msdn

Integrating Visio 2007 and Microsoft SQL Server 2005

Summary: Learn how you can integrate Microsoft Office Visio 2007 with Microsoft SQL Server 2005, to create data-driven diagrams from stored SQL data. (7 printed pages)

Visimation, Inc. [http://www.visimation.com/]

January 2007

Applies to: Microsoft Office Visio 2007, Microsoft SQL Server 2005

Contents

Overview

This article focuses on integration techniques specific to Microsoft Office Visio 2007 and Microsoft SQL Server 2005. The article Integrating Visio 2007: Introduction to Integrating Visio with Other Microsoft Programs [http://msdn2.microsoft.com/en-us/library/bb232815.aspx] presents a general discussion about integrating data between Office Visio 2007 and other Microsoft applications. An example of integrating data from SQL Server Reporting Services with Visio 2007 is presented in the article Integrating Visio 2007 and MOM, Exchange, and Reporting Services [http://msdn2.microsoft.com/en-us/library/bb264455.aspx] .

Accessing SQL Server Analysis Services Data by Using the Data Selector Wizard

Using Visio 2007, you can connect Visio shapes to data that was generated outside Visio. A large amount of data can be captured in a SQL Analysis Services database. Being able to analyze that data in a visual way enhances the value of the data. In the following example, sales data stored in an Analysis Services cube is used to generate a Visio PivotDiagram so that the data can be explored and graphically enhanced.

In Visio, you can use two menu items on the **Data** menu to bring SQL data into the Visio External Data window. You can use the **Link Data to Shapes** command to link to a Microsoft SQL Server relational database. Linking to SQL data is very similar to linking to other types of data. You can find examples in the article Integrating Visio 2007: Introduction to Integrating Visio with Other Microsoft Programs [http://msdn2.microsoft.com/en-us/library/bb232815.aspx] .

Figure 1. Data Selector dialog box
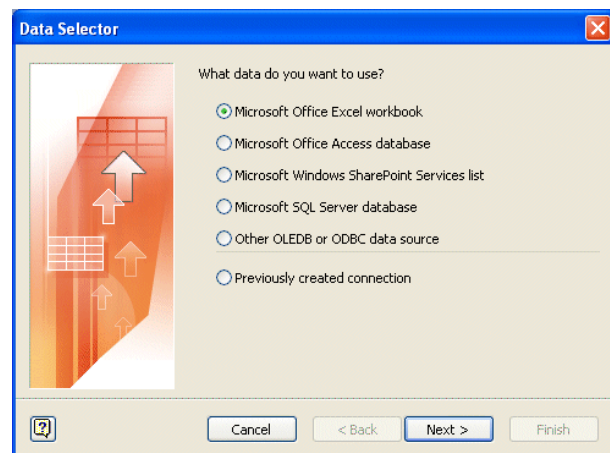
A variation of the **Data Selector** dialog box is presented if you select **Insert PivotDiagram** (also on the **Data** menu). This variation is shown in Figure 2.

Figure 2. Data Selector dialog box for PivotDiagrams
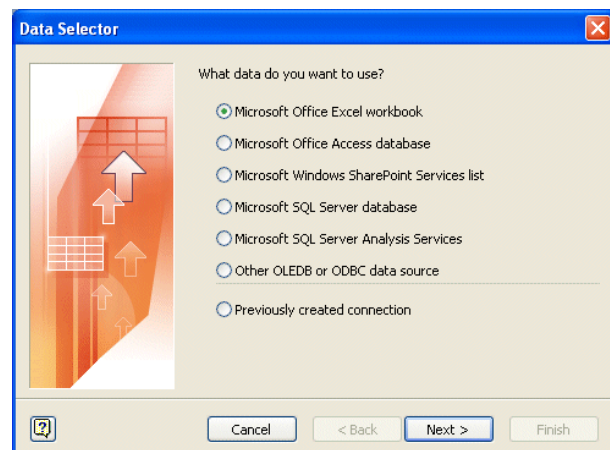
Notice the additional command on the **Data** menu, **Microsoft SQL Server Analysis Services**. You can also use this command to link to SQL data (and to other types of data). A Visio PivotDiagram based on the data is automatically created. If you select **Microsoft SQL Server Analysis Services** and then click **Next**, the **Data Connection Wizard** dialog box

appears, as shown in Figure 3. Specify the computer running SQL Server that contains the Analysis Services data and any necessary login credentials.

