

## **README: Microsoft SQL Server 2008 Integration Services SharePoint Sample**

*Please review this README before continuing.*

### **Introduction**

This sample illustrates the creation of a source and destination adapter in C#, using the public APIs for Microsoft SQL Server Integration Services. The component connects to SharePoint through a proxy dll, which wraps the complexities of working with the SharePoint Lists Webservice.

The Specific Features of the included SSIS Components are:

#### **SharePoint Source Component**

- Auto-detects all of the public fields from a SharePoint List or view of the list.
- Supports removal of fields from the output flow, which optimizes the query against SharePoint for increased performance (avail through the right-click advanced window)
- Supports restricting SharePoint data by a custom CAML query, which can be provided through a variable expression for fine tuned filtering over large lists.
- Provides a tuning feature for the # of rows to return at a time. Often with SharePoint, large lists need to be brought over in smaller batches than wide (many field) lists, or else the Webservice will die. The batchsize lets you specify a solid size that works for your list.

#### **SharePoint Credential Connection Manager Component**

- A shared custom credential manager can be created to allow end user to store custom NT Credentials to use when accessing SharePoint.
- Allows standard process credentials to be used as well.
- Can be attached to multiple Source or Destination components as needed.

#### **SharePoint Destination Component**

- Auto-detects all of the editable fields from a SharePoint List
- Only fields bound to columns from the input flow will be 'updated', for performance.
- Supports deleting rows by ID

#### **Included Bonus – SharePoint Utility**

To simplify the adapter code and keep the SharePoint routines generic, a proxy VB.Net component is included which access the SharePoint Webservice API. It includes a very flexible api for getting SharePoint List information, which is used

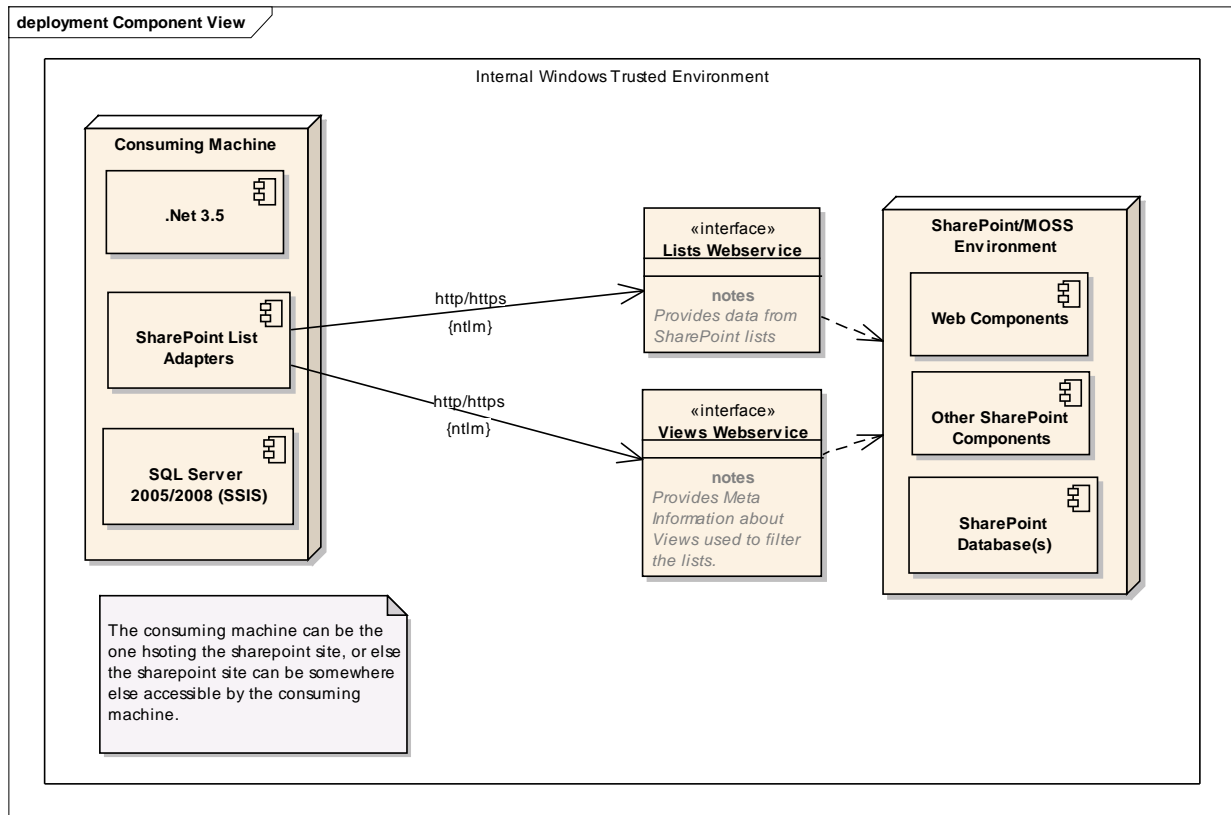
by the Components above. In addition, it can publish and remove files from a document library.

### Notes:

When working with SharePoint, you will be receiving the raw data and should update it in the similar format you receive it. One common thing is a lookup field. The data comes with an ID, a delimiter and the 'String Value'. You can use a custom script flow to split it apart as needed. When you update the value back to SharePoint, the Description for Lookup fields is NOT used (but you should include an empty string where it would go).

When doing joins with other data sets or other SharePoint lists, it helps when you only need to worry about the IDs, and not pass around the string representation every time.

## Requirements



To run this sample package, the following components are required:

- .Net 3.5 Framework
- SQL Server Integration Services 2008 (Installed as part of the SQL Server 2008 Family)

## How to Use This Sample

This installer places the source and project files for the destination into the chosen destination directory chosen. Additionally, pre-built binaries are placed into the Global Assembly Cache (GAC) and into the C:\Program Files\Microsoft SQL Server\100\DTS\PipelineComponents directory (if present).

A good way to explore the sample is to try using it, understand its behavior, and then begin reviewing the source code for the component.

If you do not see the SharePoint adapters in your toolbox automatically, then follow these instructions:

1. Open Business Intelligence Development Studio, open a package, and then click **Choose Toolbox Items** on the **Tools** menu.
2. In the **Choose Toolbox Items** dialog box, click the **SSIS Data Flow Items** tab, and then check **SharePoint Destination** and **SharePoint Source**.
3. Click **Ok**.

The SharePoint source and destination components should now appear in the toolbox for the data flow task. You can add the source and destination components to the data flow of the package.

At this point, the SharePoint source and destination components should be available in the Data Flow toolbox, and you can use them to construct data flows.

## Component Features

This sample implements unique features that may inspire you to create great custom SSIS components:

- **Extensive Validation:** Validation for this component actively goes against the SharePoint Site to verify the properties are valid.
- **Values from Expressions:** This component supports external variables, which can be expressions, and can be attached to the source component to customize the query. Similar to the CommandText for the other Sql Components
- **Linq:** The Component has been written using Linq with .net 3.5 and shows how elements such as the metadata and columns can be combined to create a readable usage in a Linq format.
- **Custom properties:** The component keeps its configuration in custom properties on itself, inputs, and input columns.
- **SharePointUtility Component:** Simplifies access to the SharePoint Lists / Views WS using Linq friendly objects

## Disclaimer

While this component is a powerful sample, it is intended only as a sample. It is specifically not intended for production use or deployment, but only as an educational tool for creating custom components.