Microsoft Script Explorer for   
Windows PowerShell

Reference Implementation

Microsoft Corporation

Abstract

This document describes:

* A reference implementation for using Google as a repository for PowerShell scripts/modules/snippets .
* A reference implementation of a database as a script repository.

Disclaimer: The sample scripts in this guide are not supported under any Microsoft standard support program or service. The sample scripts are provided AS IS without warranty of any kind. Microsoft further disclaims all implied warranties including, without limitation, any implied warranties of merchantability or of fitness for a particular purpose. The entire risk arising out of the use or performance of the sample scripts and documentation remains with you. In no event shall Microsoft, its authors, or anyone else involved in the creation, production, or delivery of the scripts be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use the sample scripts or documentation, even if Microsoft has been advised of the possibility of such damages.

Information in this document, including URL and other Internet website references, is subject to change without notice. Unless otherwise noted, the companies, organizations, products, domain names, email addresses, logos, people, places, and events depicted in examples 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. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2011 Microsoft Corporation. All rights reserved.

Microsoft, Windows, Windows Server, and Windows PowerShell are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries/regions.

All other trademarks are property of their respective owners.

Table of Contents

[1. Reference Implementation (RI) 4](#_Toc332373354)

[2. Sample Reference Implementations 4](#_Toc332373355)

[2.1 Google provider RI 4](#_Toc332373356)

[2.1.1 Create the Google provider 4](#_Toc332373357)

[2.1.2 Google Provider Setup and configuration 4](#_Toc332373358)

[2.1.3 Update the Script Explorer .config file 4](#_Toc332373359)

[2.2 Database provider RI 6](#_Toc332373360)

[2.2.1 Create the database provider 6](#_Toc332373361)

[2.2.2 SQL Server setup and configuration 6](#_Toc332373362)

[2.2.3 Create the database 6](#_Toc332373363)

[2.2.4 Update the Script Explorer .config file 7](#_Toc332373364)

# 1. Reference Implementation (RI)

Microsoft Script Explorer exposes a set of interfaces you can use to create your own Windows PowerShell script repositories. This Script Explorer Reference Implementation (RI) demonstrates how to develop a custom repository provider and integrate it into the Script Explorer application.

The Script Explorer RI is not reccomended for production use. It has been developed for demonstration purposes only. You can use the code that has been provided to either develop similar providers or extend the functionality of the Script Explorer RI.

Usage of the Script Explorer RI must adhere to the Microsoft Software License Terms, which you can find on [the CodePlex website](http://scriptexplorer.codeplex.com/license).

# 2. Sample Reference Implementations

The reference implementation described in this document uses the Script Explorer SDK, which references DLLs from the Script Explorer application.

## Google provider RI

The purpose of this sample is to demonstrate how to leverage any search engine repository in addition to the supplied Bing search engine repository for searching PowerShell scripts and other resources. The Google provider uses the Google Custom Search Engine (Google CSE) for finding PowerShell scripts. This provider also illustrates how you can use the Task Parallel Library to support asynchronous searches through the new IAsyncSearchProvider interface.

2.1.1 Create the Google provider

Use the following steps to create the Google provider:

1. Load the solution for the Reference Implementation in Visual Studio.
2. Build the GoogleProvider project.
3. Copy GoogleProvider.dll to “C:\Program Files (x86)\Microsoft Script Explorer for Windows PowerShell\LocalService\V1\bin”.

2.1.2 Google Provider Setup and configuration

The reference implementation for Google requires the use of an existing Custom Search Engine (CSE). A CSE has information that limits the scope of custom searches. The process for creating a Custom Search Engine is documented at <http://www.google.com/cse/docs>. A Stored CSE (the default) is identified by the “Search engine unique ID” (the “id” attribute in the <CustomSearchEngine> tag). Alternatively, a Linked CSE can be stored at a publicly-accessible URL referred to by the “cref” parameter POSTed to http://www.google.com/cse.

