

Chapters *To Go*



Hands-On Microsoft SQL Server 2008 Integration Services

by Ashwani Nanda
McGraw-Hill/Osborne. (c) 2011. Copying Prohibited.

Reprinted for UNNIKRISHNAN NAIR R, Cognizant Technology Solutions

Unnikrishnan.NairR@cognizant.com

Reprinted with permission as a subscription benefit of **Skillport**,
<http://skillport.books24x7.com/>

All rights reserved. Reproduction and/or distribution in whole or in part in electronic, paper or other forms without written permission is prohibited.



Chapter 1: Introducing SQL Server Integration Services

Overview

Now that the SQL Server 2008 R2 is coming over the horizon packed with self-service business intelligence features, Integration Services not only remains the core platform for the data integration and data transformation solutions but has come out stronger with several product enhancements. With more and more businesses adopting Integration Services as their preferred data movement and data transformation application, it has proven its ability to work on disparate data systems; apply complex business rules; handle large quantities of data; and enable organizations to easily comply with data profiling, auditing, and logging requirements.

The current credit crunch has left businesses in a grave situation with reduced budgets and staff yet with the utmost need to find new customers and be able to close sales. Business managers use complex analytical reports to draw up long-term and short-term policies. The analytical reports are driven by the data collected and harvested by corporate transactional systems such as customer support systems (CRM), call centers and telemarketing operations, and pre- and post-sales systems. This is primarily due to the data explosion because of the increased use of the web. People now spend more time on the web to compare and decide about the products they want to buy. Efforts to study buyer behavior and to profile activities of visitors on the site have also increased data collection. Data about customers and prospects has become the lifeblood of organizations, and it is vital that meaningful information hidden in the data be explored for businesses to stay healthy and grow.

However, many challenges remain to be met before an organization can compile meaningful information. In a typical corporation, data resides at geographically different locations in disparate data storage systems—such as DB2, Oracle, or SQL Server—and in different formats. It is the job of the information analyst to collect data and apply business rules to transform raw data into meaningful information to help the business make well-informed decisions. For example, you may decide to consolidate your customer data, complete with orders-placed and products-owned information, into your new SAP system, for which you may have to collect data from SQL Server–based customer relationship management (CRM) systems, product details from your legacy mainframe system, order details from an IBM DB2 database, and dealer information from an Oracle database. You will have to collect data from all these data sources, remove duplication in data, and standardize and cleanse data before loading it into your new customer database system. These tasks of extracting data from disparate data sources, transforming the extracted data, and then loading the transformed data are commonly done with tools called *ETL* tools.

Another challenge resulting from the increased use of the Internet is that “the required information” must be available at all times. Customers do not want to wait. With more and more businesses expanding into global markets, collecting data from multiple locations and loading it after transformation into the diverse data stores with little or no downtime have increased work pressure on the information analyst, who needs better tools to perform the job.

The conventional ETL tools are designed around batch processes that run during off-peak hours. Usually, the data-uploading process in a data warehouse is a daily update process that runs for most of the night. This is because of the underlying design of traditional ETL tools, as they tend to stage the data during the upload process. With diverse data sources and more complex transformations and manipulations, such as text mining and fuzzy matching, the traditional ETL tools tend to stage the data even more. The more these tools stage data, the more disk operations are involved, and hence the longer the update process takes to finish. These delays in the entire process of integrating data are unacceptable to modern businesses. Emerging business needs require that the long-running, offline types of batch processes be redesigned into faster, on-demand types that fit into shorter timeframes. This requirement is beyond the traditional ETL tools regime and is exactly what Microsoft SQL Server 2008 Integration Services (SSIS) is designed to do.

Microsoft SQL Server Integration Services (also referred as SSIS in this book) is designed keeping in mind the emerging needs of businesses. Microsoft SQL Server 2008 Integration Services is an enterprise data transformation and data integration solution that can be used to extract, transform, and consolidate data from disparate sources and move it to single or multiple destinations. Microsoft SQL Server 2008 Integration Services provides a complete set of tools, services, and application programming interfaces (APIs) to build complex yet robust and high-performing solutions.

SSIS is built to handle all the workflow tasks and data transformations in a way that provides the best possible performance. SSIS has two different engines for managing workflow and data transformations, both optimized to perform the nature of work they must handle. The *data flow engine*, which is responsible for all data-related transformations, is built on a buffer-oriented architecture. With this architecture design, SSIS loads row sets of data in memory buffers and can perform in-memory operations on the loaded row sets for complex transformations, thus avoiding staging of data to disks.

This ability enables SSIS to extend traditional ETL functionality to meet the stringent business requirements of information integration. The *run-time engine*, on the other hand, provides environmental support in executing and controlling the workflow of an SSIS package at run time. It enables SSIS to store packages into the file system or in the MSDB database in SQL Server with the ability to migrate the package between different stores. The run-time engine also provides support for easy deployment of your packages.

There are many features in Integration Services that will be discussed in detail in the relevant places throughout this book; however, to provide a basic understanding of how SSIS provides business benefits, the following is a brief discussion on the features and their uses.

Integration Services: Features and Uses

In order to understand how Integration Services can benefit you, let us sift through some of the features and uses that it can be put to. Integration Services provides rich set of tools, self-configurable components, and APIs that you can use to draw out meaningful information from the raw data, create complex data manipulation and business applications.

Integration Services Architecture

The Integration Services Architecture separates the operations-oriented workflow from the data transformation pipeline by providing two distinct engines. The Integration Services run-time engine provides run-time services such as establishing connections to various data sources, managing variables, handling transactions, debugging, logging, and event handling. The Integration Services data flow engine can use multiple data flow sources to extract data, none or many data flow transformations to transform the extracted data in the pipeline, and one or many data flow destinations to load the transformed data into disparate data stores. The data flow engine uses buffer-oriented architecture, which enables SSIS to transform and manipulate data within the memory. Because of this, the data flow engine is optimized to avoid staging data to disk and hence can achieve very high levels of data processing in a short time span. The run-time engine provides operational support and resources to data flow at run time, whereas the data flow engine enables you to create fast, easy-to-maintain, extensible, and reliable data transformation applications. Both engines, though separate, work together to provide high levels of performance with better control over package execution. You will study control flow in Chapters 3 to 5 and data flow components in Chapter 9 and Chapter 10.

Integration Services Designer and Management Tools

SQL Server 2008 provides Business Intelligence Development Studio (BIDS) as the development tool for developing and SQL Server Management Studio for managing Integration Services packages. BIDS includes SQL Server Integration Services Designer, a graphical tool built upon Microsoft Visual Studio 2008 that includes all the development and debugging features provided by the Visual Studio environment. This environment provides separate design surfaces for control flow, data flow, and event handlers, as well as a hierarchical view of package elements in the Package Explorer. The change in base technology of SSIS in this version from Visual Studio 2005 to Visual Studio 2008 for BIDS enables you to have both environments installed side by side on the same machine. BIDS 2008 provides several features that you will study later in this chapter and subsequently use throughout this book. SQL Server Management Studio allows you to connect to Integration Services store to import, export, run, or stop the packages and be able to see list of running packages. You will also study SQL Server Management Studio later in this chapter.

Data Warehousing Loading

At the core, SSIS provides lots of functionality to load data into a data warehouse. The Data Flow Task is a special task that can extract data from disparate data sources using Data Source Adapters and can load into any data store that allows OLE DB and ADO.NET connections. Most modern systems use these technologies to import and export data. For example, SSIS provides a Bulk Insert Task in the Control Flow that can bulk-load data from a flat file into SQL Server tables and views. While the Data Flow includes destinations such as OLE DB Destination, ADO NET Destination, and SQL Server Destination, these destination adapters allow you to load data into SQL Server or any other data stores such as Oracle and DB2. While loading a data warehouse, you may also perform aggregations during the loading process. SSIS provides Aggregate Transformation to perform functions such as SUM and Average and use Row Count transformation to count the number of rows in the data flow. Here are several other Data Flow Transformations that allow you to perform various data manipulations in the pipeline:

- SSIS provides three Transformations—*Merge*, *Merge Join*, and *Union All* Transformations—to let you combine data from various sources to load into a data warehouse by running the package only once rather than running it multiple times for each source.

- *Aggregate Transformation* can perform multiple aggregates on multiple columns.
- *Sort Transformation* sorts data on the sort order key that can be specified on one or more columns.
- *Pivot Transformation* can transform the relational data into a less-normalized form, which is sometimes what is saved in a data warehouse.
- *Audit Transformation* lets you add columns with lineage and other environmental information for auditing purposes.
- A new addition to SSIS 2008 is the Data Profiling Task, which allows you to identify data quality issues by profiling data stored in SQL Server so that you can take corrective action at the appropriate stage.
- Using the Dimension Processing Destination and the Partition Processing Destination as part of your data loading package helps in automating the loading and processing of an OLAP database.

Most data warehouses need to maintain a slowly changing dimension. Integration Services provides a Slowly Changing Dimension (SCD) Transformation that can be used in the pipeline, enabling you to maintain a slowly changing dimension easily, which otherwise is not easy to maintain. The Slowly Changing Dimension Transformation includes the SCD Wizard, which configures the SCD Transformation and also creates the data flow branches to load the slowly changing dimension with new records, with simple type 1 updates and also updates where history has to be maintained, that is, type 2 updates. Another common scenario in data warehouse loading is the early arriving facts, that is, the measures for which dimension members do not exist at the time of loading. A Slowly Changing Dimension Transformation handles this need by creating a minimal inferred-member record and creates an Inferred Member Updates output to handle the dimension data that arrives in subsequent loading.

