Integration Services Packages

**SQL Server 2008 R2**

[Other Versions](javascript:;)

A package is an organized collection of connections, control flow elements, data flow elements, event handlers, variables, and configurations, that you assemble using either the graphical design tools that SQL Server Integration Services provides, or build programmatically. You then save the completed package to SQL Server, the SSIS Package Store, or the file system. The package is the unit of work that is retrieved, executed, and saved.

When you first create a package, it is an empty object that does nothing. To add functionality to a package, you add a control flow and, optionally, one or more data flows to the package.

The following diagram shows a simple package that contains a control flow with a Data Flow task, which in turn contains a data flow.

After you have created the basic package, you can add advanced features such as logging and variables to extend package functionality. For more information, see the section about Objects that Extend Package Functionality.

The completed package can then be configured by setting package-level properties that implement security, enable restarting of packages from checkpoints, or incorporate transactions in package workflow. For more information, see the section about Properties that Support Extended Features.

[Package Content](javascript:void(0))

A control flow consists of one or more tasks and containers that execute when the package runs. To control order or define the conditions for running the next task or container in the package control flow, you use precedence constraints to connect the tasks and containers in a package. A subset of tasks and containers can also be grouped and run repeatedly as a unit within the package control flow. For more information, see [Control Flow Elements](https://technet.microsoft.com/en-us/library/ms137681(v=sql.105).aspx) and [Designing Package Control Flow](https://technet.microsoft.com/en-us/library/ms140234(v=sql.105).aspx).

A data flow consists of the sources and destinations that extract and load data, the transformations that modify and extend data, and the paths that link sources, transformations, and destinations. Before you can add a data flow to a package, the package control flow must include a Data Flow task. The Data Flow task is the executable within the SSIS package that creates, orders, and runs the data flow. A separate instance of the data flow engine is opened for each Data Flow task in a package. For more information, see [Data Flow Task](https://technet.microsoft.com/en-us/library/ms141122(v=sql.105).aspx), [Data Flow Elements](https://technet.microsoft.com/en-us/library/ms140080(v=sql.105).aspx) and [Designing Package Data Flow](https://technet.microsoft.com/en-us/library/ms137842(v=sql.105).aspx).

A package typically includes at least one connection manager. A connection manager is a link between a package and a data source that defines the connection string for accessing the data that the tasks, transformations, and event handlers in the package use. Integration Services includes connection types for data sources such as text and XML files, relational databases, and Analysis Services databases and projects. For more information, see [Integration Services Connections](https://technet.microsoft.com/en-us/library/ms140203(v=sql.105).aspx).

[Package Templates](javascript:void(0))

Packages are frequently used as templates from which to build packages that share basic functionality. You build the basic package and then copy it, or you can designate the package is a template. For example, a package that downloads and copies files and then extracts the data may include the FTP and File System tasks in a Foreach Loop that enumerates files in a folder. It may also include Flat File connection managers to access the data, and Flat File sources to exact the data. The destination of the data varies, and the destination is added to each new package after it is copied from the basic package. You can also create packages and then use them as templates for the new packages that you add to an Integration Services project. For more information, see [Creating a Package in Business Intelligence Development Studio](https://technet.microsoft.com/en-us/library/ms345180(v=sql.105).aspx).

When a package is first created, either programmatically or by using SSIS Designer, a GUID is added to its **ID** property and a name to its**Name** property. If you create a new package by copying an existing package or by using a template package, the name and the GUID are copied as well. This can be a problem if you use logging, because the GUID and the name of the package are written to the logs to identify the package to which the logged information belongs. Therefore, you should update the name and the GUID of the new packages to help differentiate them from the package from which they were copied and from each other in the log data.

To change the package GUID, you regenerate a GUID in the **ID** property in the Properties window in Business Intelligence Development Studio. To change the package name, you can update the value of the **Name** property in the Properties window. You can also use the**dtutil**command prompt, or update the GUID and name programmatically. For more information, see [Setting Package Properties](https://technet.microsoft.com/en-us/library/ms137749(v=sql.105).aspx) and [dtutil Utility (SSIS Tool)](https://technet.microsoft.com/en-us/library/ms162820(v=sql.105).aspx).

[Objects that Extend Package Functionality](javascript:void(0))

Packages can include additional objects that provide advanced features or extend existing functionality, such as event handlers, configurations, logging, and variables.

Event Handlers

An event handler is a workflow that runs in response to the events raised by a package, task, or container. For example, you could use an event handler to check disk space when a pre-execution event occurs or if an error occurs, and send an e-mail message that reports the available space or error information to an administrator. An event handler is constructed like a package, with a control flow and optional data flows. Event handlers can be added to individual tasks or containers in the package. For more information, see [Integration Services Event Handlers](https://technet.microsoft.com/en-us/library/ms140223(v=sql.105).aspx) and [Adding Package Event Handlers](https://technet.microsoft.com/en-us/library/ms140011(v=sql.105).aspx).

