THINKENTERPRISE™

# THINK Subscription Backup Stored Procedure

**Feb 2013**

THINK subscription®
Digital River™

# Table of Contents

## Revisions

| Revision Number | Date | Changes |
|---|---|---|
| 1.00 | February 5, 2013 | Original Document. |

## Overview

The user database backup stored procedure is accessible to all THINK Subscription personnel to backup databases on the THINK Subscription internal database server. It can be used to take full and differential backups as well as transaction log backups in both native and Litespeed formats. The stored procedure (sp) backs up specified databases to a central location in a consistent format so that others may use and access the backup file for restore and use. The sp also handles additional THINK Subscription specific requirements such as tracking PCI cleaned databases, logging sp usable, and overall database management auditing. This stored procedure should be used for all database backups by all departments

## Technical Specifications

The following lists technical specifications about the MSSQL server and other related items:

- The THINK Subscription internal database server's FQDN: PRVTHKDB01.mpls.digitalriver.com. This server will be referred to as the database server or DB server throughout the remainder of this document.
- There are four separate instances on the DB server:
    - An MSSQL Server 2008 R2 (MSSQL **10.5**) support instance called: *SUPPORT*. It operates over port 5150.
    - An MSSQL Server 2008 R2 (MSSQL **10.5**) Quality Assurance instance called: *QA*. It operates over port 5151.
    - An MSSQL Server 2008 R2 (MSSQL **10.5**) Development instance called: *DEV*. It operates over port 5152.
    - An MSSQL Server 2012 (MSSQL **11.0**) test instanced called: *SQL11*. It operates over port 5153.
- The user database backup sp is called: *usp_THKBackupDB* under the dbo schema (dbo.usp_THKBackupDB). It resides on the dbAdmin database on the DB server.
- Every instance has a dbAdmin database which houses server sp's as well as metadata used for the sp's and SSRS reporting.
- The db_owner of all THINK Enterprise databases uses the *thkapp* SQL login
- Shares are located on the DB Server for database backups and restores. See the Shares subsection for more information.
- Server and database information can be accessed through SSRS reporting at http://PRVTHKDB01/Reports. See the Reporting section for more details.

### Shares

Each department has their own share on the database server that employees can use for their database backups. The *usp_THKBackupDB* sp saves all backups to one of the following shares (depending on which instance the sp was executed). Security Access to each share is specified by department. Some departments have read access to multiple shares. All departments have read/write access to their own share.

| Share Name | UNC Path | Security - Read access by department |
|---|---|---|
| Support_Backup_Files | \\PRVTHKDB01\Support_Backup_Files | All departments |
| QA_Backup_Files | \\PRVTHKDB01\QA_Backup_Files | QA, Dev |
| Dev_Backup_Files | \\PRVTHKDB01\Dev_Backup_Files | Dev |

### Aliases

It is recommended that each user create an MSSQL Server Alias for all instances to which they have rights to/are going to access. An alias can be used when connecting to an MSSQL Server in place of the actual server name, instance, and port. It can also be used when creating ODBC connections or any other place that the DB server name can be entered. To create an MSSQL Server Alias follow the provided steps:

(Windows 7)

