

# Migrating from DTS

## CHAPTER 3

### IN THIS CHAPTER

- The Scope of Things
- Available Migration Resources
- Some Migration Considerations

**“WHAT HAVE YOU DONE WITH MY DTS?!”—DARREN GREEN**

### The Scope of Things

When presenting Integration Services (IS) to current or former Data Transformation Services (DTS) users, one of the first things they want to know is how Integration Services will affect their current packages. The questions come in different forms, such as the following: Will it break my current packages? Can I run IS side by side with DTS? Can I continue to use my DTS packages while building new IS packages? or Can I upgrade my DTS packages to IS packages? But, essentially, what people are asking is, “What’s the migration story?”

---

#### CAUTION

If you have DTS packages, please read this chapter thoroughly before you begin installing SQL Server 2005. It is possible that you could lose access to your existing packages if you’re not careful.

Some of the answers are simple; some are complex. For the most part, the story is pretty positive and much of the difficulty with upgrading is related to the decisions you must make about when and how to do the migration. You also have some good options for retaining your current investment in DTS packages.

This chapter explains the available resources for performing migrations and some of the problems you might encounter. By the end of the chapter, you should have answers to the previous questions and a better idea how to proceed with your upgrade and migration.

### NOTE

If you have never used DTS or you have no existing DTS packages, you can safely skip this chapter.

## SSIS Is Not DTS

The first thing everyone should clearly understand is that DTS and SQL Server Integration Services (SSIS) are completely different applications. If you were to draw rudimentary system views of DTS and SSIS on a whiteboard, it might be difficult to tell them apart, depending on how you draw them. There are still transforms, tasks, precedence constraints, connections, and variables, and the purpose for the products are closely related. But that's where the similarities end.

There is very little code shared between the two products. The architectures are very different. The scope of Integration Services is largely expanded to be a true enterprise-level Integration and Extract, Transform, and Load (ETL) platform. The data integration and transformation features in Integration Services are hugely improved and the application itself provides a more robust environment for true ETL. In short, anyone reading this should clearly understand that although Integration Services was designed upon ideas developed in DTS, it is a wholly different product.

## The Scope of Differences

This section briefly covers some of the more pronounced differences between Integration Services and Data Transformation Services. Many improvements or additional features in IS were not found in DTS. For example, the designer has been vastly improved and there are a large number of new tasks and transforms. But, because this chapter is about migrating existing DTS solutions, the discussion is constrained to those feature differences that are specifically problematic when migrating. Table 3.1 shows the list of the four most problematic feature differences.

**TABLE 3.1 Major Feature Differences Between DTS and SSIS**

Feature	DTS Solution	SSIS Solution	Description
Data transformation	Limited transforms and ActiveX script with the pump	Numerous stock transformations, error outputs, and easy extensibility with custom components in the Data Flow Task	DTS transforms were limited and ActiveX was slow. ActiveX transformations cannot be migrated.

TABLE 3.1 Continued

Feature	DTS Solution	SSIS Solution	Description
Control flow looping	ActiveX scripting using DTS object model	Built-in Loop container objects with stock enumerators	DTS had no built-in looping constructs and users were forced to hack the object model with ActiveX scripts. ActiveX looping code cannot be migrated.
Package initialization and configuration	Dynamic Properties Task and ActiveX Script Task	Package configurations and property expressions	The Dynamic Properties Task and ActiveX Script Task traversed the package object model to configure other tasks. SSIS no longer allows tasks access to the object model.
Custom behavior in packages	The ActiveX Script Task was used extensively to provide not only task behavior, but DTS runtime behavior.	The Script Task can only be used to provide task functionality and cannot access the object model or other tasks.	This is a specific instance of a general problem, which is that tasks accessing the object model are difficult to support and upgrade.

