

# **NetVault® LiteSpeed® for SQL Server**

**version 7.0.1**

User Guide

© 2012 Quest Software, Inc.  
ALL RIGHTS RESERVED.

This guide contains proprietary information protected by copyright. The software described in this guide is furnished under a software license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of the applicable agreement. No part of this guide may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of Quest Software, Inc.

The information in this document is provided in connection with Quest products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Quest products. EXCEPT AS SET FORTH IN QUEST'S TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT, QUEST ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL QUEST BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF QUEST HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Quest makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Quest does not make any commitment to update the information contained in this document.

If you have any questions regarding your potential use of this material, contact:

Quest Software World Headquarters  
LEGAL Dept  
5 Polaris Way  
Aliso Viejo, CA 92656  
email: [legal@quest.com](mailto:legal@quest.com)

Refer to our Web site ([www.quest.com](http://www.quest.com)) for regional and international office information.

### **Patents**

This product includes patent pending technology.

### **Trademarks**

Quest, Quest Software, the Quest Software logo, Simplicity at Work, LiteSpeed and NetVault are trademarks of Quest Software, Inc., and its subsidiaries. See <http://www.quest.com/legal/trademarks.aspx> for a complete list of Quest Software's trademarks. Other trademarks are property of their respective owners.

NetVault LiteSpeed for SQL Server 7.0  
User Guide  
July 28, 2012

# Table of Contents

<b>About Backing Up/Restoring with LiteSpeed</b> .....	<b>13</b>
<b>LiteSpeed User Interface</b> .....	<b>15</b>
Navigation Pane.....	15
Central Pane.....	16
Background Tasks Pane.....	17
Properties Pane.....	17
Toolbar.....	18
<b>Configure LiteSpeed for First Use</b> .....	<b>19</b>
Register Central Repositories.....	19
Select a Central Repository.....	19
Register and Group Server Instances.....	19
Register Server Instances.....	19
Create Server Groups.....	21
Assign Server Instances to Server Groups.....	22
About Categorizing Server Instances.....	22
Change Server Instance Grouping Methods.....	22
Create Categories.....	23
Assign Server Instances and Databases to Categories.....	23
Configure LiteSpeed Defaults.....	24
Processor Affinity.....	25
Configure LiteSpeed Options.....	26
LiteSpeed General Options.....	26
Backup Manager Options.....	28
Log Shipping Options.....	28
Job Manager Options.....	29
Log Reader Options.....	30

---

<b>Back Up Databases</b> .....	<b>31</b>
LiteSpeed Backup Templates .....	31
Create Backup Templates .....	31
Deploy Backup Templates .....	37
Back Up Individual Databases .....	38
Back Up Multiple Databases .....	46
LiteSpeed's Logic for Backing Up Multiple Databases .....	46
Multi-Database Backup Wizard .....	48
Fast Compression .....	48
Quick Start .....	49
Backup Files and Folders .....	49
Full Backup Conditions .....	50
Backup Escalation .....	51
Backup Verification .....	52
Cleanup .....	52
Notification .....	52
Backup Jobs .....	53
Double Click Restore Executables .....	53
Double Click Restore Naming Conventions .....	53
Compression Levels .....	54
Adaptive Compression .....	55
Encryption .....	56
Test Optimal Backup Settings .....	56
Backup Analyzer Wizard .....	57
Backup Analyzer Tab .....	57
Network Resilience .....	58
Smart Cleanup .....	58
<b>Automate Maintenance Tasks</b> .....	<b>61</b>
About Automating Maintenance Tasks .....	61

---

Legacy and SSIS Maintenance Plans .....	61
Native SQL Server and LiteSpeed Maintenance Plans .....	61
About Creating Maintenance Plans .....	62
Back Up Databases Using Maintenance Plans .....	66
Clean Up Maintenance Plans .....	71
Copy Maintenance Plans .....	71
Automate Similar Backup Tasks on Multiple Instances .....	72
<b>Restore Databases .....</b>	<b>74</b>
Restore Databases Using the Restore Wizard .....	74
Restore Double Click Restore Executables .....	77
Manually Restore a Master Database .....	77
<b>Restore Objects .....</b>	<b>80</b>
Restore Objects in the LiteSpeed UI Console .....	80
Review the Backup File Contents .....	81
Restore Tables and Schemas .....	82
Object Level Restores from TSM Backups .....	83
Execute SELECT Statements .....	83
Supported SELECT Statements .....	84
<b>View Activity and History .....</b>	<b>86</b>
View Activity and History in Backup Manager .....	86
View Maintenance Plans Activity and History .....	88
<b>Use Command-Line Interface .....</b>	<b>90</b>
About Using the Command-Line Interface .....	90
LiteSpeed Command-Line Arguments .....	90
Syntax .....	91
Arguments .....	92
TSM-Specific Arguments .....	103
Examples .....	105
Returns .....	106

---

Fast Compression Command-Line Arguments .....	106
Syntax .....	106
Arguments .....	107
Examples .....	116
Returns .....	117
SmartCleanup Command-Line Arguments .....	117
Syntax .....	117
Arguments .....	118
Example .....	121
Returns .....	121
Recast LiteSpeed Backups .....	121
Syntax .....	121
Arguments .....	122
Examples .....	128
Returns .....	129
Convert LiteSpeed Backups to SQL Server Backups .....	129
Syntax .....	129
Arguments .....	129
Examples .....	132
Returns .....	132
Restore Objects with the Command-Line Interface .....	133
Syntax .....	133
Arguments .....	133
Examples .....	137
Returns .....	137
LicenseInfoCmd Utility .....	137
Syntax .....	137
Examples .....	138

---

<b>Use Extended Stored Procedures.....</b>	<b>139</b>
About Using Extended Stored Procedures.....	139
xp_backup_database.....	140
Syntax.....	140
Arguments.....	142
Examples.....	153
Returns.....	155
xp_backup_log.....	156
Syntax.....	156
Arguments.....	157
Example.....	168
Returns.....	169
xp_delete_tsmfile.....	169
Syntax.....	169
Arguments.....	169
Returns.....	170
xp_encrypt_backup_key.....	171
Syntax.....	171
Results.....	171
xp_encrypt_restore_key.....	171
Syntax.....	171
Results.....	171
xp_memory_size.....	171
Syntax.....	171
Results.....	172
xp_objectrecovery.....	172
Syntax.....	173
Arguments.....	174
Examples.....	177

---

Returns.....	178
xp_objectrecovery_createscript.....	178
Syntax.....	179
Arguments.....	179
Examples.....	182
Returns.....	183
xp_objectrecovery_viewcontents.....	184
Syntax.....	184
Arguments.....	184
Examples.....	186
Returns.....	187
xp_objectrecovery_executeselect.....	188
Syntax.....	188
Arguments.....	189
Examples.....	193
Returns.....	193
xp_restore_attachedfilesonly.....	194
Syntax.....	194
Arguments.....	194
Examples.....	198
Returns.....	199
xp_restore_checkpassword.....	199
Syntax.....	199
Arguments.....	199
xp_restore_checksumonly.....	200
Syntax.....	200
xp_restore_database.....	200
Syntax.....	200
Arguments.....	201



---

Examples.....	211
Returns.....	212
xp_restore_filelistonly.....	212
Syntax.....	212
Arguments.....	213
Results.....	215
Returns.....	215
xp_restore_headeronly.....	215
Syntax.....	216
Arguments.....	216
Examples.....	218
Results.....	218
Returns.....	221
xp_restore_setinfo.....	222
Syntax.....	222
Arguments.....	222
Example.....	223
Results.....	224
Returns.....	224
xp_restore_log.....	224
Syntax.....	224
Arguments.....	225
Examples.....	234
Returns.....	234
xp_restore_verifyonly.....	234
Syntax.....	234
Arguments.....	235
Example.....	241
Returns.....	241

---

xp_slsCreateDCR.....	241
Syntax.....	241
Agruments.....	242
Example.....	242
Returns.....	242
xp_slsFastCompression.....	242
Syntax.....	242
Arguments.....	244
Examples.....	253
Returns.....	254
xp_slsreadprogress.....	254
Syntax.....	254
Examples.....	255
xp_slsSmartCleanup.....	255
Syntax.....	255
Arguments.....	256
Example.....	258
Returns.....	259
xp_sqlitespeed_licenseinfo.....	259
Syntax.....	259
Arguments.....	259
Examples.....	260
Result Set.....	261
xp_sqlitespeed_version.....	261
xp_view_tsmcontents.....	261
Syntax.....	262
Arguments.....	262
Example.....	264
Result Set.....	265

---

Returns.....	267
xp_view_tsmmc.....	268
Syntax.....	268
Arguments.....	268
Result Set.....	269
Returns.....	270
<b>LiteSpeed Variables.....</b>	<b>271</b>
Accepted Variables.....	271
Examples.....	272
<b>Privilege and Grant Requirements.....</b>	<b>274</b>
<b>Repository Maintenance.....</b>	<b>276</b>
LiteSpeed_DeleteActivity.....	276
Syntax.....	276
Arguments.....	276
Examples.....	278
Upgrade Repositories.....	279
Push Statistics to Central Repository.....	279
<b>Additional Resources.....</b>	<b>281</b>
LiteSpeed Community.....	281
Video Tutorials.....	281
Useful Web Resources.....	281
<b>Review Known Issues.....</b>	<b>283</b>
<b>Configure Logging in LiteSpeed.....</b>	<b>285</b>
Installer Logging.....	285
Backup/Restore Logging.....	285
Instance-Wide LiteSpeed Logging.....	286
Log File Naming and Location.....	287
LiteSpeed UI Console Activity Logging.....	288
Reporting and Logging in Maintenance Plans.....	288

---

<b>Troubleshoot LiteSpeed Activity.....</b>	<b>290</b>
Local repository is not populated.....	290
4.x jobs not displayed.....	290
<b>Troubleshoot Maintenance Plans.....</b>	<b>292</b>
Leverage SSIS and LiteSpeed advanced options.....	292
Install Backward Compatibility components.....	292
Resolve upgrade issues.....	292
Analyze log information.....	292
<b>Troubleshoot Performance-Related Issues.....</b>	<b>293</b>
<b>Troubleshoot Previous Versions of LiteSpeed.....</b>	<b>294</b>
<b>Create Support Bundles.....</b>	<b>295</b>
<b>Appendix: Contact Quest.....</b>	<b>296</b>
About Quest Software.....	296
Contact Quest Support.....	296
Contact Quest Software.....	296
<b>Index.....</b>	<b>297</b>

# About Backing Up/Restoring with LiteSpeed

NetVault LiteSpeed for SQL Server, or LiteSpeed, is a fast and flexible backup and recovery solution that allows database administrators to easily maintain complete control over the backup and recovery process. LiteSpeed's low-impact, high-performance compression and encryption technology helps reduce storage costs and protect data, while maintaining a high level of recoverability.

Review the following information about the LiteSpeed's key features to find the best practice for backing up and restoring databases.

Use this feature...	To...
Backup Analyzer	Evaluate different backup options, such as compression level, striping, and backup destinations, to determine which settings have the best compression and duration values. See "Test Optimal Backup Settings" (page 56) for more information.
Backup Wizard	Back up individual databases. This option lets you define backup options individually for each database. You can create native SQL Server or LiteSpeed backups, generate backup scripts, run backups immediately or schedule SQL Agent jobs. LiteSpeed supports the following options for backups: <ul style="list-style-type: none"><li>• encrypt (See "Encryption" (page 56) for more information.)</li><li>• compress (See "Compression Levels" (page 54) for more information.)</li><li>• mirror</li><li>• stripe</li><li>• attach files</li><li>• log activity</li><li>• and more...</li></ul> Be sure to select <b>Use LiteSpeed</b> to leverage many LiteSpeed advanced features. See "Back Up Individual Databases" (page 38) for more information.
Multi-Database Backup Wizard	Back up several databases with the same options. See "Multi-Database Backup Wizard" (page 48) for more information.
Backup Templates	Automate backing up databases on multiple server instances by deploying a LiteSpeed Backup Template. See "Create Backup Templates" (page 31) for more information.

Use this feature...	To...
Maintenance Plans	Automate backing up databases. Maintenance Plans provide flexible backup options as well as additional database maintenance options, such as CleanUp and Rebuild Indexes tasks. See "About Automating Maintenance Tasks" (page 61) for more information.
Fast Compression	Reduce backup size and decrease backup times from hours to minutes by including differential backups in the nightly backup routine. See "Fast Compression" (page 48) for more information.
Double Click Restore	Restore Double Click Restore executable files on a server instance that does not have LiteSpeed installed. See "Double Click Restore Executables" (page 53) for more information.
Network Resilience	Control network resilience options. LiteSpeed's read and write resilience can handle various failures on both network and attached storage devices. See "Network Resilience" (page 58) for more information.
Adaptive Compression	Let LiteSpeed select optimal compression based on server performance at the time of backup. Adaptive Compression can optimize backups either for speed or for small size. See "Compression Levels" (page 54) for more information.
Restore Wizard	Restore databases and attached files. See "Restore Databases Using the Restore Wizard" (page 74) for more information.
Object Level Recovery	List, query, preview and restore specific objects directly from the native SQL Server or LiteSpeed backup files. See "Restore Objects" (page 80) for more information.
Log Shipping	Automate backing up and restoring database transaction logs on one or more standby databases. See the Configure Log Shipping guide for more information.
Integration with TSM	Back up to TSM. See the Integration with TSM guide for more information.

You can back up and restore with LiteSpeed using wizards in the LiteSpeed UI Console, extended stored procedures and the command-line interface.

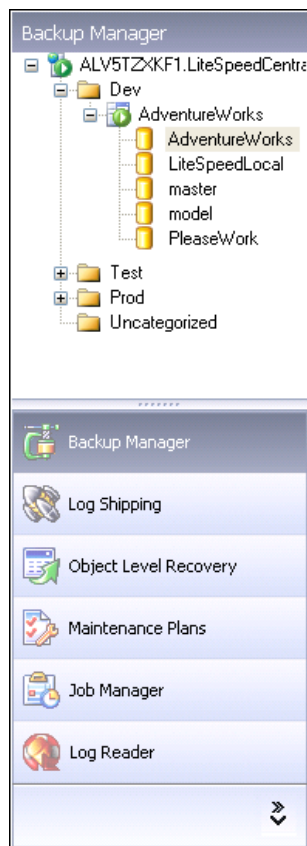
Use the LiteSpeed UI Console to view activity and history for your backups, including processes that fail or succeed, the amount of disk space you save, a list of all of the jobs for a server instance or database. See "View Activity and History in Backup Manager" (page 86) for more information.

## LiteSpeed User Interface

The LiteSpeed user interface (called the LiteSpeed UI Console) maintains the stability and accessibility of traditional LiteSpeed functionality, while implementing a user interface that integrates new LiteSpeed functions with enterprise management.

### Navigation Pane

The navigation pane has two parts: a list of LiteSpeed features at the bottom and feature-specific information at the top. Once you select a feature, the top of the navigation pane and the central pane update with the relevant information. The top of the navigation pane displays a tree of servers for Backup Manager, Log Shipping, and Maintenance Plans. It displays key tasks and options for the other features.



**Tip:** Click  to change the features that appear in the navigation.

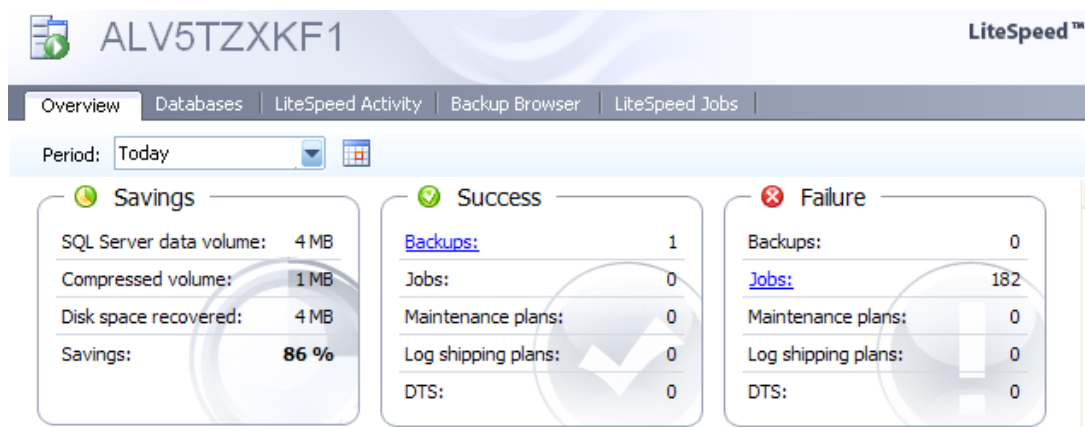
Using the Navigation pane, you can access the following features:

Feature	Description	Keyboard shortcut
Backup Manager	Manage LiteSpeed and native SQL Server backups and restores with a variety of advanced tools. You can also view detailed information about your backup and restore processes, including statistics on processes that fail or succeed, the amount of disk space you save, and a list of all of the jobs for a server instance or database.	CTRL+1
Log Shipping	Automate backing up a database (the publisher) and restoring its transaction logs on one or more standby databases (the subscribers). The process runs automatically throughout the day at the interval you specify, which creates synchronized databases	CTRL+2
Object Level Recovery	Read native SQL Server or LiteSpeed backups to view tables, query backup data and restore tables, schemas, and views. You can also generate DDL scripts for one or more databases objects. <b>Note:</b> Object level recovery is only available with the Enterprise license.	CTRL+3
Maintenance Plans	Automate routine database maintenance tasks, such as backing up databases, updating statistics, and rebuilding indexes to run on a specific day and time.	CTRL+4
Job Manager	Schedule, monitor, and manage SQL Agent jobs and Windows tasks.	CTRL+5
Log Reader	Restore data in transaction log files by rolling back any operation and reconstructing transactions. You can view recent transactions, the full database log, and all transactions in the backup file.	CTRL+6

## Central Pane



The central pane displays information based upon your selection in the navigation pane. Each feature has a unique home page and set of tabs. For a description of the Backup Manager tabs and central pane options, see View Activity and History in Backup Manager (page 86).

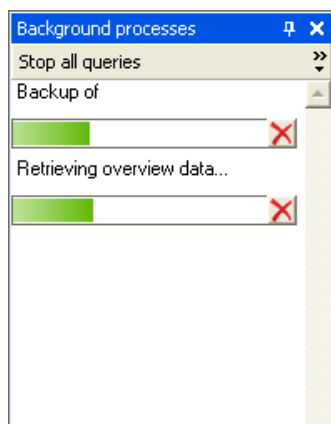




**Note:** If you selected a central repository, LiteSpeed does not display the home page for Backup Manager, Log Shipping, and Maintenance Plans.

## Background Tasks Pane


The Background Tasks pane displays processes that you selected to run in the background while you use the LiteSpeed UI Console. You can dock the pane by clicking  or cancel a process by clicking .

