# SQL_R2_Logo.jpg

Optimized Bulk Loading of Data into Oracle

SQL Server Technical Article

**Writer:** Carla Sabotta, Debarchan Sarkar

**Technical Reviewer:** Matt Masson, Jason Howell

**Published:** 04/2012

**Applies to:** SQL Server 2005 (all editions), SQL Server 2008 and SQL Server 2008 R2 (non-Enterprise editions)

**Summary:** SQL Server 2008 and SQL Server 2008 R2 (Enterprise editions) support bulk loading Oracle data using Integration Services packages with the Microsoft Connector for Oracle by Attunity. For SQL Server 2005 and the non-Enterprise editions of SQL Server 2008 and 2008 R2, there are alternatives for achieving optimal performance when loading Oracle data. This paper discusses these alternatives.

Copyright

This document is provided “as-is”. Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

Some examples depicted herein are provided for illustration only and are fictitious.  No real association or connection is intended or should be inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

© 2011 Microsoft. All rights reserved.

Contents

[ 1](#_Toc320693265)

[Introduction 4](#_Toc320693266)

[Alternatives for Optimized Loading and Unloading Oracle Data 4](#_Toc320693267)

[Customized Script Component 5](#_Toc320693268)

[Third-party Components 12](#_Toc320693269)

[Conclusion 12](#_Toc320693270)

# Introduction

SQL Server 2008 and 2008 R2 (Enterprise editions) support bulk loading Oracle data using Integration Services (SSIS) packages. The Microsoft Connector for Oracle by Attunity provides optimal performance through their high-speed connectors during the loading or unloading of data from Oracle. For more information, see [Using the Microsoft Connector for Oracle by Attunity with SQL Server 2008 Integration Services](http://msdn.microsoft.com/en-us/library/ee470675%28SQL.100%29.aspx) (http://msdn.microsoft.com/en-us/library/ee470675(SQL.100).aspx).

SQL Server 2005 and the non-Enterprise editions of SQL Server 2008 and 2008 R2 don’t provide an out-of-the box option for bulk loading Oracle data.

* The fast load options for the OLE DB destination aren’t available when you use the Oracle OLE DB provider for Oracle because the provider doesn’t implement the [IRowsetFastLoad](http://msdn.microsoft.com/en-us/library/ms131708.aspx) (http://msdn.microsoft.com/en-us/library/ms131708.aspx) interface.

In addition, the current design of SSIS is such that it makes the fast load options available only for the SQL providers. The options aren’t available for any other provider even if the provider implements the IRowsetFastLoad interface.

* The Microsoft OLE DB Provider for Oracle is deprecated and not recommended to use against Oracle versions later than 8i.
<http://support.microsoft.com/kb/244661>

In SQL Server 2005 and the non-Enterprise edition of SQL Server 2008 and 2008 R2, the out-of-the box, SSIS components implement single row inserts to load data to Oracle. When you use single row inserts, the following issues may occur.

* Long load times and poor performance
* Data migration deadlines are not met
* Timeout during the ETL process for large production databases (greater than 500 GB) with complex referential integrity

For these releases, there are alternatives for achieving optimal performance when loading Oracle data. This paper discusses alternatives for previous releases.

# Alternatives for Optimized Loading and Unloading Oracle Data

The following are alternatives for optimizing the loading of Oracle data, for SQL Server 2005 and the non-Enterprise edition of SQL Server 2008 and 2008 R2.

* Customized Script component
* Third-party components

## Customized Script Component

In this solution, a Script component is configured as a destination. The component connects to Oracle using the OLE DB provider from Oracle (OraOLEDB) and bulk loads data to an Oracle database. The Script component performs the data load in about half the time that it would take to perform single row inserts using an OLE DB destination.

The provider is included in the Oracle Data Access Components (ODAC) that is available for download on the [Oracle Data Access Components](http://www.oracle.com/technetwork/topics/dotnet/utilsoft-086879.html) site (http://www.oracle.com/technetwork/topics/dotnet/utilsoft-086879.html). An Oracle online account is required to download the software.

**Note**: You can also configure the Script component to connect to Oracle using the ODP.Net provider from Oracle.

The [System.Data.OleDb namespace](http://msdn.microsoft.com/en-us/library/system.data.oledb%28VS.80%29.aspx) (http://tinyurl.com/7zuffuf ) is used in the script, as shown in the Microsoft Visual Basic code example below. The namespace is the .NET Framework Data Provider for OLE DB.