As you can see from Table 3.1, most of the problems arise from the differences between the Pump Task and the Data Flow Task, using ActiveX and allowing tasks to access the object model in promiscuous ways. The Integration Services Data Flow Task is a whole different class of integration and transformation tool, with its own object model and new set of transformations and adapters. The differences are too numerous to list here. ActiveX scripting was a great tool for one-off, simple work. However, because it is interpreted, ActiveX scripting can be quite slow and doesn't scale well. Finally, DTS lets tasks modify the object model of an executing package, which created numerous problems, not the least of which was how to seamlessly migrate packages if the object model changes. Microsoft decided to eliminate task object model access altogether for these and many other similar reasons. Because of these and other differences, you take a one-time

migration hit with the expectation that by eliminating the previously mentioned barriers to migration, future upgrades will be much easier.

## Good News

Now, for the good news: Microsoft took great care to provide viable and reasonable migration paths for existing DTS packages. You, as the DTS user, have the ultimate control for how and when you migrate to Integration Services. As you'll see in the following sections, you have options for upgrading, migrating, embedding, sharing, calling, and/or running side by side with DTS packages so that you can continue to use your existing investment in DTS while migrating to Integration Services according to your requirements. Also, the upgrade process, no matter how you approach it, is noninvasive and keeps your packages intact.

## Available Migration Resources

Knowing that the changes in architecture and features would essentially break a large number of customers, the Integration Services team developed a set of tools and features that you can use to simplify migration. These tools can be used exclusively or in combination with each other, depending on the approach you take, as discussed in the next section.

---

### WHO IS JIM HOWEY?

---

Jim is the Program Manager (PM) on the Integration Services team who was largely responsible for, among other things, championing the cause of you, the DTS 2000 user who needs to upgrade packages. In any project of scale and complexity, there are always several priorities that compete for attention and resources. Jim constantly pushed, at times against determined opposition from the author and others, for a better DTS 2000 migration experience. It is largely through his efforts and sturdy determinism that the flexible and substantial options presented here are available. Jim managed the Upgrade Advisor, Migration Wizard, Execute DTS 2000 Package Task, and the SQL Server DTS 2000 Designer Components that make it possible to edit DTS packages in the Execute DTS 2000 Package Task.

So, the next time you successfully upgrade a DTS package or edit one in the Execute DTS 2000 Package Task UI, think of Jim.

## Upgrade Advisor

The Upgrade Advisor is a tool that interrogates your servers and builds a report that lists any potential problems you might have when migrating to SQL Server 2005. Running the Upgrade Advisor should be the first thing you do when you're considering an upgrade.

### Running the Upgrade Advisor

When you first place the SQL Server setup disk into your machine, there should be a splash screen that appears with several links on it. Figure 3.1 shows the splash screen with the link to install the Upgrade Advisor highlighted.



**FIGURE 3.1** Install the SQL Server Upgrade Advisor from the Setup Splash Screen.

---

### TIP

Some users have reported trouble getting the splash screen to show up. You can install the Upgrade Advisor from the disk directly. Under the `redist` folder, there is an `Upgrade Advisor` folder with an `msi` file called `SQLUASetup.msi`. Right-click on the `msi` file and select `Install` to install the Upgrade Advisor.

After you've stepped through the setup and have the Upgrade Advisor installed, you can launch it from the SQL Server 2005 Start menu. Figure 3.2 shows the Upgrade Advisor start page. Some links are provided for finding out more about upgrading. The bottom two links are the most interesting. The first link, `Launch Upgrade Advisor Analysis Wizard`, lets you perform the actual upgrade analysis. The second link, `Launch Upgrade Advisor Report Viewer`, lets you view reports generated from previous runs of the Upgrade Advisor.

Click the link to launch the Upgrade Advisor Analysis Wizard. Figure 3.3 shows the first page in the wizard, which tells you the steps you'll take in the wizard.

Click the `Next` button to see the page shown in Figure 3.4. Type in the name of the server and click `Detect`; the wizard autodetects what SQL Server 2000 components are installed on the server.

