Integration Services Event Handlers

**SQL Server 2008 R2**

[Other Versions](javascript:;)

At run time, executables (packages and Foreach Loop, For Loop, Sequence, and task host containers) raise events. For example, an OnError event is raised when an error occurs. You can create custom event handlers for these events to extend package functionality and make packages easier to manage at run time. Event handlers can perform tasks such as the following:

* Clean up temporary data storage when a package or task finishes running.
* Retrieve system information to assess resource availability before a package runs.
* Refresh data in a table when a lookup in a reference table fails.
* Send an e-mail message when an error or a warning occurs or when a task fails.

If an event has no event handler, the event is raised to the next container up the container hierarchy in a package. If this container has an event handler, the event handler runs in response to the event. If not, the event is raised to the next container up the container hierarchy.

The following diagram shows a simple package that has a For Loop container that contains one Execute SQL task.

Only the package has an event handler, for its **OnError** event. If an error occurs when the Execute SQL task runs, the **OnError** event handler for the package runs. The following diagram shows the sequence of calls that causes the **OnError** event handler for the package to execute.

Event handlers are members of an event handler collection, and all containers include this collection. If you create the package using SSIS Designer, you can see the members of the event handler collections in the **Event Handlers** folders on the **Package Explorer** tab of SSIS Designer.

You can configure the event handler container in the following ways:

* Specify a name and description for the event handler.
* Indicate whether the event handler runs, whether the package fails if the event handler fails, and the number of errors that can occur before the event handler fails.
* Specify an execution result to return instead of the actual execution result that the event handler returns at run time.
* Specify the transaction option for the event handler.
* Specify the logging mode that the event handler uses.

[Event Handler Content](javascript:void(0))

Creating an event handler is similar to building a package; an event handler has tasks and containers, which are sequenced into a control flow, and an event handler can also include data flows. The SSIS Designer includes the **Event Handlers** tab for creating custom event handlers. For more information, see [Adding Package Event Handlers](https://technet.microsoft.com/en-us/library/ms140011(v=sql.105).aspx).

You can also create event handlers programmatically. For more information, see [Handling Events Programmatically](https://technet.microsoft.com/en-us/library/ms135967(v=sql.105).aspx).

[Run-Time Events](javascript:void(0))

The following table lists the event handlers that Integration Services provides, and describes the run-time events that cause the event handler to run.

|  |  |
| --- | --- |
| **Event handler** | **Event** |
| **OnError** | The event handler for the **OnError** event. This event is raised by an executable when an error occurs. |
| **OnExecStatusChanged** | The event handler for the **OnExecStatusChanged** event. This event is raised by an executable when its execution status changes. |
| **OnInformation** | The event handler for the **OnInformation** event. This event is raised during the validation and execution of an executable to report information. This event conveys information only, no errors or warnings. |
| **OnPostExecute** | The event handler for the **OnPostExecute** event. This event is raised by an executable immediately after it has finished running. |
| **OnPostValidate** | The event handler for the **OnPostValidate** event. This event is raised by an executable when its validation is finished. |
| **OnPreExecute** | The event handler for the **OnPreExecute** event. This event is raised by an executable immediately before it runs. |
| **OnPreValidate** | The event handler for the **OnPreValidate** event. This event is raised by an executable when its validation starts. |
| **OnProgress** | The event handler for the **OnProgress** event. This event is raised by an executable when measurable progress is made by the executable. |
| **OnQueryCancel** | The event handler for the **OnQueryCancel** event. This event is raised by an executable to determine whether it should stop running. |
| **OnTaskFailed** | The event handler for the **OnTaskFailed** event. This event is raised by a task when it fails. |
| **OnVariableValueChanged** | The event handler for the **OnVariableValueChanged** event. This event is raised by an executable when the value of a variable changes. The event is raised by the executable on which the variable is defined. This event is not raised if you set the **RaiseChangeEvent** property for the variable to **False**. For more information, see [Integration Services Variables](https://technet.microsoft.com/en-us/library/ms141085(v=sql.105).aspx). |
| **OnWarning** | The event handler for the **OnWarning** event. This event is raised by an executable when a warning occurs. |

[Configuring an Event Handler](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 in Business Intelligence Development Studio, see [How to: Set the Properties of a Task or Container](https://technet.microsoft.com/en-us/library/ms139733(v=sql.105).aspx).

Configure an Event Handler Programmatically

For more information about programmatically setting these properties, see [DtsEventHandler](https://technet.microsoft.com/en-us/library/microsoft.sqlserver.dts.runtime.dtseventhandler(v=sql.105).aspx).