Standardizing and Enhancing Data Quality Features

Integration Services includes the following transformations that enable you to perform various operations to standardize data:

- *Character Map Transformation* allows you to perform string functions to string data type columns such as change the case of data.
- *Data Conversion Transformation* allows you to convert data to a different data type.
- *Lookup Transformation* enables you to look up an existing data set to match and standardize the incoming data.
- *Derived Column Transformation* allows you to create new column values or replace the values of existing columns based on expressions. SSIS allows extensive use of expressions and variables and hence enables you to derive required values in quite complex situations.

Integration Services can also clean and de-dupe (eliminate duplications in) data before loading them into the destination. This can be achieved either by using Lookup Transformation (for finding exact matches) or by using Fuzzy Lookup Transformation (for finding fuzzy matches). You can also use both of these transformations in a package by first looking for exact matches and then looking for fuzzy matches to find matches as detailed as you may want. Fuzzy Grouping Transformation groups similar records together and helps you to identify similar records if you want to treat the similar records with the same process, for example, to avoid loading similar records based on your fuzzy grouping criteria. The details of this scenario are covered in Chapter 10.

Converting Data into Meaningful Information

There is no reason to collect and process large volumes of data other than to draw out meaningful information from it. SSIS provides several components and transformations that you can use to draw out meaningful information from raw data. You may need to perform one or more of the following operations to achieve the required results:

- Apply repeating logic to a unit of work in the workflow using *For Loop* or *Foreach Loop* containers
- Convert data format or locale using *Data Conversion Transformation*
- Distribute data by splitting it on data values using a condition
- Use parameters and expressions to build decision logic

- Perform text mining to identify the interesting terms in text related to business in order to improve customer satisfaction, products, or services

Data Consolidation

The data in which you are interested may be stored at various locations such as relational database systems, legacy databases, mainframes, spreadsheets, or even flat files. SSIS helps you to consolidate this data by connecting to the disparate data sources, extracting and bringing the data of interest into the data flow pipeline and then merging this data together. This may sound very easy, but things can get a bit convoluted when you are dealing with different types of data stores that use different data storage technologies with different schema settings. SSIS has a comprehensive set of Data Flow Sources and Data Flow Destinations that can connect to these disparate data stores and extract or load data for you, while the Merge, Merge Join, or Union All Transformations can join multiple data sets together so that all of them can be processed using single pipeline process.

Package Security Features

A comprehensive set of security options available in SSIS enables you to secure your SSIS packages and the metadata specified in various components. The security features provided are:

- Access control
- Encrypting the packages
- Digitally signing the packages using a digital certificate

Depending on where the packages have been deployed, access control methods are provided by the underlying platform. For example, you can control access to packages saved into SQL Server using SQL Server roles for Integration Services, while Windows access control mechanisms are used if the packages are deployed to the file system. Integration Services packages can use various levels of encryption to protect sensitive information such as passwords and connection strings. You can also digitally sign your SSIS packages to establish the authenticity of the packages. Chapter 7 covers these security features in detail.

Service-Oriented Architecture

SSIS provides support for Service-Oriented Architecture (SOA) through a combination of HTTP connection manager, Web Service task, and XML source. These can be used together to pull XML data from URLs into the data flow.

SSIS Package as a Data Source

SSIS provides a DataReader destination that enables a SSIS package to be used as a data source. When you use a DataReader destination in your SSIS package, you effectively convert your SSIS package into an on-demand data source that can provide integrated, transformed, and cleansed data from multiple data sources to an external application such as SQL Server Reporting Services. You can also use this feature to connect to multiple web services, extract RSS feeds, and combine and identify interesting articles to be fed back to the application on demand. This is a very unique and powerful feature that places SSIS far ahead of other traditional ETL tools.

Programmability

SSIS provides a rich set of APIs in a native and managed form that enables you not only to extend the functionality provided by preconfigured components but also to develop new custom components using C++ or other languages supported by the .NET Framework (such as Visual C#, Visual Basic 2008). With the provision of this functionality, you can include your already-developed legacy applications or thirdparty components in SSIS processes, or you can program and extend SSIS packages by scripting or by writing your own custom components. These custom components can be developed for both Control Flow and Data Flow environments and can be included in an SSIS toolset quite easily so as to be reused in enterprise-wide development projects. Examples of custom components could be Control Flow tasks, Data Flow Sources, Data Flow Destinations, Data Flow Transformations, Log providers, Connection Managers, and so on.

Scripting

SSIS also provides scripting components in both Control Flow and Data Flow environments to allow you to add ad hoc functionality quickly within your SSIS packages using Microsoft Visual Basic 2008 and Microsoft Visual C# 2008.

Easy Management of SSIS Packages

SSIS is designed with high development productivity, easy management, and fast debugging in mind. Some of the features that contribute to achieve these goals are listed here:

- Integration Services is installed as a Microsoft Windows service, which provides storage and management functions and displays running packages for SSIS packages.
- Integration Services provides rich logging features that allow you to choose the type of information you want to log at the package level or at the component level using one of the five built-in log providers, and if you're not happy with them, you have the flexibility to custom-code one that suits more to your requirements.
- If your package fails halfway through processing, you do not need to do all the work again. Integration Services has a restart capability that allows a failed package to be restarted from the point of failure rather than from the beginning, thus saving you time.
- Integration Services provides SSIS Service and SSIS Pipeline performance objects that include a set of performance counters for monitoring the running instances of packages and the performance of the data flow pipeline. Using these counters, you can fine-tune the performance of your packages.
- SSIS provides several utilities and wizards such as the dtexec utility, dtutil utility, Execute Package Utility, Data Profiler Viewer, Package Migration Wizard, and Query Builder that help you perform the work easily and quickly.
- SSIS provides the SQL Server Import and Export Wizard that lets you quickly copy data from a source to a destination. The packages saved with SQL Server Import and Export Wizard can later be opened in BIDS and extended. You will study the SQL Server Import and Export Wizard in Chapter 2.

Automating Administrative Tasks

SSIS can automate many administrative tasks such as backing up and restoring, copying SQL server databases and objects, loading data and processing SQL Server Analysis objects when you create the required logic in a package and schedule it using SQL Server agent job or any other scheduling agent.

Easy Deployment Features

You can enable package configurations to update properties of package components dynamically with the Package Configuration Wizard and deploy packages from development to testing and to production environments easily and quickly with the Deployment Utility. You will study deployment features and facilities in Chapter 11.

Legacy Support Features

You can install SQL Server 2008 Integration Services side by side with SQL Server 2005 Integration Services and SQL Server 2000 Data Transformation Services. Alternatively, you can choose to upgrade the legacy DTS 2000 or SSIS 2005 versions to the SQL Server 2008 version. Various installation options are discussed later in this chapter, when you will do an SSIS 2008 installation Hands-On. But here it is important to understand that SQL Server 2008 is a point upgrade of SQL Server 2005 Integration Services, though enough changes have been made that you cannot modify or administer packages developed in one version from the other version. However, runtime support has been maintained in SQL Server 2008; for example, you can run SSIS 2005 packages in SQL Server 2008 using BIDS, dtexec (2008 version), or SQL Server Agent. See Chapter 14 for more details on implications of choosing to upgrade or running the side-by-side option. DTS 2000 has been deprecated in SQL Server 2008 and is not included in the default installation option. The following section describes it in more detail. DTS packages can still be used with Integration Services, as legacy support still exists, but you will have to install DTS support components separately. SSIS 2008 also provides tools to migrate your DTS packages to Integration Services to enable you to take advantage of new features. You will study backward compatibility features and migration support provided in SQL Server 2008 in Chapter 14.

What's New in Integration Services 2008

While Integration Services 2005 was not only a complete rewrite of DTS 2000 but also a new product of its kind, SSIS 2008 contains several enhancements to increase performance and productivity. In this section, you will study the major enhancements that have been included in SSIS 2008, while the others will be covered wherever we come across them. If you're new to Integration Services, you can skip this section, as this may not provide you relevant information. However, if

you've worked with SSIS 2005, this section will acquaint you with the changes that have been made to Integration Services 2008.

Better Lookup

Most data integration or data loading projects need to perform lookups against already-loaded or standardized data stores. The lookup operation has been very popular with developers since Data Transformation Services first introduced this task. Integration Services 2008 has greatly improved the usability and performance of this component over its predecessor, SSIS 2005. The continuous growth in data volume and the increased complexity of BI requirements has resulted in more and more usage of lookup operations. As Integration Services 2005 was becoming a more appealing choice in data warehouses than ever, a better performing lookup was much needed because of the limited time-window available to such operations. Think of a practical scenario: if you have to load several flat files daily, it is most likely that you will be keeping your data flow task within a looping logic. And if you're using a Lookup Transformation in a data flow task, the lookup or reference data will be loaded every time the Lookup Transformation is used within the loop in Integration Services 2005. If your reference data doesn't change that often, then this recurring loading of reference data is a redundant operation and can cause unnecessary delays. Integration Services 2008 provides a much-improved Lookup Transformation that allows you to use a cache for the reference data set, and you don't need to perform a lookup against the reference data source repeatedly as you do in SSIS 2005. You can use an in-memory cache that is built before the Lookup Transformation runs and remains in memory until the package execution completes. This in-memory lookup cache can be created in the same data flow or a separate one and used over and over until the reference data set changes, at which time you can refresh the cache again. The ability to prepopulate the cache and to repeatedly use it makes the lookup operation perform much better in this version. And this is not all: you can also extend the use of in-memory cache beyond a package execution by persisting this cache to a cache file. The cache file is a proprietary raw-format file from which the cache data can be loaded into memory much faster than from a data source. Used in this way, a cache file enables you to share the cached reference data between multiple packages. Later, when you study Lookup Transformation in Chapter 10, you will also use a cache file and the other components used to create and use a cached lookup.