Configurations

A configuration is a set of property-value pairs that defines the properties of the package and its tasks, containers, variables, connections, and event handlers when the package runs. Using configurations makes it possible to update properties without modifying the package. When the package is run, the configuration information is loaded, updating the values of properties. For example, a configuration can update the connection string of connection.

The configuration is saved and then deployed with the package when the package is installed on a different computer. The values in the configuration can be updated when the package is installed to support the package in a different environment. For more information, see[Creating Package Configurations](https://technet.microsoft.com/en-us/library/ms141132(v=sql.105).aspx).

Logging and Log Providers

A log is a collection of information about the package that is collected when the package runs. For example, a log can provide the start and finish time for a package run. A log provider defines the destination type and the format that the package and its containers and tasks can use to log run-time information. The logs are associated with a package, but the tasks and containers in the package can log information to any package log. Integration Services includes a variety of built-in log providers for logging. For example, Integration Services includes log providers for SQL Server and text files. You can also create custom log providers and use them for logging. For more information, see[Integration Services Log Providers](https://technet.microsoft.com/en-us/library/ms140246(v=sql.105).aspx).

Variables

Integration Services supports system variables and user-defined variables. The system variables provide useful information about package objects at run time, and user-defined variables support custom scenarios in packages. Both types of variables can be used in expressions, scripts, and configurations.

The package-level variables include the pre-defined system variables available to a package and the user-defined variables with package scope. For more information, see [Integration Services Variables](https://technet.microsoft.com/en-us/library/ms141085(v=sql.105).aspx).

[Package Properties that Support Extended Features](javascript:void(0))

The package object can be configured to support features such as restarting the package at checkpoints, signing the package with a digital certificate, setting the package protection level, and ensuring data integrity by using transactions.

Restarting Packages

The package includes checkpoint properties that you can use to restart the package when one or more of its tasks fail. For example, if a package has two Data Flow tasks that update two different tables and the second task fails, the package can be rerun without repeating the first Data Flow task. Restarting a package can save time for long-running packages. Restarting means you can start the package from the failed task instead of having to rerun the whole package. For more information, see [Restarting Failed Packages by Using Checkpoints](https://technet.microsoft.com/en-us/library/ms140226(v=sql.105).aspx).

Securing Packages

A package can be signed with a digital signature and encrypted by using a password or a user key. A digital signature authenticates the source of the package. However, you must also configure Integration Services to check the digital signature when the package loads. For more information, see [Using Digital Signatures with Packages](https://technet.microsoft.com/en-us/library/ms141174(v=sql.105).aspx) and [Setting the Protection Level of Packages](https://technet.microsoft.com/en-us/library/ms141747(v=sql.105).aspx).

Supporting Transactions

Setting a transaction attribute on the package enables tasks, containers, and connections in the package to join the transaction. Transaction attributes ensure that the package and its elements succeed or fail as a unit. Packages can also run other packages and enroll other packages in transactions, so that you can run multiple packages as a single unit of work. For more information, see [Ensuring Data Integrity by Using Transactions](https://technet.microsoft.com/en-us/library/ms137690(v=sql.105).aspx).

[Custom Log Entries Available on the Package](javascript:void(0))

The following table lists the custom log entries for packages. For more information, see [Implementing Logging in Packages](https://technet.microsoft.com/en-us/library/ms138020(v=sql.105).aspx) and [Custom Messages for Logging](https://technet.microsoft.com/en-us/library/ms345174(v=sql.105).aspx).

|  |  |
| --- | --- |
| **Log entry** | **Description** |
| **PackageStart** | Indicates that the package began to run.  **Note**  This log entry is automatically written to the log. You cannot exclude it. |
| **PackageEnd** | Indicates that the package completed.  **Note**  This log entry is automatically written to the log. You cannot exclude it. |
| **Diagnostic** | Provides information about the system configuration that affects package execution such as the number executables that can be run concurrently. |

[Creating a Package](javascript:void(0))

Integration Services includes two graphical tools, SSIS Designer and SQL Server Import and Export Wizard, in addition to the SSIS object model for creating packages.

For more information about how to create packages, click a topic in the following list:

* [Using the SQL Server Import and Export Wizard to Move Data](https://technet.microsoft.com/en-us/library/ms141209(v=sql.105).aspx)
* [Designing Packages in Business Intelligence Development Studio](https://technet.microsoft.com/en-us/library/ms141711(v=sql.105).aspx)
* [Building Packages Programmatically](https://technet.microsoft.com/en-us/library/ms345167(v=sql.105).aspx)

[Configuring the Package](javascript:void(0))

You can set properties in the **Properties** window of Business Intelligence Development Studio or programmatically.

For more information about how to set these properties, click the following topic:

* [Setting Package Properties](https://technet.microsoft.com/en-us/library/ms137749(v=sql.105).aspx)

Configuring the Package Programmatically

For more information about programmatically setting these properties, click the following topic:

* [Package](https://technet.microsoft.com/en-us/library/microsoft.sqlserver.dts.runtime.package(v=sql.105).aspx)