Ch-4- Enforcing Data Quality, Extending SQL Server Integration Services

Data Quality Services

SQL Server Data Quality Services (DQS) is a knowledge-driven data quality product. DQSenables you to build a knowledge base and use it to perform a variety of critical dataquality tasks, including correction, enrichment, standardization, and de-duplication ofyour data. DQS enables you to perform data cleansing by using cloud-based referencedata services provided by reference data providers. DQS also provides you with profiling That is integrated into its data-quality tasks, enabling you to analyze the integrity of yourdata.

DQS consists of Data Quality Server and Data Quality Client, both of which are installedas part of SQL Server 2012. Data Quality Server is a SQL Server instance feature that Consists of three SQL Server catalogs with data-quality functionality and storage. Data Quality Client is a SQL Server shared feature that business users, information workers,and IT professionals can use to perform computer-assisted data quality analyses andmanage their data quality interactively. You can also perform data quality processes byusing the DQS Cleansing component in Integration Services

Need of DQS

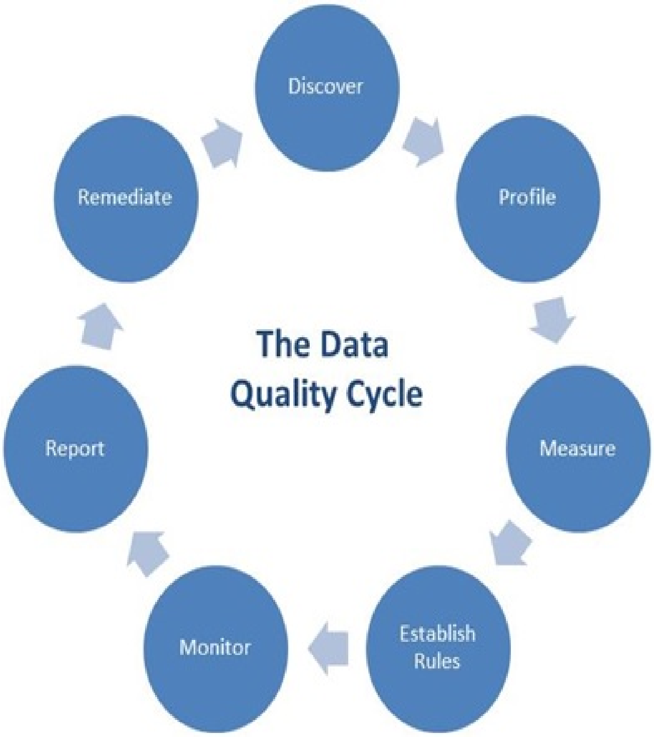
Data quality is not defined in absolute terms. It depends upon whether data isappropriate for the purpose for which it is intended. DQS identifies potentially incorrectdata, and provides you with an assessment of the likelihood that the data is in factincorrect. DQS provides you with a semantic understanding of the data so you can Decide its appropriateness. DQS enables you to resolve issues involving incompleteness,lack of conformity, inconsistency, inaccuracy, invalidity, and data duplication.

DQS provides the following features to resolve data quality issues.

- **Data Cleansing:**the modification, removal, or enrichment of data that is incorrect orincomplete, using both computer-assisted and interactive processes.
- **DataMatching:**the identification of semantic duplicates in a rules-based process thatenables you to determine what constitutes a match and performs de-duplication.

- **Reference Data Services:** verification of the quality of your data using the services ofa reference data provider. You can use reference data services from Windows Azure Marketplace DataMarket to easily cleanse, validate, match, and enrich data.
- **Profiling:** the analysis of a data source to provide insight into the quality of the dataat every stage in the knowledge discovery, domain management, matching, and datacleansing processes. Profiling is a powerful tool in a DQS data quality solution. You can create a data quality solution in which profiling is just as important as knowledgemanagement, matching, or data cleansing.
- **Monitoring:** the tracking and determination of the state of data quality activities.Monitoring enables you to verify that your data quality solution is doing what it wasdesigned to do.
- **Knowledge Base:** Data Quality Services is a knowledge-driven solution that analyses data based upon knowledge that you build with DQS. This enables you to create dataquality processes that continually enhances the knowledge about your data and in sodoing, continually improves the quality of your data.

DQS-PROCESS IMAGE



- **DQS Components**

Data Quality Services consists of Data Quality Server and Data Quality Client. These components enable you to perform data quality services separately from other SQL Server operations. Both are installed from within the SQL Server setup program. Data Quality Server is implemented as three SQL Server catalogs that you can manage and monitor in the SQL Server Management Studio (DQS_MAIN, DQS_PROJECTS, and DQS_STAGING_DATA). DQS_MAIN includes DQS stored procedures, the DQS engine, and published knowledge bases. DQS_PROJECTS includes data that is required for knowledge base management and DQS project activities. DQS_STAGING_DATA provides an intermediate staging database where you can copy your source data to perform DQS operations, and then export your processed data. Data Quality Client is a standalone application that enables you to perform knowledge management, data quality projects, and administration in one user interface. The application is designed for both data stewards and DQS administrators. It is a stand- alone executable file that performs knowledge discovery, domain management, matching policy creation, data cleansing, matching, profiling, monitoring, and server administration. Data Quality Client can be installed and run on the same computer as Data Quality Server or remotely on a separate computer. Many operations in Data Quality Client are wizard-driven for ease of use.