The [PreExecute](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dts.pipeline.scriptcomponent.preexecute%28SQL.90%29.aspx) (http://tinyurl.com/86e4exe) method is overridden to create the OleDbParameter objects for each of the input columns. The parameters are added to the OleDbCommand Object, to configure the parameterized command that the destination will use to insert the data. In the example, the input columns are CustomerID, TerritoryID, AccountNumber, and ModifiedDate. Then, the database transaction is started.

The [AcquireConnections](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dts.pipeline.scriptcomponent.acquireconnections%28SQL.90%29.aspx) (http://tinyurl.com/7qkkqvq) method is overridden to return a System.Data.OleDb.OleDbConnection from the connection manager that connects to the Oracle database.

The [ProcessInputRow](http://msdn.microsoft.com/en-us/library/ms187303%28SQL.90%29.aspx) (http://tinyurl.com/8y7vnh5 ) method is overridden to process the data in each input row as it passes through.

**To configure the Script component**

1. Add a data source to the package, such as an OLE DB Source. The data source should have fields that can be easily loaded into a target table. In this example, we’re using the Customer table in the AdventureWorks database as the data source.
2. Add a Script component and configure the component as a destination. Connect the component to the data source.
3. Double-click the Script component to open the **Script Transformation Editor**.



1. Click **Connection Managers** in the left-hand pane, and then click **Add**.



1. Change the name of the new connection to orcl\_scott to easily identify the connection manager, and then select **<New connection>** in the **Connection Manager** Field.
2. In the **Add SSIS Connection Manager** dialog box, select **ADO.NET** and then click **Add**.



1. In the **Configure ADO.NET Connection Manager** dialog box click **New**.
2. In the **Connection Manager** dialog box, select **.Net Providers for OleDb\Oracle Provider for OLE DB** in the **Provider** drop-down list.



1. Click **Test Connection** to confirm the connection, and then click **OK**.
2. In the **Script Transformation Editor**, click the newly created connection manager to select it and then click **Script** in the left-hand pane.





1. Click **Design Script** and add the following Visual Basic code.

In SQL Server 2008, **Design Script** was changed to **Edit Script** and support was added for the Microsoft Visual C# 2008 programming language.

Imports System

Imports System.Data

Imports System.Math

Imports Microsoft.SqlServer.Dts.Pipeline.Wrapper

Imports Microsoft.SqlServer.Dts.Runtime.Wrapper

Imports System.Data.OleDb

Imports System.Data.Common

<Microsoft.SqlServer.Dts.Pipeline.SSISScriptComponentEntryPointAttribute> \_

<CLSCompliant(False)> \_

Public Class ScriptMain

 Inherits UserComponent

 Dim row\_count As Int64

 Dim batch\_size As Int64

 Dim connMgr As IDTSConnectionManager100

 Dim oledbconn As OleDbConnection

 Dim oledbtran As OleDbTransaction

 Dim oledbCmd As OleDbCommand

 Dim oledbParam As OleDbParameter

 Public Overrides Sub PreExecute()

 batch\_size = 8 \* 1024

 row\_count = 0

 oledbCmd = New OleDbCommand("INSERT INTO "Customer"("CustomerID", "TerritoryID", "AccountNumber", "ModifiedDate") VALUES(?, ?, ?, ?)", oledbconn)

 oledbParam = New OleDbParameter("@CustomerID", OleDbType.Integer, 7)

 oledbCmd.Parameters.Add(oledbParam)

 oledbParam = New OleDbParameter("@TerritoryID", OleDbType.Integer, 7)

 oledbCmd.Parameters.Add(oledbParam)

 oledbParam = New OleDbParameter("@AccountNumber", OleDbType.VarChar, 7)

 oledbCmd.Parameters.Add(oledbParam)

 oledbParam = New OleDbParameter("@ModifiedDate", OleDbType.Date, 7)

 oledbCmd.Parameters.Add(oledbParam)

 oledbtran = oledbconn.BeginTransaction()

 oledbCmd.Transaction = oledbtran

 MyBase.PreExecute()

