Building
Tomorrow's Enterprise

White Paper

Infosys®

DTS to SSIS 2008 Migration

Virendra Wadekar and Phaneendra Babu Subnivis

Abstract

SQL Server Integration Services (SSIS) is an Extract – Transform – Load (ETL) tool of enterprise class, which is available as part of SQL Server 2005/ 2008. The Data Transformation Service (DTS), which existed as part of SQL Server 2000, is re-written in SQL Server 2005 as SSIS.

To avail of the improved features of SSIS, along with the SQL 2008 database engine features, existing DTS packages will have to be migrated to SSIS. However, some challenges, during the migration of complex DTS packages (these include ActiveX Scripts, data driven query tasks, etc.) to SSIS, are inevitable.

This white paper explains the upgrade process:

- Usage of Upgrade Advisor
- Different methods of upgrading DTS packages to SSIS packages
- Possible exceptions and recommended work around for those exceptions

For more information, contact askus@infosys.com

Oct 2008

Upgrade Advisor

Upgrade Advisor is a tool that is used to check readiness of SQL Server 2000/ 2005 database to be upgraded to SQL Server 2008. It reports issues that need to be rectified to ensure a successful upgrade from SQL Server 2000/ 2005 to SQL Server 2008. The server components that are analyzed for issues include:

- SQL Server
- Analysis Services
- Notification Services
- Reporting Services
- Data Transformation Services
- Integration Services

Of the above listing, this paper focuses only on DTS. Looking at the challenges we have in DTS to SSIS migration, it is recommended to use Microsoft SQL 2008 upgrade advisor to check for upgrade issues not only on the databases, but also on the DTS packages.

The Upgrade Advisor will not make any changes to the packages, but suggests the features and configuration changes that might affect the upgrade process. It also provides links to documentation that describes the issue and its resolution process. It can be used to define the scope of DTS to SSIS migration.

The upgrade advisor needs the DTS client to be installed on the machine. It needs the DTS object model to analyze the DTS package and report the issues. It can be installed from the SQL 2008 installation media, using the SQLServer2005_BC*.msi package. It installs other components as well, along with the DTS client as shown in *Figure 1*.

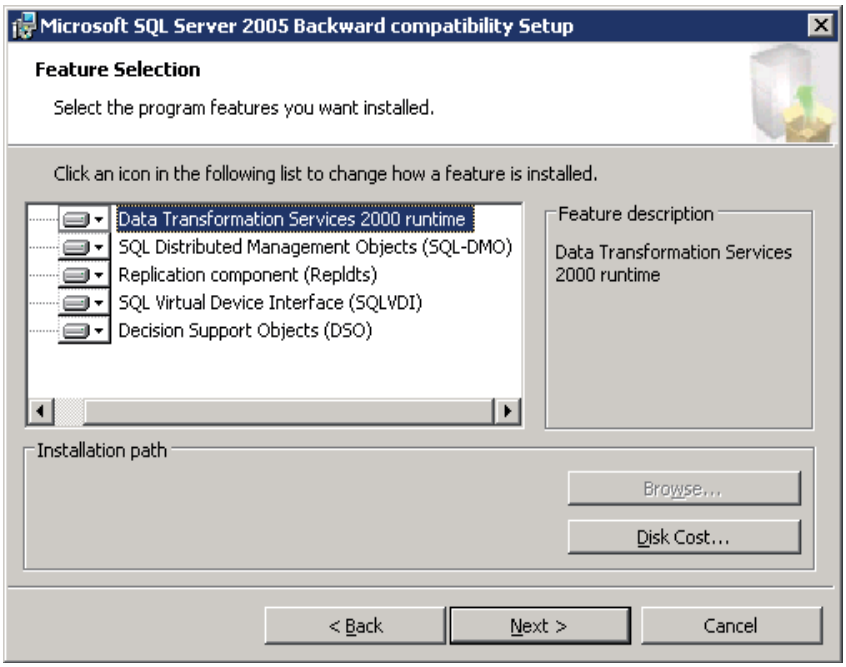

Figure 1: SQL Server 2005 Backward Compatibility Setup

The upgrade advisor takes location of the DTS packages as input from the user. After analysis, the upgrade advisor reports four types of issues:

- Pre-Upgrade issues - need to be fixed before the upgrade process
- Post-Upgrade issues - need to be fixed after the upgrade process

- Pre Or Post-Upgrade issues - can be fixed anytime
- Advisory issues - serve as information messages to the user. For example, it advices that the DTS packages are deprecated and should be moved to SSIS

Figure 2 shows the sample output when upgrade advisor is run on the DTS packages. The most common upgrade issues are listed with the upgrade advisor. It can be checked after installing the upgrade advisor.