Data Quality Functionality in Integration Services and Master Data Services Data quality functionality provided by Data Quality Services is built into a component of SQL Server Integration Services (SSIS) and into features of Master Data Services (MDS) to enable you to perform data quality processes within those services. DQS Cleansing component in Integration Services. The DQS Cleansing component in Integration Services enables you to perform data cleansing as part of an Integration Services package. When the package is run, data cleansing is run as a batch file. This is an alternative to running a cleansing project in the Data Quality Client application. You can ensure the quality of your data automatically. You do not have to perform the interactive steps of a data cleansing project within the Data Quality Client application. You can include the data cleansing process within a data flow that contains other Integration Services components. For more information, see Data Cleansing Transformation. Data Quality Processes in Master Data Services Data Quality Services functionality has been integrated into Master Data Services (MDS), so you can perform de-duplication on source data and master data within the Microsoft SQL Server 2012 Master Data Services Add-in for Microsoft Excel. To perform matching, load data managed by MDS into an Excel worksheet, combine it with data not managed by MDS, and then perform matching within Excel. The Data Quality Server components must be installed with MDS.

DQS CLEANSING

Introduced in SQL Server 2012 was a component called Data Quality Services (DQS). This is not a feature of Integration Services, but it is very much connected to the data cleansing processes within SSIS. In fact, there is a data transformation called the DQS Cleansing Task. This task connects to DQS, enabling you to connect incoming Data Flow data and perform data cleansing operations. Because this Tutorial focuses on SSIS, a full DQS tutorial is not included; however, this section provides a brief overview of DQS and highlights a few data quality examples.

SSIS can connect to DQS using the DQS Cleansing Transformation. This is one of two ways that data can be applied against the knowledge bases within DQS. (A data quality project is the primary way to process data if you are not using SSIS for ETL. This is found in the DQS client tool, but it’s not described in this Tutorial, which focuses on SSIS.)

Data Quality Service to match data

The Data Quality Services (DQS) data matching process enables you to reduce data duplication and improve data accuracy in a data source. Matching analyzes the degree of duplication in all records of a single data source, returning weighted probabilities of a match between each set of records compared. You can then decide which records are matches and take the appropriate action on the source data.

The DQS matching process has the following benefits:

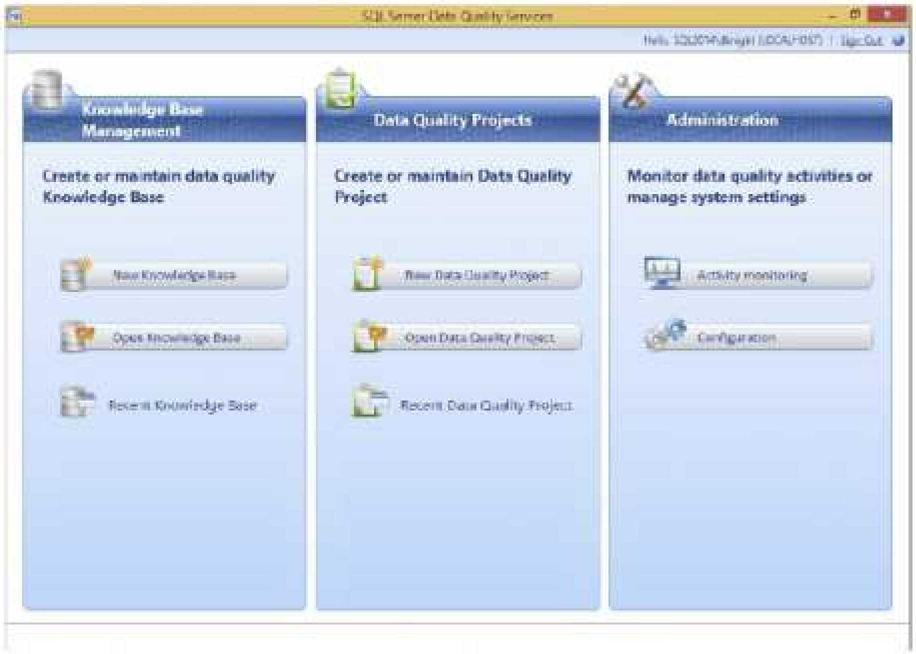
Matching enables you to eliminate differences between data values that should be equal, determining the correct value and reducing the errors that data differences can cause. For example, names and addresses are often the identifying data for a data source, particularly customer data, but the data can become dirty and deteriorate over time. Performing matching to identify and correct these errors can make data use and maintenance much easier.

