

Optimized Bulk Loading of Data into Oracle

Carla Sabotta Debarchan Sarkar

Quick Step-by-Step



Microsoft

Optimized Bulk Loading of Data into Oracle

Carla Sabotta, Debarchan Sarkar

Summary: SQL Server 2008 and SQL Server 2008 R2 (Enterprise and Developer 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 and non-Developer editions of SQL Server 2008 and 2008 R2, there are alternatives for achieving optimal performance when loading Oracle data. This paper discusses these alternatives.

The Enterprise and Developer editions of SQL Server 2012 also support bulk loading Oracle data using Integration Services packages with the Microsoft Connector for Oracle by Attunity.

Category: Quick Step-by-Step

Applies to: SQL Server 2005 (all editions), SQL Server 2008, SQL Server 2008 R2, and SQL Server 2012 (non-Enterprise and non-Developer editions)

Source: White paper ([link to source content](#))

E-book publication data: November 2012

Copyright © 2012 by Microsoft Corporation

All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

Microsoft and the trademarks listed at

<http://www.microsoft.com/about/legal/en/us/IntellectualProperty/Trademarks/EN-US.aspx> are trademarks of the Microsoft group of companies. All other marks are property of their respective owners.

The example companies, organizations, products, domain names, email addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

This book expresses the author's views and opinions. The information contained in this book is provided without any express, statutory, or implied warranties. Neither the authors, Microsoft Corporation, nor its resellers, or distributors will be held liable for any damages caused or alleged to be caused either directly or indirectly by this book.

Contents

Introduction 4

Alternatives for Optimized Loading and Unloading Oracle Data 4

 Customized Script Component..... 5

 Third-party Components..... 12

Conclusion..... 12

Introduction

SQL Server 2008, 2008 R2, and 2012 (Enterprise and Developer 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(SQL.100).aspx) ([http://msdn.microsoft.com/en-us/library/ee470675\(SQL.100\).aspx](http://msdn.microsoft.com/en-us/library/ee470675(SQL.100).aspx)).

SQL Server 2005 and the non-Enterprise and non-Developer editions of SQL Server 2008, 2008 R2, and 2012 don't provide an out-of-the box option for bulk loading Oracle data.

- The fastload 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 and non-Developer editions of SQL Server 2008, 2008 R2, and 2012, 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 500GB) with complex referential integrity

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

Alternatives for Optimized Loading and Unloading Oracle Data

The following are alternatives for optimizing the loading of Oracle data.

Alternative	SQL Server Versions
Customized Script component	SQL Server 2005 and the non-Enterprise and non-Developer editions of SQL Server 2008, 2008 R2, and 2012.
Third-party components	SQL Server 2005 and the non-Enterprise and

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://tinyurl.com/7zuffuf) (<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.

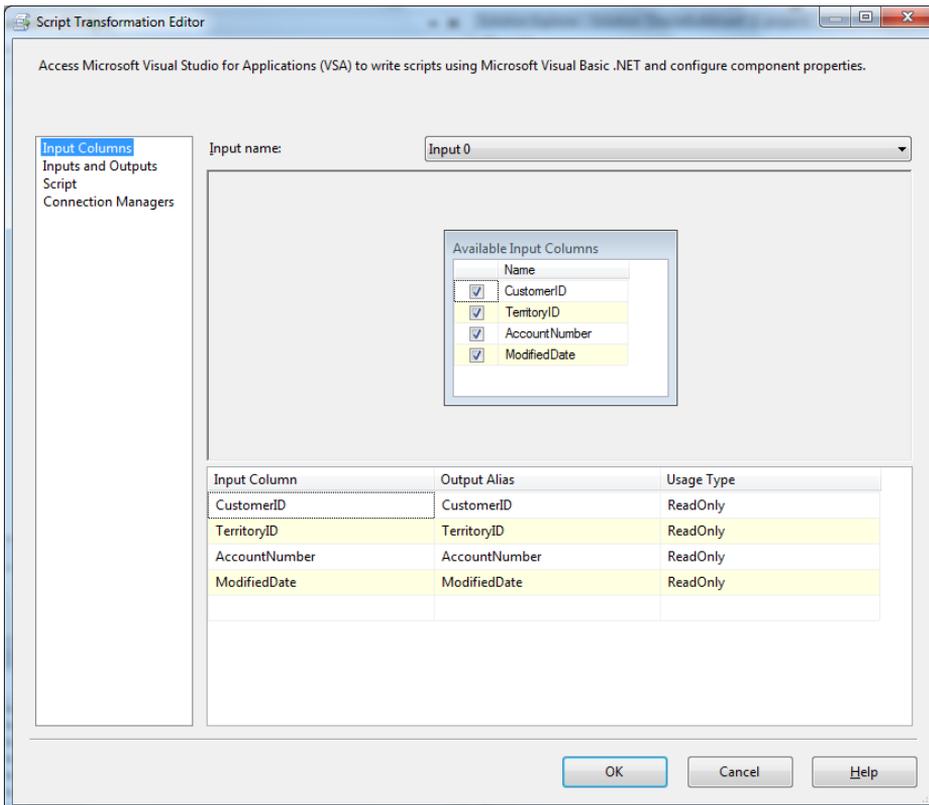
The [PreExecute](http://tinyurl.com/86e4exe) (<http://tinyurl.com/86e4exe>) method is overridden to create the OleDbParameter objects for each of the input columns. The parameters are added to the OleDbCommand Object, to configure the parameterized command that the destination will use to insert the data. In the example, the input columns are CustomerID, TerritoryID, AccountNumber, and ModifiedDate. Then, the database transaction is started.

