THINKENTERPRISE™

THINK Subscription Restore Stored Procedure

Feb 2013

INTERNAL

THINK subscription® Digital River™

Table of Contents

INTERNAL

Revisions

Revision Number	Date	Changes
1.00	November 27, 2012	Original Document.
1.01	November 30, 2012	Grammatical Change – Under Overview, changed “database” to “databases”.
2.00	November 30, 2012	Document Change – Added “Shares” subsection under “Technical Specifications” section. Under the “Technical Specifications” added additional bullet point: Shares are located on the DB Server for database backups and restores. See the Shares section for more information.
3.00	January 17, 2013	Document Change – Under “@setRecovery – OPTIONAL { ‘y’ | ‘n’ }” subsection removed the following: This option cannot be used if the @setCreate parameter is set to ‘y’. Under “Examples” subsection added the following: Also note that the ‘mit’ database is first created with the first call to the sp by setting the @setCreate parameter to ‘y’. This parameter is not specified in all other calls to the sp (which takes the default of ‘n’). Under the “@setRecovery – OPTIONAL { ‘y’ | ‘n’ }” subsection removed the **Note** and added the following paragraph: To restore differential backups or transaction log files you must first restore the database with a full backup and leave the database as non-operational by setting the @setRecovery parameter to ‘n’… Under the “@setBackupFile – REQUIRED { FREE-FORM }” subsection changed “2000” to “4000”. Grammatical Change – Added periods to several sentences that were missing them.
3.10	January 21, 2013	Document Change – Under “@setBackupFile – REQUIRED { FREE-FORM }” subsection added the following: To restore multi-file backups put the path and filename of all included files in the @setBackupFile parameter separated by a semi-colon. See the Examples section for more information. Added an example to the Examples section.
4.00	February 4, 2013	Document Change - Completed the “Connecting to the Database Server” section. Completed the “Security” section. Document Addition - Added the “Reporting” section.
4.10	February 5, 2013	Document Addition - Under the “Technical Specifications” added additional bullet point: Server and database information can be accessed through SSRS reporting at http://PRVTHKDB01/Reports. See the Reporting section for more details.

INTERNAL

		Under the “Technical Specifications” section added, “and SSRS reporting”. Under “Alias” subsection added step to create alias for Windows 8. Grammatical Change – Under the “Overview” section changed “for” to “by”. Under “Parameter List” subsection changed “section” to “subsection”. Under “@setClient – OPTIONAL { FREE-FORM }” subsection changed “80” to “128” Document Change – Under “shares” subsection italicized “usp_THKRestoreDB”.

INTERNAL

Overview

The user database restore stored procedure is accessible to all THINK Subscription personnel to restore databases, both MSSQL native and Litespeed, to the THINK Subscription internal database server. It can be used to restore over existing databases or to create new databases. The stored procedure (sp) handles additional SQL Server administrative tasks as well as THINK Subscription specific requirements such as creating SQL users to the restored database, changing bank definition settings to point to test servers, audit logging and tracking, etc. This stored procedure should be used for all database restores 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 restore sp is called: *usp_THKRestoreDB* under the dbo schema (dbo.usp_THKRestoreDB). 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. You can point to a share in the *usp_THKRestoreDB* sp when needing to restore your own backup or someone else's. 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:

INTERNAL

(Windows 7)