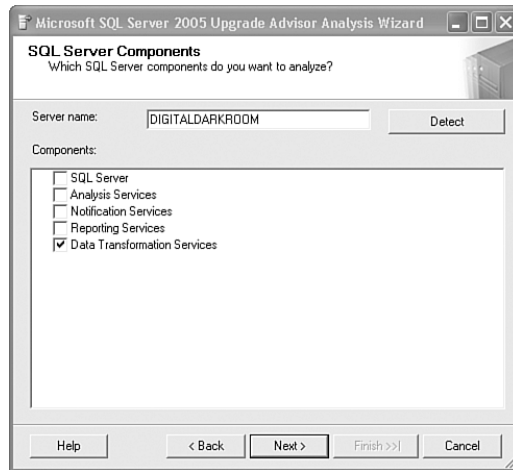
For the purposes of this exercise, you are only interested in getting advice about upgrading your DTS packages, so leave only the `Data Transformation Services` check box checked. Click the `Next` button, and input your credentials to attach to the server and the instances you want to scan. You can select multiple instances. Click the `Next` button again to see the dialog box shown in Figure 3.5.



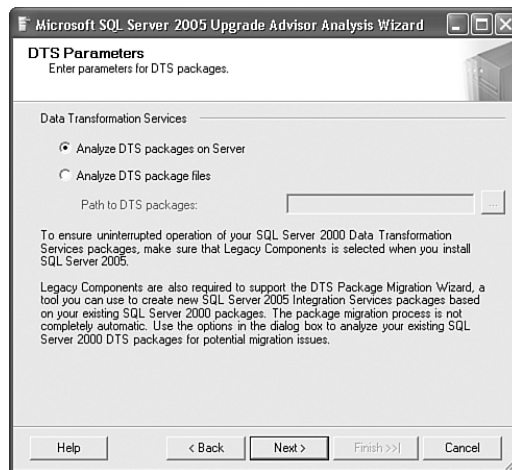
**FIGURE 3.2** The Upgrade Advisor lets you view reports or run the Upgrade Advisor Analysis Wizard.



**FIGURE 3.3** The Upgrade Advisor Analysis Wizard start page details the steps you'll follow.

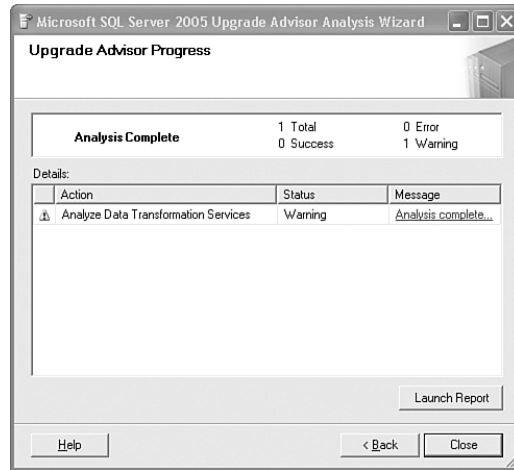


**FIGURE 3.4** The wizard autodetects SQL Server 2000 components.



**FIGURE 3.5** The wizard allows you to scan packages on the server or file system.

The wizard mentions the Package Migration Wizard, which is covered later. When you select packages on the server, all packages on the server are scanned. There is no way to scan a subset of the packages. Click the Next button and the wizard shows you a summary of the settings. Click the Run button and the wizard starts scanning and analyzing the packages on the server and instances you've selected. Figure 3.6 shows the dialog box shown after the wizard has completed the analysis.



**FIGURE 3.6** The wizard generates a report after performing the analysis.

After the wizard has completed the analysis, it writes out the report to the SQL Server 2005 Upgrade Advisor Reports folder under your My Documents folder. You can save these reports to another location and review them later if you want. However, if you leave the report in the same location, the wizard may overwrite it.