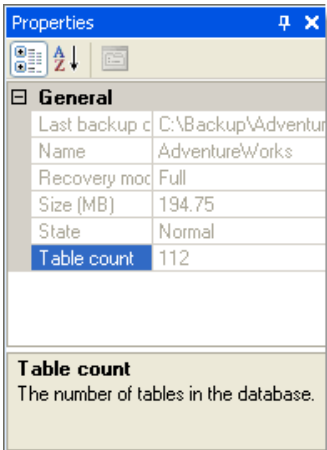


To view the pane, select **View | Background Tasks**.

**Note:** Canceling a background task does not mean that SQL Server is done rolling back the process.

## Properties Pane



The Properties pane lists properties about the item selected in the navigation pane. You can dock the pane by clicking .



To view the properties pane, select **Tools | Options** and then **Display properties in dockable window** on the General tab.

## Toolbar

The toolbar has the following parts:

Toolbar Part	Description
Navigation	Includes buttons to navigate within the LiteSpeed UI Console, including Back, Forward, Up One Level, Home, and Refresh: 
Pane-specific	Options change depending on what feature you select in the navigation pane. The following toolbar displays for the Backup Manager: 

**Tip:** To hide a toolbar, right-click and deselect it.

# Configure LiteSpeed for First Use

## Register Central Repositories

When you create or upgrade a central repository, you must register it in the LiteSpeed UI Console before you can view its activity statistics.

### *To register a repository*

1. Click ▼ beside the **Central Repository** field in the bottom of the LiteSpeed UI Console and select **Select Repository**.
2. Click **Add repository** and complete the Register New Repository dialog.
3. Click **Test connection**.

### **Tips:**

- Edit, delete, and test a repository connection by clicking the corresponding button in the Edit Repository Registration dialog. You can also import or export the repository registrations.
- You can click the **Display name**, **Authentication**, **Login**, or **Password** fields in the grid to change their value.

## Select a Central Repository

The LiteSpeed UI Console displays the activity statistics in the Backup Manager Overview tab for the selected central repository. You can register multiple central repositories in the LiteSpeed UI Console, but you can only view activity for one at a time.

### *To select a central repository*

- » Select **Tools | Options** and select the repository on the General tab.

**Tip:** You can also click ▼ beside the **Central Repository** field in the status bar of the LiteSpeed UI Console and select the repository from the list.

## Register and Group Server Instances

### Register Server Instances

You must register server instances before you can manage them in the LiteSpeed UI Console. You can also register server instances that do not have LiteSpeed installed on them and perform native backups and restores through the LiteSpeed UI Console.

**Note:** You need to register all SQL Server instances involved in log shipping in the LiteSpeed UI Console to retrieve log shipping data for them.

## Scenario

There are several server instances configured to report to a central repository and you need to register some of them in your LiteSpeed UI Console.

### *To register server instances*

1. Select **File | SQL Server Registration**.
2. Click **Add server** to add an individual server or **Import** to import server instances from a CSV file or the LiteSpeed central repository.

**Note:** You can import instances from a central repository, if the central repository is configured and selected for use.

**Scenario:** Click **Import** and then **From Central Repository**. Select the central repository to import server instances from, review the list of servers, optionally deselecting those you do not want to be registered in your LiteSpeed UI Console, click **OK** and continue with step 4.

3. (If you selected to import servers from a CSV file) Review the following for additional information:

The first line of a valid CSV file contains the following column headers:

- **DisplayName**—Specify how you want to display servers in the navigation pane tree.
- **ServerName**—Computer name or IP address following by port number: <IP\_address>, <port\_number>.
- **Authentication**—Windows or SQL Server.
- **LoginName**—User name in the following format: 'Domain\Username'.
- **Password**—May be blank.

Next lines contain server parameters, separated by semicolons (;), a separate line for a server. You can omit the LoginName and Password parameters to complete them later in the LiteSpeed UI Console.

Example:

```
DisplayName;ServerName;Authentication;LoginName;Password  
W2K3-14;W2K3-14;Windows;DOMAIN\Username;3w663k3E  
spb9884_sql_auth;spb9884;SQL Server;sa;321
```

**Note:** If you export server instances to a CSV file, it will contain SQL logins and obfuscated passwords. You need to manually edit the CSV file to remove the connection information before you import instances from this file.

4. Complete the fields as necessary. Review the following for additional information:

SQL Server	Specify server name or IP address using the following format: <IP_address>, <port_number>.
Connection parameters	Select either Windows authentication or SQL Server authentication (enter the user name and password). <b>Note:</b> Microsoft recommends using Windows authentication when possible because it is more secure than SQL Server authentication.
Display name	Enter the name to display in the navigation pane tree.

**Tips:**

- You can click the **Display name**, **Authentication**, **Login**, or **Password** fields in the grid to change their value.
- Click **Display Overview Tab** to see the Overview tab of the LiteSpeed UI Console window for the selected instance. This will let you stay on the registration dialog and continue editing or registering new server instances, but still see overview information of the various instances.

5. (Optional) In the Server tree, click and drag server instances to move them between groups.

**Tip:** In the Server tree, you can see registered SQL Server Instance nodes sorted alphabetically. To use a custom sort order, select **View | Sort Instances | Custom**.

## Scenario

There are several server instances configured to report to a central repository and you want a new colleague to manage some of them in their LiteSpeed UI Console. Define which server instances are eligible for import.

By default, LiteSpeed allows all of the instances to be imported into other clients when you select to import from a central repository, unless you specifically exclude instances.

### *To exclude a server instance from import*

1. Right-click a server in the Server tree in the navigation pane.
2. Clear the **Allow import of this instance...** checkbox and click **OK**.

## Create Server Groups

### *To create server groups*

1. Select the **Backup Manager** pane (CTRL+1).
2. Group the Backup Manager tree by server group (**View | Group Instances By | Server**

**Groups).**

3. Select one of the following options:

Create a top-level server group	Right-click <b>SQL Servers</b> at the top of the navigation tree and select <b>New SQL Server Group</b> .
Create a subgroup of an existing server group	Right-click the group in the navigation pane tree and select <b>New SQL Server Group</b> .

**Tips:**

- To create multi-level groups, create the parent group first. When you create the subgroup, the parent group will already be there for you to select.
- To edit or delete a server group, right-click the group and select the appropriate option.
- You can also group server instances by categories. To switch between server groups and categories, select **View | Group Instances By**.

## Assign Server Instances to Server Groups

You can assign server instances to server groups when you first register them, or by modifying their properties afterwards.

**Note:** You can group server instances in the navigation pane tree based on their category or server group. Categories are similar to server groups, but they offer different features. See "Change Server Instance Grouping Methods" (page 22) for more information.

*To assign a server instance to a server group*

1. Select the **Backup Manager** pane (CTRL+1).
2. Right-click the server instance and select **Edit SQL Server Registration**.
3. Click **Advanced**.
4. Select **Subgroup of** and select the parent group from the list.

## About Categorizing Server Instances

Use categories to organize server instances and databases based upon your business needs, such as function, location, division, or criticality.

**Note:** You can use categories, if the central repository is configured and selected for use. See "Select a Central Repository" (page 19) for more information.

## Change Server Instance Grouping Methods

You can group server instances in the navigation pane tree based on their category or server group. Categories are similar to server groups, but they offer different features:

- Categories can organize both server instances and databases, but server groups only organize server instances. In addition, a database does not have to be assigned to the same categories as its parent server instance.
- Databases and server instances can be assigned to more than one category. For example, if you have Location and Role categories, a database could be categorized as both North America and Development. Unlike categories, a server instance can only be assigned to one server group. In the previous example, the server instance could only be assigned to a role or location group, but not both.

#### *To change the server instance grouping method*

- » Select **View | Group Instances By**, and then select **Server Group** or **Categories**. If you select Categories, select a category to view in the **Category** field in the toolbar.

**Note:** You can use categories, if the central repository is configured and selected for use. See "Select a Central Repository" (page 19) for more information.

## Create Categories

You can create as many categories as you need to organize your server instances and databases. You can assign server instances and databases to as many or as few categories as necessary.

Categories can only be two levels deep, with top-level categories (Role or Location), and subcategories (Development/Testing or Asia/Europe).

**Note:** You can use categories, if the central repository is configured and selected for use. See "Select a Central Repository" (page 19) for more information.

#### *To create categories*

1. Select **Categories | Edit**.
2. Click **Add Category** to add a top-level category, or select the top-level category and click **Add Subcategory**.

**Tip:** To edit categories, select **Categories | Edit**. You can rename, move, and delete categories.

## Assign Server Instances and Databases to Categories

Server instances and their databases can be assigned to the same or different categories and subcategories.

**Note:** You can use categories, if the central repository is configured and selected for use. See "Select a Central Repository" (page 19) for more information.

#### *To assign a server instance or database to a category*

1. Select the Backup Manager pane (CTRL+1).
2. Right-click the server instance or database and select **Assign Categories**.

3. Select a subcategory to assign the server instance or database to.

**Note:** You can only select one subcategory per category, but you can assign server instances and databases to multiple categories.

## Configure LiteSpeed Defaults

LiteSpeed defaults specify the default values for various LiteSpeed backup parameters, such as compression level, processor affinity, max transfer size, buffer count and some other.

You do not need to specify these parameters each time you run a backup from the LiteSpeed UI Console, command-line interface or when using the extended stored procedures. LiteSpeed will use the pre-defined default values automatically, unless you supply a different value.

**Note:** LiteSpeed defaults typically result in the best performance. You should only modify advanced options after careful planning and testing.

### To set the LiteSpeed defaults

1. In the server tree, right-click the server instance and select **LiteSpeed Defaults...**
2. Select an option to change its value. Review the following additional information about the LiteSpeed defaults:

Encryption level	By default, encryption is not used. If you select to encrypt a backup using the LiteSpeed UI Console wizards, the default encryption level is <i>128-bit AES</i> .  See "Encryption" (page 56) for more information.
File Name	Location and name of a LiteSpeed backup file. LiteSpeed uses the default SQL Server backup directory. The default file name format is <i>%D_%DATETIME%.bak</i> . See "LiteSpeed Variables" (page 271) for more information.  <b>Note:</b> Fast Compression handles the naming of files automatically. See "Backup Files and Folders" (page 49) for more information.
Init Backup Set	LiteSpeed will appends the backup to an existing backup file set or tape.
Throttle	Specifies the maximum CPU usage allowed. The argument accepts an integer value between 1 and 100. The default value is 100. This is the percentage of the total amount of CPU usage (across all enabled processors) available.  <b>Tip:</b> Before you start tuning the CPU Throttle or Affinity parameters to adjust backup performance, try limiting the number of threads. If you decide to use an affinity value other than default, it is recommended that you limit the threading as well. You may also want to consider using Adaptive Compression to maintain backup




	<a href="#">performance</a> . See "Adaptive Compression" (page 55) for more information.
Priority	Select the priority of the backup over other transactions or processes running on the same server. The default is <i>Normal</i> .
Buffer Count	Number of SQL Server buffers available for a LiteSpeed operation. The default value is 20.
Max Transfer Size	Specifies the data size in bytes for each transfer when communicating with SQL Server. The size can be any multiple of 64KB in the range from 64KB to 4MB. The default is 1048576 bytes.
Processor affinity	Specifies the affinity mask for the process. The mask is a 64-bit integer value. By default, it is 0 and will utilize all CPUs. See "Processor Affinity" (page 25) for more information.
Number of Treads	Determines the number of threads used for the backup. You will achieve the best results by specifying multiple threads, but the exact value depends on several factors including: processors available, affinity setting, compression level, encryption settings, IO device speed, and SQL Server responsiveness. The default is $n-1$ threads, where $n$ is the number of processors.
Compression level	Specifies the compression level for the backup. Valid values are 0 through 8. 0 bypasses the compression routines. The remaining values of 1 through 8 specify compression with increasingly aggressive computation. 2 is the default value. See "Compression Levels" (page 54) for more information.
Path to TSM opt file	See the <i>Integration with TSM</i> Guide for more information.

**Tip:** To reset to the application default values, click **Reset Original Values**.

Where multiple SQL Server instances exist on one machine, you need to change the defaults for each instance individually.

## Processor Affinity

You can specify which processors LiteSpeed can use for the backup/restore process. They can be the same or different from the processor affinity for SQL Server.

In wizards, access the advanced options. Click  to select which processors LiteSpeed can use. The default is 0, which allows LiteSpeed to use all available system processors.

Or you can use the affinity parameter with the LiteSpeed extended stored procedures or command-line utilities. See "About Using the Command-Line Interface" (page 90) for more information.

Processor affinity designates specific processors to run LiteSpeed, while not allowing LiteSpeed to run on the remaining processors.

This argument accepts decimal values and hexadecimal values. If a value begins with "0x" it is interpreted as hexadecimal. A positive 64-bit integer value translates to a binary mask where a value of 1 designates the corresponding processor to be able to run the LiteSpeed process.

**Note:** 32-bit Windows is internally limited to a 32-bit mask.

For example, you need to select processors 2, 3, and 6 for use with LiteSpeed. Number the bits from the right to left. The rightmost bit represents the first processor. Set the second, third, and sixth bits to 1 and all other bits to 0. The result is binary 100110, which is decimal 38 or hexadecimal 0x26. Review the following for additional information:

Decimal Value	Binary Bit Mask	Allow LiteSpeed Threads on Processors
0	0	All (default)
1	1	1
3	11	1 and 2
7	111	1, 2 and 3
38	100110	2, 3, and 6
205	11001101	1, 3, 4, 7, and 8

**Tip:** Before you start tuning the CPU Throttle or Affinity parameters to adjust backup performance, try limiting the number of threads. If you decide to use an affinity value other than default, it is recommended that you limit the threading as well. You may also want to consider using Adaptive Compression to maintain backup performance. See "Adaptive Compression" (page 55) for more information.

## Configure LiteSpeed Options

### LiteSpeed General Options

Use the options pages to customize settings in LiteSpeed. There is a General options page to select basic settings, and a separate options page for each main feature in LiteSpeed.

#### *To set General options*

1. Select **Tools | Options** (Keyboard: ALT+T+O).
2. Review the following for additional information:

Use this option...	To...	Default
Select central repository	Select a central repository to view statistics in the LiteSpeed UI Console. You must register the repository before you can select it. See "Register Central Repositories" (page 19) for more information.	Not Selected
Display properties in dockable window	View properties about the item selected in the navigation pane in a dockable window.	Cleared
Show confirmation message	View confirmation message when exiting the LiteSpeed UI Console or wizards.	Cleared
Log the LiteSpeed UI Console activity	Select a logging level to define what events to log for the console.	Errors only
Show database status	<p>View if a database has been backed up recently using LiteSpeed</p> <p>The instance tree view displays databases icons of the following colors:</p> <ul style="list-style-type: none"> <li>• GREEN—If the latest database backup was successfully created within the number of days specified for GREEN.</li> <li>• YELLOW—If the latest database backup was successfully created within the number of days specified for YELLOW and NOT in the number of days specified for GREEN.</li> <li>• RED—If the latest database backup was NOT successfully created within the number of days specified for YELLOW.</li> </ul> <p><b>Note:</b> Database status can only be displayed for LiteSpeed backups and if the local repository is configured.</p>	Cleared

## Backup Manager Options

### *To set Backup Manager options*

1. Select **Tools | Options** (*Keyboard: ALT+T+O*).
2. Select the Backup Manager tab. Review the following for additional information:

Use this option...	To...	Default
View content timeout	Enter the connection timeout for when you view the backup file content.	30 seconds
TSM device retry time	Select the amount of time to wait for the device to become available, or not to retry.	No retries
Show TimeLine on LiteSpeed Activity tab	View database backups in a timeline at the server and database level.	Selected

## Log Shipping Options

### *To set Log Shipping options*

1. Select **Tools | Options** (*Keyboard: ALT+T+O*).
2. Select the Log Shipping tab. Review the following for additional information:

Use this option...	To...	Default
Read subscriber information	View subscriber information on the Monitoring tab.	Selected
Display detailed status information on the Monitoring tab	Show the details panel on the Monitoring tab. The details panel displays tips if there is something wrong with the log shipping plan.	Selected
Load log shipping plan statuses...	View current statuses of all log shipping plans when connecting to the server instance and when refreshing the Log shipping plans tab at the instance level.  If this option is off, plan status is only loaded when you double-click a plan.	Cleared

## Job Manager Options

### *To set Job Manager options*

1. Select **Tools | Options**.
2. Select the Job Manager tab. Review the following for additional information:

Use this option...	To...	Default
Scheduler Page	Set the following options.	—
Show server time	Display the local time and the time difference between the desktop and the server at the top of the Calendar tab for each server.	Selected
Show recurring executions as one item	Simplify the display by showing jobs that run multiple times as one job.	Selected
Task List Page	Set the following options.	—
Show steps	Display each job step on the Job List tab in an expanding row in the list or in a separate pane at the bottom of the window.	Under each row of a job
Show history	Display job history on the Job List tab in a separate pane at the bottom of the window. On Demand—Display history only after clicking <b>Show History</b> . (Selected.) If cleared, history displays automatically when you select a job.	Cleared
Advanced	Click <b>Advanced</b> to set the following options.	—
Maximum number of threads	Specify the maximum number of total concurrent threads to use on target instances to run the jobs.	3
Maximum appointments	Specify the maximum number of appointments that can display on the Calendar tab. If the number of appointments exceeds this value, appointments may display incorrectly.	25

**Tip:** Click **Advanced | Known Applications** to add new or change existing masks and icons. A mask is a pattern used to group jobs or tasks with similar names and display the appropriate icon for them. Masks are stored locally.

## Log Reader Options

### *To set the Log Reader options*

1. Select **Tools | Options** from the menu.
2. Select the Log Reader tab.
3. Review the following for additional information:

Use this option...	To...	Default
Reconstruct DDL commands	Select this option to display both DML and DDL commands	Selected
Request full database backup file if needed	When reading an online or offline log, the Log Reader might need to refer to the full-backup file for the database to reconstruct old log records. Select this option to enable the use of the full-backup file.	Selected
Search Depth	Select whether to read the entire log file or only a set number of records in the log file. <b>Note:</b> The minimum number of records to read is 100,000.	Read approximately <i>N</i> records  100000

## Back Up Databases

### LiteSpeed Backup Templates

A LiteSpeed backup template contains a set of backup parameters that describe the types of databases and the types of backups you want to perform. Using a backup template you define Full backup or Fast Compression backup schemes for your environment, with or without transaction log backups, and define the LiteSpeed compression and encryption options to use for the backup. When you deploy a template to a server instance, this creates a backup job or a maintenance plan that uses the parameters you specified in the template.

**Note:** This feature is only supported for disk LiteSpeed backups.

Backup Templates can be easily updated and changes deployed to all instances. LiteSpeed tracks the instances where each template is deployed, making re-deployment very easy. Backup Templates can also be easily removed from an instance if they are no longer required.