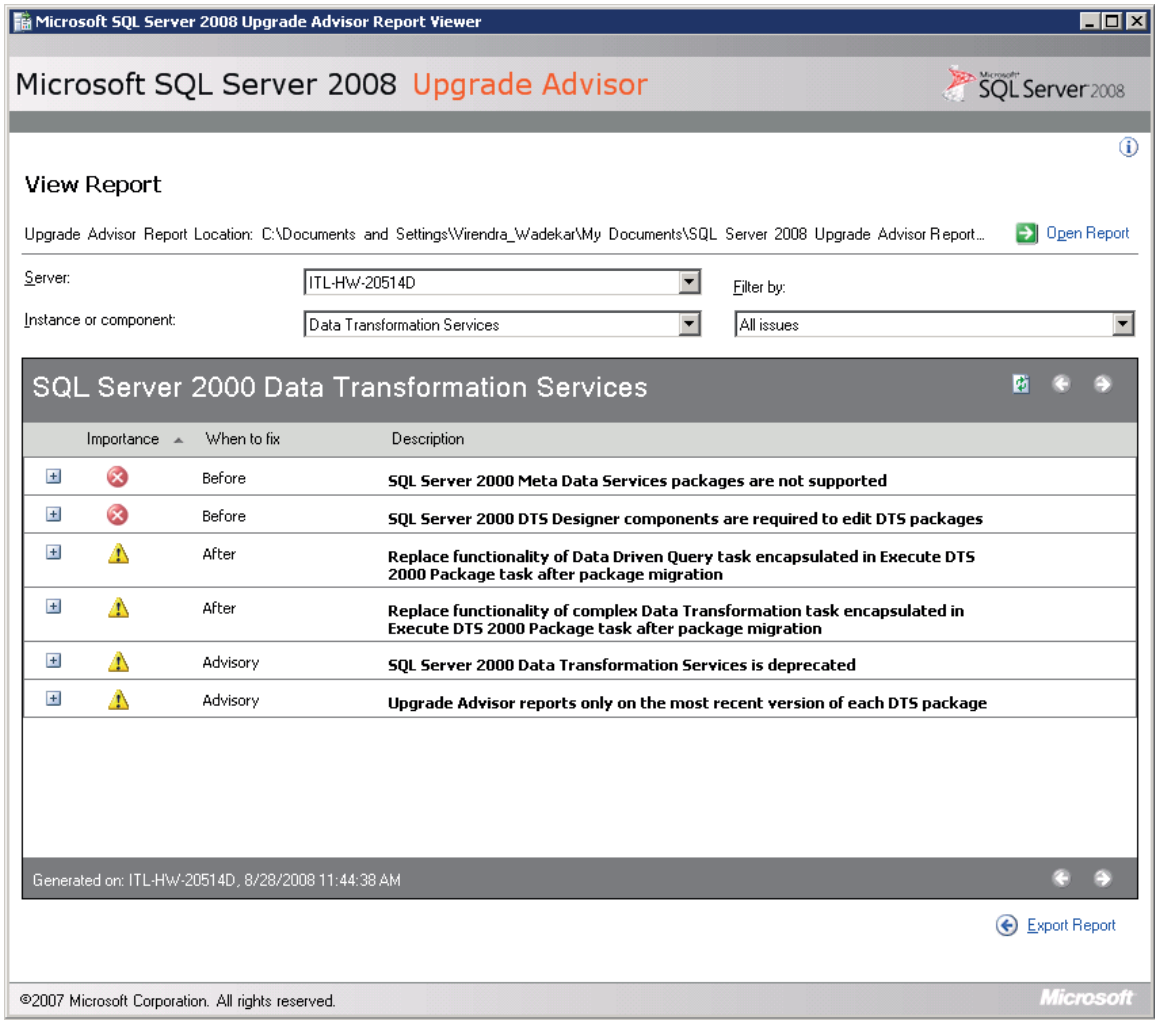

Figure 2: SQL Server 2008 Upgrade Advisor

Migration

There are three ways to handle DTS packages during your SQL 2000 to SQL 2008 upgradation.

1. Without Migration/ Upgrade
2. Migration Wizard
3. Complete Rewrite

This white paper outlines the process of the upgrade with the first two methods. It also highlights the issues faced during this upgrade and the tools that can be used for migration of DTS packages. The third approach is a complete rewrite of the DTS packages into SSIS. The document explains advantages of rewriting DTS packages to SSIS as well.

1. Without Migration/ Upgrade

In this approach, the DTS packages can be imported to SQL Server 2008 without disturbing the design and functionality. It continues to use the DTS object model. Such DTS packages run using the backward compatibility support - included in SQL Server 2008. Hence, these packages can neither use new features/ enhancements of SQL Server 2008, nor experience the performance benefits. This is not a recommended option, but can be used in the following scenarios:

- In the earlier stages of the Database upgrade to do the end-to-end testing, without migrating or rewriting the DTS packages
- To do away with the database upgrade in phase 1 first, and then DTS re-write in the following phases
- To maintain some DTS packages due to complex processing logic

In this scenario, DTS packages can be imported directly into SQL Server 2008, without any changes to logic.

With the backward compatibility components, all the DTS packages can be imported under the legacy node inside the SQL Server Management Studio (SSMS), as shown in Figure 3. The packages cannot be opened and seen after importing to SQL 2008, unless you install the "DTS Designer Components". The link for installing DTS Designer Components is included in the references.

These packages can now be scheduled using the SQL agent job, and will continue to run as is without any additional changes. Refer to the link given in the references section to know more about SSIS backward compatibility.

The DTS packages can also be maintained using the DTS designer components (explained in the later section of this white paper in detail). Refer to the link given in the references section to know more about "How to: Ensure Support for Data Transformation Services Packages".

This approach requires minimum efforts of all three approaches explained in the document.