Figure 3.7 shows a report the wizard generated for the DTS packages on the test server.

You can do a number of things with the reports. You can open existing reports, filter line items by issues or component, and even export them. Pay close attention to the When to Fix column. If it says Before, you'll need to understand that entry well, and possibly take action on it before upgrading. To get more information about each issue, click on the issue in the report. Figure 3.7 shows the expanded description for Meta Data Services packages after clicking on it. Notice that it is possible to lose access to the packages stored in Meta Data Services. Again, it is very important that you take time to scan and analyze your packages **before** you start your SQL Server installation so that you know about the preinstallation tasks before it is too late. In this case, the Upgrade Advisor is telling the user to move any packages stored in Meta Data Services to SQL Server or save them to the file system before installing SQL Server because they will no longer be accessible after installing SQL Server 2005. Certainly, this is something you'd want to handle before installing.

---

### CAUTION

Make sure you run the Upgrade Advisor Analysis Wizard before you install SQL Server 2005. It will forecast the problems you might experience when upgrading so you can make informed decisions on how to proceed.



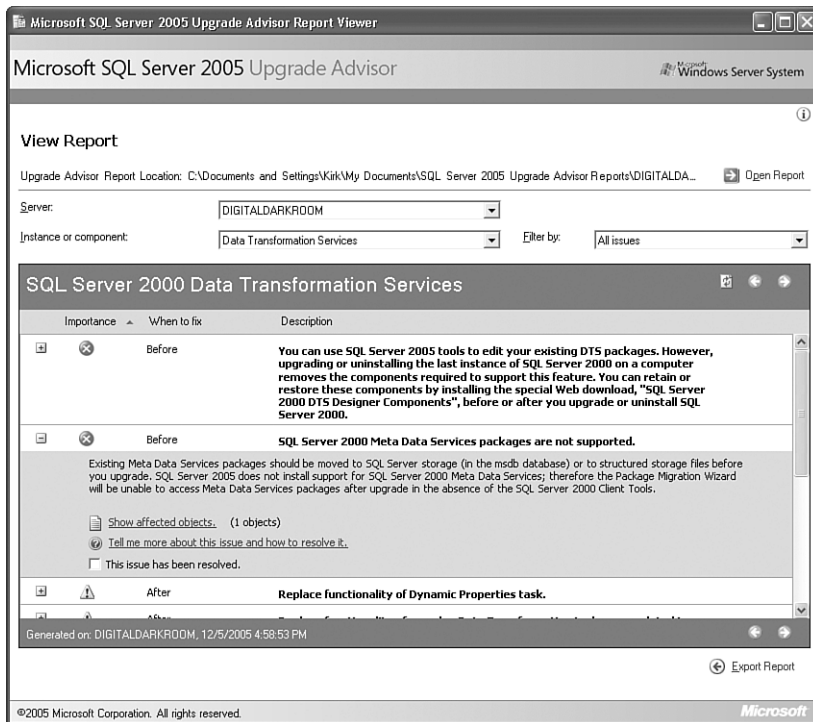


FIGURE 3.7 The Upgrade Advisor Report Viewer shows the analysis results.

## The Execute DTS 2000 Package Task

The Execute DTS 2000 Package Task is a thing of beauty for anyone who needs to use their DTS packages in Integration Services. It provides the entire DTS 2000 package editing environment inside the Business Intelligence Development Studio and a way to execute DTS packages from an Integration Services package. This task is provided to help bridge the gap when migrating from Data Transformation Services to Integration Services. Because the architecture of Integration Services is so different from DTS, few legacy DTS packages will successfully migrate and run in Integration Services. This task provides a “half” migration step so that legacy packages can be run side by side with newer Integration Services packages. Even limited interaction between the two is possible using outer variables such that existing investment in DTS packages is not lost when upgrading to Integration Services. For more information about this task, see Chapter 8, “The Stock Tasks.”