1. Start | All Programs | Microsoft SQL Server *<version>* | Configuration Tools | SQL Server Configuration Manager

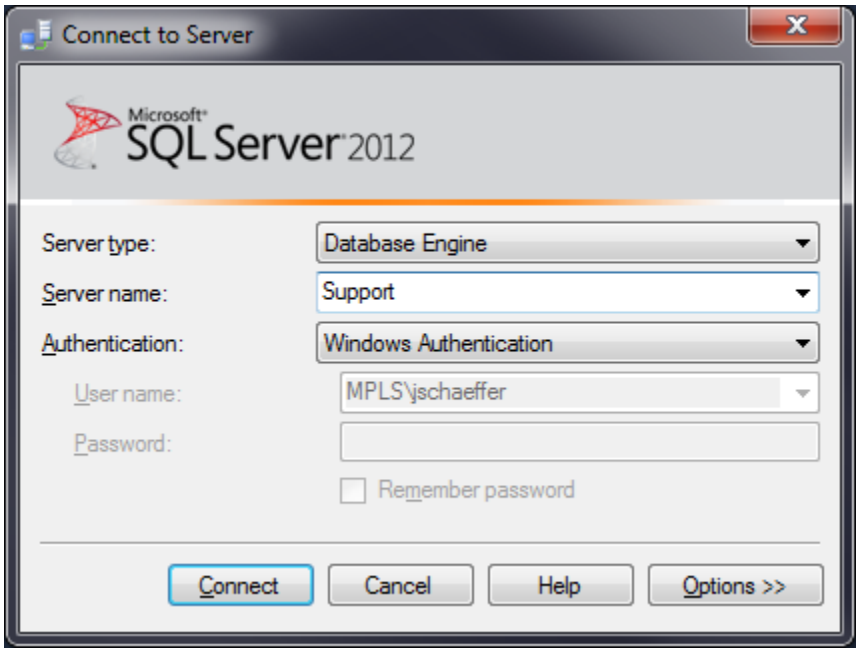
(Windows 8)

1. All Apps | Microsoft SQL Server *<version>* | SQL Server Configuration Manager
   **Note:** If you have multiple versions of MSSQL installed on your machine, select your most recent MSSQL Server version
2. On the *Sql Server Configuration Manager* window expand the "SQL Native Client *<version>* Configuration (32-bit)" node and select "Aliases"
3. Click on the Action menu and hit New Alias. The *Alias* window will appear. Enter in the requested information

| | |
|---|---|
| **Alias Name** | The name of the alias you want to create. It should be short, simple, and easy to remember. Aliases are not case sensitive. Ex: QA |
| **Port No** | The port number that the server your alias is connecting to operates over. See Technical Specifications for more details |
| **Protocol** | The protocol that the server your alias is connecting to operates on. This should be left as the default of TCP/IP. If this is not the default change it to TCP/IP. |
| **Server** | The name of the server and instance your alias will connect to. It should be given in the format *SERVERNAME\INSTANCE_NAME*. Do not include the port number. |

4. Repeat the above steps to create additional aliases

As stated above an alias can be used as an alternative to entering the full server name, instance, and port number. For example:




## Connecting to the Database Server

Connecting to the DB server can be accomplished in two ways. THINK Subscription personnel authorized to access the database server may connect using their Microsoft Windows account. Windows accounts have rights to query, insert, update, and delete data from all THINK Enterprise databases. All Windows accounts also have rights to run a certain set of stored procedures (see Reporting on how to view which stored procedures can be executed), which includes the *usp_THKBackupDB* sp. All stored procedures must be run from a Windows account.

The second method to connect to the DB Server and the databases it houses is through a single MSSQL login. All database instances have a SQL account called *thkapp*. This account is mapped to every THINK Enterprise database and has db_owner access to each of them. The account should be used for the following purposes: 1) Creating an ODBC connection to a THINK Enterprise database and 2) upgrading a THINK Enterprise Database. You are free to use the *thkapp* login for any and all purposes outside the scope of the two given above, however it is considered easier and quicker to use your Windows credentials to query and/or modify any THINK Enterprise databases. It should be noted that the *thkapp* users does not have rights to run any non-THINK Enterprise stored procedures (including the *usp_THKRestoreDB* sp) and cannot access any SSRS reports.

See the Technical Specifications section for information on database name, instance name, and port numbers

## Stored Procedure Syntax

A database is backed up by the calling the *usp_THKBackupDB* sp.

### Parameter List

The following list and subsections are parameters that the sp accepts. Note that all free form parameters must be enclosed in single quotes (‘):

- @setDbName
- @setClient
- @setBackupType
- @setBackupMethod
- @setDbType
- @setProbNbr
- @setBackupRetention

#### @setDbName – REQUIRED { FREE-FORM }

**REQUIRED:** This parameter sets the name of the database that will be backed up. You may enter a maximum of 128 characters. The parameter is required.