Using LiteSpeed backup templates in your backup routine can help you manage multiple SQL Server instances. You do not have to manage backup jobs for each database or manage maintenance plans for one server at a time. Instead, create and deploy a backup template.

If your company's policy changes, you can quickly edit the template to comply with the new standards and re-deploy. LiteSpeed versions the templates and the deployments, so it is easy to see if an instance needs to be updated.

### Create Backup Templates

#### *To create a Backup Template*

1. Select the **Backup Manager** pane (CTRL+1). Do one of the following:
  - (If the central repository is selected for use) In the Backup Templates tab, click **New** on the toolbar.
  - (If you do not use the central repository) Click **Backup Templates** on the toolbar.
2. Review the following additional information about the backup type and destination:

Backup Type	<p>Select what backups will occur after you deploy this template on a server instance.</p> <ul style="list-style-type: none"><li>• Full—Full backups and optionally transaction log backups.</li><li>• Fast Compression—Full and Differential and optionally transaction log backups. Fast Compression automatically decides when to issue a Full or Differential backup based on</li></ul>
-------------	---

	<p>the amount of database changes and some other conditions. See "Fast Compression" (page 48) for more information.</p> <p>The default backup name and description use the following information:</p> <ul style="list-style-type: none"> <li>• %D—Database name</li> <li>• %T—Backup type (Full, Diff or Log)</li> <li>• %z—Timestamp</li> </ul> <p>You can specify custom backup name and description using both the LiteSpeed variables and text. See "LiteSpeed Variables" (page 271) for more information.</p>
Select Databases	<p>Select databases you want to back up.</p> <p><b>Notes:</b></p> <ul style="list-style-type: none"> <li>• You can select the <b>Custom Database Selection at Deployment</b> option to specify individual databases for each instance at the deployment time. When you redeploy this template, you can see if a database was included in the previous deployment and change the database selection as needed. See "Deploy Backup Templates" (page 37) for more information.</li> <li>• You can use regular expressions or wildcard characters '?' and '*' to specify databases. If you use the Regular Expression option, then you are limited to deploying as a Maintenance Plan. Wildcards are supported by both Maintenance Plans and Jobs.</li> <li>• You can select whether to backup up databases participating in AlwaysOn Availability Groups. See "Backing Up SQL Server 2012 AlwaysOn Availability Groups" (page 47) for more information.</li> </ul>
Backup Destinations	<p>Decide whether you want to save a default backup destination name in the template or specify it at the deployment time. If you specify a backup destination in a template, you can override backup destinations for each instance at deployment time. Each Backup Template remembers if you override these settings to make future redeployments easier.</p> <p>Backup file name as well as the backup folder can include the %SERVER% variable, that may be particularly useful when backing up many instances to the same network share.</p> <p><b>Note:</b> Fast Compression handles the naming of files automatically. See "Backup Files and Folders" (page 49) for more information.</p>



3. Set Backup options. Review the following for additional information:

Optimize the Object Level Recovery speed	<p>Select to create an index of objects in the backup file. This option is only available for LiteSpeed backups.</p> <p><b>Note:</b> Before you can recover objects or execute a <b>SELECT</b> statement, you must read the backup file to create an index of restorable objects. If you select this option, LiteSpeed uses the index in the backup file to read the backup file, which makes the object level recovery process much faster.</p>
Create a Double Click Restore executable	<p>Select to create an executable backup file or a Double Click Restore Loader that allows you to restore a backup on a server instance that does not have LiteSpeed installed. See "Double Click Restore Executables" (page 53) for more information.</p> <p><b>Note:</b> A Double Click Restore can only be created for a disk file.</p>
Perform checksum before writing to media	<p>Select to verify checksums when a backup is created.</p> <p>Additionally, you can control the response to an error. If you select the <b>Continue on error</b> option, the backup is executed despite encountering an invalid backup checksum.</p>

Review the following if you selected the **Fast Compression** backup type.

Fast Compression Type	<p>Select whether you prefer to create a unique file for each backup or you prefer to manage a single file for each backup set (a backup set is composed of one full database backup plus all associated differential backups):</p> <ul style="list-style-type: none"> <li>• Self-Contained Backup Sets—Provides the convenience of only having to manage a single file per backup set. Only one file needs to be saved to or pulled from tape or copied from the backup location to a secondary location. If backing up more than one database, a file for each database will be created.</li> <li>• Separate Backup Files—Creates a unique file for each backup in the backup set. This option provides the convenience of having to move less data to tape or across the network when copying individual backup files. Using this option means that up to two physical files may be needed to restore the database (full backup plus the associated differential for the day in question).</li> </ul> <p>The Self-Contained Backup Sets option automatically verifies the Full backup exists. The Separate Backup Files option performs the same validation by default, but Fast Compression includes an option to not perform this check.</p>
-----------------------	---


Fast Compression Backup Options	<p>You can set the following thresholds to define when to issue a full backup:</p> <ul style="list-style-type: none"> <li>• The amount of time elapsed since the last full backup</li> <li>• The amount of database changes since the last full backup</li> </ul> <p>Fast Compression measures the amount of data change by either querying SQL Server or by comparing the size of the last differential to the last full backup. The option to Query Actual Data Pages provides the most accurate way to determine the amount of data change. If the query fails for any reason, Fast Compression will automatically run a size comparison to the last Differential backup.</p> <p>For example, set this parameter to 20%, and should the database change by 20% or more, Fast Compression will automatically run a Full backup. The larger the threshold, the larger the differential backups can grow before Fast Compression triggers the next Full backup.</p> <p>Regardless of how much underlying database data has changed, when exceeding the maximum interval (in days) between full backups, Fast Compression will force a full backup.</p> <p><b>Notes:</b></p> <ul style="list-style-type: none"> <li>• Before a differential Fast Compression backup is available, the last full backup must have been created in the Fast Compression backup or alternate backup folder.</li> <li>• When backing up the master database as part of a Fast Compression maintenance plan or job, Fast Compression always executes a full backup.</li> <li>• The copy-only full backups cannot serve as a base for differential backups.</li> </ul>
Backup Escalation	<p>This option causes LiteSpeed to issue a full backup, if one of the following problems is discovered in the current backup set:</p> <ul style="list-style-type: none"> <li>• The full backup is missing.</li> <li>• A differential backup is missing from the backup set (excludes backups automatically removed after the specified retention period).</li> <li>• LSN verification fails in the backup set.</li> <li>• Verify operation fails on full or differential backup.</li> </ul> <p><b>Note:</b> If a problem is detected and a full backup is created through escalation, an error will be returned.</p>
Verification	<p>Make sure the backup files in the backup set have integrity. This provides an added level of insurance the backup files can be restored. Verification failures appear in the LiteSpeed UI Console and, optionally, as job failure notifications. A verification failure after a</p>

	differential backup will trigger the backup escalation process, if selected.
--	--

4. Set Compression and Encryption options. Note that you can specify both the compression level and Adaptive Compression option. LiteSpeed will select and use Adaptive Compression if it is supported by the target server LiteSpeed version. Otherwise, the specified compression level will be used.

Adaptive Compression	<p>LiteSpeed will automatically select the optimal compression based on throughput and CPU usage and optimize backups either for size or for speed (default).</p> <p><b>Note:</b> Adaptive Compression is only available with LiteSpeed 6.5 or later; Enterprise license.</p>
Compression level	<p>Select 0 for no compression or 1-8 to compress the file. See "Compression Levels" (page 54) for more information.</p> <p><b>Note:</b> Higher compression levels result in smaller backup files, but they also take longer to complete. For assistance determining the best compression options, use the Backup Analyzer. See "Test Optimal Backup Settings" (page 56) for more information.</p>
Encrypt backup	<p>Select this checkbox to encrypt the backup. Then, select the encryption level and enter the encryption password. See "Encryption" (page 56) for more information.</p>

Review the following additional information about the advanced options:

Throttle	Enter the maximum percentage of CPU that LiteSpeed can use for the process. The default is <i>100</i> .
Priority	Select the priority of the backup over other transactions or processes running on the same server. The default is <i>Normal</i> .
Buffer count	Enter the number of SQL Server buffers available for a LiteSpeed operation. The default is <i>20</i> .
Max transfer size	Enter the maximum backup file size in bytes. The default is <i>1048576</i> .
Processor affinity	<p>Click  to select which processors LiteSpeed can use. The default is <i>0</i>, which allows LiteSpeed to use all available system processors.</p> <p>See "Processor Affinity" (page 25) for more information.</p>
Number of threads	Enter the number of threads used for the backup. The default is <i>n-1</i> threads, where <i>n</i> is the number of processors.

Logging level	<p>Select one of the following options:</p> <ul style="list-style-type: none"> <li>• None—LiteSpeed does not log backup/restore activity.</li> <li>• Verbose—LiteSpeed logs all backup/restore activity.</li> <li>• Verbose on failure—LiteSpeed logs all activity and only saves log files if the backup or restore operation fails. If it succeeds, LiteSpeed does not save the log.</li> </ul>
Network Resilience	<p>If LiteSpeed fails to write a disk backup or read from disk, it will wait and retry the operation. You can control the number of times to retry and the amount of time to wait before retrying.</p> <p>The default is <i>4 retries</i>, the maximum allowed setting is 1000. The default period to wait before retry is <i>15 seconds</i>, the maximum allowed setting is 300 seconds.</p> <p>See "Network Resilience" (page 58) for more information.</p>
<p><b>Tips:</b></p> <ul style="list-style-type: none"> <li>• Before you start tuning the CPU Throttle or Affinity parameters to adjust backup performance, try limiting the number of threads. If you decide to use an affinity value other than default, it is recommended that you limit the threading as well.</li> <li>• You may also want to consider using Adaptive Compression to maintain backup performance. See "Adaptive Compression" (page 55) for more information.</li> </ul>	

**Note:** LiteSpeed defaults typically result in the best performance. You should only modify advanced options after careful planning and testing. See "Configure LiteSpeed Defaults" (page 24) for more information.

5. Specify the cleanup options. See "Smart Cleanup" (page 58) for more information.
6. Complete the wizard.

Backup Templates are saved in the central repository (if one is used) and available for editing and deployment on the Backup Templates tab in the **Backup Manager** pane (CTRL+1). If the central repository is not configured, LiteSpeed will save each template as file.

In the Backup Templates tab, you can create, edit, clone, import, export and deploy templates, view the template contents, deployment details and modification history.

**Notes:**

- The Backup Templates tab is only available, if the central repository is configured and selected for use. To edit, deploy or remove a template when the central repository is not used, click ▾ beside **Backup Templates** on the toolbar and select the appropriate option.
- The template deployment history is not exported when you export a template.

## Deploy Backup Templates

### Notes:

- A LiteSpeed backup template can only be deployed on a server that has LiteSpeed installed.
- If the target server LiteSpeed version does not support Adaptive Compression, the created jobs and maintenance plans will instead use the specified compression level.
- Backup Templates can only be deployed as jobs on SQL Server 2000, because SSIS maintenance plans are not supported in this version of SQL Server.
- Regular expressions are only supported in maintenance plans.

### To deploy a LiteSpeed Backup Template

1. Select the Backup Manager pane (CTRL+1).
2. Do one of the following:
  - (If you saved the template in the LiteSpeed central repository) In the Backup Templates tab, click **Deploy** on the toolbar.
  - (If you saved the template to file and the central repository is not selected for use) Click ▾ beside **Backup Templates** on the toolbar, select the **Deploy** option and double-click the template file you want to deploy.
3. Complete the wizard. Review the following for additional information:

Select Instances	<p>Use the <b>Show selected instances only</b> option to see server instances and databases where the selected template was deployed.</p> <p><b>Note:</b> You can select individual databases only if the backup template is saved with the <b>Custom Database Selection at Deployment</b> option.</p>
Backup Options	<p>Enter backup destinations for individual instances or for several instances at a time or use the default backup destination you specified in the template.</p> <p>You can change the scheduled time manually or you can right-click a cell in the Scheduled Time column and select one of the following options:</p> <ul style="list-style-type: none"><li>• Default to template—To change the current cell back to the template default.</li><li>• Default all to template—To change all cells back to the template default.</li><li>• Stagger by n minutes—To stagger all scheduled times by n minutes, starting at the template time and adding n minutes for an instance.</li><li>• Spread over n hour(s)—To spread the scheduled times evenly over an n hour range, starting with the template time.</li></ul>

**Note:** You can only change the time for the instance.

Each Backup Template remembers if you override these settings to make future redeployments easier. For example, if you specify a different scheduled time for an instance, it will retain the job/plan start time for the next deployments and will not change if you edit this setting in the template.

For each of the selected instances, the Deployment wizard will create either a SQL Agent job or a maintenance plan. Maintenance Plans names and subplans jobs names are prefixed with the template name. SQL Agent job names have the following format: *LiteSpeed Backup Template <template\_name> (version n)*.

**Tip:** To remove a deployed template, run the Deployment wizard, select the **Remove deployed template** option on the Select Template page and complete the wizard.

## Back Up Individual Databases

The Backup Manager wizards guide you through the process of backing up a database. You can perform file, filegroup, or transaction log backups, and you can also mirror, stripe, compress, encrypt, and add attachments to the backup file.

### Notes:

- You can only back up one database at a time through the Backup wizard. To back up multiple databases at the same time, run the Multi-Database Backup wizard. See "Multi-Database Backup Wizard" (page 48) for more information.  
  
Alternately, you can create a maintenance plan. See "About Creating Maintenance Plans" (page 62) for more information.
- If you selected **Fast Compression Backup**, the wizard creates a Fast Compression job and does not create backup files immediately.
- When using this wizard to back up databases participating in AlwaysOn availability groups, LiteSpeed does not check whether the replica is preferred for backups. You can automate backing up of preferred replicas using the LiteSpeed Backup Templates feature. See "Create Backup Templates" (page 31) for more information.
- Fast Compression is not supported on secondary replicas in AlwaysOn Availability Groups. Backups must be performed on the primary because differential backups are not supported on secondaries.
- Performing full, file and filegroup backups on secondary replicas in an AlwaysOn Availability Group will always produce copy-only backups. Please refer to the SQL Server 2012 product documentation for information about supported backup types.

For information about full file backups, see <http://msdn.microsoft.com/en-us/library/ms189860.aspx>.

## Scenario

You need to perform nightly disk backups for a 500 GB database that changes 10% daily.

### *To back up a database using the Backup wizard*

1. Select the **Backup Manager** pane (CTRL+1).
2. Right-click the database and select **Backup**.
3. Review the following additional information about the Select Backup Type page:

Backup type	<p>Select one of the following backup types:</p> <ul style="list-style-type: none"><li>• Regular (Full, Differential or Transaction Log).</li><li>• Fast Compression backup. Using this option you can schedule Full and Differential and optionally Transaction log backups. Fast Compression automatically decides when to issue a Full or Differential backup based on the amount of database changes and some other conditions. See "Fast Compression" (page 48) for more information.</li></ul> <p><b>Scenario:</b> Select the Fast Compression backup type.</p>
Backup component	<p>Select <b>Database</b> or <b>Files and filegroups</b>.</p> <p><b>Note:</b> This field is only available for databases with a full recovery model.</p>
Use LiteSpeed	<p>Select to create a LiteSpeed backup. If you clear this checkbox, LiteSpeed creates a native backup script.</p>

**Tip:** You can specify custom backup name and description using variables. See "LiteSpeed Variables" (page 271) for more information.

4. Add the backup destination in the Select Backup Destination page.

**Scenario:** Select **Disk** and enter the existing or new location for Fast Compression backups.

Review the following for additional information:

Add	Select this option to add multiple backup destinations.
Overwrite	(For Full, Differential or Transaction Log backups) Select one of the following options: <ul style="list-style-type: none"> <li>• Append to media</li> <li>• Overwrite existing media</li> <li>• Create unique media - this option will generate a unique file name for a backup</li> </ul>
Add Mirror	Click <b>Add Mirrors</b> to copy the entire backup file to multiple locations.

**Notes:**

- Mirroring is not the same as striping, which divides the backup file and stores the pieces at different destinations.
- LiteSpeed automatically stripes the backup files if you include more than one backup destination.
- You can mirror backups regardless of what SQL Server version you use, but you cannot mirror TSM or tape backups.

If you selected **TSM Backup** or **TSM Archive**, review the following for additional information:

Client node	Enter the node name for the TSM session. This field is not case-sensitive.
Client owner password	Enter the access password for the specified node.
Configuration file	Select the configuration file. (Usually, dsm.opt.) <b>Note:</b> This file contains session options such as the TSM server's TCP address. If you select the <b>Use PASSWORDACCESS GENERATE from TSM configuration file</b> checkbox and your options file is configured to support this option, you do not need to specify the client node and client owner password.
Management class	Select the management class (policy) to associate with the backup object being created. LiteSpeed will use the default management class, if this option is not selected.
TSM Filespaces (Fast Compression)	Click <b>TSM Filespaces</b> and select the existing or enter new file space name(s). This step is optional, if you do not specify the file space name, Fast Compression will automatically create one. <b>Note:</b> Fast Compression handles the naming of files automatically.



backups)	See "Backup Files and Folders" (page 49) for more information.
TSM Object (Regular backups)	<p>Click <b>Select TSM Object</b>.</p> <p>Enter the filespace and the high-level and low-level names and click <b>Query TSM</b> to pick the object name from the list of available TSM objects. From the Available TSM Objects list, double-click the objects you would like to select.</p> <p><b>Note:</b> If you leave the High level and Low level fields blank, LiteSpeed will query all TSM server levels. Querying all levels may take longer to complete.</p> <p>For a new object, you can manually enter the full three-part name.</p>
Check for existing objects with same name (Regular backups)	Select this option to check for objects with the same name. LiteSpeed aborts the backup if it finds one.