The [AcquireConnections](http://tinyurl.com/7qkkqvq) (<http://tinyurl.com/7qkkqvq>) method is overridden to return a System.Data.OleDb.OleDbConnection from the connection manager that connects to the Oracle database.

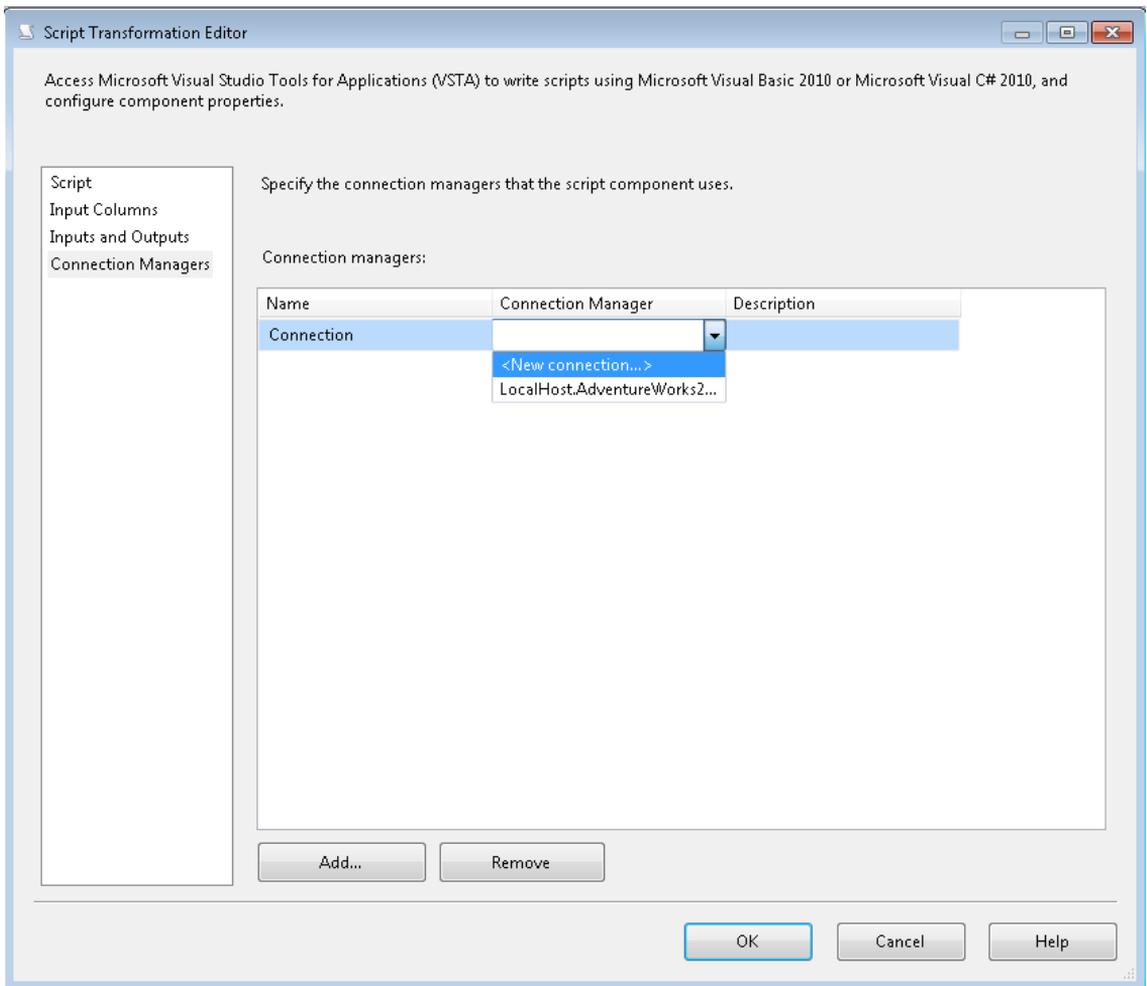
The [ProcessInputRow](http://tinyurl.com/8y7vnh5) (<http://tinyurl.com/8y7vnh5>) method is overridden to process the data in each input row as it passes through.

