myLittleAdmin for MS SQL Server
Installation Guide
version 3.5

CONTENT

OVERVIEW

myLittleAdmin is the first web-based management tool specially designed for MS SQL Server 2005 and 2008. It allows managing most objects of MS SQL Server databases and servers through a web browser. myLittleAdmin is fully integrated with the best hosting control panels (Plesk, DNP, Helm, ...) but can also be used as a stand-alone application. myLittleAdmin provides a reliable, efficient and secure solution to manage SQL Server databases in a (web) hosting environment that has already seduced hundreds of hosting providers and IT companies.

CONFIGURATION REQUIREMENTS

myLittleAdmin for MS SQL Server requires the following configuration:
- Microsoft Windows 2000, 2003, 2008
- Microsoft IIS 5, 6,7
- Microsoft SQL Server 2005 & 2008
- Microsoft .NET framework 2.0
- A web browser that supports XHTML and CSS2

myLittleAdmin for MS SQL Server **does not work with Microsoft SQL Server 2000**. An old version of myLittleAdmin with MS SQL 2000 support is still available on http://www.mylittletools.net

INSTALLATION AND CONFIGURATION

1. Open the .zip file and extract it to a folder within the scope of your site. Be sure to extract all the sub-folders.
2. Edit the config.xml file (located at the root of the application)
3. Enter your SQL Server address and name in the <sqlservers> node of the XML document:

```
<sqlservers>
   <sqlserver address="127.0.0.1" name="localhost" />
   <sqlserver address="P27SRV\SQL2K5" name="Prod. SQL Server" />
   ...
</sqlservers>
```

4. Copy your license file (license.config) in the root folder of myLittleAdmin
5. Setup your installation folder as a web application (you can do this through your IIS control panel or your web-host control

panel). If you do not have the rights to setup your installation folder as a web application, then you must copy the whole content of the /bin folder into the /bin folder of your website.

6. You can now access myLittleAdmin at the following address http://www.yourServer.com/yourInstallFolder/

ADDING A NEW INTERFACE LANGUAGE

myLittleAdmin for MS SQL Server support multi-language. Please follow the instructions below to localize myLittleAdmin in your own language.

1. Edit the config.xml file located at the root folder of myLittleAdmin. Add a new language node to the <languages> node with your language culture and name. For instance:

```
<languages>
<language culture="en-US" name="English" default="true" />
<language culture="fr-FR" name="French" />
    <language culture="ja-JP" name="Japanese" />
    ...
</languages>
```

2. Create a new folder in the **/bin** folder of myLittleAdmin. Name it with the culture name of your language. For instance, if you want to add a Japanese translation, then you create a new **ja-JP** folder.

3. Copy files **strings.resources** and **compileRes.bat** located in the **/bin** folder of myLittleAdmin in your newly created folder, and rename **strings.resources** file into **strings.RessourceName.resources**.
For instance : **strings.ja-JP.resources**

4. Edit this resource file with a resource editor application. You can download and use the free Resourcer for .NET by Lutz Roeder: http://www.aisto.com/roeder/DotNet/

5. Now you must translate all the entries from your resource file in your own language.

6. At last, you just need to compile the resource file in a DLL. To do that, you need the **Assembly Linker** application to be installed on your computer. You can find this application in the **Framework SDK** (you can download it here: http://msdn.microsoft.com/netframework/downloads/framework1_1/)

7. Edit the **compileRes.bat** file and change the line to (this sample is for ja-JP culture)

```
C:\WINNT\Microsoft.NET\Framework\v1.1.4322\al.exe
/embed:strings.ja-JP.resources,strings.ja-JP.resources
/out:strings.resources.dll /c:ja-JP
```

8. Launch the **compileRes.bat** batch file by double-clicking on it. You should now have a **strings.resources.dll** located in your language folder.