- Matching enables you to ensure that values that are equivalent, but were entered in a different format or style, are rendered uniform.
- Matching identifies exact and approximate matches, enabling you to remove duplicate data as you define it. You define the point at which an approximate match is in fact a match. You define which fields are assessed for matching, and which are not.
- DQS enables you to create a matching policy using a computer-assisted process, modify it interactively based upon matching results, and add it to a knowledge base that is reusable.
- You can re-index data copied from the source to the staging table, or not re-index, depending on the state of the matching policy and the source data. Not re-indexing can improve performance.

You can perform the matching process in conjunction with other data cleansing processes to improve overall data quality. You can also perform data de-duplication using DQS functionality built into Master Data Services

- **Data Quality Client Application**

Run Data Quality Client To run Data Quality Client on the computer where you have installed it, proceed as follows: 1. Click Start, point to All Programs, click Microsoft SQL Server 2012, click Data Quality Services, and then click Data Quality Client. 2. In the Connect to Server dialog box: a. Specify the server that you want to connect the Data Quality Client application to. Select (LOCAL) to connect to Data Quality Server on the local computer. You can



Create or maintain data quality Knowledge Base

Create or maintain Data Quality Project

Monitor data quality activities or manage system settings

You can perform three primary tasks with DQS:

Knowledge Base Management is how you define the data cleansing rules and policies.

Data Quality Projects are for applying the data quality definitions (from the knowledge base) against real data. We will not be considering projects in this Advanced Data Cleansing in SSIS Topic ; instead, you will see how to use the SSIS DQS Cleansing Task to apply the definitions.

Administration is about configuring and monitoring the server and external connections.

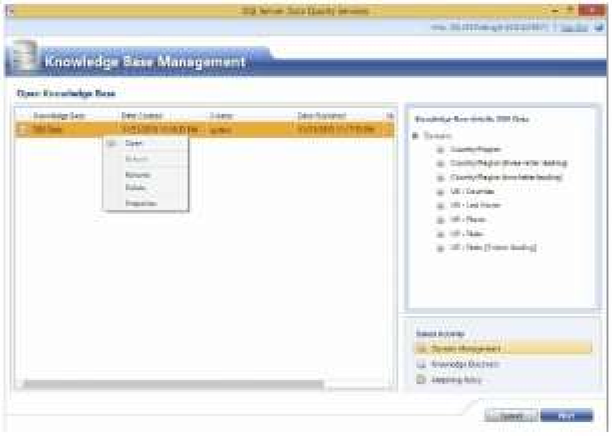
To begin the process of cleansing data with DQS, you need to perform two primary steps within the Knowledge Base Management pane:

1. Create a DQS Knowledge Base (DQS KB). A DQS KB is a grouping of related data quality definitions and rules (called domains) that are defined up front. These definitions and rules are applied against data with various outcomes (such as corrections, exceptions, etc.). For example, a DQS KB could be a set of domains

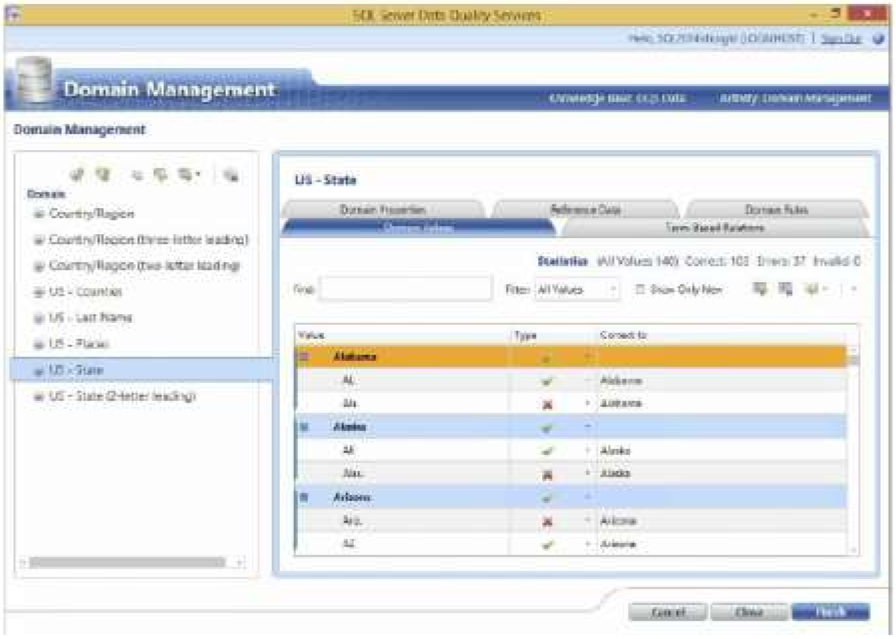
that relate to address cleansing, or a grouping of valid purchase order code rules and code relationships within your company.