Toconfigure the Script component

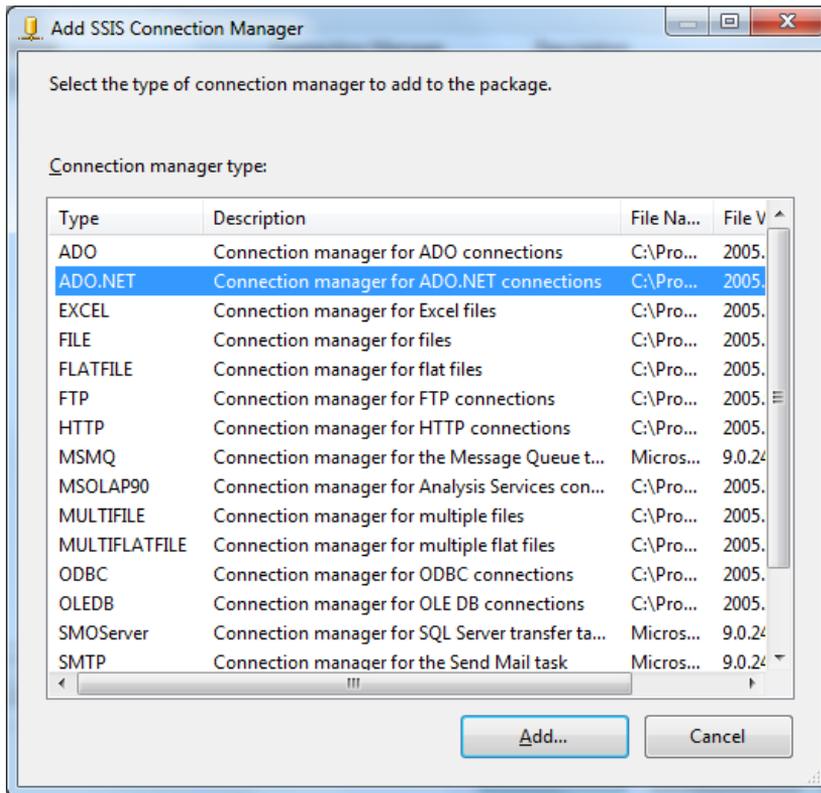
1. Add adata source to the package, such as an OLE DB Source. The data source should have fields that can be easily loaded into a target table. In this example, we're using the Customer table in the AdventureWorks database as the data source, and selecting the CustomerID, TerritoryID, AccountNumber, and ModifiedDate input columns.
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**.