Figure 3. Data Connection Wizard dialog box

After you specify the computer running SQL Server, a list of Analysis Services data sets is presented. An example is shown in Figure 4. In this example, **Internet Sales** is selected.

Figure 4. Select database and table from the Analysis Services data

Using Visio, you can capture the data set into a file. If the data file is updated and a Visio drawing is linked to it, you can refresh the data within Visio without re-linking to the drawing. The options on the **Save Data Connection File and Finish** page of the **Data Connection Wizard** (shown in Figure 5) enable you to specify a file for saving the data. When you click **Finish**, a Visio PivotDiagram is created from the selected data.

Figure 5. Save Data Connection File and Finish page

Figure 6 shows a Visio PivotDiagram created from Analysis Services data.

Figure 6. Initial Visio PivotDiagram
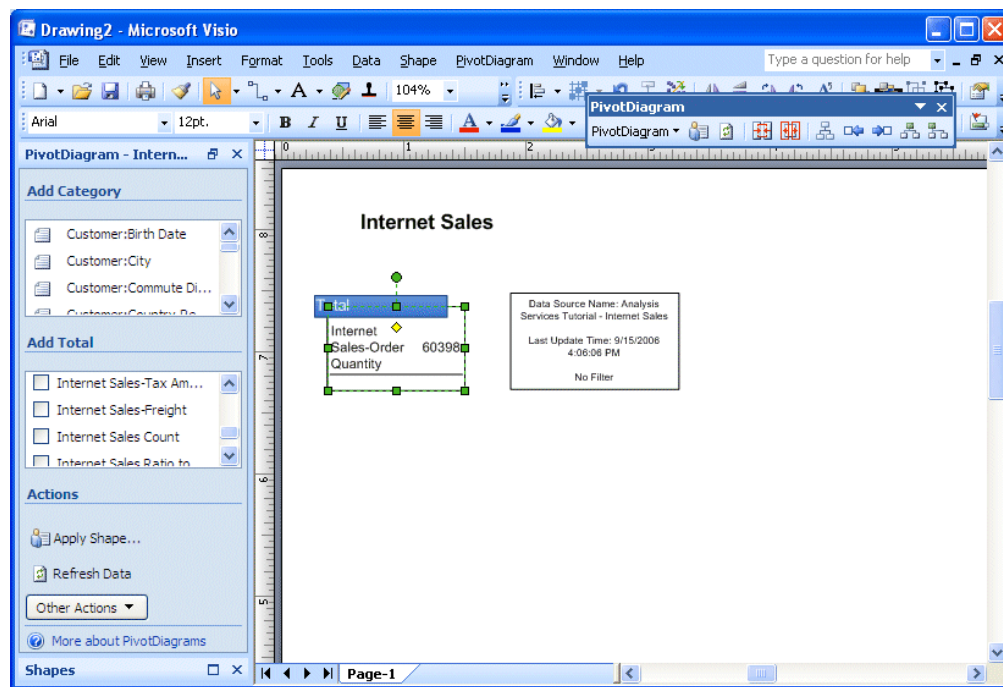
You can now explore the data. Right-click the top-level pivot shape or select the shape and choose a category from the **Add Category** list at the left of the screen. Figure 7 shows the result if you select **Customer:Gender**.