2. Define DQS domains and composite domains. A DQS domain is a targeted definition of cleansing and validation properties for a given data point. For example, a domain could be “Country” and contain the logic on how to process values that relate to countries around the world. The value mapping and rules define what names are valid and how abbreviations map to which countries.

When you select the Open knowledge base option, you are presented with a list of KBs that you have worked with. The built-in KB included with DQS, DQS Data, contains several predefined domains and rules, and connections to external data. Below screen shot shows the right-click context menu, which enables you to open the KB and see the definition details.



Knowledge bases are about domains, which are the building blocks of DQS. A domain defines what the DQS engine should do with data it receives: Is it valid? Does it need to be corrected? Should it look at external services to cleanse the data? For example, Below screen shot highlights the Domain Values tab of the State domain. It shows how values are cleansed and which values should be grouped. In this example, it lists state abbreviations and names and the Correct To value



- Country/Region
- Country/Region (three-letter leading)
- Country/Region (two-letter leading)
- US - Counties
- US - Last Name
- US - Places
- US - State
- US - State (2-letter leading)

US - State

Statistics: All Values: 140 Correct: 103 Errors: 37 Invalid: 0

Value	Type	Correct to
Alabama	✓	
AL	✓	Alabama
Ala.	✗	Alabama
Alaska	✓	
AK	✓	Alaska
Alas.	✗	Alaska
Arizona	✓	
Ariz.	✗	Arizona
AZ	✓	Arizona

The SSIS Script Task

The SSIS Script Task allows you to add functionality to your SSIS package that does not already exist with the other predefined tasks. In this tip, we look at how to get started using the SSIS Script Task with a few examples.

Solution

The SSIS Script Task is one of the most interesting tools to increase SSIS capabilities. With the script task, you can program new functionality using C# or VB. This tip is for people with limited experience in SSIS and C#. If you have SSIS experience, but you do not how to use the Script Task this tip is also for you. The next tip will include more advanced features.

Requirements

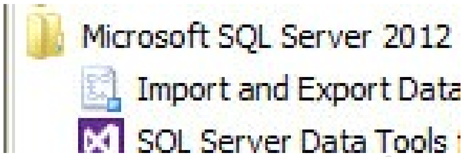
- SSIS installed
- SQL Server Data Tools Installed (SSDT) or BIDS (Business Intelligence Development Studio)
- A SQL Server database backup

- You can use SQL Server 2005 or later versions. In this example, we are using SQL Server 2014.

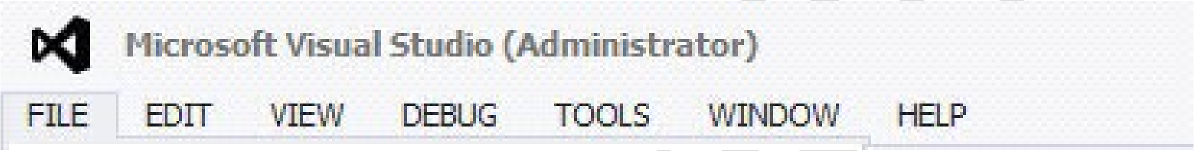
Example 1 - Hello World

Let's start with the Hello World example using a simple Script Task.

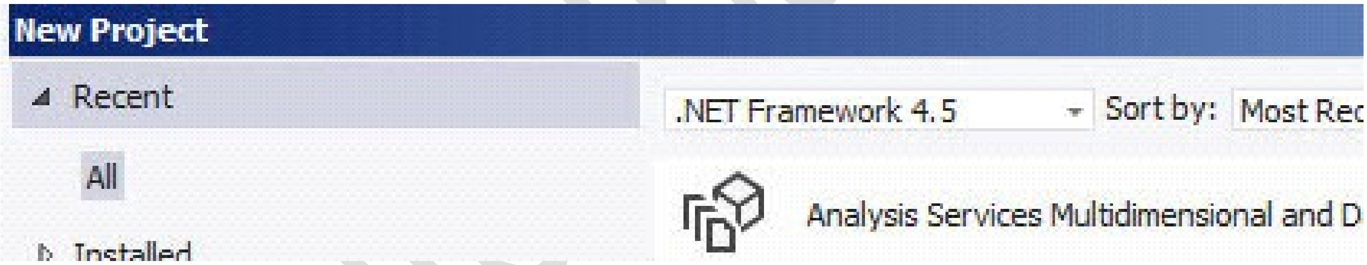
In order to start, open the SQL Server Data Tools for Visual Studio.



