SQL Server

Adhoc Backup / Restore Solution

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Author** | **Revision No** | **Description** |
| 1/17/2012 | Josh Feierman | 0.1 | Draft |

Contents

[Introduction 4](#_Toc320460421)

[General Design 4](#_Toc320460422)

[SQL Agent 4](#_Toc320460423)

[Impersonation 4](#_Toc320460424)

[Process and Limitations 5](#_Toc320460425)

[Installation Instructions 6](#_Toc320460426)

[Pre-Requisites 6](#_Toc320460427)

[Operating System Permissions 6](#_Toc320460428)

[Backup Location Permissions 6](#_Toc320460429)

[Creation of End User Login 6](#_Toc320460430)

[Entry of Configuration Details 6](#_Toc320460431)

[Execution 6](#_Toc320460432)

[Appendix A. Install Worksheet 7](#_Toc320460433)

# Introduction

A frequent request from development teams is to take an ad-hoc backup of a database and save it off for later use. Relatedly, they frequently wish to have said ad-hoc backup restored over their current database, thereby reverting the database back to some pre-test state.

In the past this required manual DBA intervention, which is not ideal for several reasons, not the least of which is that DBAs frequently have other tasks on their plates, thereby delaying the fulfillment of the request. This can hold up development testing or other efforts.

Using this automated solution, a development team can be enabled to take backups of their database whenever they wish to do so, without involving the DBA team. They are also enabled to restore said backup at their leisure.

# General Design

The solution makes use of several existing technologies within SQL Server.

## SQL Agent

SQL Agent jobs are used to actually perform the restore / backup operations. This allows for asynchronous execution, as well as robust notification / error handling routines. It also allows for the use of proxy accounts, via the SQL Agent Proxy feature. For more information on this feature, see [this MSDN article](http://msdn.microsoft.com/en-us/library/ms189064.aspx) on the subject.

## Impersonation

SQL Server allows a login to be granted the IMPERSONATION right, thereby allowing one login to effectively execute code under the context of another login. This feature is used to overcome a particular limitation of the built-in SQL Agent roles as described below.

To enable a login to start a SQL Agent job, the user must be a member of one of the following roles in the MSDB database: SQLAgentUserRole, SQLAgentReaderRole, or SQLAgentOperatorRole. For detailed descriptions of the roles and their permissions see [this MSDN article](http://msdn.microsoft.com/en-us/library/ms188283.aspx). For the purposes of this discussion, we are using the SQLAgentUserRole fixed role.

The SQLAgentUserRole role allows a user to start / stop / modify SQL Agent jobs that they own. This is good for security, however it introduces a limitation in functionality: this means that rights to start a job can only be given to one login, not series of logins or groups. This is where the IMPERSONATION feature comes into play.

By granting the end users’ logins the IMPERSONATE privilege on the “owner” login that owns the SQL Agent jobs used for the backup / restore operations, we allow all the users to execute the job by impersonating its owner.

# Process and Limitations

The solution creates two SQL Agent jobs: one to allow for the backup of the database, and one to allow for the restore operation. These agent jobs are owned by a special login, created at the time of installation, hereafter referred to as the Job Owner Login. This login is disabled and cannot be used to log into the server; it is only used for impersonation by the end users for the purpose of starting the SQL Agent jobs.

The SQL Agent jobs make use of a special proxy account (hereafter referred to as the Proxy Account), which is given certain minimal rights that allow it to backup and restore the development database.

The main limitation of the process is that it only allows for one backup to be retained. That is, when the backup job is executed, and an existing backup is already present, the existing backup will be over-written. In addition, depending upon when the backup is restored, it could break differential backup chains (a warning which is included in the generated README file that is for the developers).

# Installation Instructions

## Pre-Requisites

Before proceeding, collect all the details listed in the [Installation Worksheet](#_Appendix_A._Install) at the end of this document.

### Operating System Permissions

On the SQL Server where the databases reside, ensure that the ProxyUser is a member of the Users group, and has the “Log on as a batch job” rights in Local Security Policy. The account should not, under any circumstances, be made a local administrator. If the server is a clustered one, ensure this step is taken on all nodes.

### Backup Location Permissions

Ensure that the SQL Server Service account has read/write permissions at the BackupLocation. If the path is a UNC share, do not forget to allow these rights both on the share and the underlying NTFS folders.

### Creation of End User Login

The install scripts assume that the login specified in EndUserLogin already exists on the server, and will abort if it does not.

## Entry of Configuration Details

Open the “config.cmd” file in the build path, and enter the values of the variables collected on the Installation Worksheet in the appropriate place. The .CMD file variables are named the same as those values on the worksheet for easy identification.

## Execution

Run the “Setup.cmd” file to install the solution. If any errors are received, the console will display a notification. In this case, review the “logfile.txt” file for error details. If a failed execution occurs, make sure to run the “Cleanup.cmd” script to clean up all objects before re-running the install.

When the install completes, a README.txt file will be generated that should be sent to the requestor. This will include specifics about how to execute the jobs.

# Appendix A. Install Worksheet

The following is a list of variables necessary for the installation.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable Name** | **Description** | **How To Obtain** | **Value** |
| ServerName | The name of the server on which the database to be enabled for adhoc backup resides. | The requestor of the setup should know this value. |  |
| DatabaseName | The name of the database to be enabled for ad-hoc backup. | The requestor of the setup should know this value. |  |
| ProxyUser | An Active Directory login which the SQL Agent jobs will execute as. This will be granted certain minimum rights to enable it to backup / restore the database. | Can be an arbitrary account, but it is recommended to be an account dedicated for this sort of purpose. Under no circumstances should the SQL Agent or SQL Server service account be used, nor should any other account with sysadmin rights. |  |
| ProxyUserPassword | The password for the ProxyUser. | Obtained at time of creation of the ProxyUser account. |  |
| BackupLocation | The full UNC or local path (do not use a mapped drive letter) where the ad-hoc backups will be stored. | Can be any arbitrary location. However, it is recommended that this be a separate path from any of the following: 1. SQL Server Data / Log file location  2. SQL Server standard backup location (that is, where the regularly scheduled backups are stored). |  |
| EndUserLogin | The login or group that the end-users use to access the SQL server. This will be granted rights to impersonate the JobOwnerLogin. | Should be received at time of request. As a best practice, use an Active Directory group and not an individual login. |  |
| UseLitespeed | Indicates if the solution will use SQL Litespeed to backup / restore the databases. | If the server is licensed for SQL Litespeed, it would be recommended to use this functionality as it will improve performance. |  |
| NotificationEmail | An e-mail address that notifications will be sent to when the jobs finish. | Should be given by the requestor. As a best practice use a distribution group and not an individual address. |  |