Figure 7. PivotDiagram showing data by gender
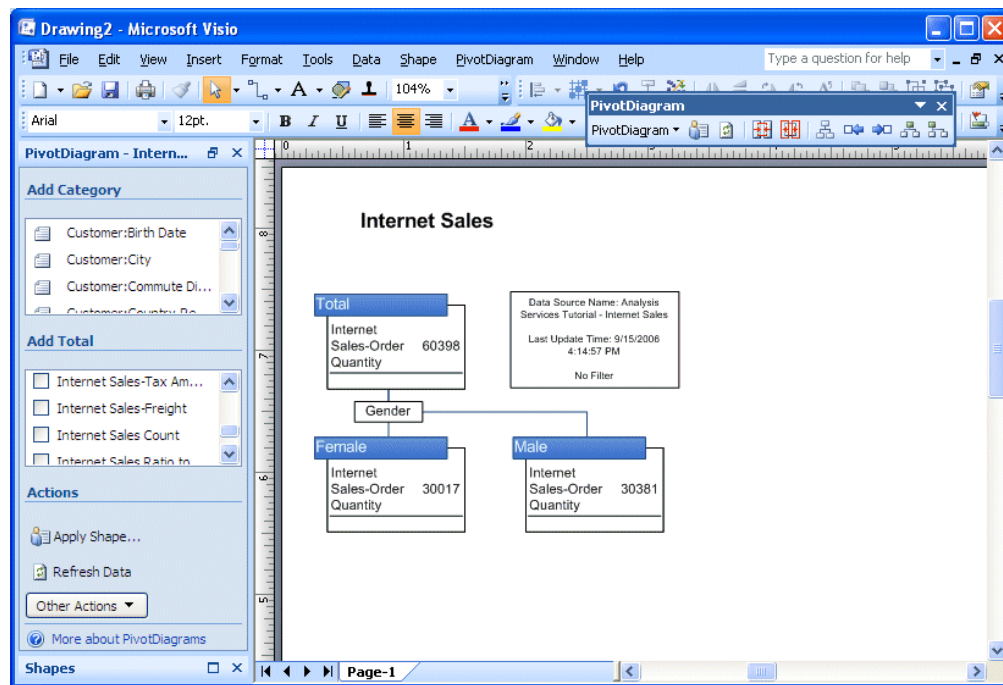
You can visually enhance the drawing by applying a shape to each node in the diagram. On the left side of the image, under **Actions**, select the shape, and then select **Apply Shape**. Figure 8 shows the resulting drawing if you add another category to show **Yearly Income** and then use **Apply Shape**.

Figure 8. The finished drawing
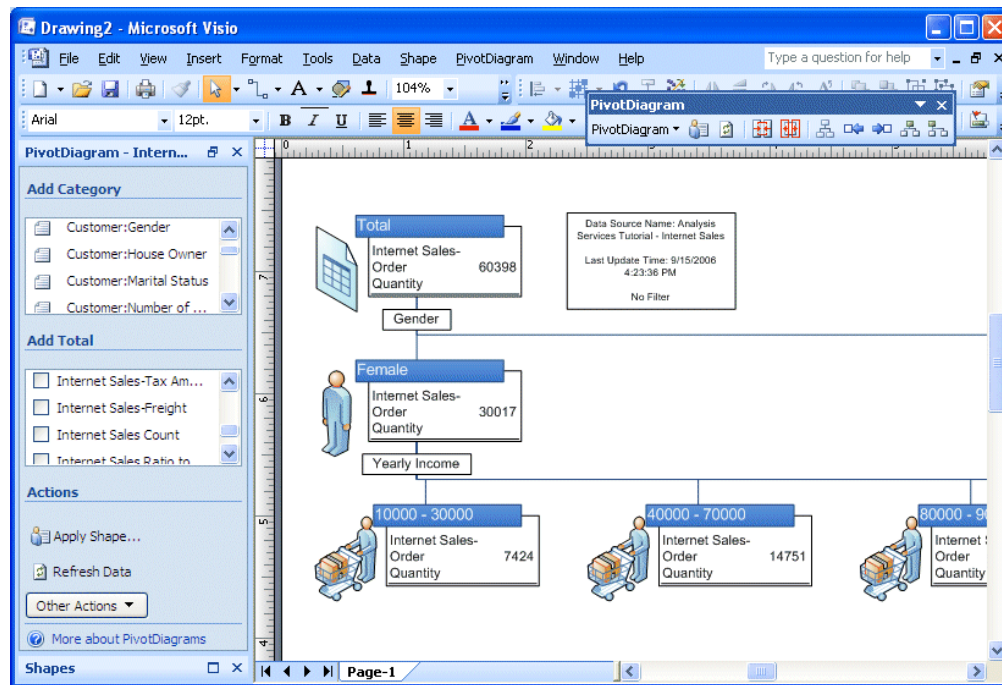
At this point, the relative values of the raw numbers are not easy to see. To add value to our drawing and to help the relative values of the numbers stand out, we next modify the data graphics that are shown with the pivot shapes to display a meter image instead of the number itself.