Improved ADO NET Components

DataReader Source and DataReader Destination components have been replaced with much improved ADO NET Source and ADO NET Destination components. DataReader adapters in SSIS 2005 allowed you to connect to ADO NET–compliant data stores; however, they were restrictive and could be configured only in an advanced editor. ADO NET adapters, on the other hand, have their own custom UI and look more like OLE DB Adapters, with the only difference being that they cannot use variables in the data access mode property. The enhanced functionality of ADO NET adapters enables SSIS 2008 to connect to ODBC destinations now.

Powerful Scripting

As mentioned earlier, BIDS is now based on VSTA (Visual Studio Tools for Applications), which is a Visual Studio 2008 IDE. This environment benefits both the Script Task and the script component by providing them a new programming IDE and an additional language, C#. In SSIS 2008 you can choose either Visual Basic 2008 or Visual C# 2008 as your preferred language. Replacement of Visual Studio for Applications (VSA) by VSTA has also made it easier to reference many more .NET assemblies and added real power to SSIS scripting.

Extended Import and Export Wizard

The Import and Export Wizard has been made more usable by extending the features it supports. You can now use ADO NET adapters within the Import and Export Wizard and take advantage of other enhancements; for instance, data type mapping information and data type conversions have been made available, along with better control over truncations and flexibility to create multiple data flows if you're dealing with several tables.

Ability to Profile Your Data

Sometimes you will receive data from external sources or from the internal lesserknown systems. You would want to check data quality to decide whether to load such data or not. May be you can build an automatic corrective action for such a data based on its quality. The ability to check quality or profile data is now included in Integration Services. The Data Profiling Task enables you to analyze columns for attributes such as column length distribution, percentage of null values, value distribution, and related statistics. You can actually identify relationship problems among columns by analyzing candidate keys, functional dependencies between columns, or value inclusion based on values in another column. SSIS 2008 provides a Data Profile Viewer application to see the results of Data Profiling Task.

Optimized Thread Allocation

The data flow engine has been optimized to create execution plans at run time. This enables data flow to allocate threads more efficiently and be able to perform better on multiprocessor machines; hence you get your packages processed quicker. You get this performance boost even without doing anything. This is an out-of-the-box improvement.

SSIS Package Upgrade Wizard

To help you upgrade your SSIS 2005 packages to the SSIS 2008 format, a SSIS Package Upgrade Wizard has been provided in this version. Though a SSIS 2005 package can be automatically upgraded to the SSIS 2008 format by opening in BIDS, this is a slow process if you have several packages in your projects. The SSIS Package Upgrade Wizard allows you to select packages from either File System or SQL Server MSDB database stores, select one or many packages at one time to upgrade, and keep a backup of the original packages in case you run into difficulties with upgraded packages.

Taking Advantage of Change Data Capture

The source systems that are used to populate a data warehouse are generally transactional systems hosting LOB applications that need the system not only to be available but also to perform at the best possible level. This virtually leaves only one option: for database developers to load a data warehouse during off-business hours. With more and more businesses using the Internet as a sales and marketing channel, either the off-business hours have reduced drastically or in many cases no off-business hours are left. This leaves very little or no time window for data warehouse processes to pick up the data from the source systems. Until recently, database developers have used triggers or timestamps to capture changed rows; however, the process makes systems complex and reduces the performance.

SQL Server 2008 includes a new feature called Change Data Capture that provides changes—that is, insert, update, and delete activities happening on the SQL Server tables—in a simple relational format in separate change tables and leaves the source systems working at their best. You will use this feature in Chapter 12 while studying the best practices for loading a data warehouse.

Benefiting from T-SQL Merge Statement

SQL Server 2008 includes a new T-SQL statement for performing insert, update, or delete operations on a table based on the differences found in another table. This enables you to perform multiple DML operations in a single statement, resulting in performance improvement due to reduction in the number of times the data is touched in source and target tables. You can use Execute SQL Task to host the MERGE statement and leverage the performance benefit provided by this statement.

Enhanced Debugging

To debug pipeline crashes or deadlocks, you can now use command prompt options with the dtexec and dtutil command prompt utilities to create debug dump files. The options /Dump and /DumpOnError can be used with dtexec to create dump files either on certain events (debug codes) or on any error. The dtutil utility contains only the /Dump option and can create dump files on occurrence of any of the specified codes.

Inclusion of New Date and Time Data Types

Last but definitely not the least, Date and Time data types have been enhanced with introduction of the three new data types:

- **DT_DBTIME2** Includes fractional seconds support over DT_DBTIME
- **DT_DBTIMESTAMP2** Includes larger fractional seconds support over DT_DBTIMESTAMP2
- **DT_DBTIMESTAMPOFFSET** Supports time zone offsets

Where Is DTS in SQL Server 2008?

You might have worked with the DTS provided with SQL Server 2000. DTS is not an independent application in itself; rather, it is tightly bound with SQL Server 2000. DTS is a nice little tool that has provided users with great functionality and components. Some developers have even extended DTS packages by writing custom scripts to the enterprise level. Yet DTS has some inherent shortcomings; for example, it is bound to SQL Server, is not a true ETL tool, has a limited number

of preconfigured tasks and components, offers a single design interface for both workflow and data flow that is limited in extensibility, and has no built-in repeating logic. Although you could fix all these shortcomings by writing a complex script, it wouldn't be easy to maintain and would be a big challenge to develop.

With the launch of SQL Server 2005 Integration Services Microsoft has replaced Data Transformation Services (addressed as DTS 2000 in this book) of SQL Server 2000. One thing you need to understand is that Integration Services is not a point upgrade of DTS rather it will be right to say that it is not an upgrade to DTS at all. The code for Integration Services has been written from scratch, thus, Integration Services has been built from ground up. DTS was deprecated in SQL Server 2005 and now in SQL Server 2008 it has been removed from the default installation process; if you want to install DTS components, you have to choose it manually. Once DTS support components have been installed, you can modify the design or run DTS packages on SQL Server 2008. However, bear in mind that backward compatibility support has been provided to enable developers and organizations to migrate existing DTS packages to Integration Services and not to encourage development of new packages on DTS. You will read more about DTS support and the migration options in Chapter 14 of this book.

Before we move on to next section, I would like to stress a couple of facts again about DTS 2000 and SSIS. SQL Server 2008 Integration Services is not an upgrade to DTS 2000. Integration Services is installed as a Windows service and Integration Services service; it enables you to see the running SSIS packages and manage storage of SSIS packages. DTS 2000 was not a separate Windows service; rather, it was managed under the MSSQLSERVER service instance. Though it is highly recommended that you migrate your DTS 2000 packages to SQL Server 2008 Integration Services to take advantage of the better-performing, more flexible, and better controlled architecture, your existing DTS 2000 packages can still run as is under Integration Services.

Integration Services in SQL Server 2008 Editions

Not all the editions of SQL Server 2008 include Integration Services; in fact only Standard, Developer, Enterprise, and Premium Data Warehouse Editions have Integration Services. However, once you've installed Integration Services, you can use any of the SQL Server editions as a data source or a destination in your SSIS packages. In the following section you will study how Integration Services is spread across various versions of SQL Server 2008.

- **SQL Server 2008 Express Edition** The Express Edition of SQL Server 2008, including its two other siblings, called SQL Server Express with Tools and SQL Server Express with Advanced Services, is an entry-level free edition and does not include Integration Services. SQL Server Express Edition includes SQL Server Import and Export Wizard only. Though you cannot use Integration Services on this edition, you can run DTS packages on an Express Edition SQL Server when you install SQL Server 2000 client tools or DTS redistributable files on the computer. Installing this legacy software will install the DTS run-time engine on the SQL Server Express Edition. DTS 2000 packages can also be modified using SQL Server 2000 client tools. Also, note that the Express Edition doesn't support SQL Server Agent and, hence, your packages can't be scheduled.
- **SQL Server 2008 Web Edition** This is a low-cost SQL Server edition designed to host and support web site databases. As in the SQL Server Express Edition, the Integration Services components are limited to support the Import and Export Wizard only. The DTS 2000 run time can be installed and used as it can with the SQL Server Express Edition.
- **SQL Server 2008 Workgroup Edition** This edition of SQL Server 2008 is targeted to be used as a departmental server that is reliable, robust, and easy to manage. This edition includes the SQL Server Import and Export Wizard, which uses Integration Services to develop simple source-to-destination data movement packages without any transformation logic. Again, Integration Services isn't supported on this server, though basic components of SSIS do exist on this server to support the wizard creating data movement packages. As in earlier-mentioned editions, DTS 2000 support software can also be installed in this edition and used in a similar way. In fact, DTS components can be installed on any edition if required; however, it will be required more on the editions that don't have Integration Services support than the ones that do. The Workgroup Edition gives you a bit more than the Express Edition by enabling you to remotely modify DTS packages using the SQL Server Management Studio, as the Workgroup Edition supports SSMS.
- **SQL Server 2008 Standard Edition** The Standard Edition of SQL Server 2008 is designed for small- to medium-sized organizations that need a complete data management and analysis platform. This edition includes the full power of Integration Services, excluding some high-end components that are considered to be of importance to enterprise operations. The Integration Services service is installed as a Windows service, and BIDS, an Integration Services development environment, is also included. The separation of Standard Edition and Enterprise Edition is only on the basis of high-end components and does not impose any limitations to performance or functionality of components. What you get in Standard Edition works exactly as it would work in Enterprise Edition. The following components have

not been included in this edition, however:

- Data Mining Query Task
 - Data Mining Query Transformation
 - Fuzzy Grouping Transformation
 - Fuzzy Lookup Transformation
 - Term Extraction Transformation
 - Term Lookup Transformation
 - Data Mining Model Training Destination
 - Dimension Processing Destination
 - Partition Processing Destination
- **SQL Server 2008 Enterprise Edition** This most comprehensive edition is targeted to the largest organizations and the most complex requirements. In this edition, Integration Services appears with all its tools, utilities, Tasks, Sources, Transformations, and Destinations. (You will not only study all of these components but will work with most of them throughout this book.)
 - **SQL Server 2008 Developer Edition** This has all the features of the Enterprise Edition.
 - **SQL Server 2008 R2 Premium Editions** With the release of R2, Microsoft has introduced two new premium editions—the Datacenter and Parallel Data Warehouse Editions, which are targeted to large-scale datacenters and data warehouses with advanced BI application requirements. These editions are covered in detail in Chapter 12.