5. If you selected the **Fast Compression** backup type, review the following:

Fast Compression Type (Disk backups only)	<p>Select whether you prefer to create a unique file for each backup or you prefer to manage a single file for each backup set (a backup set is composed of one full database backup plus all associated differential backups):</p> <ul style="list-style-type: none"> <li>• <b>Self-Contained Backup Sets</b>—Provides the convenience of only having to manage a single file per backup set. Only one file needs to be saved to or pulled from tape or copied from the backup location to a secondary location. If backing up more than one database, a file for each database will be created.</li> <li>• <b>Separate Backup Files</b>—Creates a unique file for each backup in the backup set. This option provides the convenience of having to move less data to tape or across the network when copying individual backup files. Using this option means that up to two physical files may be needed to restore the database (full backup plus the associated differential for the day in question).</li> </ul> <p>The Self-Contained Backup Sets option automatically verifies the Full backup exists. The Separate Backup Files option performs the same validation by default, but Fast Compression includes an option to not perform this check.</p> <p><b>Scenario:</b> Go with the <b>Self-Contained Backup Sets</b> option.</p>
Options	You can set the following thresholds to define when to issue a full backup:

	<ul style="list-style-type: none"> <li>• The amount of time elapsed since the last full backup</li> <li>• The amount of database changes since the last full backup</li> </ul> <p>Fast Compression measures the amount of data change by either querying SQL Server or by comparing the size of the last differential to the last full backup. The option to Query Actual Data Pages provides the most accurate way to determine the amount of data change. If the query fails for any reason, Fast Compression will automatically run a size comparison to the last Differential backup. For example, set this parameter to 20%, and should the database change by 20% or more, Fast Compression will automatically run a Full backup. The larger the threshold, the larger the differential backups can grow before Fast Compression triggers the next Full backup.</p> <p>Regardless of how much underlying database data has changed, when exceeding the maximum interval (in days) between full backups, Fast Compression will force a full backup.</p> <p><b>Notes:</b></p> <ul style="list-style-type: none"> <li>• Before a differential Fast Compression backup is available, the last full backup must have been created in the Fast Compression backup or alternate backup folder.</li> <li>• When backing up the master database as part of a Fast Compression maintenance plan or job, Fast Compression always executes a full backup.</li> <li>• The copy-only full backups cannot serve as a base for differential backups.</li> </ul> <p>The Backup Escalation option causes LiteSpeed to issue a full backup, if one of the following problems is discovered in the current backup set:</p> <ul style="list-style-type: none"> <li>• The full backup is missing.</li> <li>• A differential backup is missing from the backup set (excludes backups automatically removed after the specified retention period).</li> <li>• LSN verification fails in the backup set.</li> <li>• Verify operation fails on full or differential backup.</li> </ul> <p><b>Note:</b> If a problem is detected and a full backup is created through escalation, an error will be returned.</p> <p><b>Scenario:</b> Go with the Fast Compression defaults.</p>
Verification / Cleanup (Disk backups only)	<p>Make sure the backup files in the backup set have integrity. This provides an added level of insurance the backup files can be restored. Verification failures appear in the LiteSpeed UI Console and, optionally, as job failure notifications. A verification failure after a</p>

	<p>differential backup will trigger the backup escalation process, if selected.</p> <p><b>Scenario:</b> Make sure the <b>Enable backup escalation</b> option is selected. Select the <b>Enabled backup cleanup</b> option and specify when full/differential backups are eligible for cleanup, according to your company's retention policy. Clear the <b>Clean up transaction logs older than</b> option, since these backups are not configured in this scenario.</p>
--	---

6. Review the following additional information about the backup options:

Optimize the Object Level Recovery speed	<p>Select to create an index of objects in the backup file. This option is only available for LiteSpeed backups.</p> <p><b>Note:</b> Before you can recover objects or execute a SELECT statement, you must read the backup file to create an index of restorable objects. If you select this option, LiteSpeed uses the index in the backup file to read the backup file, which makes the object level recovery process much faster.</p>
Create a Double Click Restore executable	<p>Select to create an executable backup file or a Double Click Restore Loader that allows you to restore a backup on a server instance that does not have LiteSpeed installed. See "Double Click Restore Executables" (page 53) for more information.</p> <p><b>Note:</b> A Double Click Restore can only be created for a disk file.</p>
Perform checksum before writing to media	<p>Select to verify checksums when a backup is created.</p> <p>Additionally, you can control the response to an error. If you select the <b>Continue on error</b> option, the backup is executed despite encountering an invalid backup checksum.</p>


**Note:** Some options are only available with certain backup types.

7. Review the following additional information about the Compression page:

Adaptive Compression	<p>LiteSpeed will automatically select the optimal compression based on throughput and CPU usage and optimize backups either for size or for speed (default).</p> <p><b>Note:</b> Adaptive Compression is only available with LiteSpeed 6.5 or later; Enterprise license.</p>
----------------------	---

Compression level	Select 0 for no compression or 1-8 to compress the file. See "Compression Levels" (page 54) for more information.  <b>Note:</b> Higher compression levels result in smaller backup files, but they also take longer to complete. For assistance determining the best compression options, use the Backup Analyzer. See "Test Optimal Backup Settings" (page 56) for more information.
Encrypt backup	Select this checkbox to encrypt the backup. Then, select the encryption level and enter the encryption password. See "Encryption" (page 56) for more information.

Review the following additional information about the advanced options:

Throttle	Enter the maximum percentage of CPU that LiteSpeed can use for the process. The default is <i>100</i> .
Priority	Select the priority of the backup over other transactions or processes running on the same server. The default is <i>Normal</i> .
Buffer count	Enter the number of SQL Server buffers available for a LiteSpeed operation. The default is <i>20</i> .
Max transfer size	Enter the maximum backup file size in bytes. The default is <i>1048576</i> .
Processor affinity	Click  to select which processors LiteSpeed can use. The default is <i>0</i> , which allows LiteSpeed to use all available system processors. See "Processor Affinity" (page 25) for more information.
Number of threads	Enter the number of threads used for the backup. The default is <i>n-1</i> threads, where <i>n</i> is the number of processors.
Logging level	Select one of the following options: <ul style="list-style-type: none"> <li>None—LiteSpeed does not log backup/restore activity.</li> <li>Verbose—LiteSpeed logs all backup/restore activity.</li> <li>Verbose on failure—LiteSpeed logs all activity and only saves log files if the backup or restore operation fails. If it succeeds, LiteSpeed does not save the log.</li> </ul>
Network Resilience	If LiteSpeed fails to write a disk backup or read from disk, it will wait and retry the operation. You can control the number of times to retry and the amount of time to wait before retrying.  The default is <i>4 retries</i> , the maximum allowed setting is <i>1000</i> . The default period to wait before retry is <i>15 seconds</i> , the

	maximum allowed setting is 300 seconds. See "Network Resilience" (page 58) for more information.
<b>Tips:</b> <ul style="list-style-type: none"><li>• Before you start tuning the CPU Throttle or Affinity parameters to adjust backup performance, try limiting the number of threads. If you decide to use an affinity value other than default, it is recommended that you limit the threading as well.</li><li>• You may also want to consider using Adaptive Compression to maintain backup performance. See "Adaptive Compression" (page 55) for more information.</li></ul>	

**Note:** LiteSpeed defaults typically result in the best performance. You should only modify advanced options after careful planning and testing. See "Configure LiteSpeed Defaults" (page 24) for more information.

8. Select **Attach files and directories** on the Attached Files page to store selected files with the backup.

This feature lets you bundle items with the backup file that may be necessary to restore it. LiteSpeed appends the files to the data section of the backup file, and consequently the attached files are encrypted and compressed with the backup if those options are selected.

Some examples of when to use this feature include the following:

- Extended stored procedures packaged in DLLs may exist outside of the database, but you may need them to restore it. These external DLLs can be attached to the backup file.
- If you are going to upgrade a database application, you can back up the database along with the current version of the application. If anything goes wrong during the upgrade, you can restore both the database and the related application files.
- When sending a backup file offsite to another person (such as an auditor or consultant), you can attach schema diagrams, username and password lists, instructions, and other documentation.

To retrieve the attachment, select **Restore Attached Files and Directories** on the Attach Files page of the Restore wizard. You can also use `xp_restore_attachedfilesonly`. See "xp\_restore\_attachedfilesonly" (page 194) for more information.

**Tip:** To restore attached files only using the Restore Wizard, right-click a database in the server tree and select **Restore | Attached Files...**

9. Complete the wizard.

**Scenario:** Complete the wizard, using the default options.

**Note:** If you selected to run the backup in the background, you can view the progress in the Background Tasks pane (**View | Background Tasks**). If you scheduled the backup, you can view and edit it from the LiteSpeed Jobs tab for the database.

## Back Up Multiple Databases

You can backup multiple databases using:

- Backup Templates. See "Create Backup Templates" (page 31) for more information.
- Maintenance Plans. See "About Creating Maintenance Plans" (page 62) for more information.
- Multi-Database Backup Wizard. See "Multi-Database Backup Wizard" (page 48) for more information.

## LiteSpeed's Logic for Backing Up Multiple Databases

LiteSpeed provides you with many flexible options for selecting databases to include in a backup job or in a backup task of a maintenance plan. If there are conflicting conditions (for example, you selected to back up databases, matching the 'DirectD\*' pattern and skip read-only databases, while the name of a read-only database matches the pattern), the Exclude condition wins over Include.

**Note:** LiteSpeed wizards may have different Exclude/Include options for selecting databases.

## Include Databases

You can specify databases:

- Manually
- Using one of the following built-in groups:
  - All databases
  - System databases
  - User databases
- Using regular expressions or wildcard characters '?' and '\*'.

For example, "abc\*" will include all databases that start with "abc". LiteSpeed lets you enter multiple patterns and see a list of all databases names that match any of the specified patterns. This is an advanced feature for users who are familiar with wildcard or regular expressions. For information about regular expressions, see [msdn.microsoft.com/en-us/library/az24scfc.aspx](https://msdn.microsoft.com/en-us/library/az24scfc.aspx).

**Note:** New user databases will also be backed up as part of the "All databases" or "User databases" groups or if they match the specified pattern, unless they are excluded by an Exclude option. See "Exclude Databases" (page 47) for more information.

You cannot mix selection methods. Instead, you can schedule several subplans or jobs using different selection methods.

## Exclude Databases

Using these options you can further define databases you want to back up.

You can exclude the following databases:

- Offline
- Selected—LiteSpeed will backup all existing and new databases, except those you selected manually and those excluded by a different Exclude option.
- Deleted
- Log Shipping
- Read Only

**Note:** Mirror databases are automatically excluded.

## Backing Up SQL Server 2012 AlwaysOn Availability Groups

Versions 7.0 and later of LiteSpeed provide enhanced support for databases participating in AlwaysOn availability groups. You can include availability databases in a maintenance plan or a backup job, as well as perform backups on individual replicas.

In maintenance plans and backup templates, the following options define which replicas are eligible for backup:

- Back up preferred—This is the default option. LiteSpeed will follow the logic defined for the availability group by the database administrator and back up databases accordingly, provided the jobs or maintenance plans are configured on every server instance that hosts an availability replica. The default AlwaysOn configuration is to back up a secondary replica only, unless the primary replica is the only replica online.
- Only back up the primary—The primary is backed up each time, regardless of AlwaysOn configuration.
- Back up all (primary and secondaries)—All primaries and secondaries are backed up each time, regardless of AlwaysOn configuration.
- Back up none—AlwaysOn databases are ignored during backup.

### Notes:

- Fast Compression is not supported on secondary replicas in AlwaysOn Availability Groups. Backups must be performed on the primary because differential backups are not supported on secondaries.
- Performing full, file and filegroup backups on secondary replicas in an AlwaysOn Availability Group will always produce copy-only backups. Please refer to the SQL Server 2012 product documentation for information about supported backup types.

- In Maintenance Plans and in the Multi-Database Backup Wizard, if you select the **Differential** backup type, all secondary replicas in an AlwaysOn Availability Group are automatically excluded.

## Multi-Database Backup Wizard

The Multi-Database Backup wizard allows you to configure backup options for several databases at once.

For each database, LiteSpeed generates uniquely named backup files to avoid collision.

**Note:** Fast Compression is not supported.

### *To back up multiple databases*

1. Right-click the server instance and select **Tasks | Multi-Database Backup...**
2. Select databases for backup. See "LiteSpeed's Logic for Backing Up Multiple Databases" (page 46) for more information.
3. Complete the wizard. See "To back up a database using the Backup wizard" (page 39) for more information on backup options.

In case you select to schedule the backups to run at the specified times, the wizard creates a job with 'Multiple Databases' appended to the job name.

## Fast Compression

Fast Compression technology allows you to maximize space savings and reduce backup time considerably over nightly full backup routines by intelligently backing up only database changes rather than the entire database. Fast Compression automatically chooses to create a full or differential backup based on user-defined full backup intervals and the percentage of the database that has changed. By only backing up database changes, users will see a significant reduction in backup storage and backup time.

### **Notes:**

- Fast Compression is not available for Tape and TSM Archive.
- Fast Compression is only available with versions 5.1 and above of LiteSpeed; Enterprise license.

By default, Fast Compression selects the differential backup after first full, if the amount of changes is less than 35%.



The following table shows how much disk space can be saved for nightly Fast Compression backups for a 100 GB database that changes 5% daily and uses a 35% data change threshold.

Days	Native backup	LiteSpeed full backup	Fast Compression backup
1	100 GB	20 GB	20 GB (Full)
2	100 GB	20 GB	1 GB (Diff)
3	100 GB	20 GB	2 GB (Diff)
4	100 GB	20 GB	3 GB (Diff)
5	100 GB	20 GB	4 GB (Diff)
6	100 GB	20 GB	5 GB (Diff)
7	100 GB	20 GB	6 GB (Diff)
Total:	700 GB	140 GB	41 GB

## Quick Start

To start do one of the following:

- Run the Backup wizard and select the Fast Compression backup type. See "Back Up Individual Databases" (page 38) for more information.
- Create a maintenance plan with the Fast Compression Backup task. See "Back Up Databases Using Maintenance Plans" (page 66) for more information.

**Tip:** To create Fast Compression jobs or tasks on several instances, create and deploy a LiteSpeed backup template. See "Create Backup Templates" (page 31) for more information.

## Backup Files and Folders

Fast Compression handles the naming of files automatically. Fast Compression backups have the following format:

- **Disk backup:** *database\_name.litespeed.f#[.d#][.s#][.m#].bkp*
- **TSM backup:** *file\_space\_name\database\_name\litespeed.f#[.d#][.s#]*

where:

- *f#* is full backup index
- *d#* is diff backup index (if used)
- *s#* is stripe index (if used)
- *m#* is mirror index (if used)

**Note:** All indexes start at zero.

## Disk Backup Folders

Since the database name is incorporated into the backup name, you can safely select the same directory for all databases on an instance. If striping, you can select several directories. Also, you can add directories to mirror the entire backup file to multiple locations.

It is recommended that you create a new folder to use for Fast Compression backups. If you decide to back up to a folder that already has database backups, Fast Compression performs some validations to see if a full backup already exists:

- If the last full backup was not performed by Fast Compression, but a valid, full backup exists in the designated Fast Compression folder, then Fast Compression uses it and begins the process with a differential backup, preventing the need to run an initial full backup on the database.
- If the last full backup was not performed by Fast Compression and that full backup is either missing or has an LSN verification issue, Fast Compression starts off by executing a full backup.
- If the last Full backup was performed by Fast Compression and is either missing or has an LSN verification issue, Fast Compression escalates to a full backup (if the escalate option is selected) or continues with a differential backup.

## Disk Backup Files

Select whether you prefer to create a unique file for each backup or you prefer to manage a single file for each backup set (a backup set is composed of one full database backup plus all associated differential backups):

- Self-Contained Backup Sets—Provides the convenience of only having to manage a single file per backup set. Only one file needs to be saved to or pulled from tape or copied from the backup location to a secondary location. If backing up more than one database, a file for each database will be created.
- Separate Backup Files—Creates a unique file for each backup in the backup set. This option provides the convenience of having to move less data to tape or across the network when copying individual backup files. Using this option means that up to two physical files may be needed to restore the database (full backup plus the associated differential for the day in question).

The Self-Contained Backup Sets option automatically verifies the Full backup exists. The Separate Backup Files option performs the same validation by default, but Fast Compression includes an option to not perform this check.

## Full Backup Conditions

You can set the following thresholds to define when to issue a full backup:

- The amount of time elapsed since the last full backup
- The amount of database changes since the last full backup

Fast Compression measures the amount of data change by either querying SQL Server or by comparing the size of the last differential to the last full backup. The option to Query Actual Data Pages provides the most accurate way to determine the amount of data change. If the query fails for any reason, Fast Compression will automatically run a size comparison to the last Differential backup.

For example, set this parameter to 20%, and should the database change by 20% or more, Fast Compression will automatically run a Full backup. The larger the threshold, the larger the differential backups can grow before Fast Compression triggers the next Full backup.

Regardless of how much underlying database data has changed, when exceeding the maximum interval (in days) between full backups, Fast Compression will force a full backup.

**Notes:**

- Before a differential Fast Compression backup is available, the last full backup must have been created in the Fast Compression backup or alternate backup folder.
- When backing up the master database as part of a Fast Compression maintenance plan or job, Fast Compression always executes a full backup.
- The copy-only full backups cannot serve as a base for differential backups.

Additionally, you can prevent full backups from occurring on specified days of the week. If you select to exclude specific days of the week from Full backups and Fast Compression is set to execute the first time on an excluded day, assuming no full backup exists that can be used as described above, Fast Compression will not execute a full backup. This will continue until Fast Compression runs on a day that is not excluded.

## Backup Escalation

This option causes LiteSpeed to issue a full backup, if one of the following problems is discovered in the current backup set:

- The full backup is missing.
- A differential backup is missing from the backup set (excludes backups automatically removed after the specified retention period).
- LSN verification fails in the backup set.
- Verify operation fails on full or differential backup.

**Note:** If a problem is detected and a full backup is created through escalation, an error will be returned.

Full backup escalation is selected by default to maintain high recoverability level in the situations where recoverability may be limited (missing differential in set) or not available at all (missing full backup). This setting provides insurance against unanticipated errors. For example, if a backup file is missing from the backup set (someone accidentally deleted it), or there is some other type of issue like a Log Sequence Number (LSN) validation error or file corruption, you would not normally be able to restore the database. To correct for this potential issue with backups, Fast

Compression automatically runs a full backup to put the database in a restorable state. Errors are still noted in the LiteSpeed UI Console and alerts will still be sent via the job.

If you uncheck this option and Fast Compression discovers an issue, you will have to correct the problem manually. If the physical file for the last full backup cannot be found, a differential backup may be executed successfully, but you will not be able to recover the database using these backups unless the correct full backup is located. Correction may require forcing a full backup using the `@ForceFull` parameter. See "xp\_slsFastCompression" (page 242) for more information. Under normal operating conditions, you should not experience these types of issues as they are normally caused by accidental deletion of files or disk corruption, both of which occur very infrequently.

## Backup Verification

Make sure the backup files in the backup set have integrity. This provides an added level of insurance the backup files can be restored. Verification failures appear in the LiteSpeed UI Console and, optionally, as job failure notifications. A verification failure after a differential backup will trigger the backup escalation process, if selected.

## Cleanup

Cleanup provides a convenient way to remove old backups from disk without disrupting Fast Compression. Select this option to remove full and differential backup files and transaction log backups that are older than the specified time period.

The cleanup routine is backup set aware. This is important because the cleanup will never remove a full backup that is needed by a differential backup that is not being deleted. If you use the Separate Backup Files option in Fast Compression, you have the added flexibility of being able to remove differential backups from the active backup set that are no longer needed.

**Note:** Fast Compression does not raise errors if it detects a missing backup from a backup set that was removed via the cleanup process.

The backup retention will never delete:

- The backup files, if there are mixed backups in the same backup file. For example, if a user performs a backup of AdventureWorks and Pubs into the same mybackups.bak backup file.
- The full backup, if there are associated differential or t-log backups in the backup set that are not eligible for cleanup.