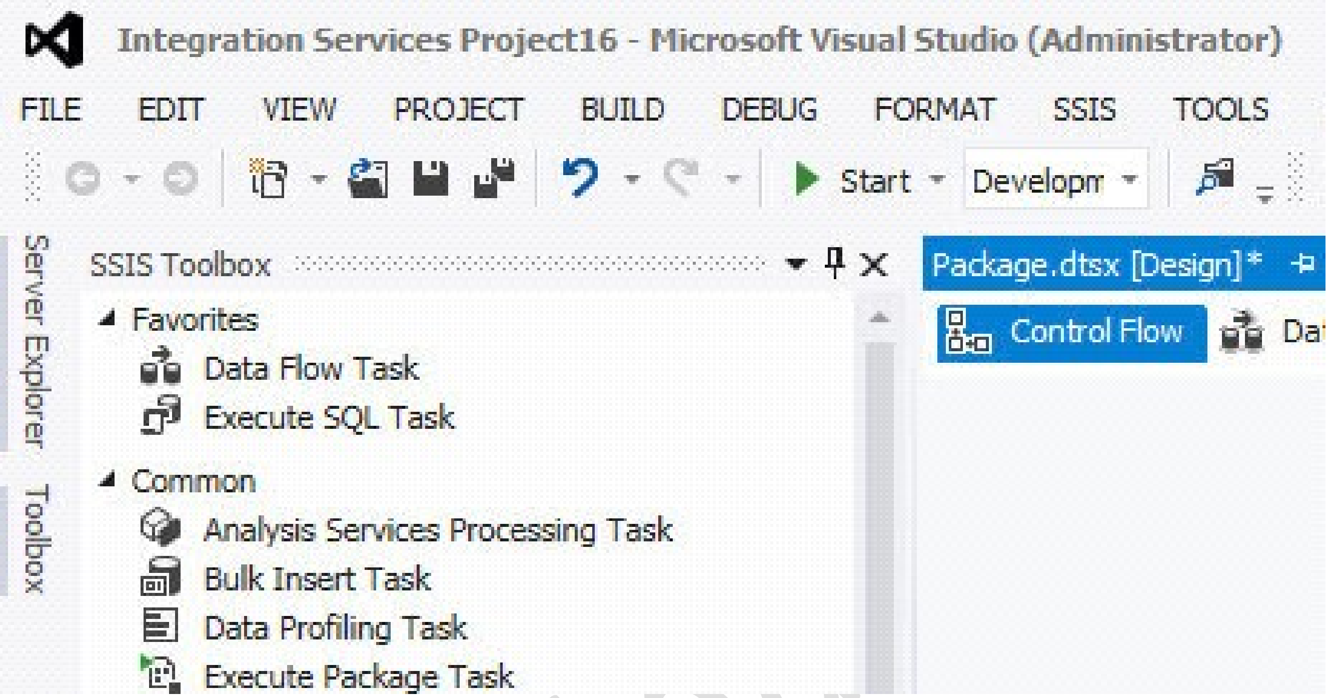
Go to File > New > Project



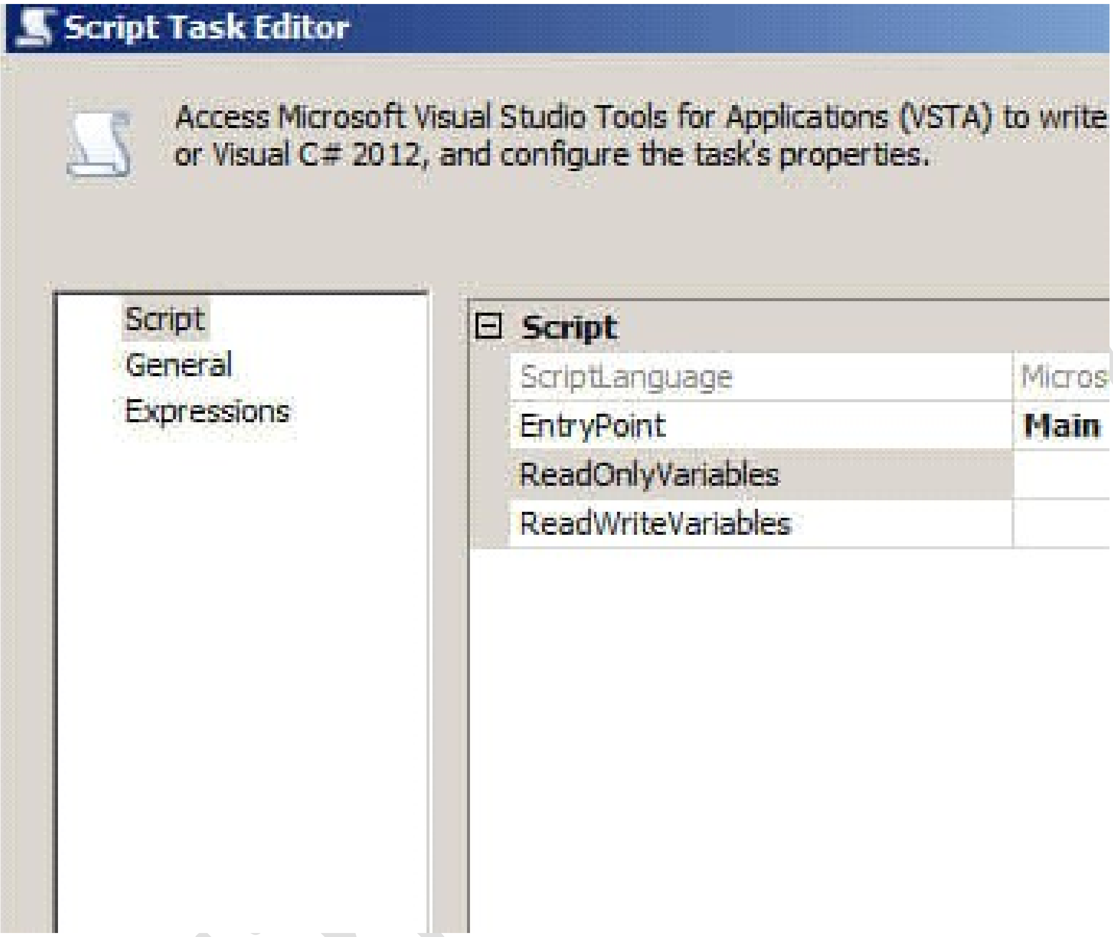
Select Integration Services Project.