9. Launch myLittleAdmin and choose your language in the **Preferences** section

ENABLING/DISABLING FEATURES

Each feature of myLittleAdmin can be enabled/disabled. Please follow the instructions below:

1. Feature settings are located in the **xml/profile/default.xml** file. Edit the **xml/profile/default.xml**.

2. Set to **true** or **false** the feature you want to enable or disable.

3. Relaunch myLittleAdmin

 Remarks :

- **System Database:** In order to disable the possibility to list **System databases**, you need to edit the **xml/ treeviewdatabase.xml** file and to remove the **D10** node

- **Security:** In order to disable the possibility to list **Logins / Server roles / Credentials**, you need to edit the **xml/ treeviewsecurity.xml** file and to remove the selected node.

- **Server Objects section:** In order to disable the possibility to list **Backup Devices / Linked servers / Triggers**, you need to edit the **xml/ treeviewserverobjects.xml** file and to remove the selected node.

CONFIGURING WEB HOSTING BACKUP AND RESTORE WIZARDS

FOR BACKUP AND RESTORE FEATURES IN A SHARED SQL SERVER HOSTING ENVIRONMENT, WE RECOMMEND TO USE MYLITTLEBACKUP.

MORE INFO ON HTTP://WWW.MYLITTLEBACKUP.COM

myLittleAdmin for SQL Server offers 2 wizards which have been specially developed for (and with) web hosting companies. These 2 wizards allow users to backup a database and download the backup file to their local machine and to upload a backup file from their local machine in order to restore their database.

1. Edit the **config.xml** file located at the root folder of myLittleAdmin and go to the **<tools>** section

```
<tools>
        <backupwh
            backupfolder="\\Server\dbbackup\"
            backupurl="http://myLittleAdmin.webhost.com/dbbackup"
        />
        <restorewh
            uploadfolder="\\Server\dbupload\"
            deleteafterrestore="true"
        />
    </tools>
```

2. Use the **<backupwh>** section to configure Backup wizard and **<restorewh>** section to configure the Restore wizard.

3. In the **backupfolder** property, specify the full path to the folder where backup files will be stored. SQL Server must be able to access this folder and have write permission. It means that you may need to use UNC path if IIS and SQL Server are not on the same box.

4. In the **backupurl** property, specify the URL to access the folder previously specify in the **backupfolder** property. To do that, you must define a new virtual folder for your myLittleAdmin virtual directory in your IIS console. For security reason, we recommend to disallow the content list.

5. In the **uploadfolder** property, specify the full path to the folder where user backup file will be uploaded. SQL Server must be able to access this folder. ASPNET user must have read/write permissions on this folder. It means that you may

need to use UNC path if IIS and SQL Server are not on the same box.

6. Set **deleteafterrestore** property to true if you want the upload file to be deleted after restore, else false.

Remarks:

- For security reasons, uploaded backup file are renamed before restore.
- Backup files are not deleted after download. You may need to manually delete backup files to retrieve disk space.
- Only databases with a single ROWS file can be restored with the restore wizard.
- In order to be able to restore a database, myLittleAdmin restricts access to the database to SINGLE_USER. Once the restore has been done, the access is set back to MULTI_USER. In case of failure, you may need to manually set the access back to MULTI_USER with the following T-SQL query: **ALTER DATABASE [*dbName*] SET MULTI_USER**

HOW TO CONFIGURE WEB HOSTING BACKUP AND RESTORE WIZARDS STEP BY STEP

CONFIGURATION #1
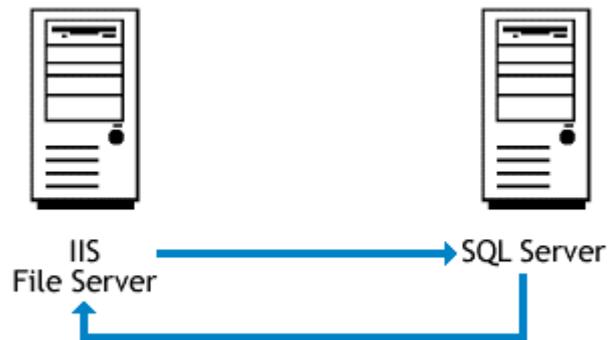
myLittleAdmin is installed on machine A.