32-Bit Editions vs. 64-Bit Editions

Technology is changing quickly, and every release of a major software platform seems to provide multiple editions and versions that can perform specific tasks. SQL Server 2008 not only introduced various editions as discussed in the preceding section but also has 32-bit and 64-bit flavors. Though SQL Server 2000 was available in a 64-bit edition, it was not a fully loaded edition and ran only on Intel Itanium 64-bit CPUs (IA64). It lacked many key facilities such as SQL Server tools on the 64-bit platform—that is, Enterprise Manager, Query Analyzer, and DTS Designer are 32-bit applications. To manage the 64-bit editions of SQL Server 2000, you must run a separate 32-bit system. Moreover, 64-bit SQL Server 2000 was available in Enterprise Edition only and was a pure 64-bit edition with less facility to switch over.

On the other hand, the SQL Server 2008 64-bit edition is a full-featured edition with all the SQL Server tools and services available on the 64-bit platform, meaning you do not need to maintain a parallel system to manage it. SQL Server 2008 64-bit edition is available for Standard Edition and Enterprise Edition. It can run on both IA64 and x64 platforms and is enhanced to run on Intel and AMD-based 64-bit servers. You can run SQL Server 2008 and its components in 64-bit native mode, or you can run 32-bit SQL Server and 32-bit components in WOW64 mode. SQL Server 2008 provides a complete implementation of Integration Services in the 64-bit edition, though there are minor tweaks here and there. The performance benefits provided by 64-bit systems outweigh the costs and efforts involved, and it is also very simple to switch over to the 64-bit edition. If you're interested in knowing more about SQL Server 2008 Integration Services 64-bit editions, detailed information is provided in Chapter 13, along with discussion of performance and issues involved with it.

Integration Services Architecture

Now you understand the benefits Integration Services provides, with its vast array of features, and also know about various versions and feature sets associated with them. Before we move further and get our hands dirty by starting working with it, it's time to know its architecture. Once you understand its architecture, you will be able to appreciate how the various components perform their jobs to successfully execute an Integration Services package. Let's start with the architecture diagram provided in Microsoft SQL Server 2005 Books Online and shown in [Figure 1-1](#).

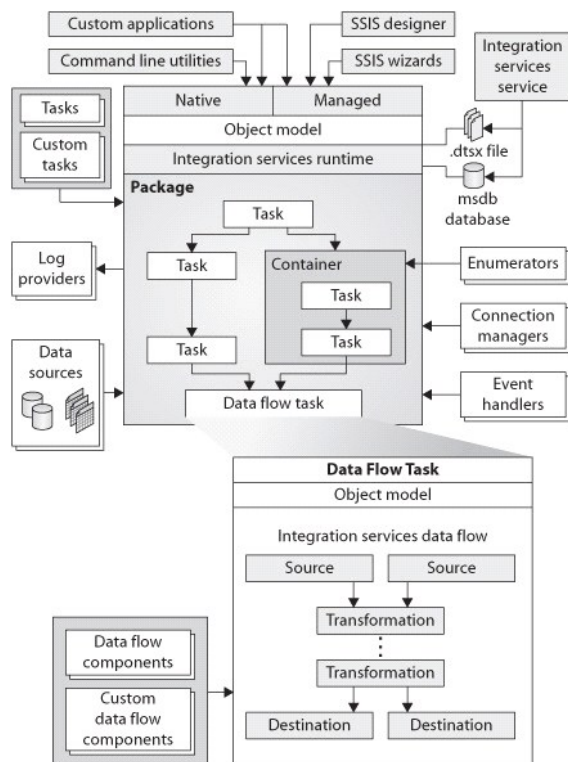


Figure 1-1: Integration Services architecture

Microsoft SQL Server 2008 Integration Services consists of the following four main components:

- Integration Services service
- Integration Services object model
- Integration Services run-time engine
- Integration Services data flow engine

Well, you've read little bit about these components earlier in this chapter as you were going through the features and uses of Integration Services. The following discussion on each of these components and their functions will clarify how Integration Services is architected.

Integration Services Service

Shown on the top-right corner of the architecture diagram (Figure 1-1), the Integration Services service is installed as a Windows service when you specifically choose Integration Services during installation. In the next section's Hands-On exercise, you will see where you make this choice and learn that choosing Integration Services specifically during installations installs other components as well. The Integration Services service allows you to execute Integration Services packages on local or remote computers, stop execution of running packages on local or remote computers, monitor running packages on local or remote computers, and connect to multiple Integration Services servers to manage multiple instances. In Figure 1-1, the Integration Services service points to a .dtsx file and MSDB database, implying that this service can manage SSIS packages stored to a file system or in an MSDB database within SQL Server 2008. The service manages SSIS packages by importing and exporting them from one type of storage location to another. You will learn a lot more about managing packages and their storage locations in Chapter 6.

You can connect to the Integration Services service using that SQL Server Management Studio, as you will do later in this chapter. Generally, with other software components, if the service is stopped, most of the components stop working. This is not true with Integration Services, because it is a component used to monitor running packages and manage their storage. You do not need to have this service running to design and run a package. You can save the newly designed package in BIDS on to the file system or in the SQL Server 2008 MSDB database and then execute it as well. However, you may find it a bit faster when running the Integration Services service, as it caches the metadata of the package and the connections. Also, if you need to monitor and list the packages using the SQL Server Management Studio, the Integration Services service must be running.

Integration Services Object Model

As mentioned earlier, Integration Services is a new product with an object model that supports both native and managed APIs. You can easily use this object model to write custom components such as tasks and transformations using C++ or any common language runtime (CLR)–compliant language. This object model provides easy accessibility for Integration Services tools, command-line utilities, and custom applications as shown on the top section of [Figure 1-1](#). You can also develop custom components, build new packages, load and modify existing packages, and then execute them programmatically. This enables you to automate maintenance and execution of your packages completely. Programming Integration Services is a vast subject and deserves a separate book altogether. A complete discussion is beyond the scope of this book.

Integration Services Run Time

The Integration Services run time provides support for the package, containers, tasks, and event handlers during package execution. It also provides run-time services such as support for logging, breakpoints, connections to data sources and data stores, and transactions. You read earlier that Integration Services has two separate engines: a runtime engine for workflow and another engine for data flow. Basically, the Integration Services run time consists of whatever you configure in the Control Flow tab of BIDS plus the run-time services.

Integration Services Data Flow

As mentioned, the second engine of Integration Services provides services to the data flow within a package. This data flow is also known as the *pipeline* due to the nature of data flowing through various transformations one after another. The Data Flow Task is a unique task provided in the Control Flow tab of BIDS that encapsulates the data flow engine and the data flow components. Integration Services Data Flow consists of one or many Data Flow Sources; none, one, or many Data Flow Transformations; and one or more Data Flow Destinations. The Data Flow engine drives the data out of Data Flow Sources, brings it into pipeline, and lets the Data Flow Transformations perform the aggregations and conversions, merge data streams, conditionally split data into multiple streams, perform lookups, derive columns, and perform several other operations before loading the data into the destination data stores using Data Flow Destinations. You will work with Data Flow components in Chapters 9 and 10 in much detail.

Installing Integration Services

Now is the time to move forward and get our hands dirty by installing Integration Services. But take a few more minutes before we do that to learn about the installation options and the implications of these options. In real life, either you will be installing Integration Services on a clean Windows platform—that is, a system where no current or previous releases of SQL Server are installed—or you will be installing it on a computer that already has SQL Server 2005 Integration Services or Data Transformation Services of SQL Server 2000 installed. You may choose to install SQL Server 2008 Integration Services alongside SQL Server 2005 Integration Services or DTS 2000, or you may choose to upgrade the existing version of SSIS 2005 or DTS 2000. All these options and their implications have been discussed in the following sections.

Installing Integration Services on a Clean System

Most of the production systems are built using this method. Administrators prefer to install SQL Server on a fresh installation of a Windows server to avoid any debugging later on because of some old component on the server that doesn't work properly with SQL Server 2008 Integration Services. I recommend you use a sandbox for doing the Hands-On and install Integration Services clean so that you don't struggle initially with unwanted issues of compatibility or coexistence. You can install Integration Services either by using the SQL Server Installation Wizard or by running setup program from the command prompt.

You'll install Integration Services using the SQL Server Installation Wizard in the following Hands-On exercise. You will be installing SQL Server 2008 database engine and Integration Services together in this exercise; however, note that Integration Services does not require SQL Server in order to work. You can develop packages in Integration Services that connect to mainframes, Oracle or DB2 database servers, and output in flat files without installing SQL Server. A couple of high-end transformations such as the Fuzzy Lookup Transformation and Fuzzy Grouping Transformation need to create temporary tables for processing data in SQL Server and hence require connection to an SQL Server. However, even in this case, you do not need to have SQL Server running on the same local machine where your Integration Services package is designed or executed. Having said that, Integration Services is a fairly independent product and does not require SQL Server to operate; however, installing the SQL Server Database on the same server might prove beneficial, as

most SSIS packages need to be run as SQL Server Agent jobs, which is a database engine feature.

Hands-On: Installing SQL Server 2008 Integration Services

This is your first Hands-On in which you will install SQL Server 2008 Integration Services using the SQL Server Installation Wizard on a clean system.