Drag and drop the Script Task to the design pane and double click on it.



The following window will open. The ScriptLanguage is used to select which language to use, either C# or Visual Basic. EntryPoint is used to select where to start in the code, by default it starts in Main. The ReadOnlyVariables and ReadWriteVariables will be explained later. Press the Edit Script button to write your code.



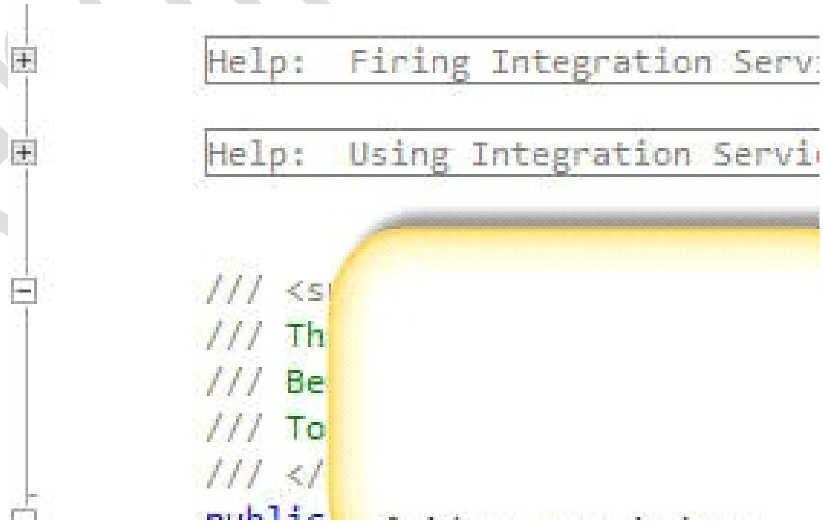
Script Task Editor

Access Microsoft Visual Studio Tools for Applications (VSTA) to write
or Visual C# 2012, and configure the task's properties.

⊟ Script

ScriptLanguage	Micros
EntryPoint	**Main**
ReadOnlyVariables	
ReadWriteVariables	

A new Window will be displayed to allow you to write the code. Go to the main procedure, by default you will create your code there.

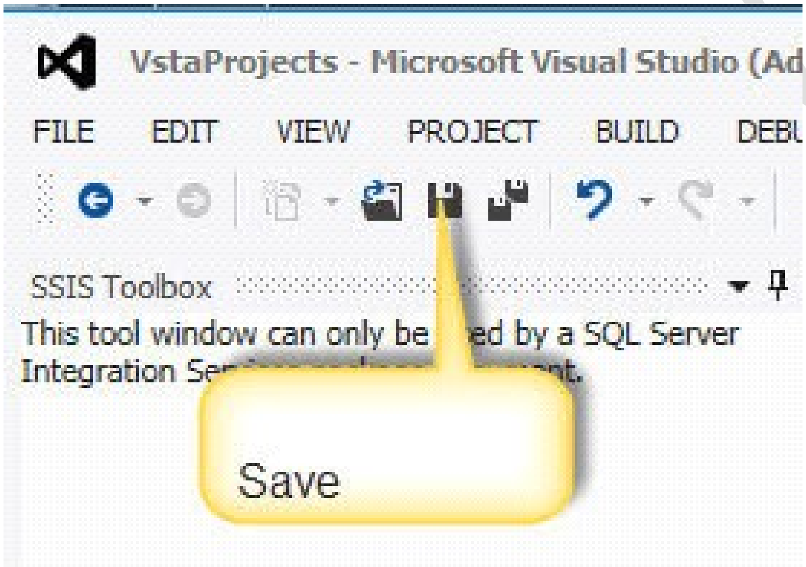


```
Help:  Firing Integration Serv

Help:  Using Integration Servi


/// <s
/// Th
/// Be
/// To
/// </
```

Add the following code in the Main section. This code will display a message with the Hello World message.