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

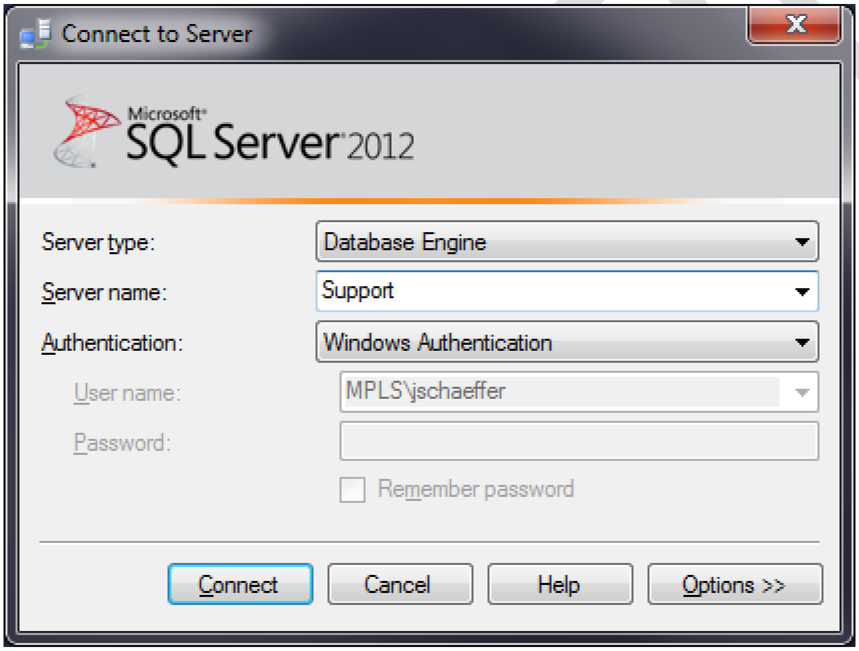
(Windows 8)

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: Support
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:



Server type: Database Engine

Server name: Support

Authentication: Windows Authentication

User name: MPLS\jschaeffer

Password:

□ Remember password

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_THKRestoreDB* sp. All stored procedures must be run from a Windows account.

FINAL

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 restored or created by the calling the *usp_THKRestoreDB* sp.

Parameters 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
- @setBackupFile
- @setRecovery
- @setCreate
- @setUserRights
- @setClient
- @setDbType

@setDbName – REQUIRED { FREE-FORM }

REQUIRED: This parameter sets the name of the database that is going to be restored. You may enter up to a maximum of 128 characters. If the @setCreate parameter is set to ‘n’ then this value must match an existing database on the server. If the @setCreate parameter is set to ‘y’ then this will be the name of the new database. This parameter is required.

Note: The following convention is recommended when creating databases on the DB server. The first character should be lower case. The database name should not include underscores (_), hypens (-), or spaces and all subsequent words, after the first, should be capitalized. Client names should be listed first (if applicable), followed by employee names/database purpose. If any numbers are present in the database name they should be put at the tail end of the string. For example:

- newsLimited
- oag
- xRats735
- ipmJustin
- ucPress55872
- internalTest74

Database name conventions are only recommended, there is no enforcement of the naming policy.

INTERNAL

@setBackupFile – REQUIRED { FREE-FORM }

REQUIRED: This parameter sets the backup directory and filename of the backup being used to restore over the database specified in the @setDbName parameter. You may enter up to a maximum of 4000 characters. To restore multi-file backups put the path and filename of all included files in the @setBackupFile parameter separated by a semi-colon. See the Examples section for more information. This parameter is required.

@setRecovery – OPTIONAL { 'y' | 'n' }

This parameter sets the recovery model the database should be in after the restore is completed. The parameter must be set to either 'y' or 'n'. If the @setRecovery parameter is set to 'y' then the database is set to RESTORE WITH RECOVERY. This leaves the database ready to use by rolling back uncommitted transactions. Additional transaction logs cannot be restored. If the @setRecovery parameter is set to 'n' then the database is set to RESTORE WITH NORECOVERY. This leaves the database non-operational and does not roll back uncommitted transactions. Additional transaction logs can be restored. Setting the @setRecovery parameter to 'n' should be used when restoring differential and transaction log backups. The default is set to 'y'.

To restore differential backups or transaction log files you must first restore the database with a full backup and leave the database as non-operational by setting the @setRecovery parameter to 'n'. Additional calls to the sp can them be made to restore any remaining transaction logs or a differential backup. In order to use the database after all restores have been completed make sure to change the database so that it is operational after the last restore. This would be done by setting the @setRecovery parameter to 'y' or leaving the @setRecovery parameter out of the last call to the sp, as it defaults to 'y'. See the Examples section for more information.

Be aware that almost all other parameters are ignored when the @setRecovery parameter is set to 'n'. However, it is still possible to create databases when restoring from multiple media sets. The @setCreate parameter should be set to 'y' when calling the first sp and then should either explicitly be set to 'n' or not be set at all on all other calls to the stored procedure. If the @setCreate parameter is set to 'y' on any subsequent calls to the sp, then an error will occur, informing you that the database has already been created.