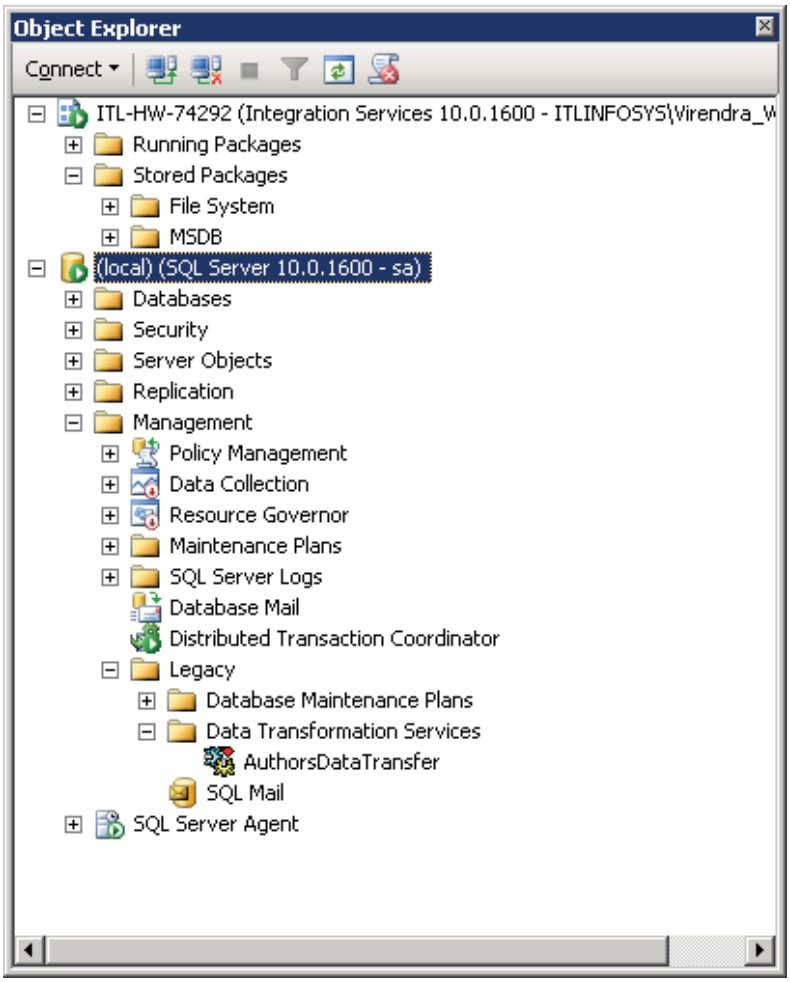

Figure 3: DTS Packages in SQL Server 2008

2. Migration Wizard

Migration Wizard helps upgrading DTS packages to SSIS packages. It converts DTS packages to SSIS packages to a large extent. Subsequently, the developer needs to rewrite those portions of packages that did not get upgraded properly, by using SSIS tasks and transformations. During this process developer/ designer involvement is required, where the migration process do not identify mapping objects of DTS package to SSIS package and retains it as DTS package tasks itself. Performance improvements are possible with this approach. A drawback of this approach is that higher effort may be needed than "Without Migration/Upgrade" approach.

The migration Wizard can be invoked from two locations:

a. In SQL Management Studio, the migration wizard can be found by right-clicking on Data Transformation Services node, as shown in Figure 4

b. In Business Intelligence Development Studio (BIDS), the migration wizard can be found by right-clicking on the Packages node, as shown in Figure 5