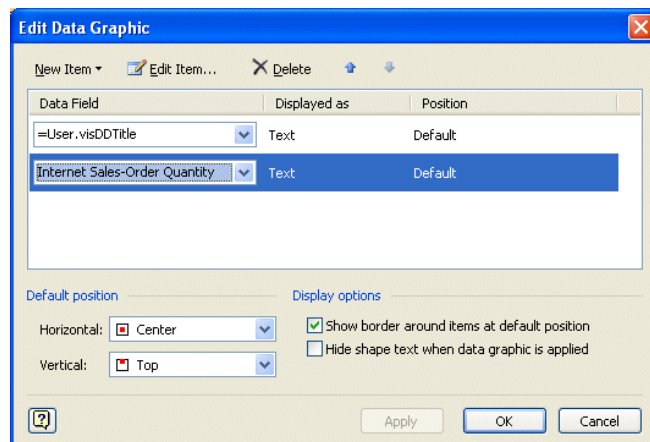**Figure 9. Edit Data Graphic dialog box**

To modify the data graphic to display a meter image

1. On the left side of the image, under **Actions**, click **Other Actions**, and then select **Edit Data Graphic**.

2. Select the item **Internet Sales-Order Quantity**, and then delete it.

3. Click **New Item**, and then select **Data Bar**. The **New Data Bar** dialog box appears, as shown in Figure 10.

   Figure 10. New Data Bar dialog box
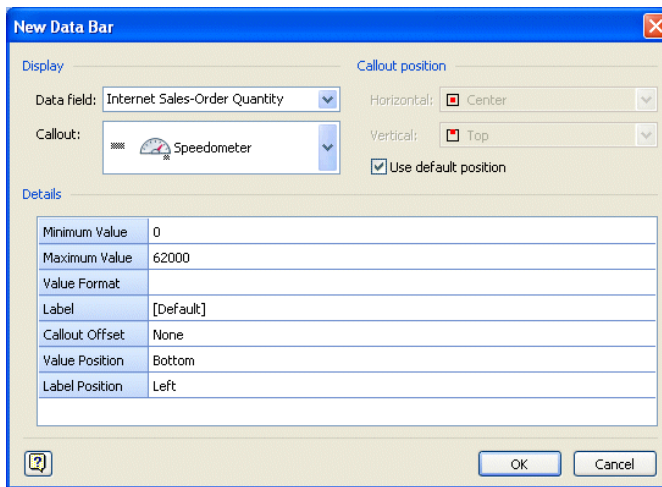
4. In the **Data field** list, select **Internet Sales-Order Quantity**.

5. In the **Callout** list, select **Speedometer**.

6. Because our data can have a value from 0 to just over 60,000, in the **Maximum Value** box, type **62000**. These selections are also shown in Figure 10.

7. Click **OK** to close the **New Data Bar** dialog box.

8. Click **OK** to close the **Edit Data Graphic** dialog box.

    The final result is shown in Figure 11.

    Figure 11. Modified data graphic
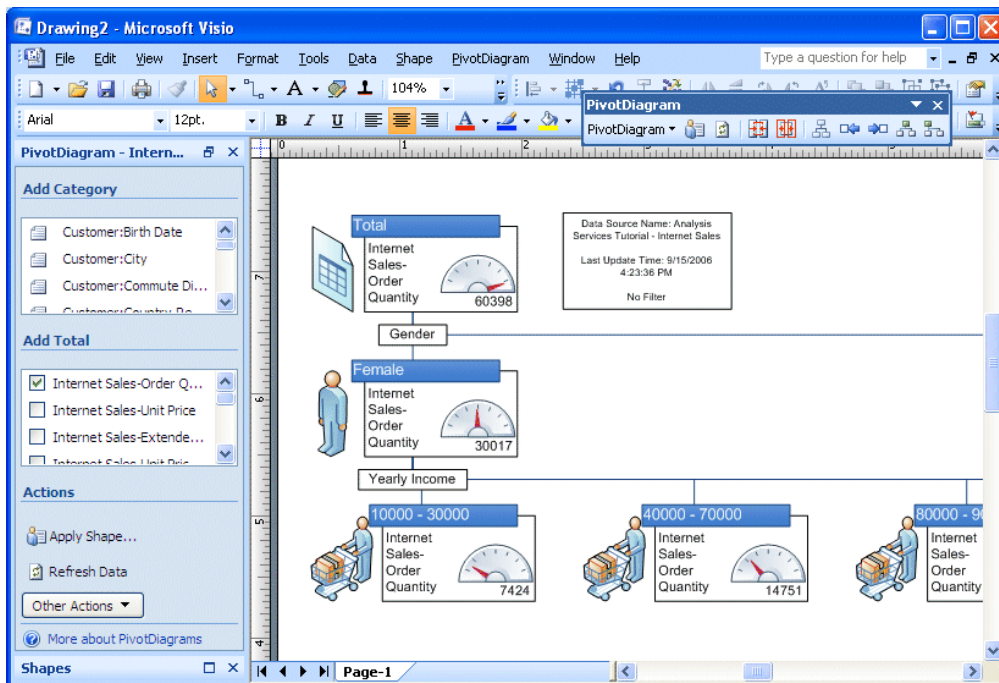
Using Automation Code to Generate a Visio Diagram from Shape and Connectivity Data Stored in SQL Tables

