



**SQL Server 数据库**

**导入/导出专家手册**

## SQL Server 数据库导入/导出专家手册

在我们建立一个数据库之后，想将分散在各处的不同类型的数据分类汇总在这个新建的数据库中时，就需要对数据进行导入与导出操作，作为数据库的基本操作之一，导入/导出对 DBA 来说也是一项极具挑战性的工作。SQL Server 作为一款主流数据库平台，提供了强大、丰富的数据导入导出功能。在本次技术手册中，我们将对这一部分进行深入的讨论，其中包括了批量数据导入、跨平台数据导入以及相关注意事项等内容。

### 数据导入/导出技巧

在 SQL Server 中主要有三种方式导入导出数据：使用 Transact-SQL 对数据进行处理；调用命令行工具 BCP 处理数据；使用数据转换服务 (DTS) 对数据进行处理。在本部分中，我们将介绍几种常用的导入导出技巧。

- ❖ SQL Server 数据库导入导出数据方式比较（一）
- ❖ SQL Server 数据库导入导出数据方式比较（二）
- ❖ 用 SSIS 和 Visual Studio 导出 SQL Server 数据到 Excel 文件（上）
- ❖ 用 SSIS 和 Visual Studio 导出 SQL Server 数据到 Excel 文件（下）
- ❖ 把 CSV 文件导入到 SQL Server 表中的方法

### 批量数据导入导出工具 BCP

SQL Server 中的批量复制工具程序 (Bulk Copy Program, BCP) 能让数据库管理员将数据批量导入表中或将数据从表中批量导入文档中。它还支持一些定义数据如何导出、导入到什么地方、加载哪些数据等选项。

- ❖ SQL Server 数据导入导出工具 BCP 介绍

- ❖ 用 BCP 工具导入和导出批量数据（上）
- ❖ 用 BCP 工具导入和导出批量数据（中）
- ❖ 用 BCP 工具导入和导出批量数据（下）

## SQL Server 专家指导

在进行 SQL Server 数据库导入导出操作时，倾听专家的建议是十分必要的。在本部分中，我们将向您推荐一些专家的经验和一些常见的错误，希望能对您有所帮助。

- ❖ 从 SQL Server 中导出至 Oracle 数据库
- ❖ 错误导入 Excel 数据到 SQL Server 2005
- ❖ SQL Server 中的导出表以及使用 FULL OUTER JOIN 的技巧
- ❖ 如何使用 BULK insert 将数据导入 SQL Server 数据库？

## SQL Server 数据库导入导出数据方式比较（一）

在我们建立一个数据库时，并且想将分散在各处的不同类型的数据库分类汇总在这个新建的数据库中时，尤其是在进行数据检验、净化和转换时，将会面临很大的挑战。幸好 SQL Server 为我们提供了强大、丰富的数据导入导出功能，并且在导入导出的同时对数据进行灵活的处理。

在 SQL Server 中主要有三种方式导入导出数据：使用 Transact-SQL 对数据进行处理；调用命令行工具 BCP 处理数据；使用数据转换服务 (DTS) 对数据进行处理。这三种方法各有其特点，下面就它们的主要特点进行比较。

### 一、使用方式的比较

#### 1. 使用 Transact-SQL 进行数据导入导出

我们很容易看出，Transact-SQL 方法就是通过 SQL 语句方式将相同或不同类型的数据库中的数据互相导入导出或者汇集在一处的方法。如果是在不同的 SQL Server 数据库之间进行数据导入导出，那将是非常容易做到的。一般可使用 SELECT INTO FROM 和 INSERT INTO。使用 SELECT INTO FROM 时 INTO 后跟的表必须存在，也就是说它的功能是在导出数据之前先建立一个空表，然后再将源表中的数据导入到新建的空表中，这就相当于表的复制（并不会复制表的索引等信息）。而 INSERT INTO 的功能是将源数据插入到已经存在的表中，可以使用它进行数据合并，如果要更新已经存在的记录，可以使用 UPDATE。

```
SELECT * INTO table2 FROM table1
```

--table1 和 table2 的表结构相同

```
INSERT INTO table2 SELECT * FROM table3
```

--table2 和 table3 的表结构相同

当在异构数据库之间的进行数据导入导出时，情况会变得复杂得多。首先要解决的是如何打开非 SQL Server 数据库的问题。

在 SQL Server 中提供了两个函数可以根据各种类型数据库的 OLE DB Provider 打开并操作这些数据库，这两个函数是 OPENDATASOURCE 和 OPENROWSET。它们的功能基本上相同，不同之处主要有两点。

(1) 调用方式不同。

OPENDATASOURCE 的参数有两个，分别是 OLE DB Provider 和连接字符串。使用 OPENDATASOURCE 只相当于引用数据库或者是服务（对于 SQL Server、Oracle 等数据库来说）。要想引用其中的数据表或视图，必须在 OPENDATASOURCE(...) 后进行引用。

在 SQL Server 中通过 OPENDATASOURCE 查询 Access 数据库 abc.mdb 中的 table1 表

```
SELECT * FROM OPENDATASOURCE('Microsoft.Jet.OLEDB.4.0',  
'Provider=Microsoft.Jet.OLEDB.4.0;Data Source=abc.mdb;Persist Security  
Info=False')...
```

table1

OPENROWSET 相当于一个记录集，可以将直接当成一个表或视图使用。

在 SQL Server 中通过 OPENROWSET 查询 Access 数据库 abc.mdb 中的 table1 表

```
SELECT * FROM OPENROWSET('Microsoft.Jet.OLEDB.4.0', 'abc.mdb';  
'admin';'', 'SELECT * FROM table1')
```

(2) 灵活度不同。

OPENDATASOURCE 只能打开相应数据库中的表或视图，如果需要过滤的话，只能在 SQL Server 中进行处理。而 OPENROWSET 可以在打开数据库的同时对其进行过滤，如上面的例子，在 OPENROWSET 中可以使用 SELECT \* FROM table1 对 abc.mdb 中的数据表进行查询，而 OPENDATASOURCE 只能引用 table1，而无法查询 table1。因此，OPENROWSET 比较 OPENDATASOURCE 更加灵活。

---

(作者: 杨延成 来源: 赛迪网)

原文标题: SQL Server 数据库导入导出数据方式比较 (一)

链接: [http://www.searchdatabase.com.cn/showcontent\\_14963.htm](http://www.searchdatabase.com.cn/showcontent_14963.htm)

## SQL Server 数据库导入导出数据方式比较（二）

### 2. 使用命令行 BCP 导入导出数据

很多大型的系统不仅仅提供了友好的图形用户接口，同时也提供了命令行方式对系统进行控制。在 SQL Server 中除了可以使用 SQL 语句对数据进行操作外，还可以使用一个命令行工具 BCP 对数据进行同样的操作。BCP 是基于 DB-Library 客户端库的工具。它的功能十分强大，BCP 能够以并行方式将数据从多个客户端大容量复制到单个表中，从而大大提高了装载效率。但在执行并行操作时要注意的只有使用基于 ODBC 或 SQL OLE DB 的 API 的应用程序才可以执行将数据并行装载到单个表中的操作。

BCP 可以将 SQL Server 中的数据导出到任何 OLE DB 所支持的数据库的，如下面的语句是将 authors 表导出到 excel 文件中。

```
bcp pubs.dbo.authors out c:\templ.xls
```

```
-c -q -S"GNETDATA/GNETDATA" -U"sa" -P"password"
```

BCP 不仅能够通过命令行执行，同时也可以通过 SQL 执行，这需要一个系统存储过程 xp\_cmdshell 来实现，如上面的命令可改写为如下形式。

```
EXEC master..xp_cmdshell 'bcp pubs.dbo.authors out
```

```
c:\templ.xls -c -q -S"GNETDATA/GNETDATA" -U"sa" -P"password"
```

### 3. 使用数据转换服务(DTS)导入导出数据

DTS 是 SQL Server 中导入导出数据的核心，它除有具有 SQL 和命令行工具 BCP 相应的功能外，还可以灵活地通过 VBScript、JScript 等脚本语言对数据进行检验、净化和转换。

SQL Server 为 DTS 提供了图形用户接口，用户可以使用图形界面导入导出数据，并对数据进行相应的处理。同时，DTS 还以 com 组件的形式提供编程接口，也就是说任何支持 com 组件的开发工具都可以利用 com 组件使用 DTS 所提供的功能。DTS 在 SQL Server 中可