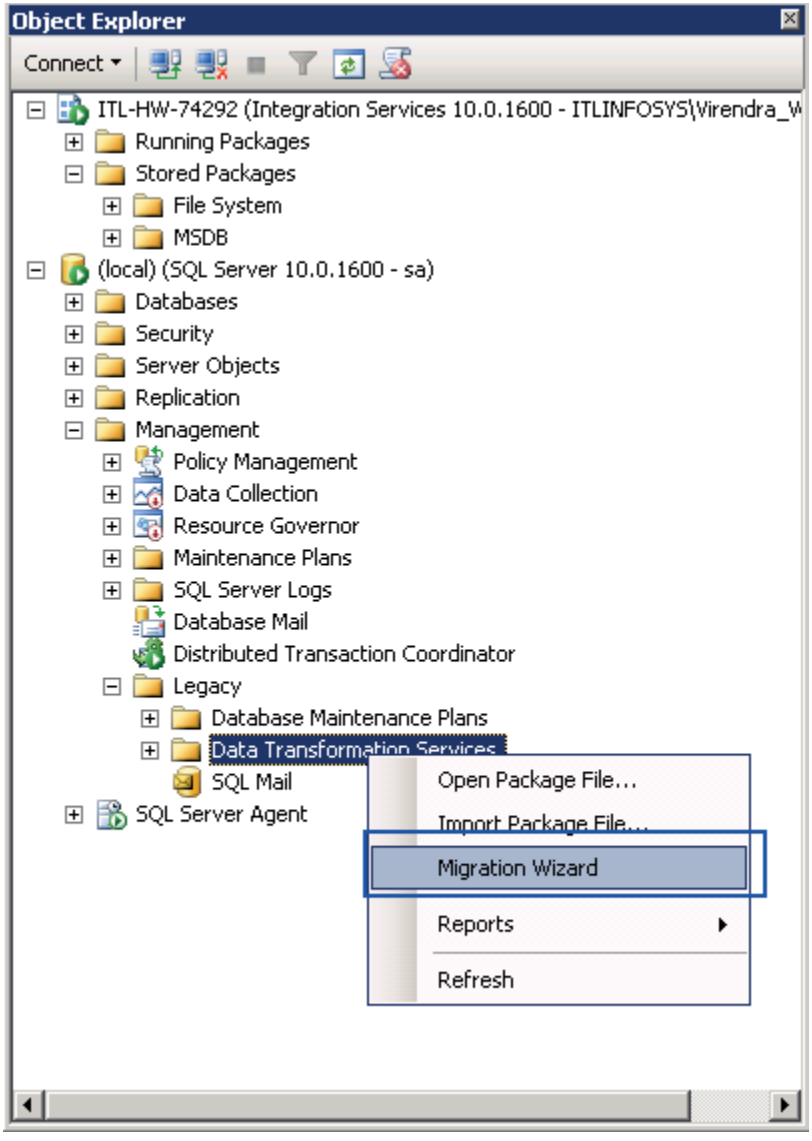

Figure 4: SSIS Migration Wizard

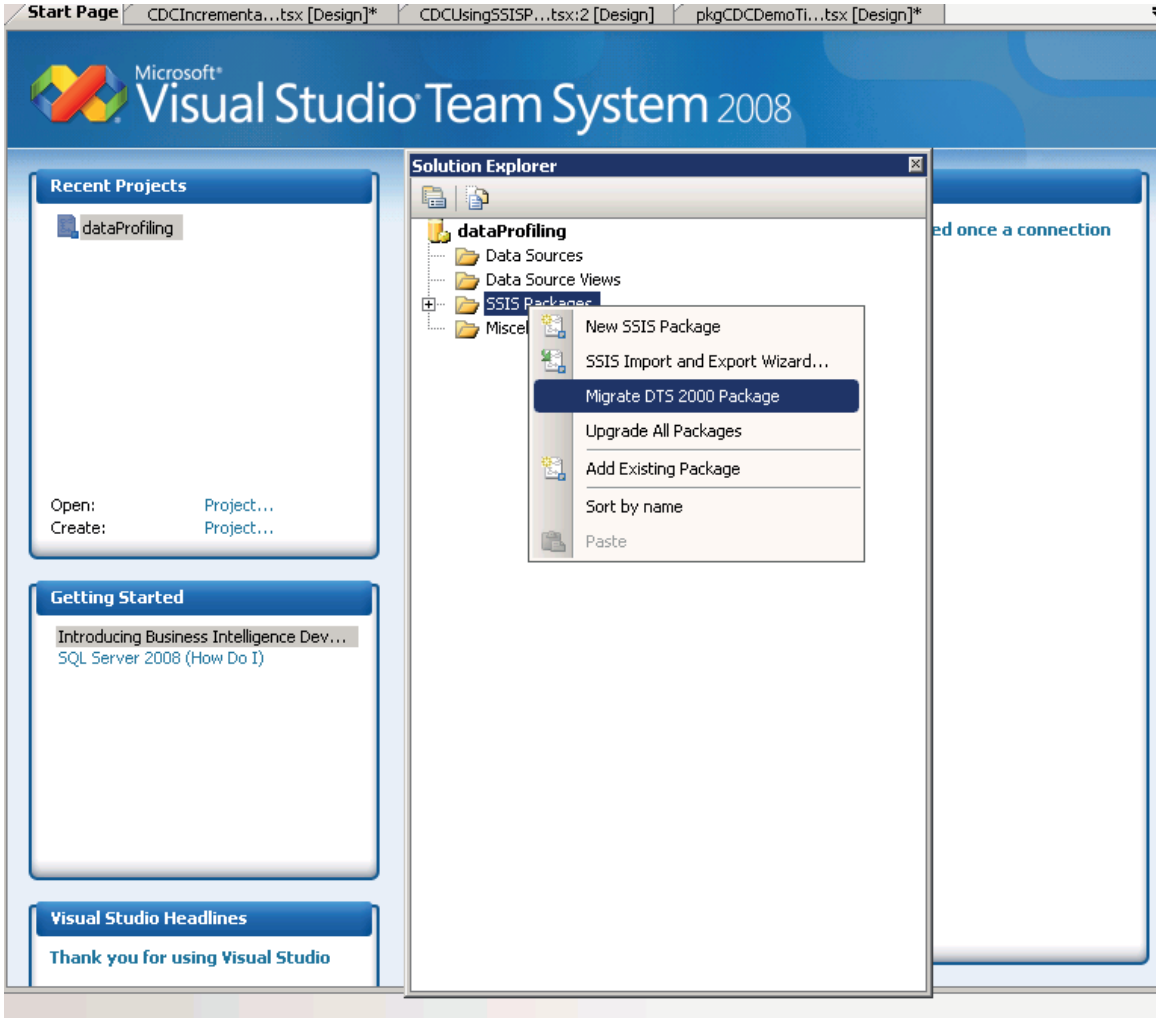

Figure 5: BIDS Migration Wizard

The migration wizard accepts the DTS package as a source that can reside either in SQL Server, or in windows folder, and the destination where migrated packages need to be stored.

The migration of DTS packages to SSIS happens as much as possible. However, few of the DTS tasks - like data driven query task, custom tasks, dynamic properties task, etc. - do not have equivalent mapping tasks available with SQL Server 2008. Even the ActiveX script task migration is not a straight forward activity. In such cases, the migration wizard attempts migration. So, in this process, either the DTS package is migrated completely or some tasks are still kept as DTS 2000 tasks.

The following two scenarios explain the behavior of the Migration Wizard during upgrade.

Scenario 1

In this scenario, the DTS package transfers data from source to destination. Using the migration wizard, the package can be successfully migrated.

When the package is migrated successfully, there will be no changes (or very minimal changes) required on the migrated package, and it works as is in the SSIS environment. In this scenario, since all the components used in DTS package have corresponding mapping components in SSIS, the package is successfully migrated. Figure 6 shows the post migration package in SQL 2008.