Method

It is important that you follow this process step-by-step, as this installation will be used throughout this book to create and run Integration Services projects. If you do not have SQL Server 2008 Enterprise Edition or Development Edition software, you can download the SQL Server 2008 Enterprise Evaluation Edition from Microsoft's download web site. This version is valid for 180 days and can be used for trial purposes. Details on non-Integration Services installation options are not covered here and are beyond the scope of this book. Refer to Microsoft SQL Server 2008 Books Online for more details on these installation options.

Exercise (Running the SQL Server Installation Wizard)

Load SQL Server 2008 DVD media in your computer's DVD drive and start the installation as follows:

1. After you load the DVD, the Autorun feature will open the Start screen, which displays various options. If Autorun doesn't do this, browse the DVD and run setup.exe from the root folder. Click the Installation hyperlink from the left sidebar and choose a new SQL Server stand-alone installation or else choose to add features to an existing installation to start the installation.
2. Setup first installs the .NET Framework 3.5 SP1 if it is not already installed. Some versions of Windows Server may require different versions of .NET Framework. Accept the License Agreement and click Install. The Installation Wizard will install .NET Framework, the SQL Server Native Client, and setup support files. Once installation completes, click Exit and the setup installs hot fixes for the operating system if it needs any. Click Finish to complete the preinstallation phase; it may require a restart to continue installation. After restart, again run setup.exe from the installation DVD and choose the New SQL Server installation link.
3. The installation program performs Setup Support Rules checks and lists pass, failure, and warning messages. Click OK to proceed further. On the Setup Support Files screen, click Install to install the required setup support files.
4. On the Feature Selection screen, choose Database Engine Services from the Instance Features section, choose Business Intelligence Development Studio, Integration Services, Client Tools Backwards Compatibility, Client Tools SDK, SQL Server Books Online, and Management Tools—Complete from the Shared Features section as shown in [Figure 1-2](#).

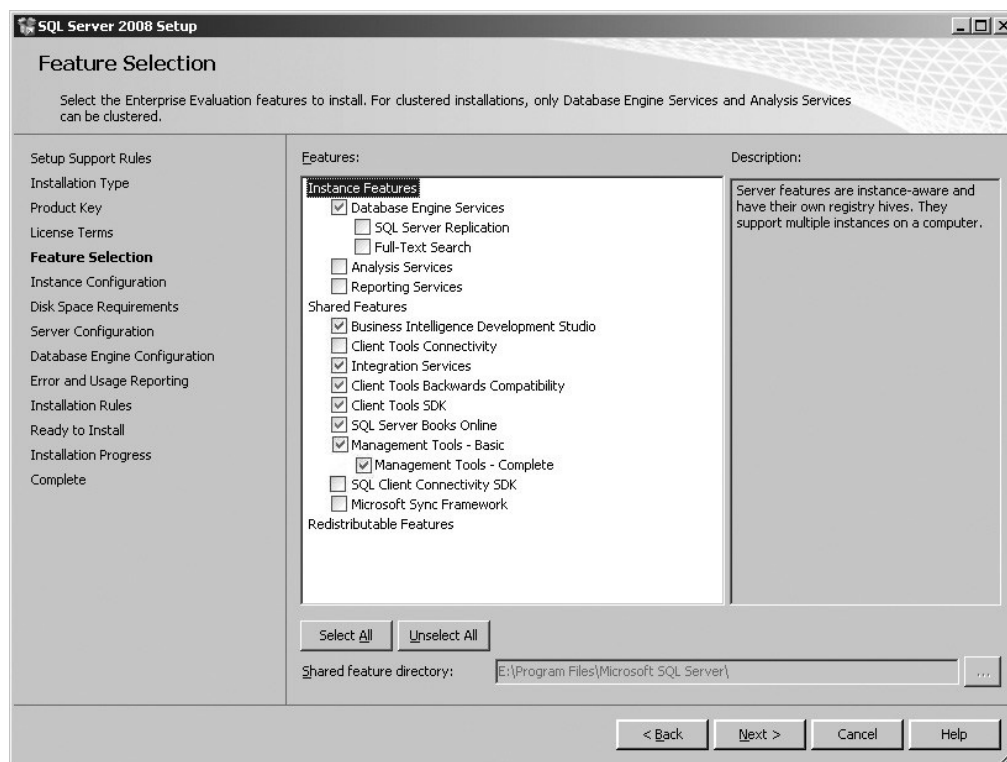


Figure 1-2: Feature Selection for Integration Services

This is the most important step in the installation process, as you choose to install Integration Services here. However, even if you do not specifically select Integration Services to install, some components of Integration Services will still be installed because of they are needed to perform specific functions for other selected components. For example, the SQL Server Import and Export Wizard will be installed when you install SQL Server Database Services only. Also, most of the tasks and transformations will be available to you for developing your packages when you install Business Intelligence Development Studio without selecting Integration Services. Bear in mind that Integration Services is not required for developing and executing packages within BIDS; however, to run packages outside the development environment, you do need to choose Integration Services specifically at this step. Integration Services is installed as a Windows service and helps to manage storage of SSIS packages in SQL Server or the file system using SQL Server Management Studio and enables you to monitor running packages in local as well as remote SSIS instances. Another benefit of installing Integration Services service is that it caches metadata of package components and speeds up loading of packages in Business Intelligence Development Studio. Selecting Integration Service also installs the ActiveX Script task and the DTS Package Migration Wizard.

Also, note that the Integration Services is not listed under Instance Features section. This means that you cannot install more than one instance of Integration Services. Though you don't have multiple instances of Integration Services on the server, it is instance aware. That is, it can connect to any instance of SQL Server and is not tied to a particular SQL Server instance. All you have to do is modify the Integration Services Configuration file to connect to a different SQL Server instance to store packages. More on this topic is covered in Chapter 6 of the book.

As mentioned earlier, Business Intelligence Development Studio is the designing tool for your packages, and selecting this installs the 32-bit design environment along with most of the tasks and transformations. Management Tools installs SQL Server Management Studio and is used to connect to Integration Services, manage storage of packages in the MSDB database and the file system, and monitor running packages.

Selecting Client Tools Backward Compatibility feature installs legacy support components and the Execute DTS 2000 Package task to enable you to run your DTS 2000 packages inside the Integration Services 2008 package. If you want the DTS 2000 run-time environment to be installed, you must install it separately. Chapter 14 covers more details on this subject. The Client Tools SDK will be required to install custom-developed managed assemblies for Integration Services, and finally, SQL Server Books Online is the documentation installation feature.

Select Shared Feature Directory and click Next.

5. Confirm that on the Instance Configuration page, Default Instance is selected and MESQLSERVER is specified in the

Instance ID. Click Next. Again click Next to confirm the Disk Space Requirements.

6. Specify Account Name and password in the Server Configuration page. By default, Integration Service installs with NT AUTHORITY\NETWORK SERVICE account. It is recommended that you choose an appropriate domain account with minimum required permissions assigned. Proceed next to the Database Engine Configuration page to choose an authentication mode. Choose Mixed Mode and specify a password for the built-in system administrator account. Click Add Current User to add yourself to the administrators group and click Next three times to reach to Ready To Install page.
7. Review the options on the Ready To Install screen and press Install when ready.
8. You will see the installation of SQL Server 2008 components in the Installation Progress screen. This may take about 15 to 30 minutes to complete on a fast machine.
9. When the process completes, you will see the installation summary log in the Complete screen. Click Close to close the Installation Wizard. If prompted, restart your computer to complete the installation process.

Review

You installed the SQL Server 2008 database engine default instance and SQL Server Integration Services in the preceding exercise. In real life, you'll be installing Integration Services on 64-bit servers, as they are becoming more affordable and prevalent these days. Note that the 64-bit version of SQL Server 2008 software installs all the 64-bit components by default. However, BIDS is a 32-bit development environment for Integration Services and selecting this feature installs a 32-bit version of Integration Services tools, enabling you to run your packages in 32-bit mode on a 64-bit server. Also, BIDS is not supported on the 64-bit Itanium operating system and hence is not installed on Itanium servers. You can now check the programs that have been installed via the Control Panel. Also, open the Services console from the Administrative Tools and note the SQL Server 2008 services have been installed along with SQL Server Integration Services 10.0 service. Now you can play around with the software components installed by the Installation Wizard in the Programs group to make yourself acquainted with various components if you haven't had a chance to look at SQL Server 2008 up until now.

Installing Integration Services from the Command Prompt

Command prompt installation can help you roll out Installation to a team, install multiple nodes of a failover cluster, or use scripted installation files as a backup plan in case the worst happens. You can install the Integration Services by executing setup.exe along with parameters from the command prompt on a local or remote server. The parameters and their values can be specified either directly in the command or by use of an .ini file. The parameter-value pairs that are relevant for installing Integration Services and its components using a command prompt are as follows:

- **Action** This is a required parameter that specifies the installation type—install, upgrade, or repair. See Books Online for more parameter values.
- **Features** Indicates the SQL Server components to be installed, for instance, (SQL) for Database Engine, (IS) for Integration Services. Other options are (AS) for Analysis Services, (RS) for Reporting Services, and (Tools) for client tools.
- **ISSVCAccount** This is a required parameter option that specifies the service account for the Integration Services service.
- **ISSVCPassword** Specify a password for the ISSVCAccount using this option.
- **ISSVCStartupType** This is an optional parameter to specify the startup type for the service: automatic, manual, or disabled.

For example, if you want to install the SQL Server Database Engine, Integration Services, and client tools and online documentation, the syntax for the command will be something like this:

```
setup.exe /q /ACTION=install /FEATURES=IS /ISSVCAccount="DomainName\
UserName" /ISSVCPASSWORD="StrongPassword"
```

To know about more available options, refer to Microsoft SQL Server 2008 Books Online.