SQL Server is installed on machine B.

We will create and upload backup files on machine A.

1.1. Step 1:

Create 2 folders on machine A:
- dbbackup
- dbupload

1.2. Step 2:

Share these folders so that machine B will be able to access them through UNC share.

1.3. Step 3:

Give ASPNER user (machine A) read/write permission on these both folders.

1.4. Step 4:

SQL Server needs to access the file server. The default set of NT credentials used by MSSQLSERVER is the Localsystem account. The Localsystem account has no access to shares on the network as it isn't an authenticated network account. Therefore **SQL Server running**

under this account cannot backup/restore to/from a normal network share. You will then need to use another account to run SQL Server.

Step 1. Create a new local user account on machine B (SQL Server)

Step 2. Launch your SQL Server Configuration manager and change the account used by SQL Server from the local system account to this newly created account. (see below)
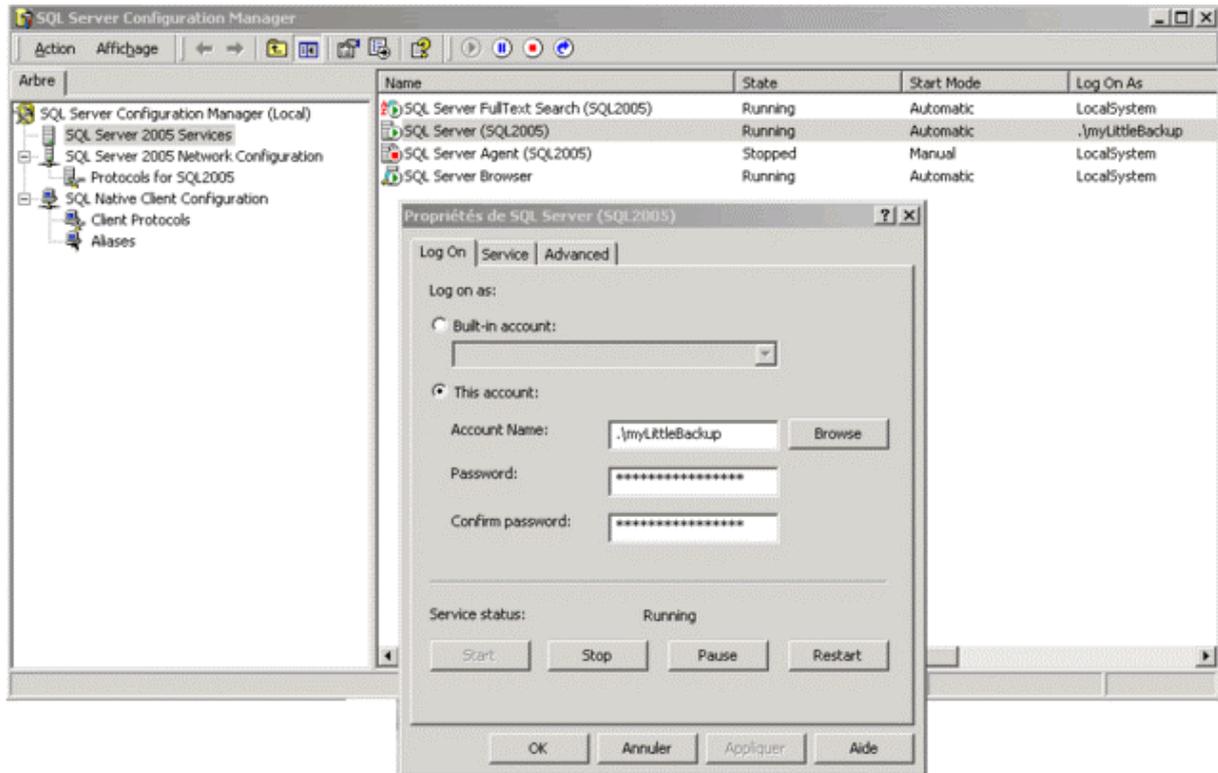
Step 3. Create a new local user account on machine A (IIS, File Server) with exactly the same name and password than the one created on machine B.