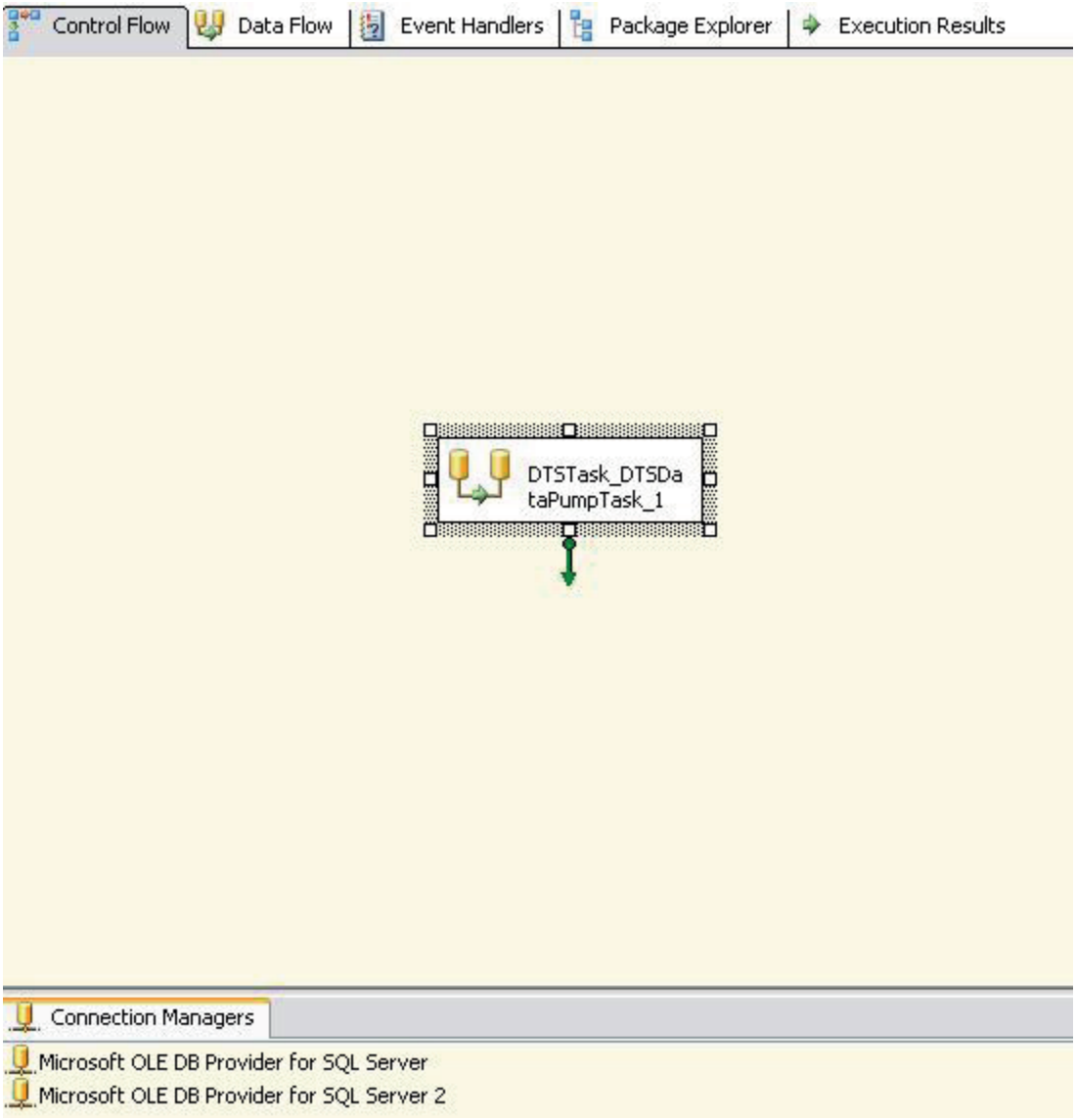

Figure 6: Migrated SSIS Packages

Scenario 2

In second scenario, the migration wizard cannot map all the components of the DTS package to SSIS equivalent components. Hence, the migration wizard keeps the component as it existed in the DTS package. Error! Reference source not found. shows that the migrated package has the execute DTS 2000 package task for the non migrated task. This is kind of partial migration and the DTS 2000 package task will continue to run using the backward compatibility support.

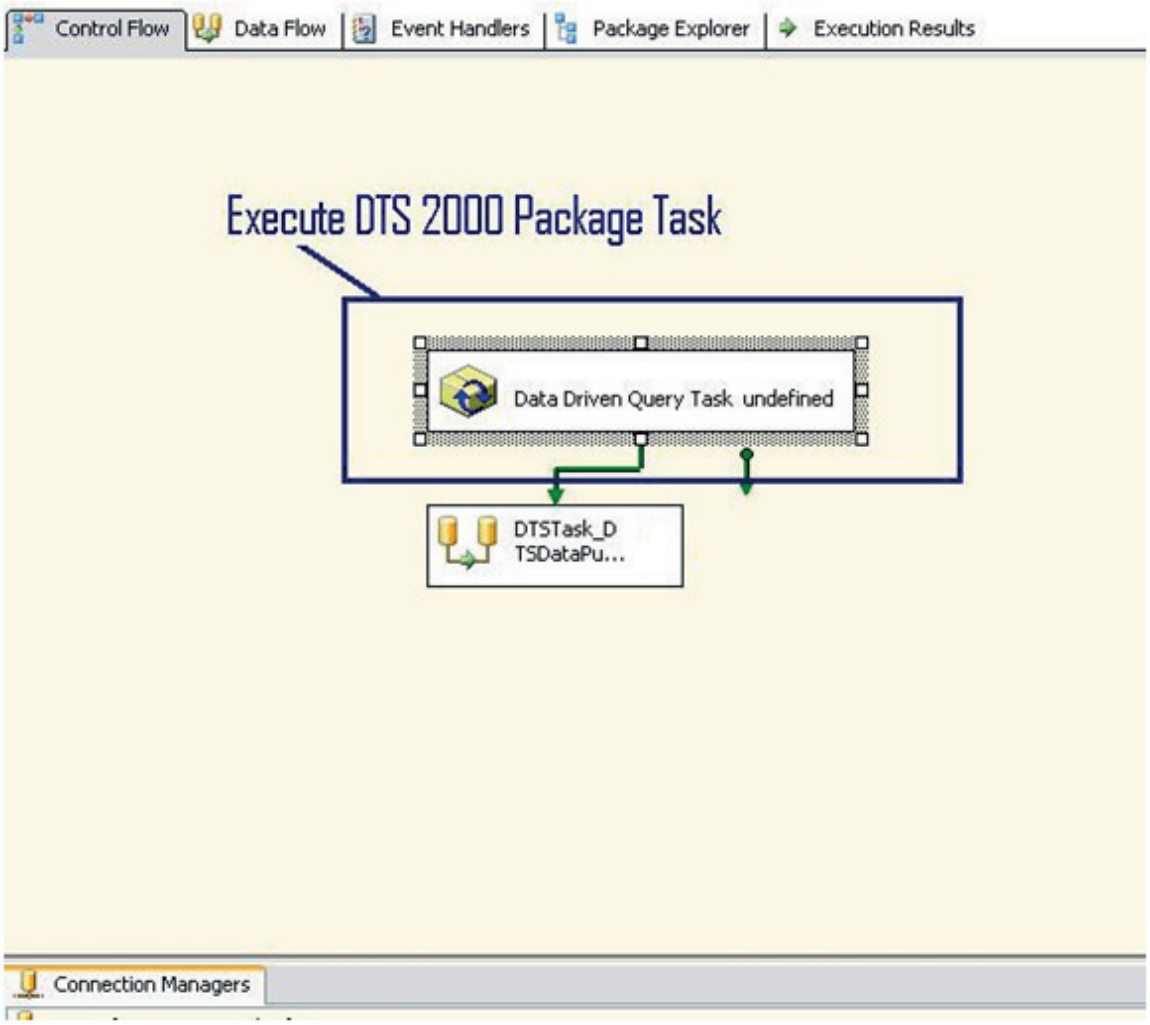

Figure 7: Execute DTS 2000 Package Task

Maintaining DTS Packages in SSIS

In the above two scenarios, the imported/ migrated DTS package will continue to work without re-writing the DTS package tasks in SSIS, with the help of Background Compatibility option discussed earlier.