以保存为不同的形式，可以是包的形式，也可以保存成 Visual Basic 源程序文件，这样只要在 VB 中编译便可以使用 DTS com 组件了。

DTS 和其它数据导入导出方式最大的不同就是它可以在处理数据的过程中对每一行数据进行深度处理。以下是一段 VBScript 代码，这段代码在处 DTS 理每一条记录时执行，DTSDestination 表示目标记录，DTSSource 表示源记录，在处理“婚姻状况”时，将源记录中的“婚姻状况”中的 0 或 1 转换成目标记录中“已婚”或“未婚”。

```
Function Main()  
  
DTSDestination("姓名") = DTSSource("姓名")  
  
DTSDestination("年龄") = DTSSource("年龄")  
  
If DTSDestination("婚姻状况") = 1 Then  
  
DTSDestination("婚姻状况") = "已婚"  
  
Else  
  
DTSDestination("婚姻状况") = "未婚"  
  
End If  
  
Main = DTSTransformStat_OK  
  
End Function
```

上述的三种数据导入导出方法各有其利弊，它们之间的相互比较如图 1 所示。

## 二、性能的比较

使用 Transact-SQL 方式。如果是 SQL Server 数据库之间的导入导出，速度将非常快，但是使用 OPENDATASOURCE 和 OPENROWSET 方法利用 OLE DB Provider 打开并操作数据库时速度会慢一些。



使用 BCP 命令方式。如果不需要对数据进行验证等操作的话，使用它还是非常快的，这是因为它的内部使用 c 接口的 DB-library，所以在操作数据库时速度有很大的提升。

使用 DTS 方式导数据应该是最好的方式了。由于它整合了 Microsoft Universal Data Access 技术与 Microsoft ActiveX 技术，因此不仅可以灵活地处理数据，而且在数据导入导出的效率是非常高的。

## 总结

SQL Server 提供了丰富的数据导入导出方法，这给我们提供了更多的选择，但是这又会给我们带来一个新问题：如何根据具体情况选择合适的数据库导入导出方法呢？我在这里提供一些个人的建议，希望能对读者起到一定的指导作用。

如果是在 SQL Server 数据库之间进行数据导入导出时，并且不需要对数据进行复杂的检验，最好使用 Transact-SQL 方法进行处理，因为在 SQL Server 数据库之间进行数据操作时，SQL 是非常快的。当然，如果要进行复杂的操作，如数据检验、转换等操作时，最好还是使用 DTS 进行处理，因为 DTS 不光导数据效率高，而且能够对数据进行深度控制。但是 DTS 的编程接口是基于 com 的，并且这个接口十分复杂，因此，使用程序调用 DTS 将变也会变得很复杂，因此，当数据量不是很大，并且想将数据导入导出功能加入到程序中，而且没有复杂的数据处理功能时，可以使用 OPENDATASOURCE 或 OPENROWSET 进行处理。

BCP 命令并不太适合通过程序来调用，如果需要使用批量的方式导数据，可以通过批处理文件调用 BCP 命令，这样做即不需要编写大量的程序，也无需在企业管理器中通过各种操作界面的切换来进行数据导入导出。因此，它比较适合在客户端未安装企业管理器或使用 SQL Server Express 时对数据进行快速导入导出的场合。

(作者：杨延成 来源：赛迪网)

原文标题：SQL Server 数据库导入导出数据方式比较（二）

链接：[http://www.searchdatabase.com.cn/showcontent\\_14964.htm](http://www.searchdatabase.com.cn/showcontent_14964.htm)

---

## 用 SSIS 和 Visual Studio 导出 SQL Server 数据到 Excel 文件（上）

---

本文选自 Jayaram Krishnaswamy 编写的《在 Visual Studio 2005 中使用 SQL Server 集成服务 (SSIS) 初学者指南》一书，讲述了创建 SSIS 包来实现把 SQL Server 2005 数据库表中数据导入到微软 Excel 电子表格的基础知识。

你也会学到一些关于使用字符映射数据转换组件的基础知识。在动手练习过程中，你会练习从 SQL Server 2005 表转移数据到微软 Excel 2003 数据表文件中。你将使用由“基于连接管理的连接到 SQL Server 2005 服务器的源组件”和“连接到 Excel 连接管理器的 Excel 目标组件”组成的数据流任务。

为了执行下列步骤，你需要一个源数据组件和一个目标数据组件：源数据是从 MyNorthwind 数据库中提取的（该数据源只是 Northwind 数据库重命名后的版本），该数据库运行于 SQL Server 2005 服务器；我们的目标是加载该数据到微软 Excel 2003 数据表文件中。你还需要建立一条路径来连接它们。此外，你也还要插入一个字符映射数据流任务，它会转换其中一个数据字段内部的文本，这样一来，该列所有字符转换后都变成了大写形式。

### 把 SQL Server 数据转移到 Excel 文件

第一步：创建 SQL Server BI 项目并添加一个数据流任务

第二步：配置数据读取组件（DataReader）的连接管理器

第三步：为 SQL Server 数据设置数据读取源组件

第四步：给 SQL Server 数据转换组件设置字符映射机制

第五步：添加 Excel 目标组件并创建连接到字符映射组件的路径

第六步：配置微软 Excel 目标组件

第七步：测试从 SQL Server 表到 Excel 的数据转移

## 第一步：创建 SQL Server BI 项目并添加一个数据流任务

在这一部分，你将创建一个商业智能项目，然后修改默认包对象的名称。既然我们要做的事与数据有关，你将会添加一个数据流任务，还会给数据流中添加一个数据读取组件（DataReader）。

按照第二章或第三章的介绍，创建一个商业智能项目 Ch5。

把默认包名称 Package.dtsx 改为 TableToXls.dtsx。

从工具箱中拖拽一个数据流任务组件到控制流页。

点击打开数据流标签页，该标签显示了数据流页。

现在你就可以访问工具箱中的数据流选项了。这些选项有：数据流源组件，数据流目标组件，数据流转换组件（参考第一章）。

从数据流源组件组中拖拽数据读取（DataReader）源组件到数据流页面。

## 第二步：配置数据读取组件（DataReader）的连接管理器

关于配置连接到本地 SQL Server 2005 数据读取（DataReader）源组件的方法已介绍过了。这里只列出几个与本章需要展示的内容相关的步骤。

在连接管理器页面下方单击右键，然后从右键菜单中选择“新建 ADO.Net 连接……”。

如果你是在读过第四章（并做过其中的练习）以后才进行本章的练习，你会看见配置 ADO.NET 连接管理器屏幕显示之前配置的连接管理器。如果你需要创建新的连接配置，你可以按照上一章中讲的步骤进行。

在配置 ADO.NET 连接管理器窗体中点击确定（OK）按钮。

连接管理器“localhost.MyNorthwind.sa”就被添加到连接管理器页面了。

## 第三步：为 SQL Server 数据设置数据读取（DataReader）源组件

在下拉菜单中右键点击数据读取源组件 (DataReader)。

在下拉菜单中选择编辑 (Edit)，就可以打开数据读取 (DataReader) 源组件高级编辑器。首先，你需要指定一个数据读取组件 (DataReader) 可用的连接管理器。

在刚显示的数据读取 (DataReader) 源组件高级编辑器中，点击连接管理器标签。

点击列表标题连接管理器下面的空白区域 (灰色区域)，在这里，你会看到你在第一步中添加的连接管理器。

选择这个连接管理器。

接下来，点击组件属性标签打开数据读取组件 (DataReader) 属性。在这里，你会发现它要求填写一个 SQLCommand (目前唯一空白的填充项)。

点击它旁边的省略号按钮会显示一个文本编辑器，在那里你可以写入 SQL 命令。

你可以直接写上下面这句 SQL 命令：

```
SELECT CustomerID, CompanyName, Address, City, PostalCode
```

```
FROM Customers
```

点击刷新按钮。

这个查询意思是让 DataReader 读取 5 列字段的数据。下图是从 SQL Server 2005 Management Studio 中得到的一个表数据示例。如果你能回想起来，上一章我们也用过同样的这些字段列。

点击列映射标签。

打开的列映射页面中会显示 DataReader 输出的列。

在编辑器的最后一个标签“输入和输出属性”中，你可以从外部列，输出列和 DataReader 错误输出中添加或者删除项目。在本教程中，我们不做修改。



The screenshot shows a SQL query window with the following query: `Select CustomerID, CompanyName, Address, City, PostalCode from Customers`. The results are displayed in a table with 5 columns: CustomerID, CompanyName, Address, City, and PostalCode. There are 4 rows of data.