#### @setClient – REQUIRED { FREE-FORM }

**REQUIRED:** The value of this parameter is used when creating the backup file name. The backup file name consists of several components the first being the value of the @setClient parameter. When restoring THINK Subscription client databases the parameter should be set to a the name of the client that provided the database, however, the @setClient parameter can be a misnomer as the value does not have to be set to an actual THINK Subscription client. In more general terms it simply refers to a one word, or concatenated short phrase that describes the database purpose. You may enter a maximum of 128 characters. This parameter is required.

#### @setBackupType – OPTIONAL { ‘full’ | ‘diff’ | ‘log’ }

This parameter affects the type of backup that will be performed on the database. The parameter must be set to ‘full’, ‘diff’, or ‘log’. If the @setBackupType parameter is set to ‘full’ then the sp will take a full backup of the database. If the @setBackupType parameter is set to ‘diff’ then the sp will take a differential backup of the database. Differential backups require a full database backup. All available database on the DB server has corresponding full backups, however if you do not know when a full backup was taken it is recommended that you review the backup history to find when the last full backup was taken (which could be as far back as the original restore). Alternatively you could create a full backup as your new reference point and then take differential backup going forward. If the @setBackupType parameter is set to ‘log’ then a transaction log backup will be taken. All THINK Enterprise databases are set to full recovery so that transaction log

backups may be taken, however, it should be noted that shrinking the log file(s) of all the THINK Enterprise databases on the DB server does occur from time to time at irregular intervals. It is best to coordinate with others if transaction log backups are needed. The default is set to 'full'.

### @setBackupMethod – OPTIONAL { 'native' | 'litespeed' }

This parameter affects how the backup formatting will be performed. The parameter must be set to either 'native' or 'litespeed'. If the @setBackupMethod parameter is set to 'native' then the sp will perform the database backup in a native MSSQL format and append a ".bak" file extension to the backup file. If the @setBackupMethod parameter is set to 'litespeed' then the sp will perform the database backup in a litespeed format and append a ".sls" file extension to the backup file. Note that all backups are performed on MSSQL 10.5 or higher, it is not possible to restore any of these backups (native or litespeed) on any SQL server running a lower version. The default is set to 'litespeed'.

### @setDbType – OPTIONAL { 'T' | 'L' | 'S' | 'C' | 'Q' | 'D' | 'O' }

Sets the type of database that is being backed up. The parameter must be set to one of the following, or NULL:

- T: Test database – customers or internal.
- L: Clients' Live database.
- S: Staging database – some customers have separate staging db's.
- C: Conversion database.
- Q: QA database – customers or internal.
- D: Development database – customers or internal.
- O: Other.

The value of this parameter is used when creating the backup file name. The backup file name consists of several components one of which is set to the value of the @setDbType parameter. The default is set to NULL.

### @setProbNbr – OPTIONAL { FREE-FORM }

This is used to associate a specific problem number to a backup. This can be helpful when taking a "snapshot" of a database for debugging, reporting, testing, or some other form of use by another individual or can be taken to archive the database for longer term storage. When the database is taken out of storage it can easily be referenced to a specific issue or problem, by the case number that was assigned to it. The default is set to NULL.

### @setBackupRetention – OPTIONAL { integer }

The @setBackupRetention parameter controls how long a database backup will be kept on the SQL Server in days. You must enter an integer between 0 and 999. Setting the @setBackupRetention to 0 will keep the database backup indefinitely. The default is 90.

## Examples

Use the following guidelines when backing up databases on the DB server.

## Backup File Naming Convention

Coming Soon!!!

# Security

Security is controlled to the DB server through several Windows groups and service accounts as well as one SQL Server login. Each department is placed under their own Windows group with appropriate users being members of those groups:

- MPLS\Commerce – THK Support Read Write
- MPLS\Commerce – THK QA Read Write
- MPLS\Commerce – THK Dev Read Write

If for any reason you should need to reference a specific instance or department you may include the following group which includes all the users under it. In addition to the Windows groups the SQL Server instances each run under their own accounts:

- The *SUPPORT* instance runs under the: **MPLS\db-thk-sup** service account
- The *QA* instance runs under the: **MPLS\db-thk-qa** service account
- The *DEV* instance runs under the: **MPLS\db-thk-dev** service account
- The *SQL11* instance runs under the: **MPLS\db-thk-sql11** service account

## Reporting

Coming Soon!!!