In order to maintain the DTS packages in SSIS environment, DTS designer components need to be installed. The DTS designer components are available at the link, under the name DTS Designer Components, in the references section. Once you install the DTS designer components, the following steps need to be followed to edit the package from SSMS and BIDS.

To ensure that the DTS designer can be used in SQl Server Management Studio:

1. Copy the files, SEMSFC.DLL, SQLGUI.DLL, and SQLSVC.DLL, from the %ProgramFiles%\Microsoft SQL Server\80\Tools\Binn folder to the %ProgramFiles%\Microsoft SQL Server\100\Tools\Binn\VSShell\Common7\IDE folder
2. Copy the files, SEMSFC.RLL, SQLGUI.RLL, and SQLSVC.RLL, from the %ProgramFiles%\Microsoft SQL Server\80\Tools\Binn\Resources\1033 folder to the %ProgramFiles%\Microsoft SQL Server\100\Tools\Binn\VSShell\Common7\IDE\Resources\1033 folder

To ensure that the DTS designer can be used in BIDS:

1. Copy the files, SEMSFC.DLL, SQLGUI.DLL, and SQLSVC.DLL, from the %ProgramFiles%\Microsoft SQL Server\80\Tools\Binn folder to the %ProgramFiles%\Microsoft Visual Studio 9.0\Common7\IDE folder

2. Copy the files, SEMSFC.RLL, SQLGUI.RLL, and SQLSVC.RLL, from the %ProgramFiles%\Microsoft SQL Server\80\Tools\Binn\Resources\1033 folder to the %ProgramFiles%\Microsoft Visual Studio 9.0\Common7\IDE\Resources\1033 folder.

Figure 8 shows the way to edit DTS 2000 packages in SSIS

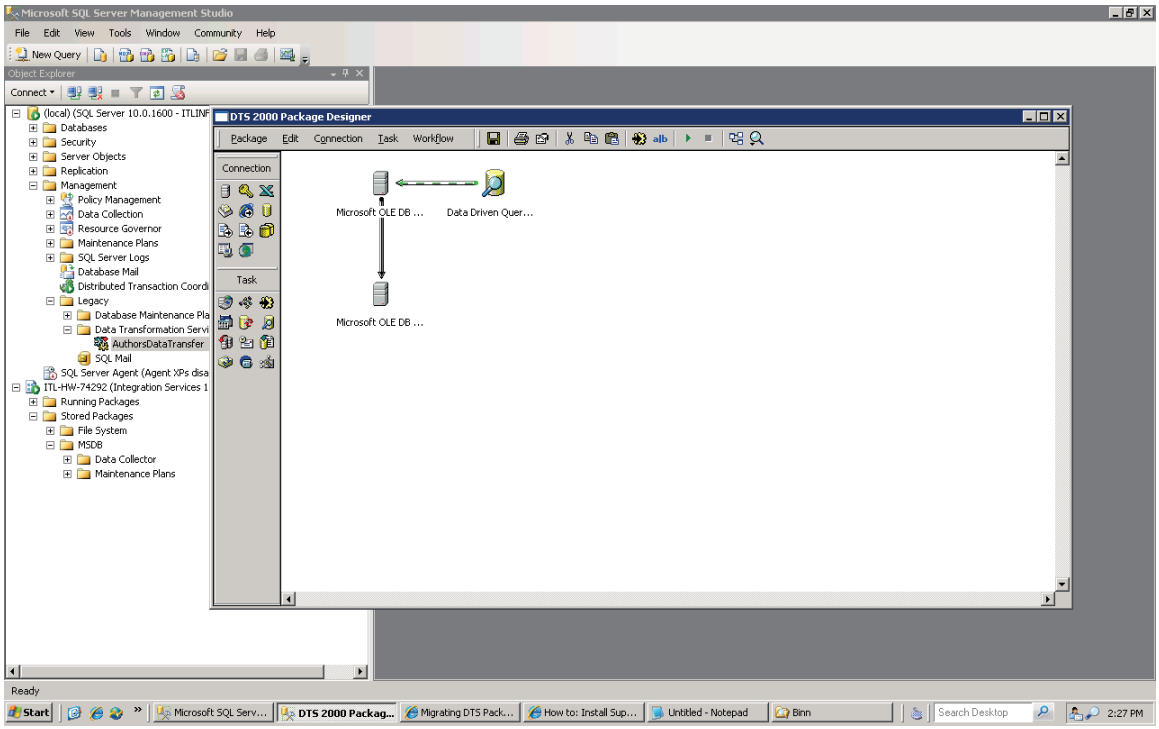

Figure 8: DTS 2000 Package Designer

Figure 9 shows DTS packages editing, using the DTS designer support in BIDS.