See "Smart Cleanup" (page 58) for more information.

## Notification

Use the Job Notification feature to be notified when any type of failure occurs, whether that failure was due to a failed backup, a validation issue like a missing Full or Differential backup, an LSN verification issue, or a failed Verify operation.

## Backup Jobs

Completing the wizard will create the Fast Compression backup job. Using the Backup wizard, you can optionally schedule transaction log backups for the database. Transaction log backups are scheduled as a separate job from Fast Compression.

## Double Click Restore Executables

LiteSpeed executables can be created for Native SQL Server and LiteSpeed disk backups. In many cases, LiteSpeed disk backups can be directly converted to Double Click Restore executables.

A Double Click Restore is an executable that has an .exe extension and performs a database restore when double-clicked. An executable file allows you to restore a backup on a server instance that does not have LiteSpeed installed.

A Double Click Restore executable is created by either writing a loader program designed to restore backup files, or by inserting the loader directly into the header of a suitable LiteSpeed backup file. If you convert a striped backup file, the first file will be the executable (.exe), and the others will remain unchanged.

## Double Click Restore Naming Conventions

Double Click Restore conversion may modify the extension of the backup file.

For LiteSpeed backups, file name conversion depends on the backup file size. If the backup file size is less than 4 GB, then the backup file name will have the .exe extension. If the backup file is larger than 4 GB, then the backup file name will not have the .exe extension and LiteSpeed will create X.exe, the empty Double Click Restore loader that restores the backup when double-clicked.

For native SQL Server backups, LiteSpeed will create the empty Double Click Restore loader that has the .exe extension and restores the backup when double-clicked.

Backup Type	Name Before Conversion	Name After Conversion
LiteSpeed (<4GB)	X.exe	X.exe (No name changes)
	X	X.exe
LiteSpeed (>4GB)	X.exe	X*
	X	X* (No name changes)
Native SQL Server	X.exe	X*
	X	X* (No name changes)

\*—X.exe is created as empty Double Click Restore loader. You can locate it in the same directory as the converted X.

*To create a new Double Click Restore executable, do one of the following:*

- Select the **Create a Double Click Restore executable...** option when creating backups in the LiteSpeed UI Console.
- Supply `@doubleclick = 1` when creating a backup using procedures. See "xp\_backup\_database" (page 140) for more information.
- Supply `--doubleclick 1` when creating a backup using command-line interface. See "LiteSpeed Command-Line Arguments" (page 90) for more information.

## Scenario

You need to restore particular compressed and encrypted LiteSpeed backups on a server that does not have LiteSpeed

*To restore LiteSpeed backups on a server that does not have LiteSpeed*

1. Define which backup files are needed for the restore and convert them to the Double Click Restore executables. Do one of the following:
  - Right-click a backup in the Backup Browser tab or in the Backup History tab and select **Convert to Double Click Restore backup**.
  - Run `exec xp_slsCreateDCR @FileName='<path>'`  
where `<path>` is the path to the backup.

2. Copy the Double Click Restore executable(s) you created to the server that does not have LiteSpeed.

**Note:** If a backup file is more than 4 GB, you need to copy both the converted backup file and the empty Double Click Restore loader.

3. Log on to the server, double-click the first Double Click Restore Executable file to restore and complete the LiteSpeed Double Click Restore dialog. Repeat for all other files.

**Note:** If you deselected and selected appended backups to restore, you may need to re-enter the encryption password.

## Compression Levels

LiteSpeed offers the following compression levels that allow you to specify compression from least compression to most compression, with a corresponding CPU trade-off.

Compression Level	Description
1	Medium Compression—for servers where minimal CPU utilization is preferred at the expense of some compression.
2	Medium-High Compression—a new highly optimized low CPU algorithm for

Compression Level	Description
	environments where low CPU utilization is preferred but with improved compression over level 1.
3, 4, 5, 6	High Compression—for databases where balanced compressed backup size and CPU utilization is important.
7, 8	<p>Extreme Compression—a new highly optimized extreme compression algorithm for databases where compressed size is very important with only a slight increase in CPU utilization over previous levels.</p> <p><b>Note:</b> Levels 9, 10 and 11 have been deprecated and mapped to the same compression algorithm as is used for the level 8.</p>

Depending on your environment, the various algorithms will yield different results. When choosing a compression level, test various options to determine the best option for your environment. See "Test Optimal Backup Settings" (page 56) for more information.

Generally, the higher the compression ratio the higher the CPU utilization and potentially more compression. That is, the higher compression levels will look for longer patterns to compress, as well as perform more passes on the data.

The higher levels do not guarantee better compression ratios as the nature of the data dictates the final result. Therefore, some databases will get varying results as the level increases.

Additionally, if a higher level gets significantly better compression, it may actually perform faster than a lower level. Typically, the higher levels require more time for the backup.

## Adaptive Compression

With Adaptive Compression you do not have to run the Backup Analyzer wizard to determine the best compression level for a database. LiteSpeed will dynamically change the compression level during a backup in order to optimize for speed or size, while maximizing use of available CPU. If the server workload changes during the backup (change in CPU or Disk IO), Adaptive Compression automatically switches compression to maintain optimal performance.

You can select to optimize backups either for size or for speed:

- Optimize for speed—Backups complete in the least amount of time possible. Available CPU is leveraged to reduce backup size, but not at the expense of increased backup time. This is the default setting.
- Optimize for size—Backups are completed with higher compression while managing overall backup time to ensure backups do not take a long time to complete (when compared to optimizing for speed). In this mode, LiteSpeed allows the backup to complete more slowly if the reduction in speed results in a smaller backup file. This mode is designed for databases where a smaller backup is desired but managing how long the backup takes to complete is important as well.

**Note:** Adaptive Compression is only available with LiteSpeed 6.5 or later; Enterprise license.

## Encryption

Encryption is a mechanism for protecting data, which applies to it a specially designed algorithm, effectively obfuscating its content by making it different from the original.

**Note:** If running Windows 2000 to utilize the higher levels of encryption, the Windows 2000 High Encryption Pack must be installed.

LiteSpeed offers the option of encrypting in the following formats:

- 40-bit RC2
- 56-bit RC2
- 112-bit RC2
- 128-bit RC2
- 168-bit 3DES
- 128-bit RC4
- 128-bit AES
- 192-bit AES
- 256-bit AES

Higher levels of encryption require slightly more CPU, but generally the impact of 256-bit AES encryption on a backup running on a modern server is very low at less than 0.5% CPU utilization. We recommend for best security of a backup that 256-bit AES be used when encryption is needed.

**Caution:** When encrypting data, take care not to lose the encryption key; a backup cannot be restored or recovered without the original encryption key.

## Test Optimal Backup Settings

Higher compression levels result in smaller backup files, but they also may consume additional CPU. If the server does not have sufficient CPU, then the backup may take longer to complete. The Backup Analyzer evaluates different settings, such as compression level, striping, and backup destinations, to help you determine optimal backup settings.

**Note:** When running the Backup Analyzer, follow these guidelines for the best results:

*Minimum:* 1 GB sample size.

*Recommendation:* 10 GB sample size. The combination of compression levels, encryption options, and backup locations should be considered. For example, if you sample 4 GB of data with 10 different tests, LiteSpeed will take the same amount of time as it takes to back up 40 GB. On very



large databases, you can speed up the analysis by reducing the number of backup combinations or by reducing the amount of sample data.

## Backup Analyzer Wizard

The Backup Analyzer wizard guides you through selecting the backup parameters to test. The wizard tests all of the different combinations of your selected parameters by backing up a portion of the database. It does not interfere with existing backup schedules or sets.

### *To run the Backup Analyzer wizard*

1. Select the **Backup Manager** pane (CTRL+1).
2. Right-click the database and select **Backup Analyzer Wizard**.



#### **Tips:**

- To test striped backups, click **Add** to add several destinations and select the **Test backup striping to selected backup locations** checkbox on the Backup Location page.
  - If you add more than two destinations, select both checkboxes to test all destination combinations.
  - If you select a large number of backup parameters, you may want to schedule the tests to run at a specified time within the LiteSpeed Analyzer job.
3. Complete the wizard.

**Note:** After the required number of bytes is received for analysis, the process is intentionally aborted. This generates the VDI error messages in the LiteSpeed log files and the SQL Server error log. Please ignore them.

## Backup Analyzer Tab

The Backup Analyzer tab presents the test results in a graph and grid:

- **Graph**—Displays the backup duration and compression amount for each test so you can easily compare the results. The test with the best duration to size ratio has stars instead of circles on the graph. If you select a test in the grid, LiteSpeed indicates the corresponding test in the graph with a yellow border around the circles. You can view previous tests by changing the **Existing test** field.
- **Grid**—Displays the important details of each test. You can double-click a row for more information about the test. Click  to export the grid to Microsoft Excel or  to print the grid.

**Tips:** For panes that have grids, you can sort, group, move, and remove the columns:

- To sort and group the results, right-click a column heading and select the appropriate options.

- To add or remove columns, right-click a column heading and select **Column Chooser**. Add a column by dragging it from the list into the column headings. Remove a column by dragging its column heading into the list.
- To move a column, drag the column heading to the new location.

## Network Resilience

LiteSpeed's read and write resilience can handle various failures on both network and attached storage devices. If LiteSpeed fails to write a backup during a backup operation or fails to read a backup during a restore operation, it will wait and retry the failed operation. If successful on a subsequent attempt, the backup or restore operation continues without interruption. Without network resilience, these operations would fail immediately on the first read or write problem encountered.

You can control the number of times to retry and the amount of time to wait before retrying.

**Note:** This functionality is only available for disk operations.

### *To configure retry options in wizards*

1. Access advanced options.
2. Select one or both network resilience options to change the default values.
3. Complete the wizard.

### *To configure retry options in procedures*

» Use @IOFLAG parameter with the following extended stored procedures:

- xp\_backup\_database (page 140)
- xp\_backup\_log (page 156)
- xp\_slsFastCompression (page 242)
- xp\_restore\_database (page 200)
- xp\_restore\_log (page 224)
- xp\_restore\_verifyonly (page 234)

### *To configure retry options from the command line*

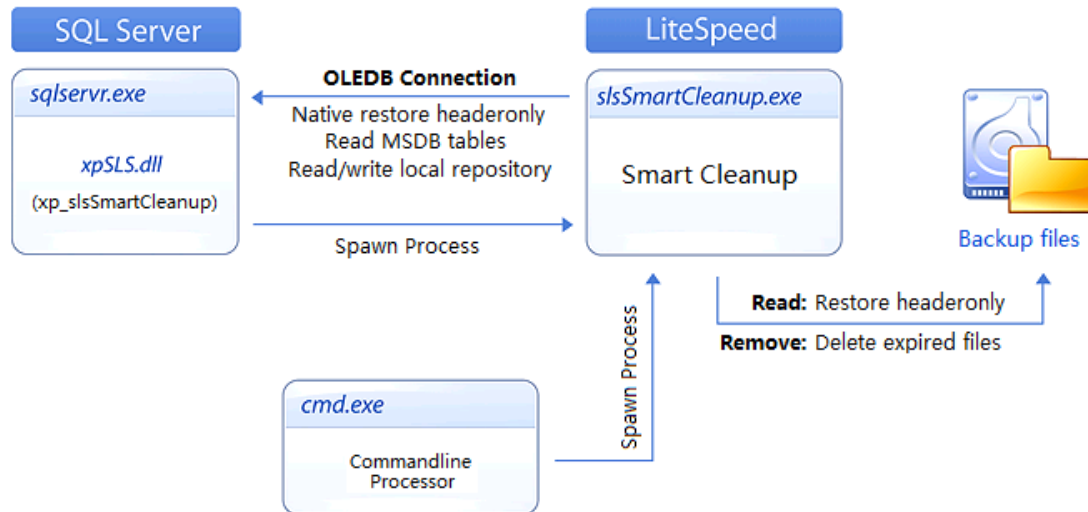
» Use -X or --IOFlags parameter. See "LiteSpeed Command-Line Arguments" (page 90) for more information.

## Smart Cleanup

Smart Cleanup provides a convenient way to remove old LiteSpeed backups. It removes full and differential backup files and optionally transaction log backups based on a user-defined period. LiteSpeed will ignore copy-only backups except on secondary replicas in AlwaysOn Availability groups, in which case it will allow deletions.

Smart Cleanup is backup set aware and will never remove a full backup that is needed by a differential backup that is not being deleted.

The diagram below shows how the LiteSpeed components communicate to SQL Server to check if backups are eligible for cleanup and delete them each time Smart Cleanup runs.



The backup retention will never delete:

- The backup files, if there are mixed backups in the same backup file. For example, if a user performs a backup of AdventureWorks and Pubs into the same mybackups.bak backup file.
- The full backup, if there are associated differential or t-log backups in the backup set that are not eligible for cleanup.
- File/FileGroup backups
- File/FileGroup differential backups
- Partial backups
- Partial differential backups

**Note:** Fast Compression does not raise errors if it detects a missing backup from a backup set that was removed via the cleanup process.

#### *To configure Smart Cleanup in Wizards*

- Open existing or create a new backup template. See "Create Backup Templates" (page 31) for more information.
- Open or create the Fast Compression Database task in a maintenance plan. See "About Creating Maintenance Plans" (page 62) for more information.
- Run the Backup wizard and select the Fast Compression backup type. See "Back Up Individual Databases" (page 38) for more information.

*To run Smart Cleanup manually, do one of the following:*

- Use the xp\_slsSmartCleanup extended stored procedure. See "xp\_slsSmartCleanup" (page 255) for more information.
- Run slsSmartCleanup.exe with the appropriate arguments from the command-line. See "SmartCleanup Command-Line Arguments" (page 117) for more information.

# Automate Maintenance Tasks

## About Automating Maintenance Tasks

Maintenance plans help you automate routine database maintenance tasks, such as backing up databases, updating statistics, and rebuilding indexes to run on a specific day and time. This ensures that your databases are stable and perform at optimal levels.

### Legacy and SSIS Maintenance Plans

In SQL Server 2000, maintenance plans create multiple agent jobs as output. These jobs contain a command-line call to the function you want to execute. SQL Server 2000 and LiteSpeed 4.x maintenance plans are considered *legacy* plans.

In SQL Server 2005 or later, maintenance plans create SSIS packages that run as agent jobs.

To take advantage of Integration Services (SSIS) in maintenance plans, Integration Services must be installed on any server instance where you want to create maintenance plans. Integration Services may be a part of the Client Tools or a part of SQL Server.

#### Notes:

- Express edition of SQL Server 2005 does not support maintenance plans.
- LiteSpeed also creates a *legacy* maintenance plan when:
  - SSIS components are not installed on server.
  - SQL Server 2005 SSIS components version is lower than 9.00.3042 (Service Pack 2).

### Native SQL Server and LiteSpeed Maintenance Plans

You can manage both native SQL Server and LiteSpeed maintenance plans using the LiteSpeed UI Console. LiteSpeed maintenance plans have the extended task set and a larger number of advanced options, including the high-performance compression technology LiteSpeed uses to create backups.

You need to have LiteSpeed installed on every server instance where you want to create LiteSpeed maintenance plans. Otherwise, only native SQL Server functionality is available.

If LiteSpeed is installed on the server, you may decide whether to use it for each particular task or not.

#### Convert to LiteSpeed

You can convert native SQL Server maintenance plans to LiteSpeed maintenance plans. Using this option, you can also convert previous LiteSpeed maintenance plans to the latest version.

### *To convert a maintenance plan*

1. Select **Maintenance Plans** in the Navigation pane (CTRL+4).
2. Right-click a maintenance plan and select **Convert to LiteSpeed**.

## About Creating Maintenance Plans

Use the Design pane to create a workflow of database maintenance tasks. Tasks can execute independent of another task's status, or can be dependent on another task's completion before they can begin execution.

You can add additional subplans in a maintenance plan to group related tasks or to schedule tasks to execute at different times.


### *To create a maintenance plan*


1. Select **Maintenance Plans** in the Navigation pane (CTRL+4).
2. Right-click a server instance and select **Tasks | Create New Maintenance Plan** in the Maintenance Plan pane.
3. Drag tasks to the Design pane to add them to the subplan. Double-click a task to specify its properties.

**Tip:** If you need to create several similar plans, you can simply copy existing plans or subplans and then make the necessary edits. See "Copy Maintenance Plans" (page 71) for more information.

### **For legacy maintenance plans only:**

- You cannot have more than one task of the same type in a subplan. To add a duplicate task, create a new subplan and add the task.
- You cannot have both a Fast Compression Backup task and Back Up Database task added to the same subplan.
- Any tasks in the subplan must use the same list of databases. If you select a different database in a task, LiteSpeed prompts you to apply the database change to the entire subplan.

Task	Add this task to...	Available for legacy plans
 Fast Compression Backup	Perform full or differential backups using the Fast Compression technology. <b>Note:</b> Fast Compression is only available with versions 5.1 and above of LiteSpeed; Enterprise license.	With LiteSpeed only

Task	Add this task to...	Available for legacy plans
	See "Back Up Databases Using Maintenance Plans" (page 66) for more information.	
 Back Up Database	Perform full, differential, or transaction log backups; with or without encryption. You can back up databases on multiple servers by adding a separate backup task to the maintenance plan for each server.  See "Back Up Databases Using Maintenance Plans" (page 66) for more information.	Yes
 Check Database Integrity	Validate the following: <ul style="list-style-type: none"> <li>• Disk space allocation</li> <li>• Page and structural integrity for tables and indexed views</li> <li>• Catalog consistency</li> <li>• Contents of indexed views</li> <li>• Service Broker data</li> </ul>	Yes
 Reorganize Index	Defragment and compact existing indexes to improve performance.	With LiteSpeed only
 Execute Job	Execute an existing SQL Server Agent job.	No
 Clean Up Maintenance Plans	Remove obsolete backup files and reports created by a maintenance plan.  See "Clean Up Maintenance Plans" (page 71) for more information.	With LiteSpeed only
 Clean Up History	Remove historical data from the <b>msdb</b> database and LiteSpeed Local and Central repositories for the following: <ul style="list-style-type: none"> <li>• Back up and restore history</li> <li>• SQL Server Agent jobs</li> <li>• Maintenance plans</li> <li>• LiteSpeed activity</li> </ul> <p><b>Note:</b> LiteSpeed activity is removed from the Central repository, only if it is located on the same server, where the Local repository resides.</p>	With LiteSpeed only

Task	Add this task to...	Available for legacy plans
	<ul style="list-style-type: none"> <li>Any information for deleted databases</li> <li>Log shipping history</li> <li>Status history (Job, DTS, Maint Plans)</li> </ul>	
 Notify Operator	<p>Add this task to send an email notification to one or more existing operators.</p> <p><b>Note:</b> This task requires the SQL Server Agent configured to send email using Database Mail. You can use the notification profiles only if they are already configured within a SQL Server instance.</p>	With LiteSpeed only
 Rebuild Index	Drop and recreate an index to improve performance.	Yes
 Shrink Database	Reduce the size of data and log files in a database that grow beyond a specified size.	Yes
 Execute T-SQL Statement	<p>Execute statements or batches on one or more databases.</p> <p><b>Note:</b> SSIS maintenance plans support only the Transact-SQL command type.</p>	Yes
 Update Statistics	Update column and index statistics.	Yes

- (SSIS Plans only) Right-click a task to execute first and select **Add Constraint** to draw a line to the dependent task.