2.1.3 Update the Script Explorer .config file

Two attributes must be configured in the repository element for the Google provider:

1. A valid Google API key for calling the CSE API. This can be found in the Google API console at <https://code.google.com/apis/console>. This key is specified in the configuration file with the “key” attribute of the repository element.
2. A reference to a Stored CSE or a Linked CSE definition. Use “customSearchId” for a Stored CSE or “customSearchReference” for a Linked CSE. Exactly one of these attributes is required.

Once these two settings have been identified, update Microsoft.iX.ScriptExplorer.exe.config to add the new search repository name **GoogleWeb**:

<adm service="wcf">

<repositories>

<repository name="GoogleWeb" displayName="Web Google Search" type="GoogleProvider.SearchProvider, GoogleProvider" key="API-KEY" customSearchId="Stored-CSE" />

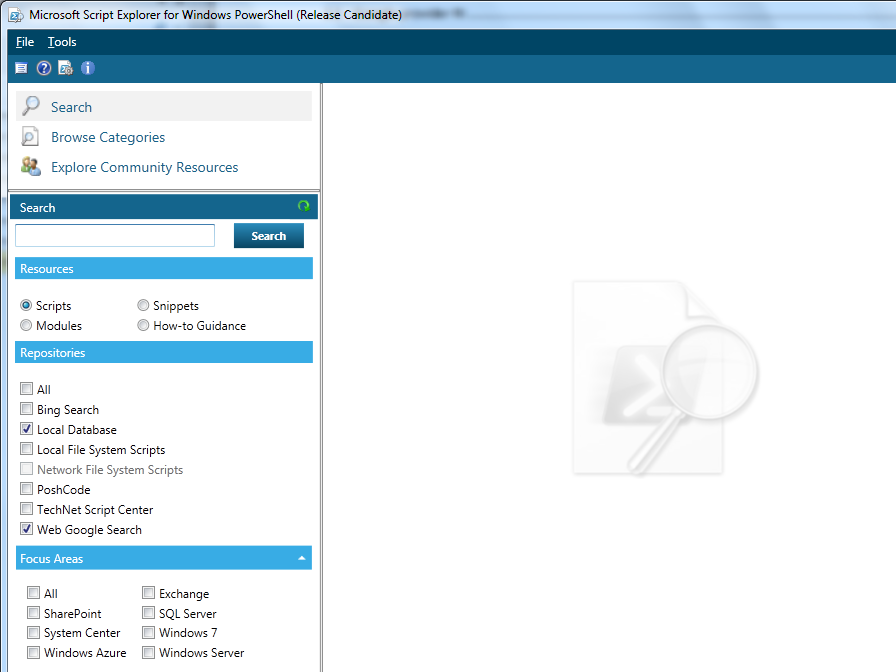
<searchProvider name="Remote" displayName="Azure Script Service" … />

  </ repositories>

</adm>

**Note** The above example demonstrates hosting the GoogleProvider repository locally.  For IIS hosting scenarios, you need to introduce the above configurations in web.config.

After you make these changes, run Script Explorer (or Windows PowerShell ISE). As shown in figure 1, you will now see **Web Google Search** under **Repositories**.



## Database provider RI

The database provider RI demonstrates the following:

* Creating a new script repository using SQL Server.
* Adding the ability to save scripts to the database.

### 2.2.1 Create the database provider

Use the following steps to create the database provider

1. Load the solution for the sample Reference Implementation in Visual Studio.
2. Build the LocalDBProvider project.
3. Copy LocalDBProvider.dll to “C:\Program Files (x86)\Microsoft Script Explorer for Windows PowerShell \LocalService\V1\bin”.

### 2.2.2 SQL Server setup and configuration

You must install Microsoft SQL Express or SQL Server 2008 R2 on the client computer before deploying the Database provider RI.