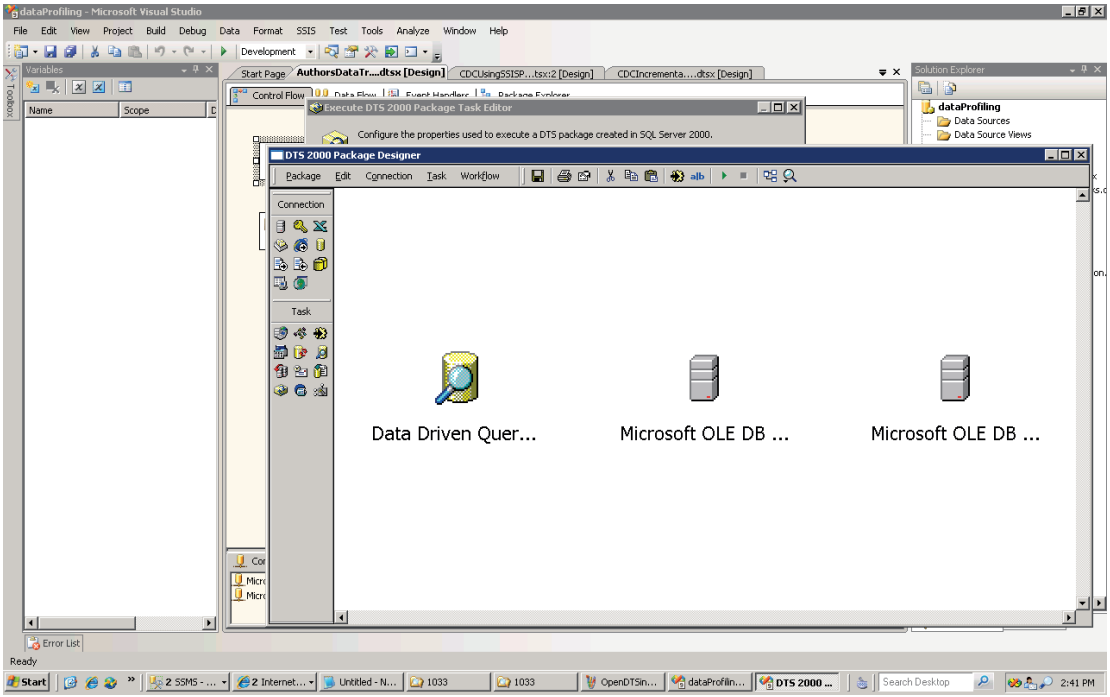

Figure 9: DTS Package in BIDS 2008

Availability of the DTS designer components in the system can be viewed from "add remove programs" option, available in Control Panel.

3. Complete Rewrite

This approach doesn't need any knowledge on the backward compatibility and DTS designer components. Even the upgrade advisor is not required. It requires strong SSIS knowledge to rewrite the DTS package, using the new tasks in SSIS.

In this approach, the DTS package will be rewritten to achieve the functionality, by making use of the rich features available with SSIS. This is the best approach and you can avail all the benefits of the new technology. The efforts required in this approach would be the highest, and it's worth putting the efforts as you would see a good amount of performance improvement. It will also reduce the complexity of your DTS packages. Most of the times; Active X script task had to be used in DTS to achieve functionality - like Looping, conditional logic, lookups, "UPSERT" statements, etc.

There are three basic reasons why the packages should be re-written in SSIS as mentioned below:

1. SSIS has major feature additions, which boosts the power of your ETL programs. Some of the important features are listed below.
    a. Enhanced Debugging features with super dump option
    b. Security Enhancements
    c. Deployment utilities
    d. Separate Data Flow and Control Flow engine
    e. More heterogeneous data sources supported
    f. Various new transformations
    g. Configuration Capabilities
    h. Data Profiling Tasks to identify the data quality issues.
    i. Cached lookup transformation
2. SSIS performance is better than DTS due to buffer oriented architecture. All the data is loaded and handled in the memory. Due to this, SSIS transformations are faster than DTS. Also, in SQL 2008, the SSIS package is executed using the parallelism.
3. DTS is deprecated by Microsoft, which means it will not be supported in future versions. SQL 2008 still supports the backward compatibility.

Migration Decision Matrix

Now, once we have understood all the three approaches and there pros and cons, the following table along with the upgrade advisor report can be useful to decide the desirable approach to follow in your environment.

Criteria	Without Migration/ Upgrade	Migration Wizard	Complete Rewrite
Efforts	Low	Medium	High
Complexity	Low	Medium	Medium
Performance Benefits	-	Low	High
Maintenance	High	Medium	Low

Post Migration Steps

This section explains possible exceptions in tasks that require manual intervention.

ActiveX Script task

ActiveX Script task allows you to write scripts using Visual Basic. These scripts can perform logic that cannot be implemented with the standard tasks available with DTS. It is recommended to replace an ActiveX Script task in DTS with a Script Task in SSIS, for the following reasons:

- Some portions of ActiveX script task access the DTS object model. This behavior is not supported in SSIS
- Script task in SSIS is well integrated with Visual Studio environment. Hence, it is possible to write script code in Visual Basic.NET, and C#, which are well known .Net programming languages.

Dynamic Properties task

Dynamic Properties task assign values to package properties from external sources like environment variables. For example, the Data Source property of an OLEDB connection manager can be assigned values from an environment variable "SERVER_NAME", using the dynamic properties task.