	CustomerID	CompanyName	Address	City	PostalCode
1	ALFKI	Alfreds Futterkiste	Obers Str. 57	Berlin	12209
2	ANATR	Ana Trujillo Emparedados y ...	Avda. de la Constitución ...	México D.F.	05021
3	ANTON	Antonio Moreno Taquería	Mataderos 2312	México D.F.	05023
4	AROUT	Around the Horn	120 Hanover Sq.	London	WA1 1DP

点击上面窗口中的确定（OK）按钮。

到此为止，配置 DataReader 从 SQL Server 2005 数据库中提取五个字段列的工作就完成了。

(作者: Jayaram Krishnaswamy 译者: 冯昀晖 来源: TT 中国)

原文标题: 用 SSIS 和 Visual Studio 导出 SQL Server 数据到 Excel 文件（上）

链接: [http://www.searchdatabase.com.cn/showcontent\\_22148.htm](http://www.searchdatabase.com.cn/showcontent_22148.htm)

## 用 SSIS 和 Visual Studio 导出 SQL Server 数据到 Excel 文件（下）

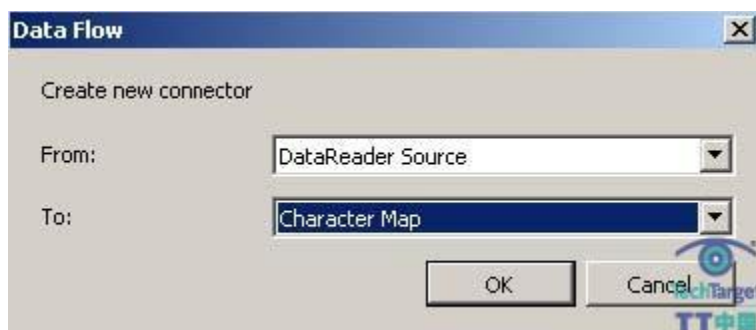
### 第四步：为 SQL Server 数据转换组件设置字符映射

字符映射转换组件是在第一章中讲到的，但是在这里我们要体验一下它的转换功能。转换组件处理接收到的文本字符串，输出为转换后的字符串。例如：在上面我们看到的图片中，公司名是大小写混合的，使用转换组件后，我们会在数据写到 Excel 之前，把出现在公司名这一列的所有字符转换为大写——例如：Alfreds Futterkiste 会转换为 ALFREDS FUTTERKISTE。

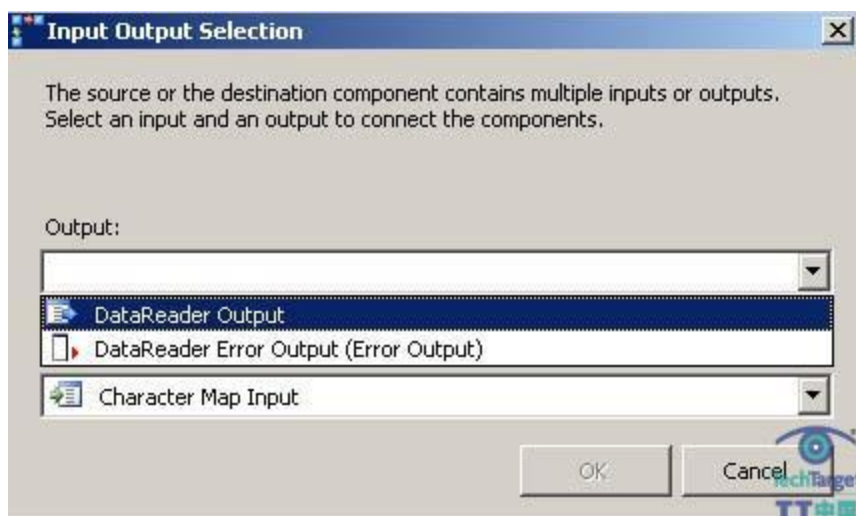
在工具箱的数据流转换组件组里选择字符映射数据流项，把它拖拽到数据流页面的设计画布上。

在 DataReader 源组件上点击右键，在右键菜单项中点击添加路径。

在弹出显示的数据流窗口中，选择转换目标为字符映射。如下图所示：



在上面的窗口中点击确定（OK），此时会显示下面的新窗口。



在这里你需要指定这个组件的输出项，从下拉列表里选择即可。

从下拉列表里选择 DataReader 输出源。

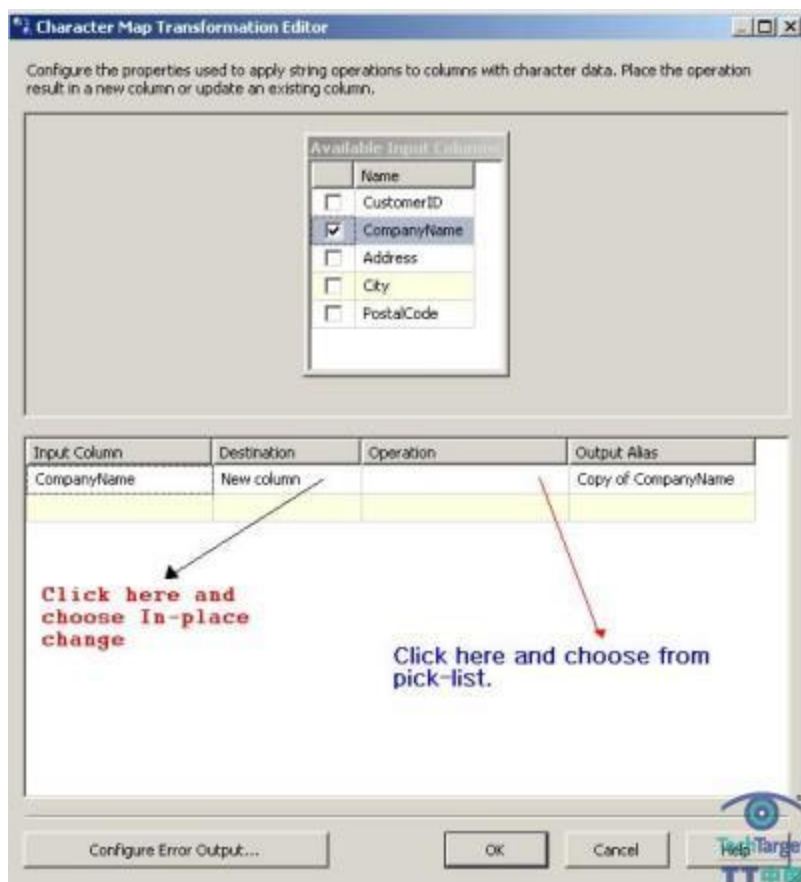
现在，在本教程中不涉及任务错误处理。选好上面的选项后，点击确定（OK）按钮就生效了。

