Web Reports Manual

FOR ORF LOG PROVIDER USERS

Revision: 1.2
Date: November 9, 2009

Preface

WHAT IS THIS MANUAL ABOUT?

This documentation describes the steps of creating web-based reports for the end-users using the **Vamsoft OLEDB Provider for ORF Log Files** (also known as **ORF Log Provider**) and an example software.

WHO SHOULD READ THIS MANUAL?

This guide is for advanced users who intend to provide reports of the filtered mail flow on a website. In this example, we will use the ORF Log Provider to parse the text logs into an SQL database, so we can use this data for the web reports.

To achieve this, we will use Microsoft .NET, C# language, and the Log Provider, thus basic knowledge of these are required.

HOW DOES THIS SOFTWARE HELP?

This software is intended to demonstrate what the parsed log data can be used for. Note that this example has demonstration purposes only and is intentionally simple: it is not built for production use, but will help you in creating your own solution for your own environment.

Please consider that unfortunately we can neither provide any support for writing your own application, nor for this example software (besides this document).

System Requirements

OPERATING SYSTEM REQUIREMENTS

The software supports the following operating system platforms:

- Microsoft® Windows® 2000/2003
- Microsoft® Windows® XP
- Microsoft® Windows® Vista

SOFTWARE REQUIREMENTS

The following should be installed to run the example:

- Microsoft® .NET™ Framework 2.0 or later
- Microsoft® ASP.NET™ runtime environment
- Microsoft® Internet Information Services®
- Microsoft® SQL Server® (or Oracle – see /usercontributed/oracle)
- Microsoft® Exchange® (within the domain)
- Vamsoft OLEDB Provider for ORF Log Files 1.5 or above (http://www.vamsoft.com/logprovider)

Note that **ORF is not required** to be installed locally, the components of the example run independently. **Administrator privileges are required** to install and uninstall the example and the above softwares.

Brief Description

The example consists of three main components.

THE WINDOWS SERVICE

This component monitors the log files and transfers all newly added entries to the database, keeping it up to date.

THE DATABASE

The database stores the logged data, so it can be accessed easily using SQL statements, stored procedures etc. (which would not be possible by using the raw log files).

THE WEB SITE

This is the actual user interface, which features a single page. It displays a list of all mails which were blacklisted in the past 24 hours for the user currently logged in to the site.

Package Contents

The package is organized into the following structure:

/database
Contains an SQL script for creating the objects in the database.

/service
Contains the source code of the service which populates the database with log data.

- **/precompiled**
  Contains a precompiled service executable and its configuration file.

/usercontributed
Contains user contributions – currently, an Oracle SQL script for creating the schema. Note that further changes may be required to get the demo work with Oracle databases.

/website
Contains a Microsoft Visual Web Developer project accessing the log database.

/readme.pdf
This document.

Installation

After extracting the contents of the package, follow the steps below.

CREATING THE DATABASE

The database should be created first, to achieve this, we will use Microsoft SQL Server Management Studio in this example. Please follow the steps below:

1. Launch Microsoft SQL Server Management Studio
2. Connect to the database server
3. Right-click "*Databases*" in the Object Explorer, and click "*New Database...*"
4. Enter a name for the database, (we will use *ORFLog in this example)*, and click OK
5. Open the *sql-orflog.sql* script located in the */database* folder of this package (*File|Open|*File in the main menu or *Ctrl+O*) If you named your database something other than ORFLog, modify the first line of the script accordingly ("*use [databasename];*")
6. Execute the script by pressing *F5*, or right-clicking the query editor window and clicking *Execute*.

COMPILING THE SERVICE

The package contains a precompiled version of the service. Building another version is needed only if you modify the source code, the easiest way is using Microsoft® Visual C# Express or Microsoft® Visual Studio® 2005 for this purpose.

Later, be sure to install the binary created in the *service/ORFLogToDbService/bin/x86/Release* folder, especially on 64-bit platforms. This assembly is marked with the 32-bit flag. Since currently there is only a 32-bit version of the log provider, a native 64-bit application would not be able to access it.

INSTALLING THE SERVICE

1. Open a command prompt, and enter the directory where Microsoft .NET Framework is installed to, e.g.

   cd C:\Windows\Microsoft.NET\Framework\V2.0.50727

2. Install the service using *installutil.exe*:

   installutil /i *package\service\precompiled*\ORFLogToDbService.exe

   where the path should be the folder where the service executable resides.

Upon successful install, “Example ORF Log Service” should appear in the Service Control Manager list (*Start|Programs|Administrative Tools|Services*).

NOTES

Before installing a new version of the service, the previous version **MUST** be uninstalled first.

LAUNCHING THE WEB SITE

If you have Microsoft Visual Web Developer installed, a simple test-run can be done: open the */package/website/website.sln* file and press *F5* to run it. In case you do not have it installed, please follow the steps below:

1. Launch the Internet Information Services (IIS) Manager (Click *Start|Programs|Administrative Tools|Internet Information Services*)
2. In the tree pane on the left, browse to the web site you would like to extend (e.g. *Default Web Site)* Right-click the site, and click *New|Virtual Directory*. Click *Next*.
3. Enter an alias for the web site (we will use *ORFWebReport* in this example), click *Next*.
4. Click *Browse* to navigate to the folder *package/website/website*. Click *Next*.
5. Allow *Read* and *Run Scripts* permissions for the site, then click *Next*, and click *Finish*.