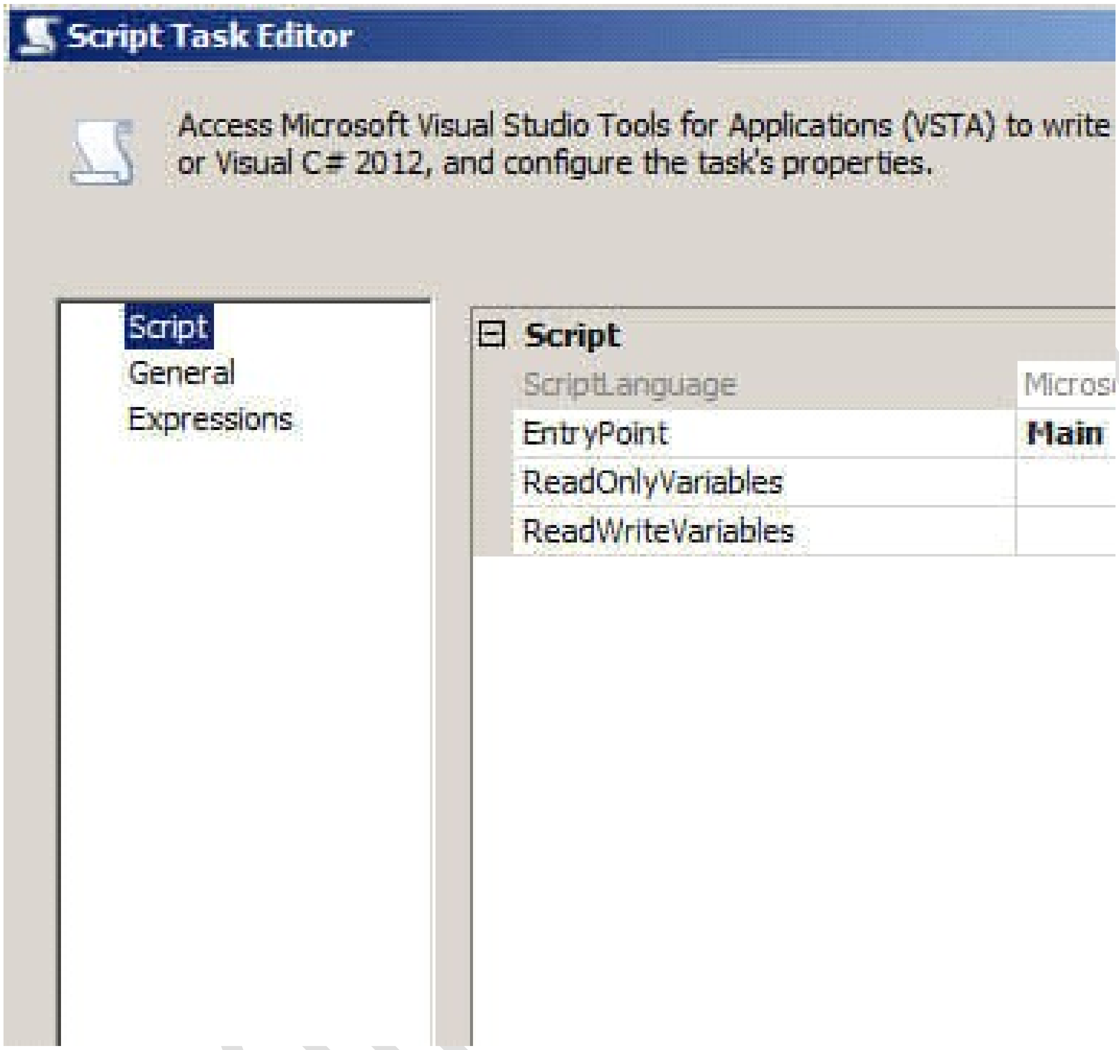
```
public void Main()
{
   // TODO: Add your code here
MessageBox.Show("Hello World");
}
```

Save the code.



This tool window can only be ed by a SQL Server Integration Se nt.

In the Script Task Editor, press OK.