Installing Side by Side

If you already have SQL Server 2005 Integration Services or SQL Server 2000 Data Transformation Services installed on a computer and don't want to remove or upgrade them, you can still install SQL Server 2008 Integration Services alongside them. They all can coexist on the same computer because all three have different execution and design environments. You may wonder, but in fact the SQL Server 2008 Integration Services has a different designer than its predecessor; for instance, BIDS in 2008 is built on a different architecture than BIDS 2005. Though they can coexist, there are some considerations for you to keep in mind when you're working with multiple versions of Integration Services.

SQL Server 2008 Integration Services has been optimized for performance over its previous version, SSIS 2005. While doing that, the development team at Microsoft has also made some underlying changes such as replacing the word "dts" with "ssis" from several places. As you can expect, this will mean that the code Integration Services works with in the 2005 version will most likely not work in the 2008 version. One such change affects storage of SSIS packages in SQL Server. The sysdtspackages90 table used in SQL Server 2005 to store packages in the MSDB database has been changed to the sysssispackages table in SQL Server 2008. It isn't hard to imagine that an Integration Services version won't be able to access packages across both versions due to different storage tables.

This also means that you cannot store packages in the MSDB database of one version that have been developed in another version of Integration Services. You must stick to the same version to open and modify packages using Business Intelligence Development Studio. To clarify a bit more, we have a new version of Business Intelligence Development Studio in SQL Server 2008, which is based on Visual Studio 2008. BIDS also has a new scripting environment built around Visual Studio Tools for Applications (VSTA), replacing the earlier environment of Visual Studio for Applications (VSA) used in BIDS 2005. These underlying changes enable you to install BIDS 2008 side by side with BIDS 2005. However, this leaves BIDS 2008 unable to save packages in SSIS 2005 format. You can load and run packages in BIDS 2008 that have been developed in BIDS 2005; however, loading BIDS 2008 converts these packages into SSIS 2008 format and hence runs the package in 2008 environment. You can save this package in SSIS 2008 format but not in SSIS 2005. Hence if you want to modify your SSIS 2005 packages and want to keep them in 2005 format, you have to use BIDS 2005.

On the other hand, BIDS 2005 cannot load the higher version—i.e., SSIS 2008, packages at all. For similar reasons, you have a new version of the dtexec utility. While dtexec can run both SSIS 2008 and SSIS 2005 packages, it actually executes only SSIS 2008 format packages, as it converts SSIS 2005 packages temporarily to SSIS 2008 format before execution. This means that if your SSIS 2005 package can't be converted to SSIS 2008 format by dtexec, it can't be run. So, you must be a bit careful when working with the packages, as it may require you to keep multiple versions of the same package for SSIS 2005 and SSIS 2008.

On the other hand, things are rather simple, as Integration Services can coexist with DTS 2000 without any issues and there are no interoperability issues as they are totally different in usability, operability, and many other respects. While Integration Services provides wizards and tools to migrate from DTS 2000 to its latest version, it still offers run-time support to run DTS 2000 packages as is. The Execute DTS 2000 Package task allows you to run a DTS 2000 package inside an Integration Services package. This task is not installed by default; you must choose the Client Tools Backward Compatibility feature during the Integration Services installation process.

Upgrading to SQL Server 2008 Integration Services

You can upgrade from SQL Server 2000 Data Transformation Services or SQL Server 2005 Integration Services. Both the Installation Wizard and the command prompt installation provides options to upgrade. With the Installation Wizard, select the Upgrade from SQL Server 2000 or SQL Server 2005 option, but for a command prompt installation, specify the /ACTION=upgrade option in the parameters.

When upgrading SQL Server 2005 Integration Services, you can upgrade either Database Engine and Integration Services together or just the Integration Services or just the Database Engine. The easiest option is to upgrade both Database Engine and Integration Services together when both are on the same computer. This option offers the fewest issues after upgrade, as the MSDB tables that store packages, metadata, and log information are moved to Integration Services 2008 format and existing tables are removed after the upgrade. There are more things for you to do while upgrading that are discussed in detail in Chapter 14.

Upgrading Data Transformation Services to Integration Services 2008 is not that straightforward, though SQL Server 2008 provides tools to upgrade. This is due to major differences in the architecture of DTS 2000 and Integration Services. Chapter 14 discusses all the options that you have for upgrading your existing DTS 2000 packages to Integration Services.

Business Intelligence Development Studio

Business Intelligence Development Studio is designed for developing business intelligence solutions, including Integration

Services packages, Reporting Services reports, and Analysis Services cubes and OLAP databases. BIDS is built on Visual Studio 2008, which allows you to design, build, test, execute, deploy, and extend Integration Services packages in an integrated development environment. Because of Visual Studio 2008 integration, BIDS provides the advantage of having integrated development features, a debugger, integrated help, and an integrated source control (such as Visual Studio Team Foundation Server or Visual SourceSafe) environment. You can use the same techniques to develop and store Analysis Services, Reporting Services, and Integration Services projects with BIDS. You can also develop a solution using BIDS that can have multiple Integration Services, Analysis Services, and Reporting Services projects.

BIDS is based on usual application design philosophy of solutions and projects. This provides lots of benefits, including the following:

- You don't need to have an SQL Server to develop an Integration Services package as was the case with DTS 2000, which required SQL Server 2000 Enterprise Manager to design and develop a package. Integration Services development and execution environments are fairly independent from the SQL Server Engine; however, the presence of SQL Server on the same server where the Integration Services service is running offers additional benefits such as ability to store packages in an MSDb database and to schedule running of SSIS packages using the SQL Server Agent.
- Your development gets an organized structure so that a number of projects can be run under the context of one solution and these projects further can be of different types such as Integration Services, Reporting Services, Analysis Services, or a C# class library project. Integration Services projects can contain data sources, data source views, SSIS packages, and other miscellaneous files. Availability of all of the support DDL and DML files at one place makes deployment a very easy task.
- Direct integration with Source Control servers such as the Visual Studio Team Foundation Server or Visual SourceSafe or one of many third-party source control servers facilitates immediate check-in whenever changes are made to projects. This feature enables a team to work on different parts of a project at the same time. BIDS doesn't work directly on packages stored in SQL Server, so during development, you should be saving your packages in the file system and checking in whenever making changes or on daily basis or on completion of a development cycle. For packages stored in SQL Server, BIDS imports them first by creating a copy on the file system and only then allows you to make changes to your code. After making changes, you will need to save those packages back into SQL Server. Hence, it is easier and better to work with the file system in development environment.
- Being built on the .NET Framework, BIDS makes it very easy to develop Custom Tasks or Components using .NET languages such as C# or Visual Basic 2008. As mentioned earlier, these custom development projects can also reside under the same solution.

The BIDS environment consists of various windows. Among the main windows are SSIS Designer, Solution Explorer, Toolbox, Variables, and Properties, in addition to other windows, such as Output, Error List, and Task List. All these windows can be docked anywhere and can be tabbed on to the main Designer window, set to autohide, or closed. These features provide a fantastic UI configuration feature that allows developers to customize their environment (to free up working space) and boost productivity.

Let's take a closer look at BIDS. In the following Hands-On exercise, you will create a blank Integration Services project and will learn various aspects of this tool.

Hands-On: Creating a Blank Integration Services Project

The objective of this exercise is to create your first blank Integration Services project and study various aspects of BIDS while working with it. You will use this project in Chapter 2 to add packages to it and take it further.

Exercise (Creating a Blank Integration Services Project)

In this part, you will learn about the Integration Services development environment within the Business Intelligence Development Studio while creating a blank project.

1. Choose Start | All Programs | Microsoft SQL Server 2008 and then click SQL Server Business Intelligence Development Studio to open this tool.
2. When the BIDS screen appears, choose File | New and then click Project. Alternatively, you can create a new project by clicking the Project URL next to the Create in Recent Projects section on the Start page. This will open the New Project dialog box, in which the Business Intelligence Projects option is selected by default in Project Types. In the

Templates pane, select Integration Services Project and then fill in the following details in the fields provided in the lower section of the dialog box (see [Figure 1-3](#)).

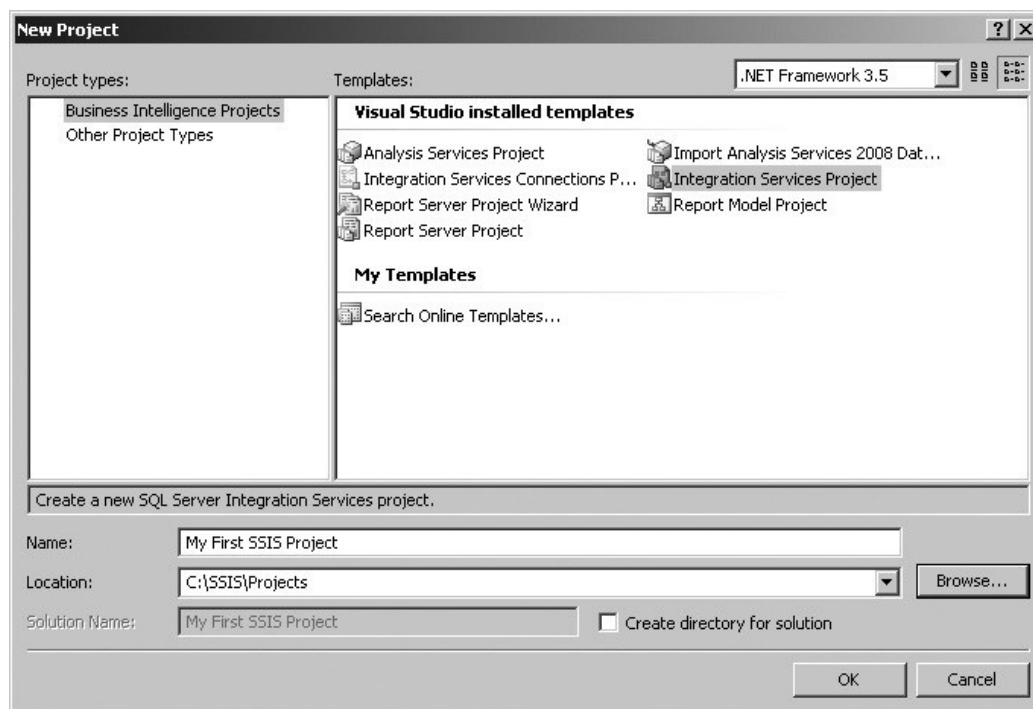


Figure 1-3: Templates for creating new projects in BIDS

Name My First SSIS Project

Location C:\SSIS\Projects

Do not select the check box for Create Directory For Solution, as we do not want a parent folder for a solution to be created in this instance. Click OK when you have filled in these details to create an Integration Services Project.