## Migration Wizard

The Migration Wizard is a way to convert your DTS packages to Integration Services packages. It is a “best efforts” migration, meaning it attempts to migrate the biggest parts of packages, such as tasks and variables. However, in the event that it fails to successfully

migrate a package, it migrates those parts of the package that it can and wraps the rest in an Execute DTS 2000 Package Task.

### When Package Migration Fails

Many DTS packages will successfully migrate, but most will not. What could cause package migration to fail? The list in Table 3.1 is a good place to start. For example, the DTS Data Transformation Task and the Integration Services Data Flow Task are not compatible and their features don't map to one another. Custom tasks won't migrate because they depend on the DTS object model. The Multiphase Pump and Dynamic Properties Task migration will fail because they have no counterpart tasks in IS and you will probably need to fix up ActiveX Script Tasks, especially if they access the DTS object model.

Because the object models are so different, the package and task properties have changed, and there are so many different package configurations, the odds are against a thoroughly clean migration. Even IS tasks that map well to their DTS counterparts will sometimes fail migration because of some setting that isn't related to the task. For example, the IS packages might be stored on a machine in a different domain without access to the same files or servers. The simple fact that you move the package to a different location can break it. The takeaway here is that there is no free ride. You might get lucky and have great success with migrating a few packages. But, as a general rule, most migrations will be a fairly labor-intensive operation. Fortunately, as discussed in the following section, you have options.

### Individual Components and Settings

When it comes to migrating packages, the success or failure of the package migration really boils down to what tasks and connections are in the DTS package. Also, a DTS package is more than a set of tasks. It's a combination of tasks, transforms, logging, and other package settings. A number of DTS-compatible components and settings in Integration Services seem to migrate pretty well with little or no user intervention. In addition, some components and settings don't migrate at all or somewhere in between. Following is a list of DTS components and settings with a very rough expectation of how often each will successfully migrate. The success probability is based on estimates on a number of factors, including the author's own tests, the number of differences between DTS and IS components, the complexity of the component, and wizard support. Notice that none of the success probabilities are estimated at 100%. This is because there are simply too many points of failure when migrating to be sure that any component will succeed all the time.

---

#### CAUTION

The estimates given here for probability of success are the author's. They are strictly based on his experience. You can use them to sanity check the effort it will take to migrate your packages, but please don't rely on these numbers for any substantial commitments. Only by testing in your environment with a representative subset of your packages can you accurately gauge what your migration effort will be.

**Migrating Tasks** There seem to be three categories of DTS tasks. The first category is tasks that migrate straight across with few complications and if they don't run successfully the first time, a tweak or two usually fixes any lingering issues. Table 3.2 shows the tasks in this category listed with the IS task to which they migrate.

**TABLE 3.2** Tasks with High Probability of Migration Success

DTS Task	IS Task	Success Probability
File Transfer Protocol	File Transfer Protocol	90%
Execute SQL	Execute SQL	95%
Execute Process	Execute Process	95%
Copy SQL Server Objects	Transfer SQL Server Objects	90%
Send Mail	Send Mail (New SMTP Info)	95%
Bulk Insert	Bulk Insert	70%
Execute Package	Execute DTS 2000 Package	90%
Message Queue	Message Queue	80%
Transfer Error Messages	Transfer Error Messages	95%
Transfer Databases	Transfer Database (one for each database)	85%
Transfer Master Stored Procedures	Transfer Master Stored Procedures	95%
Transfer Jobs	Transfer Jobs	95%
Transfer Logs	Transfer Logs	95%

The second category contains those tasks that might migrate successfully, but often fail because they are either too complex to migrate or they do something that is no longer supported. The ActiveX Script Task often fails to migrate because it contains code that attempts to access the DTS Object Model. The Data Pump Task fails if it is too complex or contains ActiveX Script. The Data Driven Query does not migrate, but the Migration Wizard wraps it in its own DTS package and calls the package from the Execute DTS 2000 Package Task.