Configuration

Configuring the example requires administrator privileges.
After the example has been installed, continue with the following steps.

IIS CONFIGURATION

Turning authentication on in IIS is required in order to identify the recipients.

1. Right-click the web site (*ORFWebReport*) in IIS, choose *Properties*.
2. Switch to the *Directory Security* tab, and in the *Anonymous Access and Authentication control* box click *Edit*.
3. Uncheck *Anonymous access*.
4. Check *Integrated Windows Authentication* if it was unchecked. Click *OK* twice to apply the changes.

SQL SERVER CONFIGURATION

The account which ASP.NET (IIS) uses to execute the web page needs access to the database. If it is the default ASPNET user, create a database login for ASPNET, and grant permissions on the *ORFLog* database:

1. In Microsoft SQL Management Studio, browse to *Security|Logins* in the Object Explorer, right-click *Logins*.

2. Select *New login*, select the user, and click *OK*.

3. Browse to *Databases|ORFLog|Security*, right-click *Users* and click *New User*.

4. Enter a user name, choose the recently created login. Switch to page *Securables*, add the *ORFLog* database to the list.

5. Grant (check) *Execute* and *Select* explicit permissions. Click *OK*.

CONFIGURING THE SERVICE

Open the */package/precompiled/ORFLogToDbService.exe.config* file in a text or XML editor and modify the values according to your settings:

Setting	Description
ORFLogFolder	The directory where the ORF log files reside
ConnectionString	Connection string to the database. Some values should be updated according to your settings, i.e. **Data Source=*machinename\service*;** **Initial Catalog=databasename;**
RefreshInterval	This value tells the Service how often it should check the logs and update the database with the newly added entries. The default value is **1 second,** which should be used for testing purposes only: when used in production (i.e. Large (100+ MB) log files) **60-120 seconds is suggested**. To find out what value is the most suitable in your environment, check how many seconds it takes to open a complete log file in the Log Viewer, and multiply this value by three.

NOTES

The service does not start after installation, but later automatically on system startup. In order to apply your new configuration, the service should be restarted.

CONFIGURING THE WEB SITE

Configure the connection string to the database:

1. Open the */package/website/web.config* file in a text or XML editor.

2. Find the *MSSQLConnectionString* tag, and modify the *connectionString* value according to the database server and database name chosen:

   Data Source=*machine\service*;Initial Catalog=*ORFLog*;...

   where *machine\service* is the name of the database server, and *ORFLog* is the name of the database.

The user account executing the ASP.NET page also needs to have sufficient rights to browse the Active Directory.

Uninstallation

UNINSTALLING THE SERVICE

1. Open a command prompt, and enter the directory where Microsoft .NET Framework is installed to, e.g.

   cd C:\Windows\Microsoft.NET\Framework\V2.0.50727

2. Use *installutil.exe* to uninstall the service:

   installutil /u *package\precompiled*\ORFLogToDbService.exe

   where the path should be the folder where the service executable resides.

REMOVING THE WEB SITE

1. Launch the IIS Manager (Click *Start|Programs|Administrative Tools|Internet Information Services*)

2. In the tree pane on the left, browse to the virtual directory created for the web site (*ORFWebReport*).

3. Right-click the website, and click *Delete*. Confirm the dialog by clicking *Yes*.

REMOVING MICROSOFT SQL SERVER OBJECTS

1. Launch Microsoft SQL Server Management Studio and connect to the database server hosting the database.

2. Browse to the *ORFLog* database in the Object Explorer, right-click it, click *Delete*. Confirm the dialog by clicking *OK*.

3. If a login was created for the *ASPNET* user, browse to that (*Security|Logins*), right-click and delete it.

Testing the Installation

STARTING THE SERVICE

The Service can be either started from command prompt:

net start ORFLDSVC

Or using the Service Control Manager (*Start|Programs|Administrative Tools|Services*). In the latter case, select "*Example ORF Log Service*", and click "*Start the service*" (if it is not running).

If a new ORF log file is added to the folder specified by the *ORFLogFolder* setting (or if a new entry is added to an existing log file), the service will start processing the new entries within the interval specified by the *RefreshInterval* setting.

OPEN THE WEB PAGE

Open the URL of the web page in a web browser (if the components are installed locally, and you used the default values: http://localhost/ORFWebReport/ORFWebReport.aspx).

All those blacklisted mails should show up which were sent to the recipient currently logged into Windows (from the last 24 hours).

CHECKING THE EVENT LOG

The service logs failures into the Windows Event Log. You can check it by opening *Start|Programs|Administrative Tools|Event Viewer*, and clicking *Application*. Recent entries with the source "*ORFLDSVC*" and the type "*Error*" should be searched.

When troubleshooting a problem, do not forget that you need to restart the service after the config file (*OrfLogToDbService.exe.config*) was modified.

Notes

We tried to keep this example as simple as possible, so in order to optimalize it for production uses, some modifications should be applied:

- While only a few fields of the log file are required for this example, all columns are transferred to the database. You might want to reduce the number of columns affected.

- While only emails from the last 24 hours are listed to the user, the log records never expire in the database, so the size of the database grows continously. Besides consuming disk space, it can also slow down queries. You might want to periodically delete all unnecessary records. Moreover, the database is far from being normalized.

- It might happen that the system running the service crashes, and restarts only the next day. Since some unprocessed entries may be left in the log file of the previous day, you might want to implement a failover for this case.