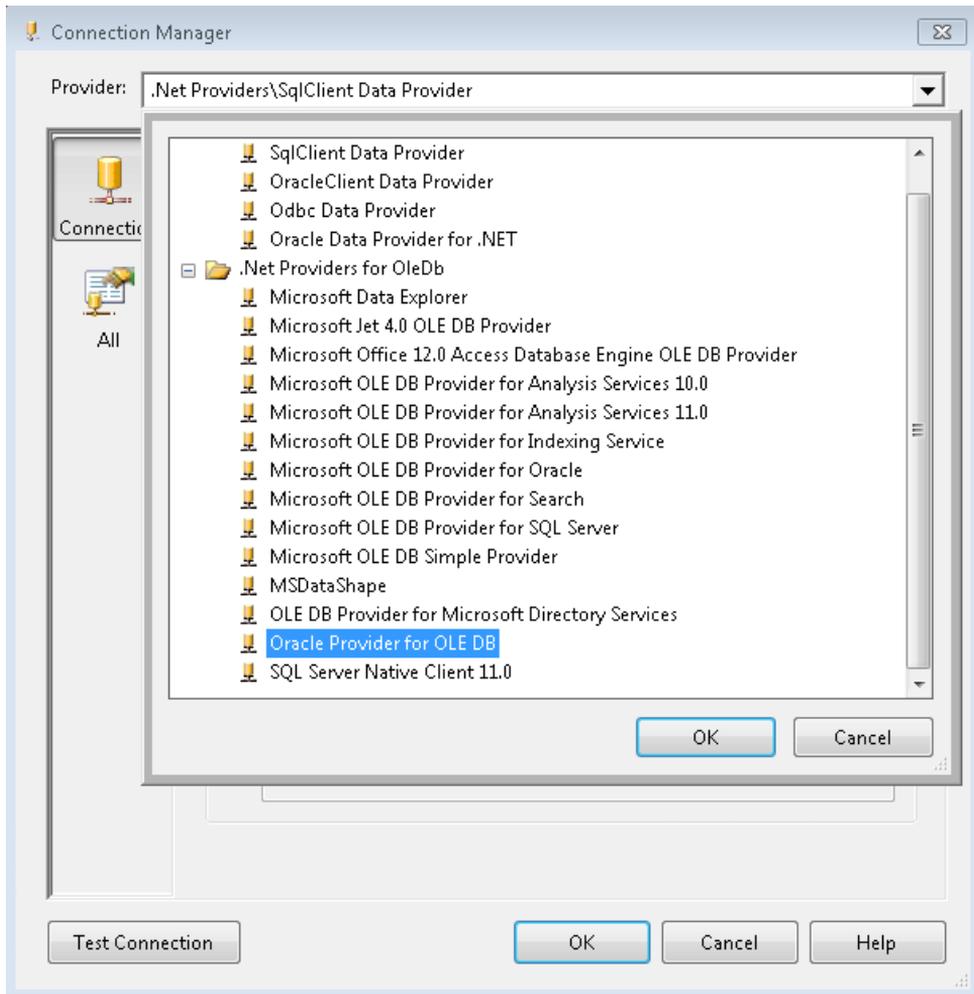
4. Click **Connection Managers** in the left-hand pane.



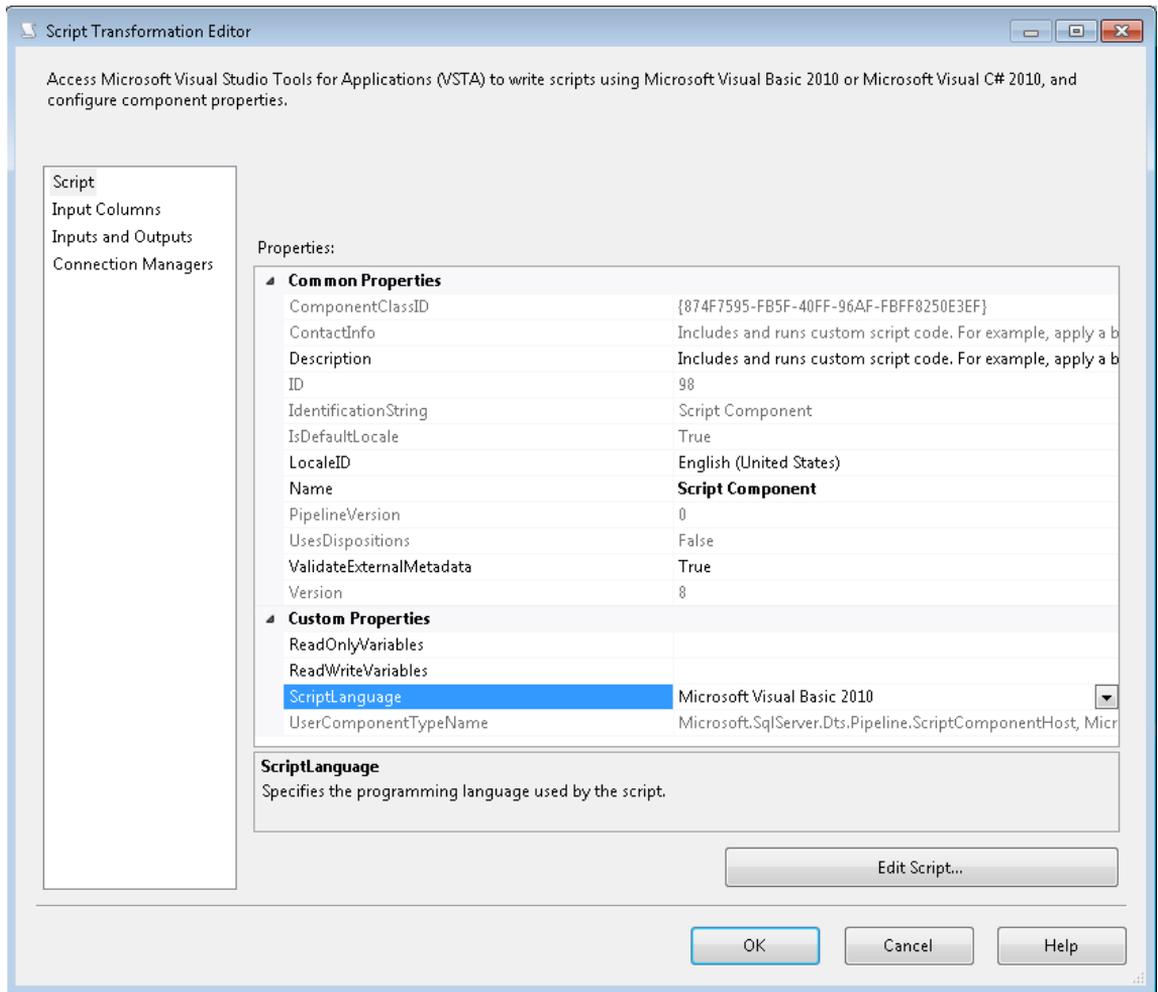
5. Click **Add**, and then select **<New connection>** in the **Connection Manager** field. The **Add SSIS Connection Manager** dialog box appears.
6. In the **Add SSIS Connection Manager** dialog box, select **ADO.NET** in the **Connection manager type** area, and then click **Add**.