在上面的窗口中点击确定（OK）按钮。

这就建立了从 DataReader 源到字符映射数据流组件的通道。通道建立好了，但仍然需要进一步配置。

在字符映射组件上点击右键，在右键菜单中选择编辑。

此时会弹出如下图所示的字符映射编辑器。选中列公司名前面的复选框。



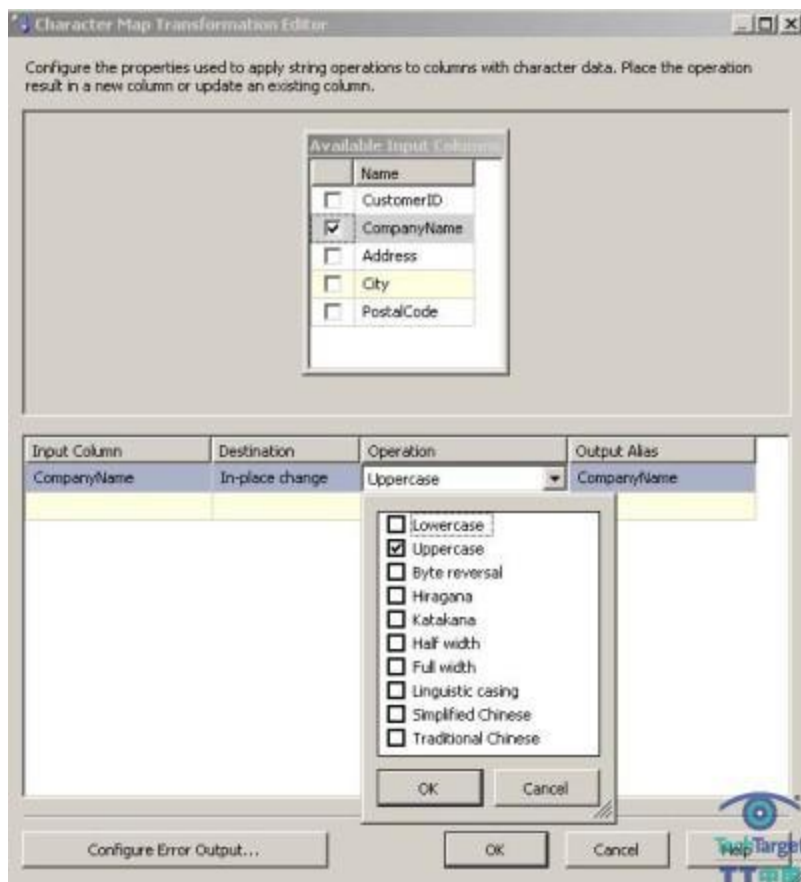
如果如上图所示的默认就被选中了，那么输出部分就会多出来一列。我们从下拉列表中选择 “In-place change” 选项。

点击上面窗体中目标列下面的新建列单元格。在下拉列表中选择 “In-place change”。

在操作（Operation）下面点击，在出现的下拉列表中选中大写。如下图所示。

输出别名不用修改，因为它在选中公司名时已经被关联修改了。





在选择列表中点击确定（OK）按钮，然后点击字符映射转换编辑器中的确定（OK）按钮。

字符映射配置就完成了。

#### 第五步：添加 Excel 目标组件并给字符映射组件创建路径

在这一步中，我们将添加一个 Excel 目标组件。然后，我们会建立一个从字符映射组件到 Excel 目标组件的路径。

从工具箱的数据流目标组件组选择 Excel 目标组件，把它添加到数据流页面上。

可以通过在工具箱中双击该组件或者通过拖拽操作来完成这一步。在字符映射组件上右击，在弹出的邮件菜单中选择添加路径。

此时会打开一个数据流窗口，在这里你可以建立一个数据流路径，显示“From:”路径为字符映射。

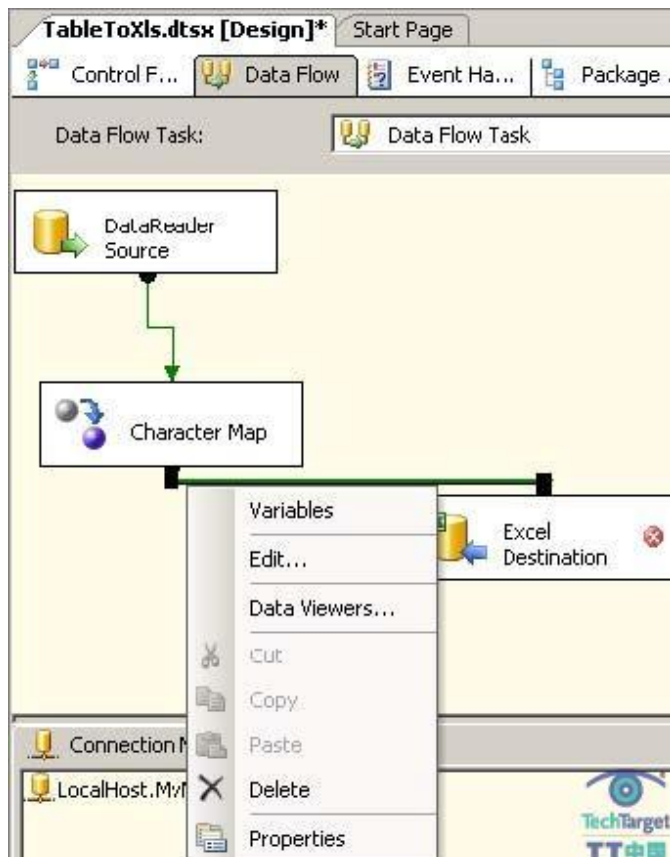
点击“To:”旁边的下拉列表，会显示 Excel 目标组件和 DataReader 源组件。

选择 Excel 目标组件，然后点击屏幕上的确定（OK）按钮。

此时会打开一个输入输出选择窗口，窗口中会显示可用的输出和输入。输出窗口是空的，而输入显示 Excel 目标输入。路径应该从字符映射到 Excel 目标输入连接。

选好上面的选项，然后点击屏幕上的确定（OK）按钮。

你会看到有一条绿色的线从字符映射数据流组件连向 Excel 目标组件，如下图所示。建立连接路径的过程也可以被简化，可以通过在数据流页面上从字符映射到 Excel 目标组件拖拽一条绿箭头完成，与前面的做法一样的效果。如下图所示，你可以通过右击这条绿色箭头线，在右键菜单中选择“编辑……”来编辑路径。



## 第六步：配置微软 Excel 目标组件

前面步骤中的绿箭头代表了数据传递的路径。Excel 目标组件也需要一个连接管理器。

Excel 目标组件连接在连接管理器中定义的链接属性来连接你硬盘上的微软 Excel 文件。

在 Excel 目标组件上右击，在右键菜单中选择编辑。

此时会弹出 Excel 目标组件编辑器。Excel 需要一个 OLEDB 连接管理器，如果（你或者其他之前的用户）没有配置连接管理器，下拉列表就是空的。

点击新建按钮。

Excel 连接管理器窗口如下面的图片显示。这里，你需要使用浏览按钮选择一个 Excel 文件作为转移数据的目标。包程序运行时，数据会被写到这个目标文件里。

打开 Windows 资源管理器，在 C 盘创建一个 Excel 文件。在本教程中，我们创建一个名为 TableToXls.xls 的 Excel 文件。

除了连接到一个已经存在的文件，Excel 连接管理器还支持在你用浏览按钮选择的本机文件夹创建文件。

现在就用浏览按钮找到刚刚创建的文件，并选中该文件。

在 Excel 连接管理窗口中点击确定（OK）按钮。

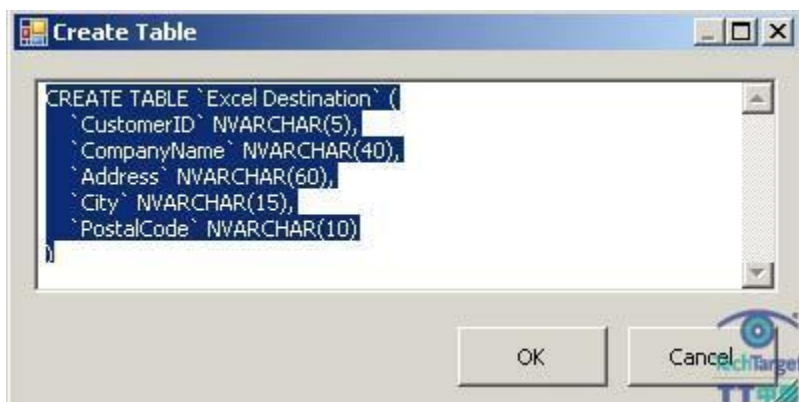


对于数据访问模式这一项，采用默认模式，表或者视图。

你必须指定要使用的 Excel sheet 页的名称。（不要点击下拉列表来指定 sheet 页，下拉列表中会显示新创建 Excel 文件的三个 Excel sheet 页，它们都只有一列。）