@setCreate – OPTIONAL { 'y' | 'n' }

Sets whether a new database should be created and the backup file specific in the @setBackupFile parameter should restore over the newly created database, or sets if the backup file should be restored over an existing database. The parameter must be set to either 'y' or 'n'. If the @setCreate parameter is set to 'y' then a new database is going to be created on the DB server with the name specified in the @setDbName parameter (see the @setDbName parameter note). If the @setCreate parameter is set to 'n' then a new database will not be created on the database server and the backup file specified in the @setBackupFile parameter will be used to restore over an existing database. The default is set to 'n'.

@setUserRights – OPTIONAL { 0 | 1 | 2 }

This parameter affects how MSSQL users are restored or kept on the original database being restored over and the database that is restored. The parameter must be set to 0, 1, or 2. If the @setUserRights parameter is set to 0 then the sp will save the current list of MSSQL users on the original database as long as those users have corresponding MSSQL logins on the server. Once the database is restored, the sp will then restore those users (and only those users) to the database. This option should only be used for specific, needed situations. If the @setUserRights parameter is set to 1 then the sp will delete all current MSSQL users of the restored database and then subsequently add a predefined list of users to the restored database. This predefined list varies depending on which instance the database is restored on. It is also possible to skip the user deletion portion of the sp by setting the @setUserRights parameter to 2. This will simply ignore the current MSSQL

INTERNAL

users on the restored database and just add the users from the predefined list, regardless of what is already there. The default is set to 1.

@setClient – OPTIONAL { FREE-FORM }

This is the name of the client whose database is being restored. You may enter up to a maximum of 128 characters. This parameter is not used in the actual restoring, cleaning, or logging process and does not need to be set in most cases. See the backup stored procedure documentation for more practical uses of this parameter. The default is set to NULL.

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

Sets the type of database a THINK Subscription client provided. 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.

This parameter is not used in the actual restoring, cleaning, or logging process and does not need to be set in most cases. See the backup stored procedure documentation for more practical uses of this parameter. The default is set to NULL.

Examples

Use the following examples as guidelines when restoring or creating databases on the database server.

This example shows the sp with its bare minimum requirements. It restores over an existing database on the DB server called "baseline" with the backup file called "baseline_sql08_735131.bak" located on the database server under the folder "Support_Backup_Files"

```
USE [dbAdmin];
GO

EXEC dbo.usp_THKRestoreDB @setDbName = 'baseline'
       ,@setBackupFile = '\\PRVTHKDB01\Support_Backup_Files\baseline_sql08_735131.bak';
GO
```

The next example creates a database called "blackwell' and restores the backup "think_database_backup.bak" from the "QA_Backup_Files" directory on the DB server over the created database.

```
USE [dbAdmin];
GO

EXECUTE dbo.usp_THKRestoreDB @setDbName = 'blackwell'
       ,@setBackupFile = '\\PRVTHKDB01\QA_Backup_Files\think_database_backup.bak'
       ,@setCreate = 'y';
GO
```

The below illustrates how to restore a Litespeed database from the hosted center (DC2) as well as its corresponding differential backup. The sp restores over the database "genAm" which is already present on the database server. Notice

INTERNAL

that the @setRecovery parameter is set to ‘n’ for the first call to the sp (the backup of the full database) and ‘y’ for the second call to the sp (the differential backup), even though, technically, the @setRecovery parameter did not explicitly need to be set for the second call. Users should note that the ordering below was intended. The full backup must be specified first.

```
USE [dbAdmin];
GO

EXEC dbo.usp_THKRestoreDB @setDbName = 'genAm'
       ,@setBackupFile =
'\\PRVTHKDB01\Support_Backup_Files\DC2THKHDBSQL01$THKPRD02_GA24190_Live_201211250202_full.sls'
       ,@setRecovery = 'n';
GO
EXEC dbo.usp_THKRestoreDB @setDbName = 'genAm'
       ,@setBackupFile =
'\\PRVTHKDB01\Support_Backup_Files\DC2THKHDBSQL01$THKPRD02_GA24190_Live_201211270200_diff.sls'
       ,@setRecovery = 'y';
GO
```