**Tips:**



- You can add multiple constraints for a single task, and can execute those tasks concurrently or based on the outcome of the previous task.
- To use edit the constraint or use an expression to evaluate precedence, right-click the constraint line and select **Edit**.



**Note:** Constraints are not available for legacy plans. Legacy plan tasks are executed in the following sequence:

- Clean Up History
- Check Database Integrity
- Rebuild Index





4. Shrink Database
5. Update Statistics
6. Reorganize Index
7. Fast Compression Backup or Back Up Database
8. Clean Up Maintenance Plans
9. Notify Operator
10. Execute T-SQL

5. Click  to enter or select a schedule for executing the current subplan.
6. Click  and repeat steps 2, 3 and 4 for additional subplans you want to add.
7. (Optional) Set reporting options.

Reporting and Logging	Click  to set the reporting and logging options. See "Reporting and Logging in Maintenance Plans" (page 288) for more information.
Notifications	Click  to notify an operator of job (subplan) status when a job fails, succeeds, or completes.

#### Tips:


- To manually execute, edit, or delete a maintenance plan, right-click the plan in the Maintenance Plan pane and select an option.
- To change the plan owner, open the plan in the Design pane and select the owner from the drop-down list in the upper-right corner of the pane. Note that the plan owner is only responsible for creating and editing plans. The account that executes packages is the SQL Agent service account (or a proxy account). See "Privilege and Grant Requirements" (page 274) for more information.
- To remove a subplan, select it and click  on the toolbar in the maintenance plan designer.
- To disable a subplan, double-click the subplan in the Design pane.
- To define any connections you want to use for tasks in a maintenance plan, click  **Manage Connections**. Once you add a connection, you can select it from the Connection drop-down list in any task. You can also define a connection at the task level that applies to other tasks and subplans. For legacy plans, you can add only one connection for remote logging and you must use Windows Authentication.

## Back Up Databases Using Maintenance Plans

**Note:** Backup options may vary depending on the LiteSpeed and SQL Server version and plan type (legacy or SSIS; native or LiteSpeed). See "About Automating Maintenance Tasks" (page 61) for more information.

### To configure database backups

1. Drag the **Back Up Database** task or the **Fast Compression Backup** task to the Design pane.
2. Double-click the task and review the following for additional information:

Databases	Click  to select databases you want to include in and exclude from the maintenance plan. See "LiteSpeed's Logic for Backing Up Multiple Databases" (page 46) for more information.
Back up database across one or more files	If you selected <b>Disk</b> , this option creates one backup file for all selected databases.  If you want to create separate disk backups for databases, select the <b>Create backup file for every database</b> option.
LiteSpeed file format	You can specify custom backup name and description using both the LiteSpeed variables and text. See "LiteSpeed Variables" (page 271) for more information.  The default backup file format uses the following information: <ul style="list-style-type: none"> <li>• %D—Database name</li> <li>• %T—Backup type (Full, Diff or Log)</li> <li>• %Y-%m-%d-%H%M%S—Date and time</li> <li>• %EXT%—File extension</li> </ul> <p><b>Note:</b> Fast Compression handles the naming of files automatically. See "Backup Files and Folders" (page 49) for more information.</p>

If you selected **TSM Backup** or **TSM Archive**, review the following for additional information:

Client node	Enter the node name for the TSM session. This field is not case-sensitive.
Client owner password	Enter the access password for the specified node.
Configuration file	Select the configuration file. (Usually, dsm.opt.)  <b>Note:</b> This file contains session options such as the TSM server's TCP

	address. If you select the <b>Use PASSWORDACCESS GENERATE from TSM configuration file</b> checkbox and your options file is configured to support this option, you do not need to specify the client node and client owner password.
Management class	Select the management class (policy) to associate with the backup object being created. LiteSpeed will use the default management class, if this option is not selected.
TSM Filespaces (Fast Compression backups)	This step is optional. You can click <b>TSM Filespaces</b> to select the existing or enter new file space name(s). <b>Note:</b> Fast Compression handles the naming of files automatically. See "Backup Files and Folders" (page 49) for more information.
TSM Objects (Regular backups)	Click <b>TSM Objects</b> and specify the filesystem, the high-level name and the format of the low-level name.

3. Select the Options page.

If you selected the **Backup Database** task, review the following:

Remove files older than	This option only removes files that match the specified <b>extension</b> from the specified <b>destination folder</b> . <b>Tip:</b> To remove obsolete files from different locations, use the Clean Up task. See "Clean Up Maintenance Plans" (page 71) for more information.
Verify backup when finished	Select this option to verify that LiteSpeed successfully wrote all backup files and can read them.
Set native Backup compression	Select one of the following options: <ul style="list-style-type: none"> <li>• Use the default server setting</li> <li>• Compress backup</li> <li>• Do not compress backup</li> </ul>

If you selected the **Fast Compression Backup** task, review the following:

Fast Compression Backup Options	You can set the following thresholds to define when to issue a full backup: <ul style="list-style-type: none"> <li>• The amount of time elapsed since the last full backup</li> <li>• The amount of database changes since the last full backup</li> </ul>
---------------------------------	--


	<p>Fast Compression measures the amount of data change by either querying SQL Server or by comparing the size of the last differential to the last full backup. The option to Query Actual Data Pages provides the most accurate way to determine the amount of data change. If the query fails for any reason, Fast Compression will automatically run a size comparison to the last Differential backup.</p> <p>For example, set this parameter to 20%, and should the database change by 20% or more, Fast Compression will automatically run a Full backup. The larger the threshold, the larger the differential backups can grow before Fast Compression triggers the next Full backup.</p> <p>Regardless of how much underlying database data has changed, when exceeding the maximum interval (in days) between full backups, Fast Compression will force a full backup.</p> <p><b>Notes:</b></p> <ul style="list-style-type: none"> <li>• Before a differential Fast Compression backup is available, the last full backup must have been created in the Fast Compression backup or alternate backup folder.</li> <li>• When backing up the master database as part of a Fast Compression maintenance plan or job, Fast Compression always executes a full backup.</li> <li>• The copy-only full backups cannot serve as a base for differential backups.</li> </ul>
Backup Escalation	<p>This option causes LiteSpeed to issue a full backup, if one of the following problems is discovered in the current backup set:</p> <ul style="list-style-type: none"> <li>• The full backup is missing.</li> <li>• A differential backup is missing from the backup set (excludes backups automatically removed after the specified retention period).</li> <li>• LSN verification fails in the backup set.</li> <li>• Verify operation fails on full or differential backup.</li> </ul> <p><b>Note:</b> If a problem is detected and a full backup is created through escalation, an error will be returned.</p>
Verification	<p>Make sure the backup files in the backup set have integrity. This provides an added level of insurance the backup files can be restored. Verification failures appear in the LiteSpeed UI Console and, optionally, as job failure notifications. A verification failure after a differential backup will trigger the backup escalation process, if selected.</p>

4. Select the LiteSpeed page. Review the following additional information about the

encryption and compression options:

Adaptive Compression	<p>LiteSpeed will automatically select the optimal compression based on throughput and CPU usage and optimize backups either for size or for speed (default).</p> <p><b>Note:</b> Adaptive Compression is only available with LiteSpeed 6.5 or later; Enterprise license.</p>
Compression level	<p>Select 0 for no compression or 1-8 to compress the file. See "Compression Levels" (page 54) for more information.</p> <p><b>Note:</b> Higher compression levels result in smaller backup files, but they also take longer to complete. For assistance determining the best compression options, use the Backup Analyzer. See "Test Optimal Backup Settings" (page 56) for more information.</p>
Encrypt backup	<p>Select this checkbox to encrypt the backup. Then, select the encryption level and enter the encryption password. See "Encryption" (page 56) for more information.</p>

Review the following additional information about the advanced options:

Throttle	Enter the maximum percentage of CPU that LiteSpeed can use for the process. The default is <i>100</i> .
Priority	Select the priority of the backup over other transactions or processes running on the same server. The default is <i>Normal</i> .
Buffer count	Enter the number of SQL Server buffers available for a LiteSpeed operation. The default is <i>20</i> .
Max transfer size	Enter the maximum backup file size in bytes. The default is <i>1048576</i> .
Processor affinity	<p>Click  to select which processors LiteSpeed can use. The default is <i>0</i>, which allows LiteSpeed to use all available system processors.</p> <p>See "Processor Affinity" (page 25) for more information.</p>
Number of threads	Enter the number of threads used for the backup. The default is <i>n-1</i> threads, where <i>n</i> is the number of processors.

Logging level	<p>Select one of the following options:</p> <ul style="list-style-type: none"> <li>• None—LiteSpeed does not log backup/restore activity.</li> <li>• Verbose—LiteSpeed logs all backup/restore activity.</li> <li>• Verbose on failure—LiteSpeed logs all activity and only saves log files if the backup or restore operation fails. If it succeeds, LiteSpeed does not save the log.</li> </ul>
Network Resilience	<p>If LiteSpeed fails to write a disk backup or read from disk, it will wait and retry the operation. You can control the number of times to retry and the amount of time to wait before retrying.</p> <p>The default is <i>4 retries</i>, the maximum allowed setting is 1000. The default period to wait before retry is <i>15 seconds</i>, the maximum allowed setting is 300 seconds.</p> <p>See "Network Resilience" (page 58) for more information.</p>
<p><b>Tips:</b></p> <ul style="list-style-type: none"> <li>• Before you start tuning the CPU Throttle or Affinity parameters to adjust backup performance, try limiting the number of threads. If you decide to use an affinity value other than default, it is recommended that you limit the threading as well.</li> <li>• You may also want to consider using Adaptive Compression to maintain backup performance. See "Adaptive Compression" (page 55) for more information.</li> </ul>	

**Note:** LiteSpeed defaults typically result in the best performance. You should only modify advanced options after careful planning and testing. See "Configure LiteSpeed Defaults" (page 24) for more information.

Review the following additional information about the verification and recovery options:

Optimize the Object Level Recovery speed	<p>Select to create an index of objects in the backup file. This option is only available for LiteSpeed backups.</p> <p><b>Note:</b> Before you can recover objects or execute a SELECT statement, you must read the backup file to create an index of restorable objects. If you select this option, LiteSpeed uses the index in the backup file to read the backup file, which makes the object level recovery process much faster.</p>
Create a Double Click Restore executable	<p>Select to create an executable backup file or a Double Click Restore Loader that allows you to restore a backup on a server instance that does not have LiteSpeed installed. See "Double Click Restore Executables" (page 53) for more information.</p> <p><b>Note:</b> A Double Click Restore can only be created for a disk file.</p>

Perform checksum before writing to media	Select to verify checksums when a backup is created.  Additionally, you can control the response to an error. If you select the <b>Continue on error</b> option, the backup is executed despite encountering an invalid backup checksum.
--	--

## Clean Up Maintenance Plans

Use this task to remove obsolete backup files and maintenance plan reports.

### *To clean up maintenance plan data*

1. Drag the Clean Up Maintenance Plans task to the Design pane.
2. Double-click the task and review the following for additional information:

Delete files of the following types	Select the type of files to clean up: <ul style="list-style-type: none"><li>• Backup files</li><li>• Maintenance plan text reports</li><li>• Any files</li></ul>
File location	Select the location of the files you want to delete.
Delete specific file	Select this option to delete a specific file.
Search folder and delete files based on extension	Select this option to remove any files with a specific file extension.
File extension	Enter the extension of the files you want to remove.
Include all subfolders	Select this option to include all subfolders when searching for files to remove.

## Copy Maintenance Plans

To set up similar plans, you do not necessarily have to start from scratch. Create a plan and copy it to any server instances where you want the plan to run.

**Note:** Copying and importing plans and subplans between servers may require additional manual steps, if the source and target servers have:

- Different SQL Server versions—You may need to review and edit the selected databases list. See "Back Up Databases Using Maintenance Plans" (page 66) for more information.




Also, when copying a subplan between SQL Server 2005 (or 2008) and SQL Server 2000 (or 7), multiple Backup Database and Fast Compression (SmartDiff) Backup tasks are not copied due to the difference between the SSIS and legacy plans. For legacy plans, you cannot have more than one Backup task in a subplan. You may want to create a new subplan and add a backup task to this subplan.

- Different LiteSpeed versions—Some options may be lost if they are not supported by the previous LiteSpeed version.

#### *To copy a maintenance plan*

1. In the Server tree, right-click a maintenance plan you want to copy and select **Copy Maintenance Plan**.
2. Right-click the instance where you want to paste the plan and select **Tasks | Paste Maintenance Plan**.

#### *To copy a subplan*

1. In the Design pane, select a subplan and click  **Copy**.
2. Open a new or existing maintenance plan in the Design pane.
3. (Optional) Click  on the Design pane toolbar to add a new subplan.
4. Select a subplan and click  **Paste**.

**Tip:** You can also save a plan to file and import it later. Open a plan you want to export in the Design pane and click the appropriate toolbar icon. To import a plan, right-click an instance in the server tree in the Maintenance Plans pane (CTRL+4) and select **Tasks | Import Maintenance Plan...**

## Automate Similar Backup Tasks on Multiple Instances

### Scenario

You need to schedule full, differential and t-log database backups on several servers and automate backup cleanup according to your company's retention policy.

You are going to create a maintenance plan on one SQL Server instance and then simply copy it to the other server instances.

#### *To automate backups on server instances*

1. Select **Maintenance Plans** in the Navigation pane (CTRL+4).
2. In the Server tree, right-click an instance and select **Tasks | Create New Maintenance Plan**.
3. Drag and drop the Back Up Database task in the middle of the Design pane and double-click it.



a. Select the **Full** backup type.


b. Click  and select **User databases (excluding master, model, msdb)**.


**Note:** This scenario describes how to back up all user databases. Similarly, you can configure maintenance plans to back up system databases and databases matching wildcard or regular expressions.


a. Select the **Create backup file for every database** option and specify the Destination folder and file extension for the backups.

b. Select the Options tab and select the **Remove files older than** option. Specify when the full backups are eligible for cleanup according to your company's retention policy.


c. Click **Ok** to save the task.

4. Click  **Copy** on the tool bar. The task you just created will serve as a base for differential and t-log backups.

5. Create two new subplans. To create a subplan, click , then **Ok**.


6. Select each of the two new subplans and click  **Paste**.

7. Configure the copied subplan tasks to create the differential and t-log backups instead of full backups. Specify when the differential and t-log backups are eligible for cleanup according to your company's retention policy.

8. Double-click every subplan in the plan. Enter the name and description. Click  to set schedule properties. Each subplan should have its own reoccurring schedule. For example,

- Full backups occurring every week
- Differential backups occurring every day
- T-log backups occurring every 20 minutes

When you are done editing the subplan properties, select **Enabled** and click **Ok**.

9. Save the plan, click  **Copy**.

10. For all instances where you want to paste the plan, right-click the instance and select **Tasks | Paste Maintenance Plan**.

# Restore Databases

## Restore Databases Using the Restore Wizard

The Restore wizard guides you through the process of restoring a database (full or differential), transaction log, files, or filegroups.

### Notes:

- For information about performing file restores, see <http://msdn.microsoft.com/en-us/library/ms190710.aspx>.
- For a description of restoring file and filegroup backups in SQL Server, see <http://support.microsoft.com/kb/281122>.

### Scenario

You need to restore a LiteSpeed disk backup to a new database on another SQL Server. Copy the backup files needed for restore to another server and run the Restore wizard.

### To run the Restore wizard

1. Select the **Backup Manager** pane (CTRL+1).
2. Right-click a database, select **Restore** and then **Database, Files and Filegroups, Transaction Log** or **Attached Files** (to restore attached files only).
3. Review information on the Restore Destination page. Select the **Kill all current connections before restore** checkbox to obtain exclusive access to the selected database.

**Scenario:** Select the server instance to restore the backup to and enter a new database name.

4. On the Backup Source page, select **Database** to restore from a specific database's backup history or **Device** to manually select files to restore.


**Scenario:** Select **Device** and then **Disk** and specify the backup files you copied.

If you selected **TSM Backup** or **TSM Archive**, review the following for additional information:


Client node	Enter the node name for the TSM session. This field is not case-sensitive.
Client owner password	Enter the access password for the specified node.
Configuration	Select the configuration file. (Usually, dsm.opt.)

file	<p><b>Note:</b> This file contains session options such as the TSM server's TCP address. If you select the <b>Use PASSWORDACCESS GENERATE from TSM configuration file</b> checkbox and your options file is configured to support this option, you do not need to specify the client node and client owner password.</p>
TSM Object	<p>Click <b>Select TSM Object</b>.</p> <p>Enter the filespace and the high-level and low-level names and click <b>Query TSM</b> to pick the object name from the list of available TSM objects. From the Available TSM Objects list, double-click the objects you would like to select.</p> <p><b>Note:</b> If you leave the High level and Low level fields blank, LiteSpeed will query all TSM server levels. Querying all levels may take longer to complete.</p>

5. Review the following additional information about the Backup Content page:

Encryption	If the backup is encrypted, enter the password in the Encrypted column in the grid.
Point in time restore	<p>Drag the slider to select a specific point in time to restore to. If used with the IntelliRestore option, backup files needed for the point-in-time restore are automatically selected in the grid below when you move the slider.</p> <p>Click  to restore to a specific point in time or a mark.</p> <p><b>Note:</b> You cannot restore filegroups to a point in time.</p>
IntelliRestore	Select to have LiteSpeed automatically select the backups needed to restore the database successfully.
Verify backups	Select to verify the backup file integrity before completing the wizard.