点击新建按钮。

你现在正在创建一个新的 Excel sheet 页。弹出的创建表格的窗口中显示了即将导入组件的列，如下图所示。

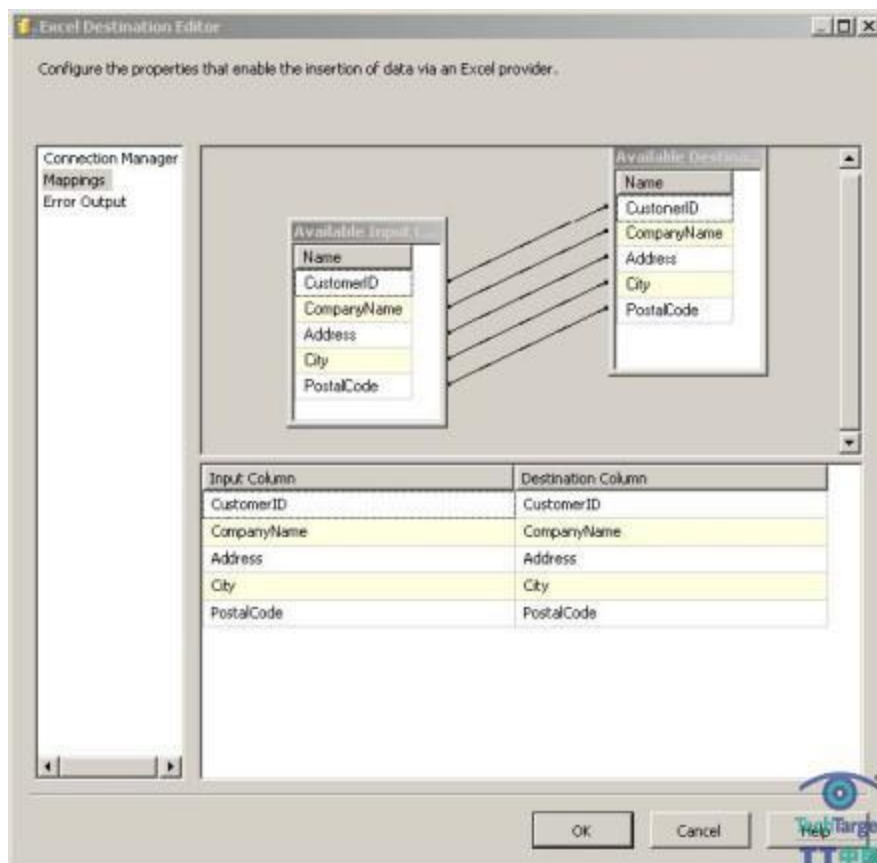


点击上图中的确定（OK）按钮。

一个新的名为 Excel Destination 的 Excel sheet 也会被添加到 TableToXLS.xls 文件中。如果你打开检查一下该文件（TableToXLS.xls），你会看到列表的表头都已经添加到该页中了。

在 Excel 目标组件编辑器面板左侧点击映射，会在右侧显示从输入到输出的映射。

这里显示了从字符映射数据流组件导向目标文件的所有列，如下图所示。



点击该窗口的确定（OK）按钮。现在，包已经完全配置好，可以运行了。

### 第七步：测试从 SQL Server 数据库表向 Excel sheet 页执行数据转移

在解决方案浏览器中右击，在右键菜单中点击执行包。

数据流页面上的三个组件都变成了绿色，这表示包执行成功，没有错误。你可以检查“Canvas”中的进度标签，那里会显示包执行过程的所有细节。

现在打开 TableToXls.xls 文件看看。

下图中显示了该文件中的几行数据。注意字符映射数据转换组件已经把公司名这一列的所有字符转换为大写了。

Microsoft Excel - TableToXls.xls

File Edit View Insert Format Tools Data Window Help

B1 'CompanyName'

	A	B	C	D	E
1	CustomerID	CompanyName	Address	City	PostalCode
2	ALFKI	ALFREDS FUTTERKISTE	Obere Str.	Berlin	12209
3	ANATR	ANA TRUJILLO EMPAREDAD	Avda. de la	México D.	05021
4	ANTON	ANTONIO MORENO TAQUER	Mataderos	México D.	05023
5	AROUT	AROUND THE HORN	120 Hanov	London	WA1 10P
6	BERGS	BERGLUNDS SNABBKÖP	Berguvsvä	Luleå	S-958 22
7	BLAUS	BLAUER SEE DELIKATESSE	Forsterstr.	Mannheim	68306

(作者: Jayaram Krishnaswamy 译者: 冯昀晖 来源: TT 中国)

原文标题: 用 SSIS 和 Visual Studio 导出 SQL Server 数据到 Excel 文件 (下)

链接: [http://www.searchdatabase.com.cn/showcontent\\_22150.htm](http://www.searchdatabase.com.cn/showcontent_22150.htm)

## 把 CSV 文件导入到 SQL Server 表中的方法

有时候我们可能会把 CSV 中的数据导入到某个数据库的表中，比如做报表分析的时候。

对于这个问题，我想一点也不难不倒程序人员吧！但是要是 SQL Server 能够完成这个任务，岂不是更好！

对，SQL Server 确实有这个功能。

首先先让我们看一下 CSV 文件，该文件保存在我的 D:盘下，名为 csv.txt，内容是：



现在就是 SQL Server 的关键部分了；

我们使用的是 SQL Server 的 BULK INSERT 命令，关于该命令的详细解释，请点击[此处](#)；

我们先在 SQL Server 中建立用于保存该信息的一张数据表，

```
CREATE TABLE CSVTable(
```

```
    Name NVARCHAR(MAX),
```

```
    Email NVARCHAR(MAX),
```

```
    Area NVARCHAR(MAX)
```

)

然后执行下面的语句：

```
BULK INSERT CSVTable
```

```
FROM 'D:\csv.txt'
```

```
WITH(
```

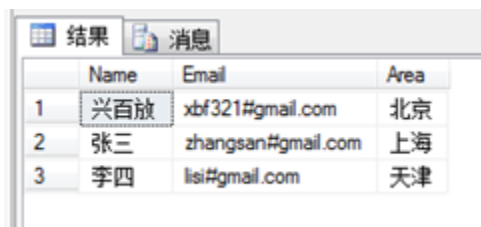
```
FIELDTERMINATOR = ',',
```

```
ROWTERMINATOR = '\n'
```

)

```
SELECT * FROM CSVTable
```

按 F5, 执行结果如下：



	Name	Email	Area
1	兴百放	xbf321#gmail.com	北京
2	张三	zhangsan#gmail.com	上海
3	李四	lisi#gmail.com	天津

怎么样?是不是比用程序简单!

但是现在有几个问题需要考虑一下：

1, CSV 文件中有的列值是用双引号, 有的列值则没有双引号:





如果再次运行上面的语句，得到结果就和上一个结果不同了：

结果			
	Name	Email	Area
1	"兴百放"	xbf321#gmail.com	"北京"
2	"张三"	zhangsan#gmail.com	"上海"
3	"李四"	lisi#gmail.com	"天津"

其中有的列就包含双引号了，这应该不是我们想要的结果，要解决这个问题，我们只能利用临时表了，先把 CSV 导入到临时表中，然后在从这个临时表中导入到最终表的过程中把双引号去掉。

2, CSV 文件的列值全部是由双引号组成的：



这个问题要比上一个稍微复杂点，除了要先把 CSV 文件导入到临时表中，还必须修改一下在把 CSV 文件导入到临时表的代码：

```
BULK INSERT CSVTable
FROM 'D:\csv.txt'
WITH(
    FIELDTERMINATOR = '"',
    ROWTERMINATOR = '\n'
)
SELECT * FROM CSVTable
```

注意圈中的部分。

3, CSV 文件的列要多于数据表的列:



而我们的数据表只有三列，如果在执行上面的导入代码，会产生什么结果呢？

结果就是：

结果		消息
Name	Email	Area
1 兴百放	xbf321@gmail.com	北京,男,100081
2 张三	zhangsan@gmail.com	上海,女,123456
3 李四	lisi@gmail.com	天津,男,456789

它把后边的全部放在了 Area 列中了，要处理这个问题，其实也很简单，就是我们要把想要的列值在数据表中都按顺序建立一列，而把不需要的列值，也在数据表中建立一个，只不过只是一个临时列，在把这个数据表导入到最终表的时候，忽略这个临时列就行了。

(作者: 兴百放 来源: Cnblog)

---

原文标题：把 CSV 文件导入到 SQL Server 表中的方法

链接：[http://www.searchdatabase.com.cn/showcontent\\_32666.htm](http://www.searchdatabase.com.cn/showcontent_32666.htm)