Step 4. Give read/write permissions to this newly created user on the backup and upload folders.

1.5.   Step 5:

Define a new virtual directory in your IIS console so that one can access dbbackup folder through Http.

1.6.   Step 6:

Edit your config.xml file

```
<sqlservers>
   <sqlserver
        address="MachineB\SQL2005"
        name="SQL 2005"
        sysadminconnection="true"
        sysadminconnectionid="1"
   />
</sqlservers>

[… snip …]

<tools>
   <backupwh
        backupfolder=="\\MachineA\dbbackup\"
        backupurl="http://serverA/mylittleadmin/dbbackup"
        usesysadminconnection="true"
   />
   <restorewh
        uploadfolder="\\MachineA\dbupload\"
        deleteafterrestore="true"
        usesysadminconnection="true"
   />
</tools>
```

1.7.   Step 7:

Edit your web.config file and define your sysadmin connection login and password.

```
<appSettings>
   <add key="Version" value="3.1b" />
   <add key="Release" value="0018" />
   <add key="Build" value="2484" />
   <!-- Sysadmin connection value //-->
   <add key="SysAdminLogin_1" value="sa" />
   <add key="SysAdminPassword_1" value="sapwd" />
</appSettings>
```

CONFIGURATION #2

myLittleAdmin is installed on machine A.

SQL Server is installed on machine B.

We will create and upload backup files on machine B.

1.8.   Step 1:

Create 2 folders on machine B:

- - dbbackup
- - dbupload

1.9.   Step 2:

Create a new local user account on machine A (IIS)

1.10. Step 3:

Create a local user account on machine B (File server) with exactly the same name and password than the one created on machine A (IIS).

1.11. Step 3b:

If you are using Windows XP or Windows .NET Server 2003 you don't need do anything. If you have Windows 2000, go to Local Security Setting and browse to "User Rights assignment" and locate "Act as part of Operating System" policy. Double click and add the newly created account. You also need to set permission for impersonated user for full control on C:\winnt\Microsoft.NET\Framework\v2.0.50727\Temporary ASP.NET Files.

1.12. Step 4:

Share the backup and upload folders on machine B.

1.13. Step 5:

Give read/write permissions to this newly created user on the backup and upload folders.

1.14. Step 6:

Edit web.config file to use impersonation
Add the following line in the configuration section

```
<identity impersonate="true" userName="UserName" password="Password" />
```

where UserName is the name of the account you created and Password its password.

1.15. Step 7:

Define a new virtual directory in your IIS console so that one can access dbbackup folder through Http. (you will need to use UNC to define this virtual directoty, and you will the same username/Password to access it)

1.16. Step 8:

Edit your config.xml file

```
<sqlservers>
   <sqlserver
        address="MachineB\SQL2005"
        name="SQL 2005"
        sysadminconnection="true"
        sysadminconnectionid="1"
   />
</sqlservers>

[… snip …]

<tools>
   <backupwh
        backupfolder=="\\MachineB\dbbackup\"
        backupurl="http://serverA/mylittleadmin/dbbackup"
        usesysadminconnection="true"
   />
   <restorewh
        uploadfolder="\\MachineB\dbupload\"
        deleteafterrestore="true"
        usesysadminconnection="true"
   />
</tools>
```

1.17. Step 7:

Edit your web.config file and define your sysadmin connection login and password.

```
<appSettings>
   <add key="Version" value="3.1b" />
   <add key="Release" value="0018" />
   <add key="Build" value="2484" />
   <!-- Sysadmin connection value //-->
   <add key="SysAdminLogin_1" value="sa" />
   <add key="SysAdminPassword_1" value="sapwd" />
</appSettings>

<identity impersonate="true" userName="UserName" password="Password" />
```