 End Sub

 Public Overrides Sub AcquireConnections(ByVal Transaction As Object)

 connMgr = Me.Connections.Connection

 oledbconn = CType(connMgr.AcquireConnection(Nothing), OleDb.OleDbConnection)

 End Sub

 Public Overrides Sub Input0\_ProcessInputRow(ByVal Row As Input0Buffer)

 With oledbCmd

 .Parameters("@CustomerID").Value = Row.CustomerID

 .Parameters("@TerritoryID").Value = Row.TerritoryID

 .Parameters("@AccountNumber").Value = Row.AccountNumber

 .Parameters("@ModifiedDate").Value = Row.ModifiedDate

 .ExecuteNonQuery()

 End With

 row\_count = row\_count + 1

 If (row\_count Mod batch\_size) = 0 Then

 oledbtran.Commit()

 oledbtran = oledbconn.BeginTransaction()

 oledbCmd.Transaction = oledbtran

 End If

 End Sub

 Public Overrides Sub PostExecute()

 MyBase.PostExecute()

 End Sub

 Public Overrides Sub ReleaseConnections()

 oledbtran.Commit()

 MyBase.ReleaseConnections()

 End Sub

End Class

1. Save your changes to the Script component.

The SSIS package now contains the custom script component, configured as a destination to bulk load data to the Oracle data source.

**Note:** The above script component connects to Oracle, but it can be used to connect to other third-party data sources such as Sybase and Informix. The only change that you need to make is to configure the connection manager to use the correct OLE DB providers available for Sybase and Informix.

## Third-party Components

In addition to the Script component solution discussed in this paper, there are third-party components that you can use to achieve optimal performance when loading Oracle data. The following components work with both SQL Server 2005 and SQL Server 2008.

* Oracle Destination and ODBC Destination components from CozyRoc. For more information, see the [CozyRoc](http://www.cozyroc.com/) web site.
* Oracle Bulk Loader SSIS Connector from [Persistent](http://www.persistentsys.com/). For more information, contact Persistent.
* Progress DataDirect Connect and DataDirect Connect64 components from Progress DataDirect. For more information, see the [DataDirect](http://www.datadirect.com/index.html) web site.

# Conclusion

SQL Server 2008 and 2008 R2 (Enterprise editions) support bulk load of Oracle data using SSIS packages. For SQL Server 2005 and the non-Enterprise editions of SQL Server 2008 and 2008 R2, the following are alternatives for optimizing the loading of Oracle data.

* Script component bulk loads data to Oracle using the Oracle OLE DB provider from Oracle
* Third-party components that connect to Oracle, from CozyRoc, Persistent, and DataDirect

**For more information:**

[Connectivity and SQL Server 2005 Integration Services](http://msdn.microsoft.com/en-us/library/bb332055%28SQL.90%29.aspx) ([http://msdn.microsoft.com/en-us/library/bb332055(SQL.90).aspx](http://msdn.microsoft.com/en-us/library/bb332055%28SQL.90%29.aspx) )

[SSIS with Oracle Connectors](http://social.technet.microsoft.com/wiki/contents/articles/1957.ssis-with-oracle-connectors.aspx) (<http://social.technet.microsoft.com/wiki/contents/articles/1957.ssis-with-oracle-connectors.aspx> )

[SQL 2012 Connectors for Oracle & Teradata Coming Soon](http://blogs.msdn.com/b/mattm/archive/2012/03/09/sql-2012-connectors-for-oracle-amp-teradata-coming-soon.aspx) (http://blogs.msdn.com/b/mattm/archive/2012/03/09/sql-2012-connectors-for-oracle-amp-teradata-coming-soon.aspx)

[SSIS and Netezza: Loading data using OLE DB Destination](http://www.rafael-salas.com/2010/06/ssis-and-netezza-loading-data-using-ole.html) (http://www.rafael-salas.com/2010/06/ssis-and-netezza-loading-data-using-ole.html)

Did this paper help you? Please give us your feedback. Tell us on a scale of 1 (poor) to 5 (excellent), how would you rate this paper and why have you given it this rating? For example:

* Are you rating it high due to having good examples, excellent screen shots, clear writing, or another reason?
* Are you rating it low due to poor examples, fuzzy screen shots, or unclear writing?

This feedback will help us improve the quality of white papers we release.

Send feedback.