6. Review the following additional information about the Data Files page:

Restore database files as	<p>Although you can manually enter DATA and LOG locations, including secondary data files locations, it is recommended that you use locations generated by LiteSpeed.</p> <p>If the source and target locations do not match or if they are other than default, LiteSpeed allows you to select one of the following options:</p> <ul style="list-style-type: none"><li>• Keep original database locations—To use DATA and LOG directories of the existing database you are restoring the backup to.</li><li>• Restore to locations from backup set—To use DATA and LOG directories of the database which backup you are restoring.</li><li>• Use SQL Server instance default locations—To use the default DATA and LOG directories that are defined within SQL Server you are restoring the backup to.</li></ul>
CPU affinity	<p>Click  to select which processors LiteSpeed can use. The default is 0, which allows LiteSpeed to use all available system processors.</p>
Logging level	<p>Select one of the following options:</p> <ul style="list-style-type: none"><li>• None—LiteSpeed does not log backup/restore activity.</li><li>• Verbose—LiteSpeed logs all backup/restore activity.</li><li>• Verbose on failure—LiteSpeed logs all activity and only saves log files if the backup or restore operation fails. If it succeeds, LiteSpeed does not save the log.</li></ul>
Network Resilience	<p>If LiteSpeed fails to write a disk backup or read from disk, it will wait and retry the operation. You can control the number of times to retry and the amount of time to wait before retrying.</p> <p>The default is <i>4 retries</i>, the maximum allowed setting is 1000. The default period to wait before retry is <i>15 seconds</i>, the maximum allowed setting is 300 seconds.</p> <p>See "Network Resilience" (page 58) for more information.</p>

7. If you added an attachment to the backup file, select the attachments to restore on the Attached Files page.

8. Complete the wizard.

## Restore Double Click Restore Executables

*To restore a Double Click Restore executable, do one of the following:*

- Double-click the Double Click Restore executable and complete the LiteSpeed Double Click Restore dialog.
- Run the command line, change the directory until you are in the directory containing the Double Click Restore executable and run the following:

```
backup.exe -R database -F backup_file -W replace
```

where

*backup.exe* is the name of the Double Click Restore executable.

*backup\_file* specifies the path to the file containing backup data. You can supply multiple instances of this argument. Use this argument to list all backup files except the executable being run:

- The filename of the backup if there is a Double Click Restore loader created for this backup
- The filenames of any other stripes that were not converted to an executable

**Note:** The syntax is exactly the same as that for `sqlitespeed.exe`. See "LiteSpeed Command-Line Arguments" (page 90) for more information.

- Restore as any other backup using the Restore Wizard, command-line interface or procedures.

**Note:** If logging is enabled during a restore, the log file is written to:

- The default output directory—See "Configure Logging in LiteSpeed" (page 285) for more information.
- The root of `C:\`—On a server that does not have LiteSpeed installed.

## Manually Restore a Master Database

To restore the master database from a LiteSpeed backup device, start Microsoft SQL Server in single-user mode.

1. From a command prompt, change the directory until you are in the directory containing `sqlservr.exe` (Usually, `C:\Program Files\Microsoft SQL Server\MSSQL\Binn`).
2. Run the following:

```
sqlservr.exe -c -m
```

**Note:** You must switch to the appropriate directory (for the instance of Microsoft SQL Server you want to start) in the command window before starting sqlservr.exe.

2. Using another command prompt, change the directory until you are in the directory containing SQLLiteSpeed.exe (Usually, *C:\Program Files\Quest Software\LiteSpeed\SQL Server\Engine*).
3. Execute the LiteSpeed restore statement to restore the master database backup.

**Note:** The master database can only be restored from a backup created on an instance of SQL Server 2000 and newer. You cannot restore master database backups which were made on SQL Server version 7.0 or earlier.

**Examples:**

- a. Restore a master database.

```
sqlllitespeed.exe -Rdatabase -T -Dmaster -  
F"C:\MSSQL\Backup\master.bak"
```

- b. Restore to named instance.

```
sqlllitespeed.exe -Rdatabase -T -S "<Server\Instance>" -Dmaster -  
F"C:\litespeed\backup\full\master.bak" -L
```

- c. Restore with encryption.

```
sqlllitespeed.exe -Rdatabase -T -Dmaster -Kpassword -  
FC:\litespeed\backup\full\master.bak
```

- d. Restore with replace.

```
sqlllitespeed.exe -Rdatabase -T -Dmaster -  
FC:\litespeed\backup\full\master.bak -WREPLACE -W"MOVE 'master' TO  
'C:\Program Files\Microsoft SQL  
Server\MSSQL.1\MSSQL\Data\master.mdf'" -W"MOVE 'master_log' TO  
'C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\master_  
Log.ldf'"
```

- e. Restore using the Tivoli Storage Manager.

```
sqlllitespeed.exe -Rdatabase -T -Dmaster -i"fsMH\nw\testcmd" -  
c"nodename" -k"password" -j"c:\program  
files\Tivoli\TSM\baclient\dsm.opt"
```

4. Restart SQL Server and LiteSpeed
5. (If the process hangs) Stop the following services and retry:
  - Alert
  - Cluster
  - Computer Browser
  - Event Log

- License Logging
- Logical Disk Manager
- Messenger
- Net Logon
- NTLM Security Support Provider
- Plug and Play
- Remote Procedure Call (RPC) Locator
- Remote Procedure Call (RPC)
- Server
- Print Spooler
- TCP/IP NetBIOS Helper
- Windows Time
- Workstation

## Restore Objects

LiteSpeed helps you restore specific objects from a native or LiteSpeed backup file. Before you can recover objects or execute a SELECT statement, you must read the backup file to create an index of restorable objects. Then, you can execute a SELECT statement or restore tables, schemas, and other objects.

You can:

- Use the LiteSpeed UI Console to restore objects from the backup files. See "Restore Objects in the LiteSpeed UI Console" (page 80) for more information.
- Restore objects using extended stored procedures. See "xp\_objectrecovery" (page 172) for more information.
- Restore objects using command line. See "Restore Objects with the Command-Line Interface" (page 133) for more information.
- Query backups. See "Execute SELECT Statements" (page 83) for more information.

**Note:** Object Level Recovery is only available with the Enterprise license.

## Restore Objects in the LiteSpeed UI Console

Before you can recover objects or execute a SELECT statement, you must read the backup file to create an index of restorable objects. The index is an .lsm file that is saved in the same directory as the backup file.

**Notes:**

- You cannot restore objects directly from TSM files or tape backups. See "Object Level Restores from TSM Backups" (page 83) for more information.
- Object Level Recovery does not support SQL Server 2008 Transparent Data Encryption (TDE).
- LiteSpeed may take a long time to read the backup file for large databases, often with little response in the LiteSpeed UI Console. To prevent this, select the **Optimize Object Level Recovery speed** option on the Backup wizard Options page to create the index during the backup process.
- Objects are recovered as they existed at the time they were backed up. You cannot recover data to a random point in time.

*To run the Object Level Recovery wizard*

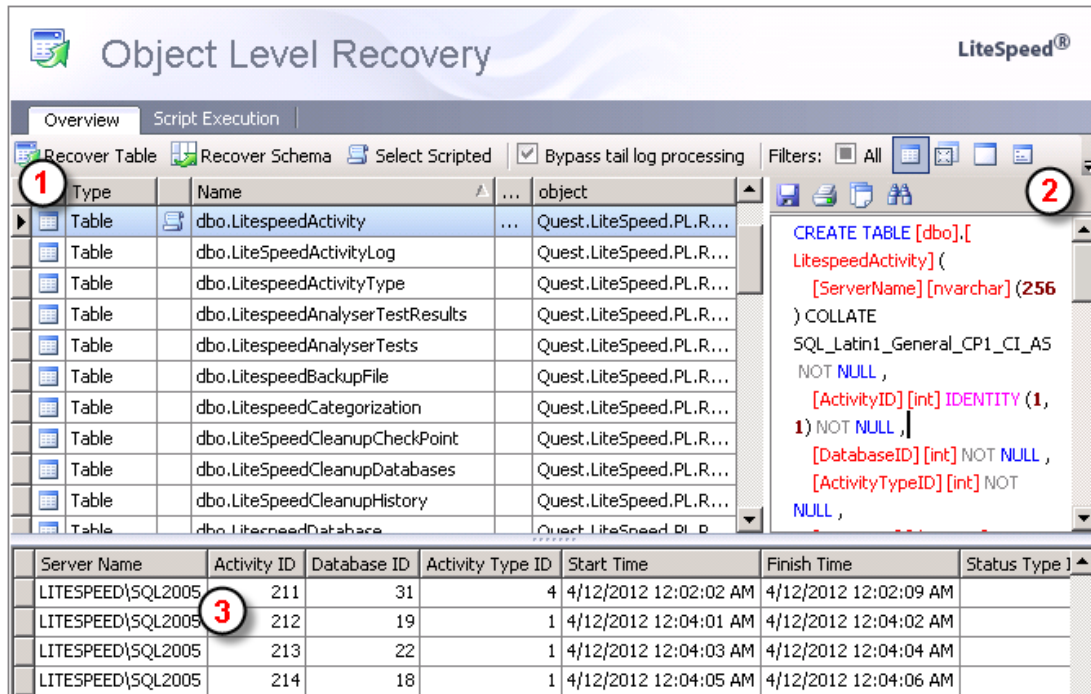
1. Select the Object Level Recovery pane (CTRL+3).
2. Select **Object Level Recovery Wizard**.
3. Complete the wizard.



**Note:** LiteSpeed must be installed on the server instance you select on the Backup Source page.

## Review the Backup File Contents

After you read the backup file, its contents appear in the Overview tab.





The Overview tab has the following panes:

### 1. Objects Grid

The grid displays all of the restorable objects in backup file. You can filter the objects that appear in the list by selecting the appropriate options in the toolbar.



### 2. Script Preview

The script preview displays the DDL script. To script an object, right-click it in the objects grid and select **Generate DDL Script**. For tables, you can also include constraints, indexes, and triggers in the script by selecting the appropriate options in the Object Level Recovery navigation pane. You can script more than one object, and scripted objects have  beside them in the objects grid. To view all of the scripts in the script preview, click  **Select Scripted**.

**Note:** You can save, copy, print, and search the script in LiteSpeed, but you cannot edit or execute it in the script preview pane.

### 3. Table Data Preview

The table data preview displays the contents of the table. To preview a table's data, right-click the table in the objects grid and select **Preview Data**. You can only preview the data of one table at a

time, and the previewed table has  beside it in the objects grid. A table that you preview and script has  beside it.

**Tips:** For panes that have grids, you can sort, group, move, and remove the columns:



- To sort and group the results, right-click a column heading and select the appropriate options.
- To add or remove columns, right-click a column heading and select **Column Chooser**. Add a column by dragging it from the list into the column headings. Remove a column by dragging its column heading into the list.
- To move a column, drag the column heading to the new location.

## Restore Tables and Schemas

Restoring a table in the LiteSpeed UI Console restores the table's schema and data.

### To restore tables

1. Right-click a table in the objects grid and select **Recover Table**.
2. Complete the fields as necessary. Review the following for additional information:

Select Database	If you select the same server instance and database as the original table, you must use a different table name.
Temporary directory	Enter the path or click  to navigate to it. This option restores the table and is generally used when the default directory does not have enough free disk space. <b>Note:</b> You can specify the default temp directory using the TempPath parameter in the [LiteSpeed] section of the LiteSpeedSettings.ini file. (Usually, <i>C:\Documents and Settings\All Users\Application Data\Quest Software\LiteSpeed\SQLServer\LiteSpeedSettings.ini</i> .)
Ship directory	Enter the path or click  to navigate to it. This option stores the object at the directory but doesn't restore it.
Filegroup	Select the filegroup. This option associates the restored object with the filegroup.

### To restore the schema only

- » Select one or more objects in the objects grid and select **Recover Schema**.

**Tips:**

- CTRL-click objects in the grid to select multiple objects that are not next to each other in the grid.
- In scenarios, when you do not require any transaction log backups and the tail log backup, you can select to bypass tail log processing. Object Level Recovery operations may work much faster in this case. You can enable and disable this option from the toolbar and when reading a backup.

## Object Level Restores from TSM Backups

You cannot do object level restores directly from a TSM backup, because TSM does not allow for randomly accessing the data.

*To work around this issue*

1. Do one of the following:
  - Convert a TSM backup to LiteSpeed disk backup. See "Recast LiteSpeed Backups" (page 121) for more information.
  - Extract a TSM backup to disk as a native uncompressed backup, using the extractor tool. See "Convert LiteSpeed Backups to SQL Server Backups" (page 129) for more information.

**Note:** To be able to extract TSM backups you need the Extractor tool delivered with LiteSpeed 5.2 or higher. This tool is fully backward compatible.

2. Restore objects from the converted or native backup files using the Object Level Recovery tool.

## Execute SELECT Statements

Before you can recover objects or execute a SELECT statement, you must read the backup file to create an index of restorable objects. The index is an .lsm file that is saved in the same directory as the backup file.

**Notes:**

- You cannot restore objects directly from TSM files or tape backups. See "Object Level Restores from TSM Backups" (page 83) for more information.
- Object Level Recovery does not support SQL Server 2008 Transparent Data Encryption (TDE).
- LiteSpeed may take a long time to read the backup file for large databases, often with little response in the LiteSpeed UI Console. To prevent this, select the **Optimize Object Level Recovery speed** option on the Backup wizard Options page to create the index during the backup process.

- Objects are recovered as they existed at the time they were backed up. You cannot recover data to a random point in time.

#### *To run the Object Level Recovery wizard*

1. Select the Object Level Recovery pane (CTRL+3).
2. Select **Object Level Recovery Wizard**.
3. Complete the wizard.

**Note:** LiteSpeed must be installed on the server instance you select on the Backup Source page.

#### *To execute a SELECT statement*

1. Select the Script Execution tab.
2. Enter the statement.
3. Click **Execute**.

**Note:** Be sure to use fully qualified names when you write a select statement.

## Supported SELECT Statements

LiteSpeed only supports a small subset of the possible T-SQL SELECT statements. In addition, it does not support computed columns and OUTER JOIN.

LiteSpeed supports the following syntax to execute SELECT statements against a backup file:

```
SELECT statement ::=
SELECT <select_list>
FROM <select_source>
[ WHERE <search_condition> ]

<select_list> ::=
<select_item>
| <select_list> , <select_item>

<select_item> ::=
column
| column_wild
| column alias
| alias = column

<select_source> ::=
<table_source>
| <select_source> , <table_source>
| <select_source> JOIN <table_source> ON <search_condition>
| <select_source> INNER JOIN <table_source> ON <search_condition>

<table_source> ::=
table
```

```
| table alias
| table AS alias

<search_condition> ::=
{ [ NOT ] <predicate> | ( <search_condition> ) }
[ { AND | OR } [ NOT ] { <predicate> | ( <search_condition> ) } ][,...n]

<predicate> ::=
<expression> { = | > | < | >= | <= | <> | !< | != | !> } <expression>
| expression [ NOT ] LIKE string_constant [ ESCAPE 'escape_char' ]
| expression [ NOT ] BETWEEN expression AND expression
| expression IS [ NOT ] NULL

<expression> ::=
constant
| column
```

## View Activity and History

### View Activity and History in Backup Manager

The LiteSpeed UI Console provides information about your backup processes in the Backup Manager tabs. Depending on what tree level and tab you select, you can view statistics on processes that fail or succeed, the amount of disk space you save, a list of all of the jobs for a server instance or database, and additional information.

You can view information about a category, subcategory, database, server instance, server group, or all of your server instances by selecting the appropriate level in the tree.

Tab Name	Available Level	Description
Overview	All	Displays information about the backup volume savings, successful jobs, and failed jobs. You can view information about different dates by changing the <b>Period</b> field or select the dates from a calendar.
List	Server group	Lists the instances in the server group with their display name, authentication method, SQL Server version, LiteSpeed version, and number of databases.
Databases	Server instance	Lists all of the server instance's databases with their state, recovery model, backup device, and backup destination.
LiteSpeed Activity	All	<p>Displays all activity for the selected parameters in a grid and timeline.</p> <p>You can change the parameters in the following fields:</p> <ul style="list-style-type: none"><li>• <b>Period</b>—Select the time span, or select dates from a calendar.</li><li>• <b>Report type</b>—Select the type of activity to display, such as backup or maintenance plans.</li><li>• <b>Status</b>—Select the status of the activity, such as failed or completed.</li><li>• <b>Source</b>—Select the type of backups to display, such as LiteSpeed or native SQL Server backups.</li></ul> <p><b>Note:</b> If you want to view information about native SQL Server backups and restores in the LiteSpeed UI Console, configure the local repository to be updated via a SQL Server job. Run the Instance Configuration wizard from the start menu.</p>

Tab Name	Available Level	Description
		<p>If you selected a database or server instance in the navigation pane, the tab also displays a timeline of activity. The timeline displays backups and restores from the past and those scheduled to occur in the future.</p> <p>You can hover over an item for additional information about it. Right-click in the timeline to navigate to a date, change the time scale, or rotate the name of the databases.</p>
Backup Templates	All	<p>Displays the LiteSpeed Backup templates stored in the central repository. See "Create Backup Templates" (page 31) for more information.</p> <p>In the Backup Templates tab, you can create, edit, clone, import, export and deploy templates, view the template contents, deployment details and modification history.</p> <p><b>Notes:</b></p> <ul style="list-style-type: none"> <li>The Backup Templates tab is only available, if the central repository is configured and selected for use. To edit, deploy or remove a template when the central repository is not used, click ▾ beside <b>Backup Templates</b> on the toolbar and select the appropriate option.</li> <li>The template deployment history is not exported when you export a template.</li> </ul>
Backup Browser	Server instance	<p>Displays backup directories and current backup files. You can restore the backup file or begin object level recovery from the files listed in the tab. You can also add directories, delete directories, or set a default directory.</p>
LiteSpeed Jobs	Server instance and Database	<p>Lists all scheduled jobs. You can select different job types in the <b>Jobs filter</b> field.</p> <p><b>Tip:</b> Right-click a job to edit, stop, or start it.</p>
Backup History	Database	<p>Lists all backups with their date and destination, including native SQL Server backups performed through LiteSpeed. The list also includes backups that have been replaced. You can view information about different dates by changing the <b>Period</b> field or select the dates from a calendar.</p>
Backup Analyzer	Database	<p>Analyzes different settings, such as compression level, striping, and backup destinations, to determine which settings have the best compression and duration values. See "Test Optimal Backup Settings" (page 56) for more information.</p> <p><b>Note:</b> When running the Backup Analyzer, follow these</p>

Tab Name	Available Level	Description
		<p>guidelines for the best results:</p> <p><i>Minimum:</i> 1 GB database size or 5 minute backup duration</p> <p><i>Recommendation:</i> 10 GB database size or 15 minute backup duration</p>

You can group server instances in the navigation pane tree based on their category or server group. Categories are similar to server groups, but they offer different features. See "Change Server Instance Grouping Methods" (page 22) for more information.


**Tips:** For panes that have grids, you can sort, group, move, and remove the columns:

- To sort and group the results, right-click a column heading and select the appropriate options.
- To add or remove columns, right-click a column heading and select **Column Chooser**. Add a column by dragging it from the list into the column headings. Remove a column by dragging its column heading into the list.
- To move a column, drag the column heading to the new location.
- After you refine the report criteria, you can print the results or export them to Excel. In addition, the tab has a timeline that displays the activity for the selected parameters by their date.

## View Maintenance Plans Activity and History

You can view information about the current state of existing maintenance plans and their execution history (CTRL+4).