3. The BIDS will create a blank Integration Services Project and will show you blank designer surface and the Package.dtsx SSIS package created in the Solution Explorer window as shown in [Figure 1-4](#).

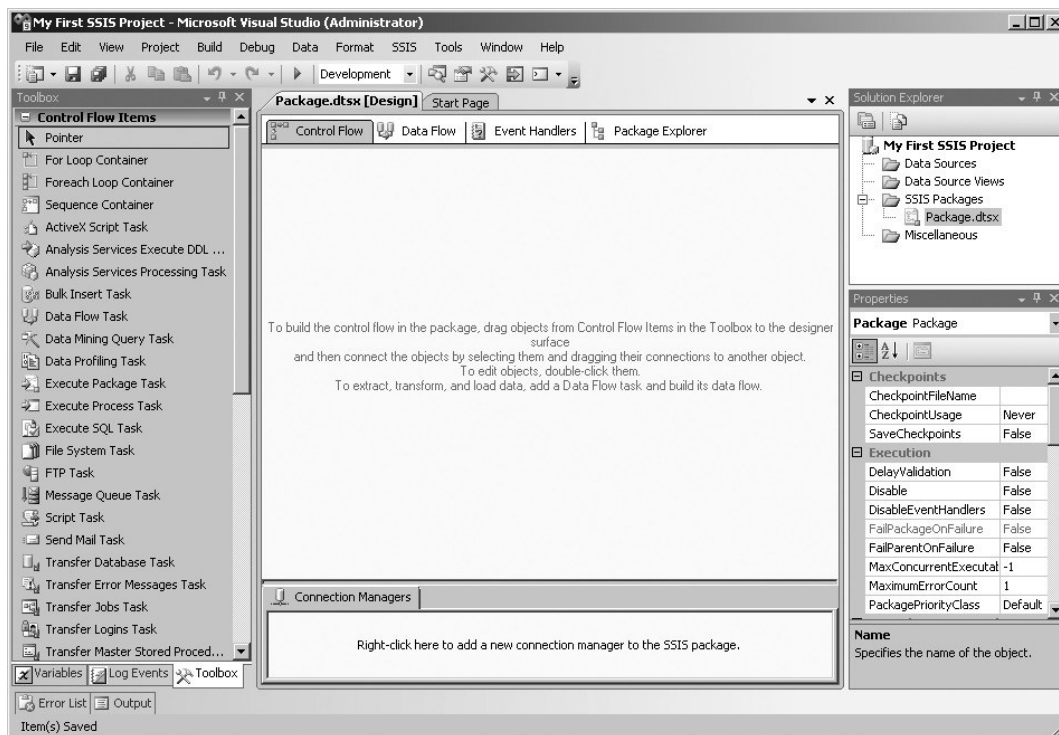


Figure 1-4: Your first blank Integration Services project

Let's take this opportunity to learn a bit more about the windows that make up an Integration Services project in the BI Development Studio.

SSIS Designer

Notice the central area in the BIDS that has four tabs: Control Flow, Data Flow, and Event Handlers along with the design surfaces, and one Package Explorer tab to show components used in the package in a hierarchical structure; it is called the SSIS Designer. These are the design surfaces for developing Integration Services packages, while BIDS provides different designer surfaces for other BI project types such as Reporting Services and Analysis Services projects. The SSIS Designer provides a graphical view of objects that make data movement, workflow, and Integration Services development work possible with minimal or no programming at all.

At the top of the designer, the name of the package, Package.dtsx, is shown. It has three design surfaces—Control Flow, Data Flow, and Event Handlers—along with one Package Explorer tab to show components used in the package in a hierarchical structure. You can have multiple packages open at one time, each with its own design surfaces displayed in a tabular form. The layout of SSIS Designer displaying design surfaces in tab groups makes it particularly useful for handling large projects with many packages open at the same time. There are other windows in the BIDS such as Solution Explorer, Toolbox, Properties, Error List, and Output. Not all the windows display at all times.

The workspace in BI Development Studio can be completely managed by the user as all the windows other than SSIS Designer can be docked anywhere in the working area, can float, can be tabbed in the SSIS Designer, can be set to autohide, or can be completely hidden or closed. If you click the down arrow button displayed in the Solution Explorer's menu bar, you will see these options. The other two buttons—a pushpin and a cross—are used to set windows to autohide or hide (close) respectively. By default, the Toolbox window is in autohide mode, hanging on the left side of the SSIS Designer, and the Solution Explorer is docked on the right side of the SSIS Designer. This is also shown in the [Figure 1-4](#).

Solution Explorer

The Solution Explorer window provides an organized view of projects and files associated with them. The active solution lies at the top of the logical container and contains one or more different types of projects below it. You can add various other projects to a solution such as analysis services projects or reporting services projects to organize your different units of work all in one place.

Because the different projects store files in different ways, the Solution Explorer window does not reflect the physical storage of files. After you create an Integration Services project, you can attach it to a solution later on. You can attach a project or a file to a solution, but a project allows you to attach only files. You add files or projects by right-clicking the

solution or the project in the Solution Explorer window and choosing Add from the context menu.

Properties Window

The Properties window is located at the lower right of the BIDS interface (see [Figure 1-4](#)). This is also a context-sensitive window and shows the properties of the item or object you have selected. Having the Properties window open during design time is a great time-saver. Like other windows, you can close the window or move it around on the desktop. You can open this window again from the View menu or by pressing **F4**.

Toolbox

You probably noticed a tabbed window on the left side of the SSIS Designer (see [Figure 1-4](#)) called the Toolbox. The Toolbox contains preconfigured tasks provided by Integration Services in the Control Flow tab, and the Data Flow tab has Data Flow Sources, Data Flow Transformations, and Data Flow Destinations. The tasks are organized in two sections in the Toolbox window, which shows the tasks that are relevant to the project type you are working with. When you are in the Control Flow tab of the Designer window and open the Toolbox, you will see Control Flow Items; if you're in the Data Flow tab of the Designer, you will see Data Flow Sources, Data Flow Transformations, and Data Flow Destinations sections.

Control Flow Tab

The Control Flow tab consists of the tasks, arranged in the order in which they are performed—that is, precedence constraints and the looping structure defined by looping structure containers For Loop, Foreach Loop, and Sequence. You can draw objects on the graphical surface of the Designer and link them with other objects by dragging and dropping an arrow that extends from one object to another. This arrow signifies a precedence constraint and can be of type OnSuccess, which appears in green; OnCompletion, which appears in blue; and OnFailure, which appears in red. By defining the tasks and the precedence constraints between them, you design the control flow of your package and thus define the workflow in your package. You can logically group tasks to simplify complex packages and annotate them with text boxes to provide an explanation of the task. You will study more about precedence constraints and other control flow components in Chapters 3 to 5.

Data Flow Tab

The Data Flow Designer consists of the source adapters that extract data from heterogeneous data sources; the transformations that modify, aggregate, or extend data; and the destination adapters that load the transformed data into the final data stores. A package must include at least one Data Flow task in order to implement a data flow. You can create multiple Data Flow tasks in a package and create multiple data flows within a Data Flow task. The data is extracted from a source using source adapters and loaded to the destination using destination adapters. In between source adapters and destinations adapters you use transformations to modify, aggregate, and extend column data and to apply business logic to convert data. The flow of data from source to destination with transformations along the way is linked together with green or red lines called *data flow paths*. Adding data viewers to a path enables you to see the data as it moves from source to destination. This helps you debug and locate a troublemaking component that is converting data incorrectly. Extensive error handling can be included in the data flow task; for instance, error rows can be routed to a different destination whenever there is a row-level fault, to capture, analyze, and maybe correct and feed back to the main data flow.

Event Handlers Tab

You can extend package functionality by using *event handlers*, which are helpful in managing packages at run time. Event handlers are like subpackages waiting for the events to be raised so that they can come to life. They are powerful tools that can extend the package functionality greatly when properly implemented. Event handlers are created for the packages, Foreach Loop container, For Loop container, and Sequence container, as well as for the tasks in the same way as you create packages. Once created, event handlers can be explored in the Package Explorer tab by first expanding the Package and then expanding the Executables and finally expanding the Event Handlers node.

Package Explorer Tab

The Package Explorer represents the container hierarchy of the SSIS object model and lists all the package objects. This is the interface through which you can execute a package and monitor the running package. When you click the Package Explorer tab, your package appears at the top of the hierarchy. Click the Package to expand and expose the Variables, Executables, Precedence Constraints, Event Handlers, Connection Managers, and Log Providers objects. Event handlers are members of the Event Handlers collection, and all executables include this collection. When you create an event handler, SSIS adds the event handler to the Event Handlers collection. The Package Explorer tab in SSIS Designer lists the event handlers for an executable. Expand the Event Handlers node to view the event handlers that executable uses.