7. In the **Configure ADO.NET Connection Manager** dialog box click **New** to create a new data connection for the connection manager.
8. In the **Connection Manager** dialog box, click the arrow next to the **Provider** drop-down list.



9. Expand the **.Net Providers for OleDb** folder, click **Oracle Provider for OLE DB**, and then click **OK**.
10. Click **Test Connection** to confirm the connection, and then click **OK**.
11. In the **Configure ADO.NET Connection Manager** dialog box, click the data connection you've created, and then click **OK**.
12. In the **Script Transformation Editor**, click **Input Columns** in the left-hand pane, and click the CustomerID, TerritoryID, AccountNumber, and ModifiedDate columns in the **Available Input Columns** box.
13. Click **Script** in the left-hand pane.



14. Confirm that the **ScriptLanguage** property value is Microsoft Visual Basic, and then Click **Edit Script**.

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

Add the following Visual Basic code.

```
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()
EndSub

Public Overrides Sub AcquireConnections(ByVal Transaction As Object)
    connMgr = Me.Connections.Connection
    oledbconn = CType(connMgr.AcquireConnection(Nothing),
OleDb.OleDbConnection)
EndSub

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()
EndWith

    row_count = row_count + 1
If (row_count Mod batch_size) = 0 Then
    oledbtran.Commit()
    oledbtran = oledbconn.BeginTransaction()
    oledbCmd.Transaction = oledbtran
EndIf
EndSub

Public Overrides Sub PostExecute()
 MyBase.PostExecute()
EndSub

```

```

PublicOverridesSub ReleaseConnections()
MyBase.ReleaseConnections()
EndSub
EndClass

```

15. 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](#) web site.
- Oracle Bulk Loader SSIS Connector from [Persistent](#). For more information, contact Persistent.
- Progress DataDirect Connect and DataDirect Connect64 components from Progress DataDirect. For more information, see the [DataDirect](#) web site.

Conclusion

SQL Server 2008, 2008 R2, and 2012 (Enterprise and Developer editions) support bulk loading Oracle data using SSIS packages.

For other SQL Server versions and editions, the following are alternatives for optimizing the loading of Oracle data when using SSIS packages.

Alternative	SQL Server versions and editions
Script component bulk loads data to Oracle using the Oracle OLE DB provider from Oracle	SQL Server 2005 and the non-Enterprise and non-Developer editions of SQL Server 2008, 2008 R2, and 2012.
Third-party components that connect to Oracle, from CozyRoc, Persistent, and DataDirect	SQL Server 2005 and the non-Enterprise and non-Developer editions of SQL Server 2008 and 2008 R2.

For more information:

[Connectivity and SQL Server 2005 Integration Services](http://msdn.microsoft.com/en-us/library/bb332055(SQL.90).aspx)([http://msdn.microsoft.com/en-us/library/bb332055\(SQL.90\).aspx](http://msdn.microsoft.com/en-us/library/bb332055(SQL.90).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 Server 2012: Microsoft Connectors V2.0 for Oracle and Teradata](http://www.microsoft.com/en-us/download/details.aspx?id=29283)

(<http://www.microsoft.com/en-us/download/details.aspx?id=29283>)

[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.](#)