## SQL Server 数据导入导出工具 BCP 介绍

BCP 是 SQL Server 中负责导入导出数据的一个命令行工具，它是基于 DB-Library 的，并且能以并行的方式高效地导入导出大批量的数据。BCP 可以将数据库的表或视图直接导出，也能通过 SELECT FROM 语句对表或视图进行过滤后导出。在导入导出数据时，可以使用默认值或是使用一个格式文件将文件中的数据导入到数据库或将数据库中的数据导出到文件中。

下面将详细讨论如何利用 BCP 导入导出数据。

### 1. BCP 的主要参数介绍

BCP 共有四个动作可以选择。

#### (1) 导入。

这个动作使用 in 命令完成，后面跟需要导入的文件名。

#### (2) 导出。

这个动作使用 out 命令完成，后面跟需要导出的文件名。

#### (3) 使用 SQL 语句导出。

这个动作使用 queryout 命令完成，它跟 out 类似，只是数据源不是表或视图名，而是 SQL 语句。

#### (4) 导出格式文件。

这个动作使用 format 命令完成，后面跟格式文件名。

下面介绍一些常用的选项：

```
-f format_file
```

`format_file` 表示格式文件名。这个选项依赖于上述的动作，如果使用的是 `in` 或 `out`，`format_file` 表示已经存在的格式文件，如果使用的是 `format` 则表示是要生成的格式文件。

`-x`

这个选项要和 `-f format_file` 配合使用，以便生成 xml 格式的格式文件。

`-F first_row`

指定从被导出表的哪一行导出，或从被导入文件的哪一行导入。

`-L last_row`

指定被导出表要导到哪一行结束，或从被导入文件导数据时，导到哪一行结束。

`-c`

使用 `char` 类型做为存储类型，没有前缀且以 `"\t"` 做为字段分割符，以 `"\n"` 做为行分割符。

`-w`

和 `-c` 类似，只是当使用 Unicode 字符集拷贝数据时使用，且以 `nchar` 做为存储类型。

`-t field_term`

指定字符分割符，默认是 `"\t"`。

`-r row_term`

指定行分割符，默认是 `"\n"`。

`-S server_name[ \instance_name]`

指定要连接的 SQL Server 服务器的实例，如果未指定此选项，BCP 连接本机的 SQL Server 默认实例。如果要连接某台机器上的默认实例，只需要指定机器名即可。

-U login\_id

指定连接 SQL Sever 的用户名。

-P password

指定连接 SQL Server 的用户名密码。

-T

指定 BCP 使用信任连接登录 SQL Server。如果未指定-T，必须指定-U 和-P。

-k

指定空列使用 null 值插入，而不是这列的默认值

## 2. 如何使用 BCP 导出数据

(1) 使用 BCP 导出整个表或视图。

BCP AdventureWorks.sales.currency out c:\currency1.txt -c -U"sa" -P"password" --使用密码连接

或

BCP AdventureWorks.sales.currency out c:\currency1.txt -c -T --使用信任连接

下面是上述命令执行后的输出结果

Starting copy...

105 rows copied.

Network packet size (bytes) : 4096

Clock Time (ms.) Total : 10 Average : (10500.00 rows per sec.)

下面是 currency1.txt 的部分内容

```
AED Emirati Dirham 1998-06-01 00:00:00.000
```

```
AFA Afghani 1998-06-01 00:00:00.000
```

```
... ..
```

```
... ..
```

```
ZWD Zimbabwe Dollar 1998-06-01 00:00:00.000
```

在使用密码登录时需要将-U 后的用户名和-P 后的密码加上双引号。

注：BCP 除了可以在控制台执行外，还可以通过调用 SQL Server 的一个系统存储过程 xp\_cmdshell 以 SQL 语句的方式运行 BCP。如上述第一条命令可改写为

```
EXEC master..xp_cmdshell 'BCP AdventureWorks.sales.currency out  
c:\currency1.txt -c -U"sa" -P"password"
```

执行 xp\_cmdshell 后，返回信息以表的形式输出。为了可以方便地在 SQL 中执行 BCP，下面的命令都使用 xp\_cmdshell 执行 BCP 命令。

(2) 对要导出的表进行过滤。

BCP 不仅可以接受表名或视图名做为参数，也可以接受 SQL 做为参数。通过 SQL 语句可以对要导出的表进行过滤，然后导出过滤后的记录。

```
EXEC master..xp_cmdshell 'BCP "SELECT TOP 20 * FROM  
AdventureWorks.sales.currency" queryout c:\currency2.txt -c -U"sa" -  
P"password"
```

BCP 还可以通过简单地设置选项对导出的行进行限制。

```
EXEC master..xp_cmdshell 'BCP "SELECT TOP 20 * FROM
AdventureWorks.sales.currency" queryout c:\currency2.txt -F 10 -L 13 -c -U"sa"
-P"password"'
```

这条命令使用了两个参数-F 10 和-L 13，表示从 SELECT TOP 20 \* FROM AdventureWorks.sales.currency 所查出来的结果中取第 10 条到 13 条记录进行导出。

### 3. 如何使用 BCP 导出格式文件

BCP 不仅可以根据表、视图导入导出数据，还可以配合格式文件对导入导出数据进行限制。格式文件以纯文本文件形式存在，分为一般格式和 xml 格式。用户可以手工编写格式文件，也可以通过 BCP 命令根据表、视图自动生成格式文件。

```
EXEC master..xp_cmdshell 'BCP AdventureWorks.sales.currency format nul -f
c:\currency_format1.fmt -c -T'
```

上述命令将 currency 表的结构生成了一个格式文件 currency\_format1.fmt，下面是这个格式文件的内容。

9.0

3

1 SQLCHAR 0 6 "\t" 1 CurrencyCode SQL\_Latin1\_General\_CP1\_CI\_AS

2 SQLCHAR 0 100 "\t" 2 Name SQL\_Latin1\_General\_CP1\_CI\_AS

3 SQLCHAR 0 24 "\r\n" 3 ModifiedDate

这个格式文件记录了这个表的字段（共 3 个字段）类型、长度、字符和行分割符和字段名等信息。

BCP 还可以通过-x 选项生成 xml 格式的格式文件。

```
EXEC master..xp_cmdshell 'BCP AdventureWorks.sales.currency format nul -f
c:\currency_format2.fmt -x -c -T'
```



xml 格式文件所描述的内容和普通格式文件所描述的内容完全一样，只是格式不同。

#### 4. 如何使用 BCP 导入数据

BCP 可以通过 in 命令将上面所导出的 currency1.txt 和 currency2.txt 再重新导入到数据库中，由于 currency 有主键，因此我们将复制一个和 currency 的结构完全一样的表。

```
SELECT TOP 0 * INTO AdventureWorks.sales.currency1 FROM  
AdventureWorks.sales.currency
```

将数据导入到 currency1 表中

```
EXEC master..xp_cmdshell 'BCP AdventureWorks.sales.currency1 in  
c:\currency1.txt -c -T'
```

导入数据也同样可以使用 -F 和 -L 选项来选择导入数据的记录行。

```
EXEC master..xp_cmdshell 'BCP AdventureWorks.sales.currency1 in  
c:\currency1.txt -c -F 10 -L 13 -T'
```

在导入数据时可以根据已经存在的格式文件将满足条件的记录导入到数据库中，不满足则不导入。如上述的格式文件中的第三个字段的字符长度是 24，如果某个文本文件中的相应字段的长度超过 24，则这条记录将不被导入到数据库中，其它满足条件的记录正常导入。

使用普通的格式文件

```
EXEC master..xp_cmdshell 'BCP AdventureWorks.sales.currency1 in  
c:\currency1.txt -F 10 -L 13 -c -f c:\currency_format1.fmt -T'
```

使用 xml 格式的格式文件

```
EXEC master..xp_cmdshell 'BCP AdventureWorks.sales.currency1 in  
c:\currency1.txt -F 10 -L 13 -c -x -f c:\currency_format2.fmt -T'
```

#### 总结

---

BCP 命令是 SQL Server 提供的一个快捷的数据导入导出工具。使用它不需要启动任何图形管理工具就能以高效的方式导入导出数据。当然，它也可以通过 xp\_cmdshell 在 SQL 语句中执行，通过这种方式可以将其放到客户端程序中（如 delphi、c#等）运行，这也是使客户端程序具有数据导入导出功能的方法之一。

(作者: wlfkong1 来源: TT 中国)