Progress Tab or Execution Result Tab

This tab doesn't show up during design time when you are still developing the package. When you run an Integration Services package in BIDS, a Progress tab will appear within the SSIS Designer. This Progress tab converts to an Execution Result tab once the package execution is completed and you've switched back to the design mode. Integration Services writes extensive information in this tab while executing a package. All sorts of information such as validation of tasks, start time, finish time, warnings, errors, failure messages, suggestions, and execution details, including the number of rows affected, are all written here. Once you become more familiar with Integration Services, you may find that this tab contains a bit more information than what you would like to see. For these reasons, you may find that Output window contains more concise and relevant information. However, if you are new to Integration Services, this tab provides some interesting information, such as the optimization hints in case more columns have been extracted than used in the data flow, and you may find these kinds of messages a learning aid to design better and more efficient packages.

Connection Managers Area

In this area, you add various connection managers, depending on the requirements of your package. For example, if your package needs to connect to a flat file, you will add a Flat File connection manager; if your package needs to connect to an SQL Server database, you will add an OLE DB connection manager.

BIDS provides many other windows for additional informational purposes. Some of these are described next, while others will be introduced wherever used.

Code Window

You can see the code of a package or an object in the SSIS Designer. To see this, go to the Solution Explorer window, right-click Package.dtsx, and choose View Code from the context menu. An additional tab on the SSIS Designer surface appears with a listing of code in XML form.

Task List Window

In the Task List window, you can add notes for descriptive purposes or as a follow-up for later development; you can also organize and manage the building of your application. To open the task list, choose View | Task List. These tasks can be filtered and sorted based on the predefined options provided.

Output Window

The Output window displays informational messages, including errors that occur during building, deployment, or run time. Third-party external tools can list their output to the Output window as well. Press the green triangle (play) button on the Standard toolbar to debug the package. The package completes execution immediately, as there is no task within the package, and you can see Output window open at the bottom of the screen among other windows. If you don't see it there, go to View menu and click Output from the option list to open it. Note that the Output window shows package start, finish, and success messages. The Output window has multiple panes to show information specific to the process that is sending informational messages. These panes can be accessed from the Show Output From drop-down box in the Output window. Click the down arrow and select Build to display messages from the build process.

Error List Window

This window provides detailed description of validation errors during design time. It also shows errors, warnings, and messages for the package you are developing. To open this window, choose View | Error List.

Locals Window

This window displays information about the local expressions and is available only when the package is in debug mode. When you use a breakpoint to stop package execution to debug, you can come to the Locals window and see values of variables at that particular moment. This is very useful and effective when you're debugging your package.

Watch Window

Like the Locals window, the Watch window is also available during debug mode and displays information only about the expressions that you've selected. For example, you can watch a variable to see how its value changes as the package execution progresses. You can open up to four watch windows.

1. Stop debugging of the package by pressing SHIFT-F5.
2. Using Windows Explorer, go to the C:\SSIS\Projects folder and note that various files have been created under the

My First SSIS Project folder. Note the extension and type of these files, as each one of them represents a different function in the Integration Services project.

- The *.sln file is the main file for a solution or project and contains solution configuration settings and the list of projects the solution contains.
- The *.dtproj files are similar files, but contain information for project configurations and the items such as packages they contain.
- The *.database file contains information required for internal use by BIDS.
- The .dtsx files are the code files for your packages. These files are independent and self-contained and are not hard-bound with the solution in which they are created. This means that they can be freely copied between projects and folders.
- The file *.suo, called Visual Studio Solution User Options, and the file *.dtproj.user, called Visual Studio Project User Options, are the two user settings files used by Integration Services at the solution and project levels.
- The bin folder keeps the backups of the previous build versions of packages.

Congratulations! You have completed your first Integration Services project Hands-On.

Review

You've had your first encounter with the BIDS and created a blank Integration Services Project. Although you haven't developed much in the package, you have understood enough about the environment and the package itself that you're almost ready to get creative and develop a working package.

SQL Server Management Studio

SQL Server Management Studio is the main tool used to manage SQL Server databases and run T-SQL queries against the tables and the views. This tool also enables you to connect to the Integration Services service and perform management operations such as run packages, monitor running packages, manage package storage on the file system as well as the MSDB database, import and export packages from one storage area to another, assign Package Roles to control access, and upgrade packages. Let's start using this tool straightaway with a very simple Hands-On.

Hands-On: Connecting to Integration Services Service

In this exercise, you will connect to Integration Services and will also explore where DTS 2000 packages can be managed within SQL Server Management Studio.

Exercise (Using SQL Server Management Studio)

You have used BIDS earlier to create a blank project. While BIDS provides a development environment for developing SSIS packages, SQL Server Management Studio enables you to manage the deployed packages. In this part, you will connect to Integration Services using SQL Server Management Studio.

1. From Start | All Programs | Microsoft SQL Server 2008 and then click SQL Server Management Studio.
2. When the SQL Server Management Studio loads, you will see the Connect To Server dialog box, where you can choose a server type to which you want to connect and provide your authentication details. Click in the Server Type field and select Integration Services from the drop-down list. Type **localhost** in the Server Name field to connect to the local server and press Connect.
3. SQL Server Management Studio will connect to Integration Services and show you Running Packages and Stored Packages folders under Integration Services in the Object Explorer, as shown in [Figure 1-5](#).

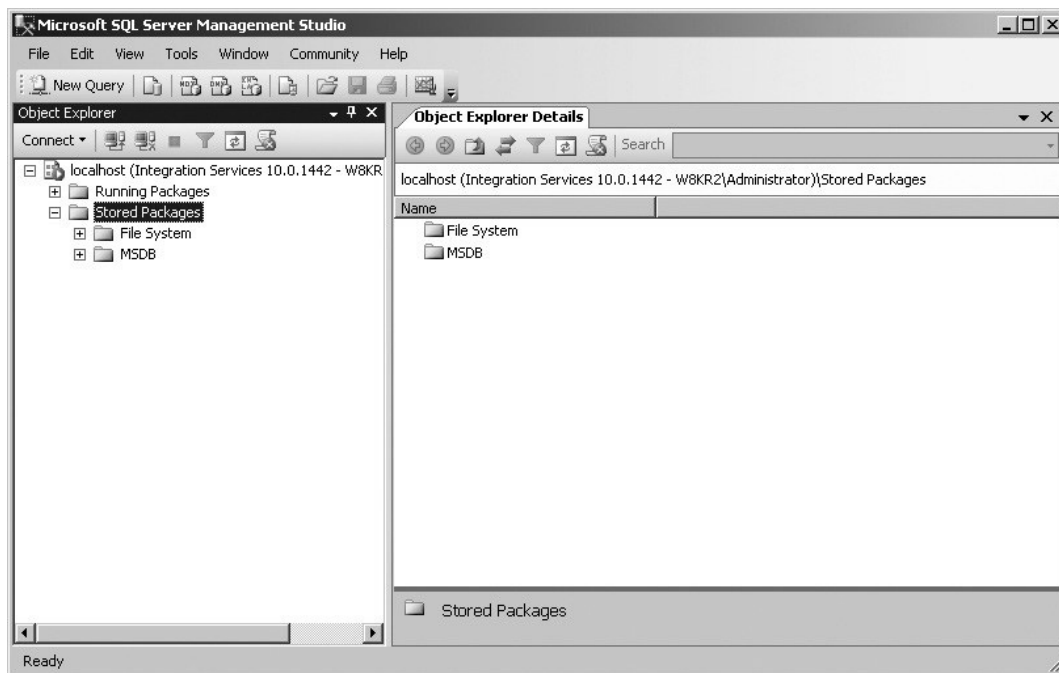


Figure 1-5: Connecting to the Integration Services service

4. Expand these folders. You will not see any packages listed under Running Packages because no packages have yet been created. Click the Stored Packages folder and you'll see the File System and MSDB folders. This is where you will be managing the packages you've stored in the file system or the MSDB database in SQL Server. Managing these folders and the packages is covered in detail in Chapter 6.
5. Let's do a little more research to see where DTS 2000 packages go. In the Object Explorer window, click Connect and choose Database Engine from the drop-down list. Note that you can connect to Integration Services from here as well. Leave localhost as the Server Name and verify that Windows Authentication is selected in the Authentication field. Click Connect to connect to the local database server.
6. Under the database server, expand the Management node and then the Legacy node. You will see a Data Transformation Services node. This is where you will be managing DTS 2000 packages that are imported into SQL Server 2008. How you run existing DTS 2000 packages or migrate them to the Integration Services format is covered in depth in Chapter 14.
7. Close SQL Server Management.

Review

Though you haven't yet built anything from the development point of view, you've seen quite a lot. If you've used DTS 2000, it may have already answered many of your questions, but if you are new to SQL Server, you can appreciate how easy it is to use SQL Server Management Studio to connect to Integration Services and manage SSIS packages storage locations. Also, you know where you can see DTS 2000 packages imported into SQL Server 2008. However, if you haven't installed the Integration Services as we did in earlier exercise or are trying to connect to an already existing Integration Services service on a remote server and run into issues, you may like to refer to the section "Connecting to Integration Services Service" in Chapter 6 to understand the issues involved in connecting to the Integration Services service.

Summary

You have been introduced to Integration Services by following a couple of simple Hands-On exercises and reading a description of its architecture, features, and uses. You understand how various components of Integration Services work together to provide the manageability, flexibility, and scalability you need for your SSIS packages. You also appreciate the fact that all the components can be programmatically controlled and that custom tasks and transformations can be written using languages such as C++ or any CLR-compatible language. You know that Integration Services has two separate engines that provide workflow and data flow functions. You are now ready to launch into the realm of bigger challenges.