There is no mapping task for dynamic properties task of DTS in SSIS. When migrated, the dynamic properties task is encapsulated in a dummy script task with the same name. The functionality of dynamic properties task is done using package configurations. Post migration; invoke the Package Configuration wizard to set property values from external sources. An example of setting configurations using Package Configuration Wizard is shown in *Figure 10*

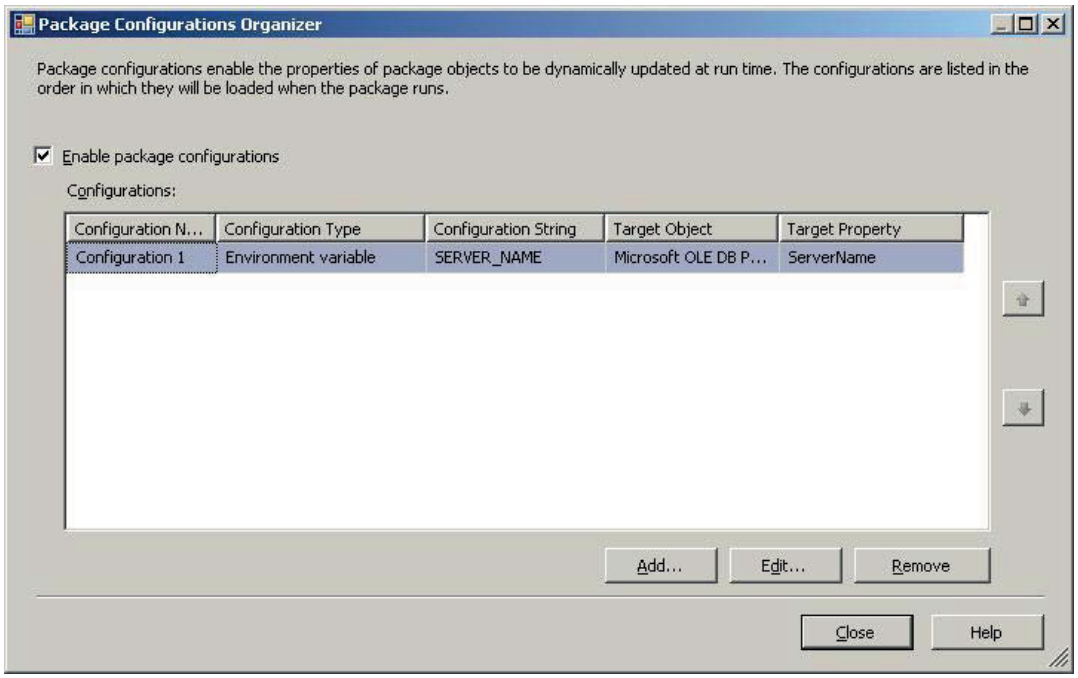

Figure 10: Package Configuration Wizard

Looping

ActiveX Script Task is used to implement looping in DTS. It accesses the DTS object model to achieve the looping functionality. A global variable is used to implement the loop variable. The script checks if the loop variable is less than the maximum iterations and accesses the package object to invoke an appropriate step within the workflow. The code for looping is available in Appendix A

SSIS has a feature called looping container. Any task that needs to be looped should be placed in the looping container.

Data Driven Query Task

The Data Driven Query task in DTS is used to perform SQL operations on data. For each row in the source, the data driven query task executes an SQL statement. SSIS has features like Slowly Changing Dimension (SCD) transformation, the OLEDB Command transformation, or the Conditional Split transformation that can be used as a replacement for a data driven query task.

Execute DTS 2000 Package Task

In SSIS, DTS packages can be run as it is, using the "Execute DTS 2000 package" task. This option is useful when certain packages cannot be migrated immediately.

Conclusion

Use upgrade advisor to check readiness of the source database for an upgrade. This will be helpful in scoping and effort estimation for DTS migration to SSIS. Based on the number of issues, one can decide upon the desirable DTS packages to be migrated and the one's that should be maintained and run with the DTS 2000 designer components support.

Upgrade advisor is a tool that helps in performing the upgrade. After the upgrade, DTS packages are migrated to SSIS, using the Package Migration Wizard.

Appendix A - Looping in DTS

The following code implements looping in DTS. It loops for 10 times and moves execution to the

task named DTSStep_DTSExecutePackageTask_1

```
'**********************************************************
'Visual Basic ActiveX Script
'**********************************************************
Function Main()
        Dim pkg
        Dim stpbegin
        'Increase the count by 1 after each execution of the Transfer Data Task
        DTSGlobalVariables("Count"). Value = DTSGlobalVariables("Count"). Value + 1

        'decide if we need to loop
        if DTSGlobalVariables("Count").value < 11 then

        set pkg = DTSGlobalVariables. Parent
        'the name of the task can be obtained by right click on the task, go to Workflow
Properties, then
        'choose the options tab.
        set stpbegin = pkg.Steps("DTSStep_DTSExecutePackageTask_1 ")
        stpbegin. ExecutionStatus = DTSStepExecStat_Waiting
    end if
    Main = DTSTaskExecResult Success
End Function
```

References

1. Why Upgrade to SQL 200S - http://www.microsoft.com/sqlserver/2008/en/us/oltp.aspx
2. Microsoft SQL 2005 Upgrade Advisor- http://www.microsoft.com/downloads/details.aspx?familyid=1470E86B-7E05-4322-A677-95AB44F12D75&displaylang=en
3. How to: Ensure Support for Data Transformation Services Packages - http://msdn.microsoft.com/en-us/library/ms143755.aspx
4. SSIS backward compatibility - http://msdn.microsoft.com/en-us/library/ms143706.aspx
5. Mainstream support for SQL 2005 sp1 and SQL 2000 sp4.- http://blogs.msdn.com/sqlreleaseservices/archive/2008/02/15/end-of-mainstream-support-for-sql-server-2005-sp1-and-sql-server-2000-sp4.aspx
6. DTS Designer Components- http://www.microsoft.com/downloads/details.aspx?FamilyID=50b97994-8453-4998-8226-fa42ec403d17&displaylang=en

Acknowledgements

We would like to acknowledge the support provided by the following:

1. Naveen Kumar (nkumar@infosys.com), Atul Gupta (atulg@lnfosys.com) - Principal Architects, Microsoft Technology Center. Their timely reviews and suggestions were very valuable to get this paper in its current shape.

Infosys® | Building Tomorrow's Enterprise

For more information, contact askus@infosys.com

About Infosys

Many of the world's most successful organizations rely on Infosys to deliver measurable business value. Infosys provides business consulting, technology, engineering and outsourcing services to help clients in over 30 countries build tomorrow's enterprise.

For more information about Infosys (NASDAQ:INFY), visit www.infosys.com.