**TABLE 3.3** Tasks with Low Probability of Migration Success

DTS Task	IS Task	Success Probability
ActiveX Script	ActiveX Script	40%
Transform Data	Data Flow	40%
Data Driven Query	Execute DTS 2000 Package	40%

The third and final category of tasks is those that simply will not migrate because they are designed to work against a different API or there is no compatible Integration Services task. For example, the Dynamic Properties Task is built upon the premise that it can

traverse the package hierarchy and modify other task properties directly. This won't work for two reasons:

- The runtime doesn't allow promiscuous package access any longer.
- Even if it did allow such access, the object model has changed.

**TABLE 3.4** Tasks That Will Not Migrate Successfully

DTS Task	IS Task	Success Probability
Analysis Services	Analysis Services Processing	0%
Dynamic Properties	ActiveX	0%
Data Mining Prediction	Data Mining Query	0%

**Migrating Connections** Connections are important in DTS and if the connection doesn't migrate successfully, you've got a broken package just the same as if the task didn't successfully migrate. The good news is that the OLEDB connections, the most common connections in DTS packages, have a very high migration success rate because the settings for the IS OLEDB Connection Manager are identical to its DTS counterpart.

**TABLE 3.5** Various Connection Managers and Their Probability to Successfully Migrate

DTS Connection	IS Connection Manager	Success Probability
OLEDB	OLEDB : OLEDB Provider	99%
Access	OLEDB : Jet 4.0 Provider	95%
Excel	OLEDB : Jet 4.0 Provider	95%
Dbase 5	OLEDB : Jet 4.0 Provider	N/A
HTML File	OLEDB : SQL Native	0%
Paradox	OLEDB : Jet 4.0 Provider	N/A
Text File (Source)	Flat File	0%*
Text File (Dest)	Flat File	0%*
ODBC	OLEDB : MSDASQL Provider	30%

*\*The Text File connections migrate to the Flat File Connection Manager, but they will never be complete until you open them in the new package and finish their configuration.*

**Migrating Settings** The package has a number of settings that impact how the package should run. The Migration Wizard generally ignores these settings. For example, it ignores all the package properties, with the exception of Global Variables. So properties like Package Priority Class, Logging Settings, Fail Package on First Error, Write Completion Status to Event Log, Use Transactions, and so on do not get migrated or in any way affect the properties of the destination IS package. The message here is: Even if your package appears to successfully migrate and you get no errors from the Migration Wizard or the new IS package, this is no guarantee that the package will behave exactly as it did when running under the DTS environment. If you want to retain the values for these properties

or the behavior they represent, you need to manually review the package settings and make appropriate settings in the IS package. The Migration Wizard does not do it for you.

### Where to Launch the Migration Wizard

You can launch the Migration Wizard from the Business Intelligence Development Studio by right-clicking on the SSIS Packages node in an Integration Services project and selecting the Migrate DTS 2000 Package menu option. You can also launch it from SQL Server Management Studio by right-clicking on the Data Transformation Services node of the Legacy branch in the Management node of SQL Server 2000 servers. The Migration Wizard is a standalone executable, so it can also be launched outside of the designer. On most machines, the Migration Wizard is in the `\Program Files\Microsoft SQL Server\90\DTS\Binn\` directory and is named `DTSMigrationWizard.exe`.

---

### CAUTION

The Migration Wizard generally fails to migrate a task if the task is missing any settings or if it cannot make a connection. To increase the likelihood of migration success, double check your packages before migrating them to ensure that all the tasks and connections are correctly configured and the servers they reference are online.

## Some Migration Considerations

When upgrading, SQL Server setup performs some installs and uninstalls of which you should be aware. The following are some key points you should understand before migrating your packages.