原文标题: SQL Server 数据导入导出工具 BCP 介绍

链接: [http://www.searchdatabase.com.cn/showcontent\\_24641.htm](http://www.searchdatabase.com.cn/showcontent_24641.htm)

## 用 BCP 工具导入和导出批量数据（上）

微软 SQL Server 中的批量复制工具程序 (Bulk Copy Program, BCP) 能让数据库管理员将数据批量导入表中或将数据从表中批量导入文档中。它还支持一些定义数据如何导出、导入到什么地方、加载哪些数据等选项。

本技巧讨论一些用 bcp 命令批量复制数据迁入或迁出 SQL Server 表的示例。这些示例在 SQL Server 2005 和 SQL Server 2008 上已经测试过。并且我还用了 AdventureWorks 样本数据库。

### 用 bcp 工具导入数据

一个最简单的操作就是你可以用 bcp 工具将数据从 SQL Server 表 bulk-copy 到文本文件。在 Windows 命令提示符中插入命令，你就可以运行 bcp 命令了。例如以下命令，从 AdventureWorks 数据库里的 Sales.vSalesPerson 视图复制数据到 C:\Data\SalesPerson.txt 文件：

```
bcp AdventureWorks.Sales.vSalesPerson out C:\Data\SalesPerson.txt -c -T
```

如同你看到的一样，bcp 命令以工具名称开头，后面为完全合格表名 database.schema.table。接下来就是 out 关键字，关键字告诉 bcp 工具数据将会从该表中导出。目标文本文件的路径和文件名称紧跟 out 关键字之后。注意本文中列出的命令例子可能包括很多行，但是所有的例子应该像一个单独的命令一样运行。

除了这些基本参数，bcp 工具还支持控制工具行为的 switch。在以上例子中，无论数据是以何种方式存储在源表中的，-c switch 表示所有的数据都应是字符数据。如果你没有指定 -c 开关或其他相关类型的 switch，你就需要在进入 bcp 命令后指定每个列的 switch 类型。

上述例子中另一个 switch 就是 -T，它主要是告诉 bcp 工具使用可靠连接来关联 SQL Server 示例。如果你没有指定 -T，你就必须提供用户名（-U switch）和密码（-P switch），或者你需要提供相关信息。

因为在先前列举的例子中没有指定实例，bcp 工具就在本地机上使用的默认实例。要指定一个 SQL Server 实例，就要用到 -S switch，后面紧跟的是服务器名称，如下所示：

```
bcp AdventureWorks.Sales.vSalesPerson out C:\Data\SalesPerson.txt -c -T -S Server01
```

Bcp 工具现在和 Server01 上的默认实例连接。如果你想连接到具体实例而不是默认的实例，你就必须指定实例名称和服务器名称，如 Server01\Sq1Srv。

通过默认，bcp 工具使用制表符分隔目标文件中里的域。但你也可以用 -t switch 调过这一操作，如下：

```
bcp AdventureWorks.Sales.vSalesPerson out C:\Data\SalesPerson.csv -c -T -t,
```

在这种情况下，-t switch 后有一个逗号，意思就是说数据域现在由逗号分开。这样做可让你将数据保存到 .csv 文件，便于你在 Microsoft Excel 文件中查看这些数据。

以上一些例子只限于将数据从表中导出。但是你还可以用 bcp 命令运行 Transact-SQL 查询、导出查询条件。例如以下 bcp 命令，包括只从 vSalesPerson 视图中检索 SalesPersonID、FirstName 和 LastName 的 SELECT 语句：

```
bcp "SELECT SalesPersonID, FirstName, LastName FROM AdventureWorks.Sales.vSalesPerson" queryout C:\Data\SalesPerson.csv -c -T -t,
```

这种情况下引号里的查询通过的是 bcp 命令而不是表名称。此外，queryout 取代了 out 关键字。但是命令其他部分和先前的例子相同。结果，SalesPerson.csv 文件现在只包含三个指定列。你还可以让查询更加精炼：例如你可以包括限定只从源表中返回那些行的 WHERE 子句。

(作者: Robert Sheldon 译者: 孙瑞 来源: TT 中国)

---

原文标题：用 BCP 工具导入和导出批量数据（上）

链接：[http://www.searchdatabase.com.cn/showcontent\\_20432.htm](http://www.searchdatabase.com.cn/showcontent_20432.htm)

## 用 BCP 工具导入和导出批量数据（中）

### 用 bcp 工具导入数据

Bcp 工具使得导入数据和导出一样简单。要在这部分里运行这个示例，首先就要执行下面的 T-SQL 脚本，在 AdventureWorks 数据库里创建 SalesPeople 表：

```
USE AdventureWorks
GO
IF OBJECT_ID (N'SalesPeople', N'U') IS NOT NULL
    DROP TABLE dbo.SalesPeople
GO
CREATE TABLE dbo.SalesPeople (
    SalesPersonID INT IDENTITY PRIMARY KEY,
    FirstName NVARCHAR(50) NOT NULL,
    LastName NVARCHAR(50) NOT NULL
)
```

要导出数据，你需要一个源文件从中复制数据。例如下面的命令就用了最近创建的 SalesPerson.csv 文件加载数据到 SalesPeople 表：

```
bcp AdventureWorks.dbo.SalesPeople in C:\Data\SalesPerson.csv -c -T -t,
```

首先，你必须指定目标表，这种情况下紧跟其后的为代替 out 或 queryout 的 in 关键字。其次，你必须指定源文件的路径和文件名称，后跟任意可用的 switch。

在运行这一命令并查看结果时，要注意源文件如果包括售货员的 ID，这些值就不能插入到 SalesPersonID 列。该列定义为 IDENTITY 列，这样我们就可以忽视源数据。要保留原始值，你就必须在命令里增加 -E switch，如下面的例子中所示：

```
bcp AdventureWorks.dbo.SalesPeople in C:\Data\SalesPerson.csv -c -T -t, -E
```

现在表中就包含了你想要得到的数据。

## 使用格式文件

在导入或导出数据时，你会发现源数据架构和目标数据架构不匹配。例如，文本文件中的这些列可能和目标表中的列的顺序不一致，或者说这些列的多少还不一样。你可以通过创建格式文件映射源文件和目标架构解决这个问题。我们通过以下示例看看它是如何工作的：

假如你使用了以下命令从 vSalesPerson 视图中导出数据到 SalesPeople.txt 文件：

```
bcp "SELECT LastName, FirstName, SalesPersonID FROM AdventureWorks.Sales.vSalesPerson"
      queryout C:\Data\SalesPeople.txt -c -T -t,
```

该命令使用了先前例子中同样的参数。但是注意这些列从视图中检索的顺序：LastName、FirstName、最后是 SalesPersonID。

现在假设你打算使用该文件导入数据到 SalesPeople 表。SalesPeople 表中列的顺序和文本文件中列顺序不同。要解决这个问题，你可以创建一个格式文件将这些列从源文件映射到目的文件。以下命令说明如何创建一个格式文件：

```
bcp AdventureWorks.dbo.SalesPeople format nul -f C:\Data\SalesPeople.fmt -c -T -t,
```

(作者: Robert Sheldon 译者: 孙瑞 来源: TT 中国)

原文标题：用 BCP 工具导入和导出批量数据（中）

链接：[http://www.searchdatabase.com.cn/showcontent\\_20480.htm](http://www.searchdatabase.com.cn/showcontent_20480.htm)

## 用 BCP 工具导入和导出批量数据（下）

先前的例子表明，命令首先指定了目标表。而这一次表明后面为关键字 `format nul`，表示 `bcf` 工具应该创建格式文件。`-f` 参数用于指定格式文件的路径和文件名称，其后为 `switch`。最后在你运行该命令时，就生成了包括 `SalesPeople` 表架构的格式文本。

下面的数据显示上述命令生成的 `SalesPeople.fmt` 格式文本内容：

```
10.0
3
1 SQLCHAR 0 12 "," 1 SalesPersonID ""
2 SQLCHAR 0 100 "," 2 FirstName SQL_Latin1_General_CP1_CI_AS
3 SQLCHAR 0 100 "\r\n" 3 LastName SQL_Latin1_General_CP1_CI_AS
```