The article Integrating Visio 2007 and Access 2007 [http://msdn2.microsoft.com/en-us/library/aa722522.aspx] describes how to generate a Visio diagram by using Automation code against shapes and connectivity data that is stored in a Microsoft Office Access 2007 database; for complete documentation of the code listed at the end of this article, refer to this article. The code segments in this article document only the differences required to make this same Automation code work against an SQL database.

The sample code generates a Visio diagram from data stored in SQL database tables. The database contains two tables: an Items table that contains information about which shapes are included in the drawing and a Connections table that contains information about connectivity between shapes. The shapes are placed in the drawing from a template. A Visio dynamic connector is used to connect the shapes in the Visio drawing. The location of the shapes on the drawing page is not stored in the Access tables. Instead, the Visio **Layout** method is used to position the shapes on the drawing page.

There are fields in the tables that are copied into the Shape Data of the shapes. The shapes use this information to set shape text and properties.

Benefits of the Visio Template
Building capabilities such as Shape Data and text fields into the shapes reduces the amount of code required to create automated solutions. For example, no code is required in this example to display text on the shapes because that is handled by text fields defined within the shape masters. Also, you do not need to set the color of the shape because that is derived from the **Status** field for the shape. The formulas for deriving the color are also built into the shapes.

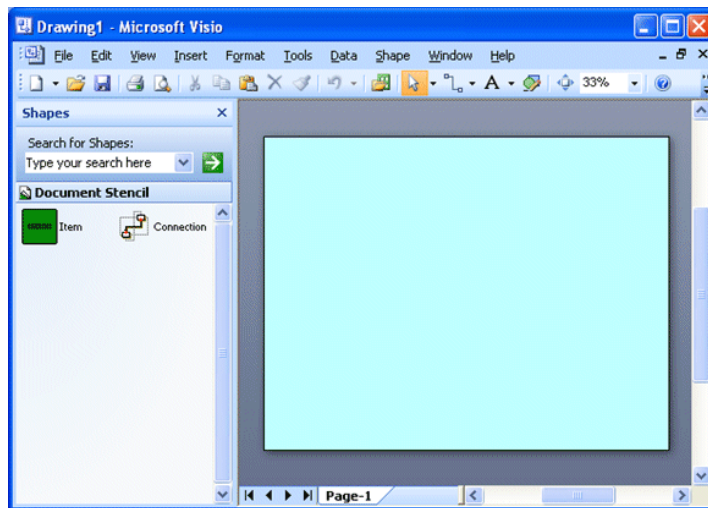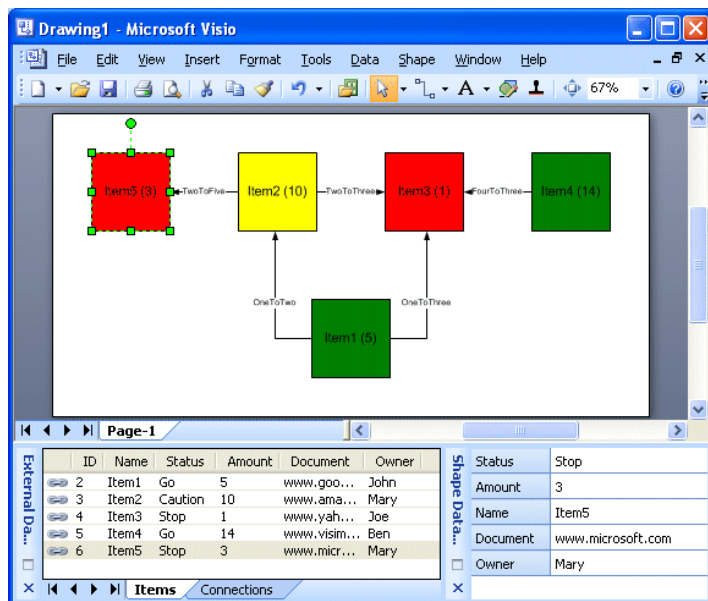**Figure 12. Visio template with preconfigured shapes**

Figure 13. Finished drawing showing one selected shape

Modifying the Sample Code to Enable the Reading of SQL Data
There are two changes to the Access sample code that are required before we can access SQL data instead of Office Access 2007 data. First, we change the connection string to reference SQL data.