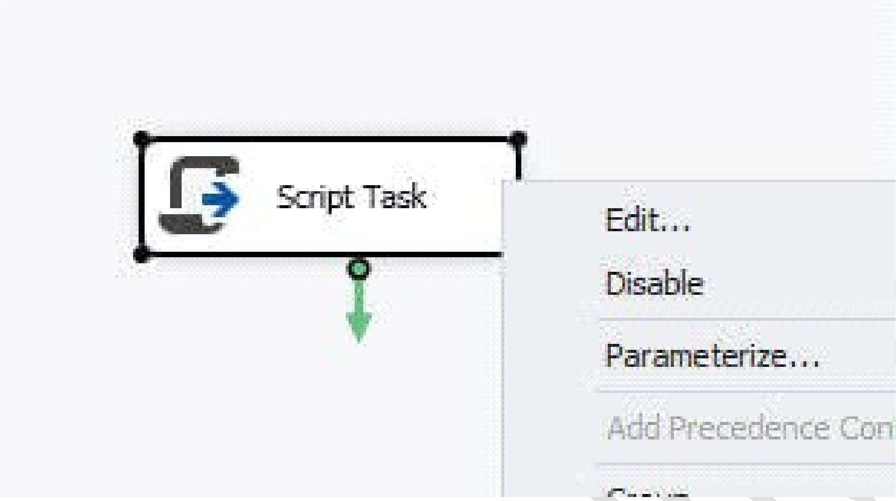


Access Microsoft Visual Studio Tools for Applications (VSTA) to write
or Visual C# 2012, and configure the task's properties.

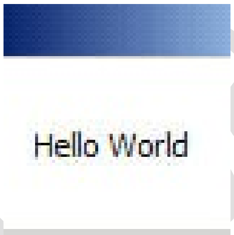
⊟ Script

ScriptLanguage	Micros
EntryPoint	Main
ReadOnlyVariables	
ReadWriteVariables	

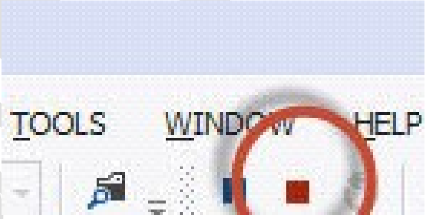
Right click on the Script Task and select the Execute Task option.



If everything is done correctly, you will receive the following pop-up message:


Hello World

Once finished, you can stop the package as shown below.



Differences between the Script Task and the Script Component

The Script task and the Script component have the following noteworthy differences.

Feature	Script Task	Script Component
Control flow / Data flow	The Script task is configured on the Control Flow tab of the designer and runs outside the data flow of the package.	The Script component is configured on the Data Flow page of the designer and represents a source, transformation, or destination in the Data Flow task.
Purpose	A Script task can accomplish almost any general-purpose task.	You must specify whether you want to create a source, transformation, or destination with the Script component.
Execution	A Script task runs custom code at some point in the package workflow. Unless you put it in a loop container or an event handler, it only runs once.	A Script component also runs once, but typically it runs its main processing routine once for each row of data in the data flow.
Editor	The **Script Task Editor** has three pages: **General**, **Script**, and **Expressions**. Only the **ReadOnlyVariables**and **ReadWriteVariables**, and **ScriptLanguage**properties directly affect the code that you can write.	The **Script Transformation Editor** has up to four pages: **Input Columns**, **Inputs and Outputs**, **Script**, and **Connection Managers**. The metadata and properties that you configure on each of these pages determines the members of the base classes that are autogenerated for your use in coding.
Interaction with the package	In the code written for a Script task, you use the **Dts**property to access other features of the package. The **Dts**property is a member of the **ScriptMain**class.	In Script component code, you use typed accessor properties to access certain package features such as variables and connection managers. The **PreExecute**method can access only read-only variables. The **PostExecute**method can access both read-only and read/write variables.

Using variables	The Script task uses the **P:Microsoft.SqlServer.Dts.Tasks.ScriptT**	The Script component uses typed accessor properties of the autogenerated

Feature	Script Task	Script Component
	ask.ScriptObjectModel.Variablesproperty of the **Dts**object to access variables that are available through the task's **P:Microsoft.SqlServer.Dts.Tasks.ScriptTask.ScriptTask.ReadOnlyVariables**and **P:Microsoft.SqlServer.Dts.Tasks.ScriptTask.ScriptTask.ReadWriteVariables**properties. For example: Dim myVar as String myVar = Dts.Variables("MyStringVariable ").Value.ToString string myVar; myVar = Dts.Variables["MyStringVariable "].Value.ToString();	based class, created from the component's **P:Microsoft.SqlServer.Dts.Pipeline.ScriptComponent.ReadOnlyVariables**and **P:Microsoft.SqlServer.Dts.Pipeline.ScriptComponent.ReadWriteVariables**properties. Forexample: Dim myVar as String myVar = Me.Variables.MyStringVariable string myVar; myVar = this.Variables.MyStringVariable;
Using connections	The Script task uses the **P:Microsoft.SqlServer.Dts.Tasks.ScriptTask.ScriptObjectModel.Connections**property of the **Dts**object to access connection managers defined in the package.	The Script component uses typed accessor properties of the autogenerated base class, created from the list of connection managers entered by the user on the Connection Managers page of the editor.