### SQL Server Instances and Enterprise Manager

When upgrading to SQL Server 2005, SQL Server 2005 setup uninstalls certain deprecated components. Based on how you do your installs, it's possible that the Enterprise Manager environment will get uninstalled and take the package designer with it—essentially leaving your legacy DTS packages stranded with no way to edit them. To avoid this situation, you have a number of options, which are discussed in the following sections.

---

### NOTE

When upgrading to SQL Server 2005, all packages stored in the SQL Server 2000 instance in the `msdb.dbo.sysdtspackages` table get migrated to a table by the same name in the SQL Server 2005 instance. So, the packages are not deleted or lost.

### Installing an Instance

If you install SQL Server 2005 as an instance side by side with your SQL Server 2000 instances, all the SQL Server 2000 tools remain intact and you'll have access to the packages for editing and so forth. All packages that are scheduled to run with `DTSRun` and/or `Agent` continue to execute without issue.

### Leaving One SQL Server 2000 Instance

This option is very similar to the previous one. Except instead of installing SQL Server 2005 as a different instance, you upgrade all SQL Server 2000 instances to SQL Server 2005 except one—leaving the tools intact. So long as there is a SQL Server 2000 instance on a machine, all the SQL Server 2000 tools you have used remain intact.

### Upgrading and Installing Designer Components

When you select Integration Services for installation, setup also installs support for DTS packages, including the DTS runtime and DTS package enumeration in SQL Server Management Studio. This happens as part of the SQL Server 2005 installation, but only the runtime components are installed, not the designer components.

For the design time components, Microsoft has created a special distribution called the Microsoft SQL Server 2005 DTS Designer Components. The Designer Components distribution replaces the components that get uninstalled when you upgrade your last instance of SQL Server 2000. If you want to upgrade all your SQL Server 2000 instances, this is the best and only option for recovering the DTS designer. To download the designer components, go to the download section of Microsoft.com and search for Knowledge Base article 339810.

### Meta Data Services

Meta Data Services has been deprecated and is no longer supported as a package store. SQL Server 2005 does not install or use the Repository, so you cannot enumerate or open packages stored there. You can still access packages stored in Meta Data Services using DTSRUN, but not using the designer tools. The SQL Server 2000 Data Transformation Services (DTS) designer and the dtsrun.exe utility continue to support DTS packages that were saved to Meta Data Services.

SQL Server 2005 Integration Services supports the Repository only in the Upgrade Advisor and the Package Migration Wizard, and only if SQL Server 2000, the SQL Server 2000 tools, or the Repository redistributable files are installed on the local computer. When the Repository files are present, the Upgrade Advisor can scan and the Package Migration Wizard can migrate DTS packages that were saved to Meta Data Services. When the Repository files are not present, the Upgrade Advisor can only scan and the Package Migration Wizard can only migrate DTS packages that were saved to SQL Server or to structured storage files.

The Integration Services Execute DTS 2000 Package Task cannot execute a DTS package that was saved to Meta Data Services. As a workaround, when the Repository files are present, you can create a DTS package that uses an Execute Package Task to execute the Meta Data Services package, save this new parent package to SQL Server or as a structured storage file, and execute the parent package from the Execute DTS 2000 Package Task.

## CAUTION

If you have packages stored in Meta Data Services, it is recommended that you move them to SQL Server or structured storage files before upgrading to make it easier to migrate them later. They will remain safe and unchanged through the upgrade.

## Migrating Packages with Passwords

If the DTS packages that you need to migrate have been protected with passwords, you need to know the passwords to migrate them. The passwords are not retained, nor does the Migration Wizard change the package protection setting on the resulting IS package. You need to manually modify the IS package to enable passwords again.

## Where Are My Packages?

One thing you should understand is that regardless of how you upgrade to SQL Server 2005—whether you install side by side with a separate instance, upgrade the last instance, or retain the last SQL Server 2000 instance—SQL Server setup keeps your packages intact. Depending on how you upgrade, the packages can be found in a different location.