This is the connection string as defined for SQL.

C#   📋 Copy Code

```
private const string _ADOConnectionString =
    @"Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security Info=True;Data Source=localhost\sqlexpress;Initial Catalog=WhitePaperSamples";
```

This is the connection string as defined for Access.

C#   📋 Copy Code

```
private const string _ADOConnectionString =
    "Provider=Microsoft.Jet.OLEDB.4.0;Data Source={0};User Id=admin;Password=;";
```

Modifying the readData Method
The second change is in the **readData** method. In Access, the file name is passed as an argument to the **readData** method. This is not done with the SQL version of **readData**, which is shown below. In the Access sample, the connection string has a placeholder, which is filled in with the file name in the **readData** method.

C#   📋 Copy Code

```
private static bool readData(
    Visio.Document visioDocument)
{
    bool itemsReadSuccessful = false;
    bool connectionsReadSuccessful = false;

    try
    {
        // Link the Items to the Visio document by adding a data
        // recordset with the connection and command information.
        visioDocument.DataRecordsets.Add(
            _ADOConnectionString,
            string.Format(_ADOSelectString, _ItemsTableName),
            0,
            _ItemsTableName);

        _ItemsDataRecordsetIndex = visioDocument.DataRecordsets.Count;
```

```
        itemsReadSuccessful = true;

        // Link the Connections to the Visio document by adding a data
        // recordset with the connection and command information.
        visioDocument.DataRecordsets.Add(
            _ADOConnectionString,
            string.Format(_ADOSelectString, _ConnectionsTableName),
            0, _ConnectionsTableName);

        _ConnectionsDataRecordsetIndex = visioDocument.DataRecordsets.Count;
        connectionsReadSuccessful = true;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }

    return (itemsReadSuccessful && connectionsReadSuccessful);
}
```

Conclusion

The techniques for linking to SQL data from Visio 2007 are very similar to the techniques for linking to data from Microsoft Office Excel, Office Access, and other sources. For more information and examples, see the following articles:

- Integrating Visio 2007 and Access 2007 [http://msdn2.microsoft.com/en-us/library/aa722522.aspx]
- Integrating Visio 2007 and Excel 2007 [http://msdn2.microsoft.com/en-us/library/aa701255.aspx]
- Integrating Visio 2007: Introduction to Integrating Visio with Other Microsoft Programs [http://msdn2.microsoft.com/en-us/library/bb232815.aspx]

In this article, you learned about using Visio PivotDiagrams to visually explore data from a SQL Analysis Services cube. You also saw an example of using Automation code to generate drawings from data stored in an SQL database.

About the Author

Since 1997, Visimation has helped companies improve their productivity by providing visual software tools to ease business and technical tasks, and offered a broad range of consulting services focusing on Microsoft Office Visio as a platform for rapid development of efficient Automation programs.

Additional Resources

For more information, see the following resources:

- Visio 2007: Software Development Kit [http://www.microsoft.com/downloads/details.aspx?familyid=373d34b8-5ef7-4e6e-a582-c8d6b5ee4e33&displaylang=en]
- Microsoft Office Developer Center: Visio Developer Portal [http://msdn2.microsoft.com/en-us/office/aa905478.aspx]

For more information about Visio 2007 integration, see the following articles:

- Integrating Visio 2007: Introduction to Integrating Visio with Other Microsoft Programs [http://msdn2.microsoft.com/en-us/library/bb232815.aspx]
- Integrating Visio 2007 and Access 2007 [http://msdn2.microsoft.com/en-us/library/aa722522.aspx]
- Integrating Visio 2007 and Active Directory [http://msdn2.microsoft.com/en-us/library/bb245820.aspx]
- Integrating Visio 2007 and Excel 2007 [http://msdn2.microsoft.com/en-us/library/aa701255.aspx]
- Integrating Visio 2007 and MOM, Exchange, and Reporting Services [http://msdn2.microsoft.com/en-us/library/bb264455.aspx]
- Integrating Visio 2007 and Project 2007 [http://msdn2.microsoft.com/en-us/library/aa827350.aspx]
- Integrating Visio 2007 and SharePoint Products and Technologies [http://msdn2.microsoft.com/en-us/library/bb229690.aspx]

Community Content