Restoring transaction log files is very similar to restoring differential backups. The below illustrates how to restore three transaction logs and a full database backup. Again, note the ordering. Also note that the ‘mit’ database is first created with the first call to the sp by setting the @setCreate parameter to ‘y’. This parameter is not specified in all other calls to the sp (which takes the default of ‘n’).

```
USE [dbAdmin];
GO

EXEC dbo.usp_THKRestoreDB @setDbName = 'mit'
       ,@setBackupFile =
'\\PRVTHKDB01\Support_Backup_Files\DC2THKHDBSQL01$THKPRD02_M20036_Live_201211250238_full.sls'
       ,@setCreate = 'y'
       ,@setRecovery = 'n';
GO
EXEC dbo.usp_THKRestoreDB @setDbName = 'mit'
       ,@setBackupFile =
'\\PRVTHKDB01\Support_Backup_Files\DC2THKHDBSQL01$THKPRD02_M20036_Live_201211250307_log.sls'
       ,@setRecovery = 'n';
GO
EXEC dbo.usp_THKRestoreDB @setDbName = 'mit'
       ,@setBackupFile =
'\\PRVTHKDB01\Support_Backup_Files\DC2THKHDBSQL01$THKPRD02_M20036_Live_201211250407_log.sls'
       ,@setRecovery = 'n';
GO
EXEC dbo.usp_THKRestoreDB @setDbName = 'mit'
       ,@setBackupFile =
'\\PRVTHKDB01\Support_Backup_Files\DC2THKHDBSQL01$THKPRD02_M20036_Live_201211250507_log.sls'
       ,@setRecovery = 'y';
GO
```

The next example shows how to restore a multi-file backup. This is a backup set that has more than one media family in the same media set. There are three media families (files) in the media set; each is separated by a semi-colon. Notice that each file must be preceded by its path.

```
USE [dbAdmin];
GO
```

INTERNAL

```
EXEC dbo.usp_THKRestoreDB @setDbName = 'baseline'
       ,@setBackupFile =
'\\PRVTHKDB01\QA_Backup_Files\sales_adm_20130117_1517MST_7.3.5.134_O_clean_00000_full_1_of_3.sls;\\PRVTHKD
B01\QA_Backup_Files\sales_adm_20130117_1517MST_7.3.5.134_O_clean_00000_full_2_of_3.sls;\\PRVTHKDB01\QA_Bac
kup_Files\sales_adm_20130117_1517MST_7.3.5.134_O_clean_00000_full_3_of_3.sls'
```

If for some reason MSSQL users need to be kept on the original database and added to the restored database you can change the @setUserRights parameter. The below restores over a database called “ebsco”.

```
USE [dbAdmin];
GO

EXECUTE dbo.usp_THKRestoreDB @setDbName = 'ebsco'
       ,@setBackupFile = '\\PRVTHKDB01\Dev_Backup_Files\degruyterDB.bak'
       ,@setRecovery = 'y'
       ,@setUserRights = 0;
GO
```

The below example includes all options. It restores a Litespeed database backup called “fairfax_kturpin_20121101_1530MST_7.3.5.134_S_clean_55866.sls” over an existing database on the database server called “fairfaxOld” with a recovery model of WITH RECOVERY (@setRecovery parameter) and specifies to just restore a specified list of MSSQL users to the database without deleting whatever current users are already added to the database. In addition it also specifies who the client is in the @setClient parameter (not used) and what the database type is (in this case staging) in the @setDbType parameter (not used).

```
USE [dbAdmin];
GO

EXECUTE dbo.usp_THKRestoreDB @setDbName = 'fairfaxOld'
       ,@setBackupFile =
'\\PRVTHKDB01\Dev_Backup_Files\fairfax_kturpin_20121101_1530MST_7.3.5.134_S_clean_55866.sls'
       ,@setRecovery = 'y'
       ,@setUserRights = 2
       ,@setCreate = 'n'
       ,@setClient = 'FairFax'
       ,@setDbType = 'S';
GO
```

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:

INTERNAL

- 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!!!

INTERNAL