文件(10.0)第一行确定了目前使用的 `bcf` 版本。第二行(3) 确定了表中的列数，接下来的三行为列的相关信息：

- \* 第一个域为列在源文件中出现的顺序。
- \* 第二个域显示每个列的源文件数据类型。因为在生成文件时指定了 `-c switch`，在从数据文件中提取时，所有的域都用字符型数据类型。插入数据时，SQL Server 会将数据转换到正确的类型。
- \* 第三个域表示域的前缀长度，SQL Server 通常会用它来提供最紧凑的文件存储。在创建格式文件时如果你指定 `-c switch`，那么就会自动用到 0。
- \* 第四个域代表特殊域数据类型字节长度。
- \* 第五个域表明如何终止行和列。由于在创建格式文件时用了 `-t switch`，源文件的域值就必须通过逗号终止。
- \* 第六个域映射这些列在 SQL Server 表中的排列顺序。
- \* 第七个和最后一个域提供了 SQL Server 表中字符列的整理信息。



要用格式文件将数据导入 SalesPeople 表中，我们必须如下修改文件：

```
10.0
3
1  SQLCHAR  0  100  "\",\"  3  LastName  SQL_Latin1_General_CP1_CI_AS
2  SQLCHAR  0  100  "\",\"  2  FirstName  SQL_Latin1_General_CP1_CI_AS
3  SQLCHAR  0  12   "\\r\\n\"  1  SalesPersonID  ""
```

你可以看到，列的顺序已经进行了修改，这一顺序就是它们在格式文件中的排列顺序。SalesPersonID 列现在排在最后，并且以\r\n 结尾。LastName 列现在排在开头并且以逗号结尾。

修改、保存格式文件后，准备在 bcp 命令中用了。下面的例子说明如何调用格式文件：

```
bcp AdventureWorks.dbo.SalesPeople in C:\Data\SalesPeople.txt -
f C:\Data\SalesPeople.fmt -T
```

注意，你从 SalesPeople.txt 文件中导入数据时，还必须用到 -f switch 调用格式文件。还要注意到你现在已经不需要包括 -t 和 -c switch，因为现在在格式文件中已经包括了这些信息。

无论你用的是格式文件还是只运行基本命令，你现在都应该更好地了解到了如何使用 bcp 工具。记住 bcp 工具支持的 switch 比我所列举的要多得多。你可以在《SQL Server 联机丛书》上了解更多有关 bcp 工具的信息。同时，本技巧还提供了有关启用 bcp 工具、轻松将数据导入到 SQL Server 表以及导出数据到文本文件的比较充足的信息。

(作者: Robert Sheldon 译者: 孙瑞 来源: TT 中国)

原文标题：用 BCP 工具导入和导出批量数据（下）

链接：[http://www.searchdatabase.com.cn/showcontent\\_20482.htm](http://www.searchdatabase.com.cn/showcontent_20482.htm)

## 从 SQL Server 中导出至 Oracle 数据库



**问：**我们可以把所有的存储过程和触发器从 SQL Server 中导出到 Oracle8i 吗？

**答：**答案是不能。即使你能完成导出，但导入是不能实现的。原因是 MS SQL Server 使用 T-SQL 作为编程语言，而 Oracle 使用 PL/SQL。T-SQL 和 PL/SQL 是不一样的。T-SQL 不能在 Oracle 中使用，同样，PL/SQL 也不能在 SQL Server 中使用。你必须查找你的 SQL Server 存储过程和触发器代码并在 Oracle 中重写。

*(作者: Brian Peasland 译者: ecubestudio 来源: TT 中国)*

原文标题：从 SQL Server 中导出至 Oracle 数据库

链接：[http://www.searchdatabase.com.cn/showcontent\\_21364.htm](http://www.searchdatabase.com.cn/showcontent_21364.htm)

## 错误导入 Excel 数据到 SQL Server 2005

问：我才刚刚开始学习 SQL Server 2005，我简直不敢相信 SSIS 和 DTS 比起来竟然这么难。我想在两个小时之内将一份 Excel 表里的数据导入进去（用 DTS 的话只需要两分钟），但是我仍然不断收到错误提示。由于不支持 DT\_R8 和 DT\_STR 数据类型之间的转换，所以不能插入列。但是我不知道什么是 DT\_R8？

我想把 Excel 数据导入 SQL Server 中。我已经尝试过导入现存的表、并创建一个新表。我还尝试过改变所有的单元格。但是仍然无济于事。你可以帮我解决这个问题吗？

答：只要客户发送文件给我，我就会自通过 DTS 或 SSIS 包进行数据抽取。但是我会经常因为某些原因避开 Excel 表。如果是给我一份 csv 文件，一份 tab 界定的文件或者是 '|' 界定的文件，但不要选 Excel 文件。除 Excel 之外有很多都可以尝试。你可以列出一些能和它们匹配的数据类型。仔细检查你的 Excel 文件中的这些数据类型——特别是不能导入的列。在单元格上点击右键、选择格式查看数据类型。你的目标表中的列必须和这些数据类型相匹配。如果它们不能够匹配，你就考虑一下在插入之前用 SSIS Convert 任务改变数据类型。

(作者: Joe Toscano 译者: 孙瑞 来源: TT 中国)

原文标题: 错误导入 Excel 数据到 SQL Server 2005

链接: [http://www.searchdatabase.com.cn/showcontent\\_16634.htm](http://www.searchdatabase.com.cn/showcontent_16634.htm)

## SQL Server 中的导出表以及使用 FULL OUTER JOIN 的技巧

问：我的问题是关于 FULL OUTER JOINs 导出表的。查询如下：

```
select tbl1.usr, tbl1.ADD, tb2.UPD from
  (select usr, count(nin) ADD
   from RGS8 where act='ADD'
  and dte=20080310 group by usr)  tbl1
  FULL OUTER JOIN
  (select usr, count(nin) UPD
   from RGS8 where act='UPD'
  and dte=20080310 group by usr)  tb2
  ON tbl1.usr=tb2.usr
```

该查询在使用左右外联结时执行很成功，但是结果却不显示完整的外联结。我在想是否有一些带有 FULL OUTER JOIN 的外联结？

答：我不知道为什么你这么不行。除非你使用的是 MySQL，不支持 FULL OUTER JOIN，否则就不会出现那种情况。但是在这种情况下，你应该收到了一个错误信息，而不是没有任何结果。

以下是供你选择的策略之一：

```
SELECT usr
, SUM(CASE WHEN act = 'ADD'
        THEN 1
        ELSE 0 END) AS ADD
, SUM(CASE WHEN act = 'UPD'
        THEN 1
        ELSE 0 END) AS UPD
FROM RGS8
WHERE act IN ('ADD', 'UPD')
AND dte = 20080310
GROUP
  BY usr
```

---

这样做可能会好些，因为它只需要该表的一个入口。

(作者: Rudy Limeback 译者: 孙瑞 来源: TT 中国)

原文标题: SQL Server 中的导出表以及使用 FULL OUTER JOIN 的技巧

链接: [http://www.searchdatabase.com.cn/showcontent\\_14808.htm](http://www.searchdatabase.com.cn/showcontent_14808.htm)

## 如何使用 BULK insert 将数据导入 SQL Server 数据库？

**问：**我想用存储程序将数据从.txt 或者.xls 文件导入 SQL Server 数据库。我想使用该文件复制临时表并从临时表迁移到数据库。你能举一个关于该程序的例子吗？

**答：**将数据导入 SQL Server 中有很多种方法，选择哪一种方法在很大程度上取决于你的具体情况。如果你仅仅想加载你的文本文件的内容，那么使用下面的 BULK INSERT 指令就能够达到你的要求。如果你文件中的数字和列的顺序和目标表中的一样，那你就可以使用下面的简单语句：

```
BULK INSERT MyTable
FROM 'c:datamydata.txt'
WITH
(
    FIELDTERMINATOR = '|',
    ROWTERMINATOR = '\n'
)
```

如果你想排除一些列或者更改一些列的顺序，那么你可以创建一个格式文件、说明它是 BULK INSERT 的一个参量。你还可以查阅 Books Online 了解更多有关使用 BULK INSERT 以及如何创建一个格式文件的信息。

(作者: Roman Rehak 译者: 孙瑞 来源: TT 中国)

原文标题: 如何使用 BULK insert 将数据导入 SQL Server 数据库？

链接: [http://www.searchdatabase.com.cn/showcontent\\_14404.htm](http://www.searchdatabase.com.cn/showcontent_14404.htm)