata is not available DTS xChange automatically creates flat file column with 255 characters. By default dataflow fails when truncation occurs. You can change this behavior by changing Truncation Row Disposition option of the column (Right click on the source adapter and select Advanced Editor - > Input/Outputs Tab -> Expand Source Outputs -> Expand Output Columns -> Change column property "Truncation Row Disposition")

Resolve this warning by completing the following steps:

1. In the dataflow right click on the source adapter and select **Show Advanced Editor**.
2. Select **Input and Output Properties** tab.
3. Expand the <source> output.
4. Expand the **Output** columns node.
5. Change **Length** to match with Target column.
6. Select **OK** to save your changes.
7. Repeat these steps for all columns with similar warnings. Check Error List for any possible Warnings/Errors.

### Removing unused columns may increase performance

W0002 is a low level warning that occurs when you have columns selected in the source adapter that are not used anywhere in the transformation or mappings. Unwanted columns increase the SSIS buffer size and slow down DataFlow processing. The following is an example warning you may see in the validation log:

**The output column "CustomerID" (42) on output "Flat File Source Output" (2) and component "Flat File Source" (1) is not subsequently used in the Data Flow task. Removing this unused output column can increase Data Flow task performance.**

Resolve this warning by completing the following steps:

1. In the dataflow right click on the source adapter and select **Edit**.



2. Select the **Columns** tab. Unselect the unnecessary columns from the **Available External Columns list**.
3. Select **OK** to save your changes.
4. Repeat these steps for all the columns with similar warnings. Check the Error List for any possible Warnings/Errors/.