### 2.2.3 Create the database

1. Use the script named PSDatabase.sql distributed with the source code to create the database. PSDatabase is located in <folder>\ LocalDBProvider\Database.
2. Run the following command (located at %ProgramFiles%\Microsoft SQL Server\100\Tools\Binn): **SQLCMD.exe –i PSDatabase.sql**
3. If you want to test the Database provider, you can use the script provided in <folder>\ LocalDBProvider\Database\PopulateScript.sql to populate your table with sample data.

### 2.2.4 Update the Script Explorer .config file

Update the Microsoft.iX.ScriptExplorer.exe.config or web.config file to add the new search repository name **LocalDatabase** and add the connection string details:

<adm service="wcf">

    <repositories>

<repository saveEnabled="true" name="LocalDatabase" displayName="Local Database" connectionString="localStorage" type="LocalDBProvider.SearchProvider, LocalDBProvider"/>

<searchProvider name="Remote" displayName="Azure Script Service" … />

  </ repositories>

  </adm>

  <connectionStrings>

    <!-- Uncomment and update the connection string so it points to a valid SQL Server -->

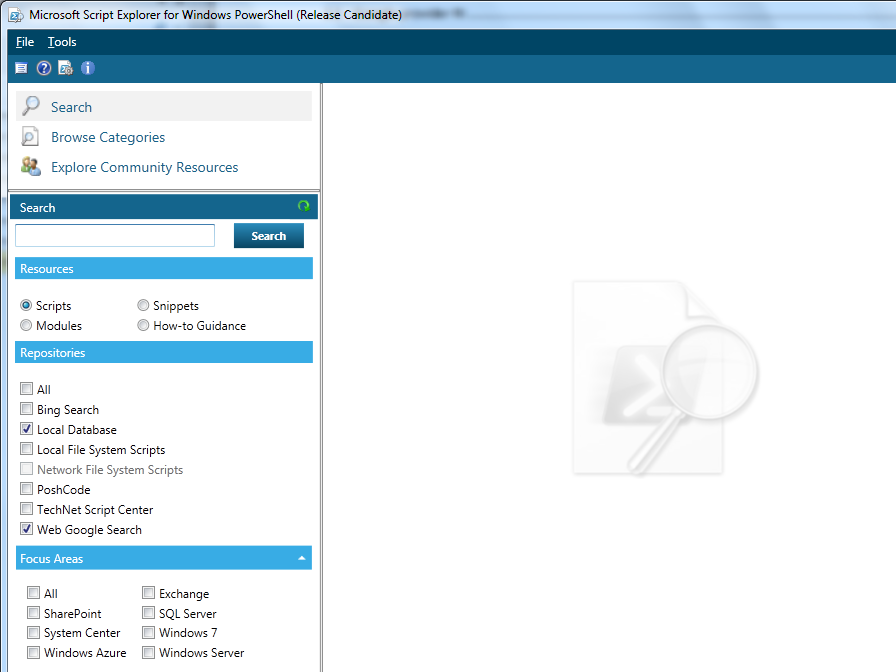
<add name="localStorage" connectionString="SERVER=contosocomputer2;DATABASE=PSDatabase;INTEGRATED SECURITY=true" providerName="System.Data.SqlClient"/>

  </connectionStrings>

**Note** The above example demonstrates hosting the Database repository locally.  For IIS hosting scenarios, you need to introduce the above configurations in web.config.

**Security Recommendation:** The **connectionString** should be encrypted for additional security.

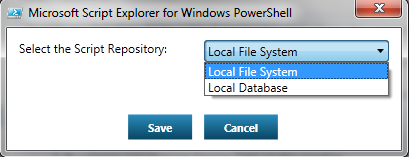
After you make these changes, run Script Explorer (or Windows PowerShell ISE). Figure2 shows **Local Database** under **Script Repositories**.



To enable saving scripts to the database, set **saveEnable** is to **true** in the config file..

The following steps illustrate the save functionality of database providers:

1. A user searches for scripts in the local file system and receives a list of results from which to select a script. The user can choose and save a script.
2. When the user selects to save the script, they see the following dialog box.



The user can choose to save the script to the **Local Database**, and can search for saved scripts in the database repository at a later time.