## File System

Packages remain on the file system undisturbed. You can migrate them with the Package Migration Wizard or load them up in the Execute DTS 2000 Package Task.

## Install Instance

If you install SQL Server 2005 as an instance side by side with your SQL Server 2000 instances and you were storing your packages in SQL Server, you will still have the Enterprise Manager environment intact so the packages will be where they have always been, in the Local Packages node of the Data Transformation Services tree of Enterprise Manager. Figure 3.8 shows DTS packages stored there.

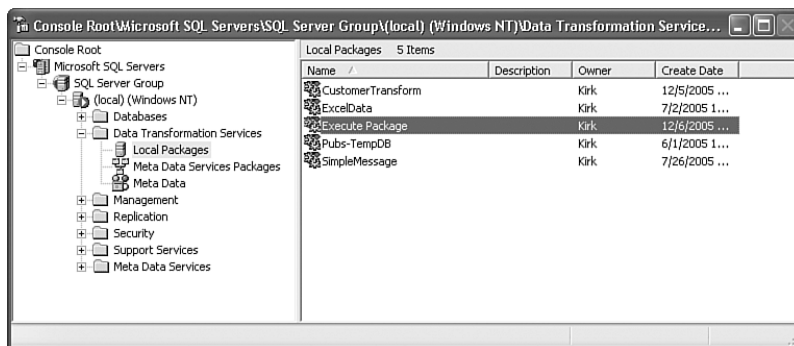


FIGURE 3.8 DTS packages remain in local packages.

Because Management Studio can enumerate packages in SQL Server 2000 instances, you can also find the DTS packages in the Object Explorer in Management Studio under the

Management tree, Legacy, Data Transformation Services node. Figure 3.9 shows the packages in the Legacy node.

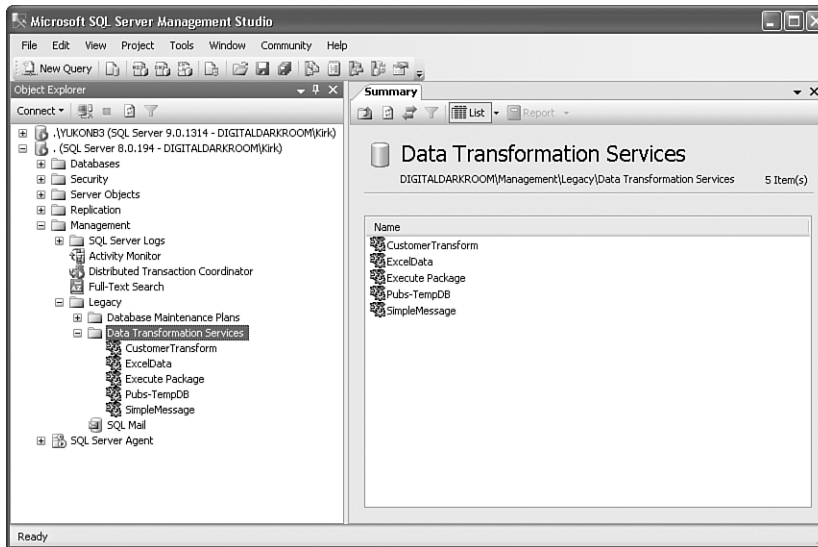


FIGURE 3.9 DTS packages remain in the Legacy node.

## Summary

You have several options available for migrating to Integration Services. The Upgrade Advisor tells you what to expect when you migrate, the Migration Wizard gets you started on migrating packages, and the Execute DTS 2000 Packages Task allows you to embed DTS packages into Integration Services packages. You can choose to keep all your DTS packages and run side by side with Integration Services while slowly migrating packages at your leisure, or you can migrate all your packages at once.

For more information, see the MSDN article “Migrating Data Transformation Services Packages” on MSDN at <http://msdn2.microsoft.com/> and search for “Migrating Data Transformation Services Packages.”