Select a group, instance, or maintenance plan in the tree view to display the following tabs:

Tab	Description
Overview	<ul style="list-style-type: none"> <li>• At a group and instance level—Displays information about the backup volume savings, successful jobs, and failed jobs.</li> <li>• At a maintenance plan level—Displays maintenance plan latest status, name, owner, creation date, and last run date.</li> </ul> <p>You can view information about different dates by changing the <b>Period</b> field or clicking  to select the dates from a calendar.</p>
Maintenance Plans	Lists all maintenance plans for the server instance.
History	<p>Displays execution history of the maintenance plans for the server instance.</p> <p><b>Note:</b> To view execution history of every task in a subplan, configure</p>



Tab	Description
	extended logging. See "Reporting and Logging in Maintenance Plans" (page 288) for more information.
Design	Create and edit maintenance plans.

**Note:** If you receive a message that the server does not exist or access is denied, make sure the instance is registered and connected.

## Use Command-Line Interface

### About Using the Command-Line Interface

LiteSpeed allows you to perform various tasks directly from the command-line interface (CLI).

#### *To perform tasks using the CLI*

1. Change the directory until you are in the directory containing the LiteSpeed command-line utility (Usually, *C:\Program Files\Quest Software\LiteSpeed\SQL Server\Engine*).
2. Run a LiteSpeed utility with appropriate arguments.

If you want to...	Use...
Perform backup/restore tasks	SQLLiteSpeed—See "LiteSpeed Command-Line Arguments" (page 90) for more information.
Back up a database using Fast Compression technology	SLSSFastCompression—See "Fast Compression Command-Line Arguments" (page 106) for more information.
Delete old backups	SLSSmartCleanup—See "SmartCleanup Command-Line Arguments" (page 117) for more information.
Change backup options for the existing LiteSpeed backups	SLSSRecast—See "Recast LiteSpeed Backups" (page 121) for more information.
Convert LiteSpeed backups to the native SQL Server backups	Extractor—See "Convert LiteSpeed Backups to SQL Server Backups" (page 129) for more information.
Restore database objects	OLR—See "Restore Objects with the Command-Line Interface" (page 133) for more information.
View currently installed license or to register a new key	LicenseInfoCmd—See "LicenseInfoCmd Utility" (page 137) for more information.

**Note:** You must run the utility on the server instance on which you want to perform activity.

### LiteSpeed Command-Line Arguments

The LiteSpeed command-line utility (sqlritespeed.exe or sqlritespeedx32.exe) allows you to conduct LiteSpeed backups and restores directly from your operating system command-line. You

must run the utility on the server that you are backing up or restoring. You may need to use `sqlritespeedx32.exe` if you have a 32-bit SQL Server on a 64-bit operating system.

## Syntax

```
sqlritespeed.exe  
[-?]  
[  
{  
{-U login_id [-P password]} | -T  
}
```

### Connection Arguments:

```
[-S server_name [\instance_name]]  
[-T]
```

### SQL Server Operations:

```
[-B backup_operation]  
[-R restore_operation]  
[--nowrite]  
[-D db_name]  
[-f file_name]  
[-g filegroup_name]
```

### Options:

```
[-K encryption_key]  
[-t number_of_threads]  
[-I]  
[-X parameters]  
[-N file_number]  
[-A affinity ]  
[-L 0 | 1 ]  
[-W with_arguments]  
[--restoreasreadonly]  
[-n backup_name]  
[-d backup_description]  
[-p priority ]  
[--adaptivecompression option]  
[-C compression_level ]  
[-e encryption_level]  
[-Y comment ]  
[--doubleclick]  
[--attachedfile path/file]  
[--checksum]  
[--verify]
```

### Optimization Arguments:

```
[-h throttle]  
[-o buffer_count]  
[-x maximum_transfer_size]
```

### TSM Connection Options:

```
[
-j TSM_configuration_file
-i TSM_object
[-c TSM_client_node ]
[-k TSM_client_owner_password]
[-M TSM_connection_user_name]
[-J TSM_connection_password ]
[-l TSM_filespace]
[-q TSM_query]
[-a delete]
[-z TSM_management_class]
[--tsmpointtime yyyy-mm-dd hh:mm:ss]
[--tsmarchive]
]
```

#### Backup File:

```
[-F backup_device_name]
```

#### Tape Arguments:

```
[-m format_tape]
[-w rewind]
[-u unload]

]
```

## Arguments

#### Notes:

- Single-letter arguments are case-sensitive, and they can be preceded by a figure dash '-' or '/'.
- Verbose multi-letter arguments are not case-sensitive, they must be preceded by double dashes '--'.

-Argument	--Argument	Description
(none)	-- AdaptiveCompression	Automatically selects the optimal compression level based on CPU usage. See "Compression Levels" (page 54) for more information.  You can tell Adaptive Compression to optimize backups either for size or for speed. This argument accepts one of the following values: <ul style="list-style-type: none"> <li>• Size</li> <li>• Speed</li> </ul>
-A	--Affinity	Processor affinity designates specific processors to run LiteSpeed, while not allowing LiteSpeed to run on the remaining processors.  This argument accepts decimal values and

-Argument	--Argument	Description																					
		<p>hexadecimal values. If a value begins with "0x" it is interpreted as hexadecimal. A positive 64-bit integer value translates to a binary mask where a value of 1 designates the corresponding processor to be able to run the LiteSpeed process.</p> <p><b>Note:</b> 32-bit Windows is internally limited to a 32-bit mask.</p> <p>For example, you need to select processors 2, 3, and 6 for use with LiteSpeed. Number the bits from the right to left. The rightmost bit represents the first processor. Set the second, third, and sixth bits to 1 and all other bits to 0. The result is binary 100110, which is decimal 38 or hexadecimal 0x26. Review the following for additional information:</p> <table> <tr> <th>Decimal Value</th><th>Binary Bit Mask</th><th>Allow LiteSpeed Threads on Processors</th></tr> <tr> <td>0</td><td>0</td><td>All (default)</td></tr> <tr> <td>1</td><td>1</td><td>1</td></tr> <tr> <td>3</td><td>11</td><td>1 and 2</td></tr> <tr> <td>7</td><td>111</td><td>1, 2 and 3</td></tr> <tr> <td>38</td><td>100110</td><td>2, 3, and 6</td></tr> <tr> <td>205</td><td>11001101</td><td>1, 3, 4, 7, and 8</td></tr> </table> <p><b>Tip:</b> Before you start tuning the CPU Throttle or Affinity parameters to adjust backup performance, try limiting the number of threads. If you decide to use an affinity value other than default, it is recommended that you limit the threading as well. You may also want to consider using Adaptive Compression to maintain backup performance. See "Adaptive Compression" (page 55) for more information.</p>	Decimal Value	Binary Bit Mask	Allow LiteSpeed Threads on Processors	0	0	All (default)	1	1	1	3	11	1 and 2	7	111	1, 2 and 3	38	100110	2, 3, and 6	205	11001101	1, 3, 4, 7, and 8
Decimal Value	Binary Bit Mask	Allow LiteSpeed Threads on Processors																					
0	0	All (default)																					
1	1	1																					
3	11	1 and 2																					
7	111	1, 2 and 3																					
38	100110	2, 3, and 6																					
205	11001101	1, 3, 4, 7, and 8																					
(none)	--AttachedFile	Specifies filepaths to include in both backup and restore operations. The filepath can be either a single file or a directory. If it is a directory, then LiteSpeed recursively includes all files and subdirectories. All attached files are encrypted and compressed, with all pertinent backup parameters																					

-Argument	--Argument	Description
		<p>supported. This feature works for disk, tape, TSM, and Double Click Restore as well. You can supply multiple instances of this argument.</p> <p>When used within the context of a restore operation, the path parameter can be expanded to include a new destination. This form will take the syntax of &lt;file_path&gt; to &lt;new_file_path&gt;. The new filepath can be used to specify a new location but cannot rename a file.</p> <p>This argument only restores the attached files. It does not restore the database, just the files that were attached to that backup.</p> <p><b>Notes:</b></p> <ul style="list-style-type: none"> <li>• The original entire directory path need not be supplied (e.g. c: to c:\testadSattsm is allowed).</li> <li>• c:\testad to testadr would restore all files in directory c:\testad to c:\testadr.</li> </ul>
-B	--Backup	<p>Backup operation. This argument accepts one of the following values:</p> <ul style="list-style-type: none"> <li>• Database—Back up database</li> <li>• Log—Back up transaction log</li> </ul>
-d	--BackupDescription	<p>Specifies a description to store with the backup. This argument accepts variables. See "LiteSpeed Variables" (page 271) for more information.</p>
-F	--BackupFile	<p>Location and name of the backup file or device to back up or restore. You can supply multiple instances of this argument.</p> <p>This argument accepts variables. See "LiteSpeed Variables" (page 271) for more information.</p> <p>Examples:</p> <p>UNC Path: \\servername\share\path\filename</p> <p>Local path: c:\filedirectory\filename</p> <p>For TSM backups and TSM archives, this argument accepts the following formats:</p> <ul style="list-style-type: none"> <li>• tsmbkp:&lt;filespace&gt;\&lt;high&gt;\&lt;low&gt;</li> <li>• tsmarc:&lt;filespace&gt;\&lt;high&gt;\&lt;low&gt;</li> </ul>
-N	--BackupIndex	<p>Specifies the particular backup to use when</p>

-Argument	--Argument	Description
		recasting, restoring, extracting or reading from files with multiple appended backups. You can run xp_restore_headeronly to query the files contained within the backup set given by backup_file_name.
-n	--BackupName	Specifies the name of the backup set. This argument accepts variables. See "LiteSpeed Variables" (page 271) for more information.
-o	--BufferCount	Number of SQL Server buffers available for a LiteSpeed operation. The default value is 20. <b>Note:</b> LiteSpeed defaults typically result in the best performance. You should only modify advanced options after careful planning and testing.
-Y	--Comment	Appends a user comment to the backup. This argument accepts variables. See "LiteSpeed Variables" (page 271) for more information.
-C	--CompressionLevel	Specifies the compression level for the backup. Valid values are 0 through 8. 0 bypasses the compression routines. The remaining values of 1 through 8 specify compression with increasingly aggressive computation. 2 is the default value. When choosing a compression level, it is best to try various options using your equipment and data to determine the best option for your environment. Use the Backup Analyzer to test the performance of different compression levels. See "Test Optimal Backup Settings" (page 56) for more information. <b>Note:</b> If both the compression level and Adaptive Compression option are passed in, LiteSpeed will not error out and will select and use Adaptive Compression.
-D	--Database	Name of database to be backed up or restored.
(none)	--DoubleClick	Creates a Double Click Restore executable. This argument accepts one of the following values: <ul style="list-style-type: none"> <li>1—Converts a LiteSpeed backup to a Double Click Restore executable, if the size of the backup is less than 4GB. If a backup is a native SQL Server backup or it is larger than 4 GB, it creates a Double Click Restore loader in the same location.</li> </ul>

-Argument	--Argument	Description
		<ul style="list-style-type: none"> <li>2—Creates a Double Click Restore loader in the same location.</li> </ul> <p>See "Double Click Restore Executables" (page 53) for more information.</p>
(none)	--EncBackupKey	Encrypts the specified key. The encrypted key is suitable for use with -jobp in a backup operation.
(none)	--EncRestoreKey	Encrypts the specified key. The encrypted key is suitable for use with -jobp in a restore operation.
-e	--EncryptionLevel	<p>Specifies encryption level. Works in conjunction with the Key (K) parameter.</p> <p>This argument accepts one of the following values:</p> <ul style="list-style-type: none"> <li>0—40-bit RC2</li> <li>1—56 bit RC2</li> <li>2—112 bit RC2</li> <li>3—128 bit RC2</li> <li>4—168 bit 3DES</li> <li>5—128 bit RC4</li> <li>6—128 bit AES</li> <li>7—192 bit AES</li> <li>8—256 bit AES</li> </ul>
-y	--Expiration	<p>Specifies the date and time when the backup expires. LiteSpeed will not overwrite this file until expiration datetime is passed. The argument accepts one of the following formats:</p> <ul style="list-style-type: none"> <li>yyyy-mm-dd</li> <li>yyyy-mm-dd hh:mm:ss</li> </ul>
-f	--File	Specifies a logical database file used for file or filegroup backups. You can supply multiple instances of this argument.
-g	--FileGroup	<p>Specifies a database filegroup to include in the backup or restore. You can supply multiple instances of this argument.</p> <p>A filegroup backup is a single backup of all files in the filegroup and is equivalent to explicitly listing all files in the filegroup when creating the backup. Files in a filegroup backup can be restored</p>



-Argument	--Argument	Description
		individually or as a group.
-X	--IOFlags	<p>Specifies if LiteSpeed should wait and retry the read or write operation on failure. You can define retry options using the following parameters:</p> <ul style="list-style-type: none"> <li>• <b>DISK_RETRY_COUNT</b>—Specifies the number of times that a specific operation will be retried on failure. The default is 4 retries, the maximum allowed setting is 1000.</li> <li>• <b>DISK_RETRY_WAIT</b>—Specifies the number of seconds to wait immediately following a failure before retrying. The default is 15 seconds, the maximum allowed setting is 300.</li> </ul> <p><b>Note:</b> This functionality is only available for disk operations.</p> <p>See "Network Resilience" (page 58) for more information.</p>
(none)	--JobP	Specifies an encrypted key. (Similar to -K).
-K	--Key	String used to encrypt database backups and used for restore authorization. Default is No Encryption.
-L	--LogLevel	<p>Creates a log file. This argument accepts one of the following values:</p> <ul style="list-style-type: none"> <li>• 0—Logging off.</li> <li>• 1 or any odd value—Logging on. Log file is removed on success.</li> <li>• 2 or any even value—Logging on.</li> </ul> <p>The default output directory is <i>C:\Documents and Settings\All Users\Application Data\Quest Software\LiteSpeed\SQL Server\Logs</i>. To log to a different directory run this utility with the following argument: <code>--trace logpath = "path"</code>.</p> <p>See "Configure Logging in LiteSpeed" (page 285) for more information.</p>
(none)	--LSECompatible	Produces a backup that is compatible for use with LiteSpeed Engine for SQL Server. The parameter can be used whenever a new backup file is created and should only be set when backups are needed for

-Argument	--Argument	Description
		<p>cross-compatibility between the products. This switch will force modifications to internal settings such as the thread count, striping model, and encryption levels. In some cases, performance may be degraded. The parameter is ignored when appending to a backup file created without the switch.</p> <p>This argument accepts one of the following values:</p> <ul style="list-style-type: none"> <li>• 0—False</li> <li>• 1—True</li> </ul>
-x	--MaxTransferSize	<p>Specifies the data size in bytes for each transfer when communicating with SQL Server. The size can be any multiple of 64KB in the range from 64KB to 4MB. The default is 1048576 bytes.</p>
-E	--Mirror	<p>Mirrors the backup file (copies the backup to multiple locations). If you back up the primary to a set of striped files, all mirrored backups must match the primary in the number of stripes in each mirror.</p> <p>This argument accepts variables. See "LiteSpeed Variables" (page 271) for more information.</p>
(none)	--NoWrite	<p>The argument is similar to backup log xxx to disk = 'NUL'. When the backup is completed, it is not written to disk.</p> <p><b>Notes:</b></p> <ul style="list-style-type: none"> <li>• You need to supply a filename (-F). The MSDB history tables are updated with the filename specified, but the file will not get created and no IO is performed.</li> <li>• If compression or encryption parameters are specified, then the data will get compressed or encrypted before being thrown away.</li> </ul>
(none)	--OLRMap	<p>Generates a map file during a backup for object level recovery. This argument accepts one of the following values:</p> <ul style="list-style-type: none"> <li>• 0—False</li> <li>• 1—True</li> </ul>

-Argument	--Argument	Description
-I	--Overwrite	Re-initializes (overwrites and replaces) the target backup files. For TSM backups, this will create the TSM object and version the backup based on the retention policy.
-P	--Password	A user-specified password. Passwords are case-sensitive.
-p	--Priority	Specifies the priority of the LiteSpeed process compared to other processes running on the same server. This argument accepts one of the following values: <ul style="list-style-type: none"> <li>• -1—Below Normal</li> <li>• 0—Normal (Default)</li> <li>• 1—AboveNormal</li> <li>• 2—High</li> </ul>
(none)	--Read_Write_Filegroups	Specifies a partial backup, which includes the primary filegroup and any read-write secondary filegroups.
-r	--RetainDays	Specifies a number of days to retain the backup. LiteSpeed will not overwrite this file for this number of days.
-R	--Restore	Restore operation. This argument accepts one of the following values: <ul style="list-style-type: none"> <li>• AttachedFilesOnly —Restore attached files without restoring the database</li> <li>• Database—Restore database backup</li> <li>• Log—Restore log backup</li> <li>• VerifyOnly—Verify backup</li> <li>• HeaderOnly—Provide backup details</li> <li>• FileListOnly—Provide database file details</li> <li>• CheckPassword—Check password/key</li> <li>• CheckSumOnly—Checksum a backup file</li> <li>• AttachedFileNamesOnly—List names of all attached files</li> </ul>
(none)	--RestoreAsReadOnly	Instructs the restore operation to leave the database in read-only mode. Using this option, you can

-Argument	--Argument	Description
		<p>safely restore a database into a NTFS-compressed folder. This argument accepts one of the following values:</p> <ul style="list-style-type: none"> <li>• 0—False (default)</li> <li>• 1—True</li> </ul>
-S	--Server	<p>Specifies the instance of Microsoft SQL Server to connect to. This argument accepts one of the following values:</p> <ul style="list-style-type: none"> <li>• server_name</li> <li>• server_name\instance_name</li> </ul> <p>If no server is specified, the LiteSpeed command-line utility will connect to the default instance of SQL Server on the local computer.</p>
-?	--ShowHelp	Displays the syntax summary of the LiteSpeed command-line utility.
-s	--Skip	Skips normal retention checks before overwriting the backup. 0—False (default). 1—True, will overwrite the backup that has not expired.
-m	--TapeFormat	<p>Applies to tape backups and restores. Initializes the media on the device. Takes one of the following four values:</p> <ul style="list-style-type: none"> <li>• 0—No format (default)</li> <li>• 1—Quick erase</li> <li>• 2—Strict erase</li> <li>• 3—Low level format</li> </ul> <p>All the above values other than 0, format the tape in LiteSpeed format.</p>
-w	--TapeRewind	<p>Applies only to backing up and restoring tape. This argument accepts one of the following values:</p> <ul style="list-style-type: none"> <li>• 0—Leave the tape unwound (default)</li> <li>• 1—Rewind the tape after writing/reading</li> </ul>
-u	--TapeUnload	<p>Applies to tape backups and restores. This argument accepts one of the following values:</p> <ul style="list-style-type: none"> <li>• 0—Keep tape loaded (default)</li> <li>• 1—Unload and eject tape from the drive after operation</li> </ul>

-Argument	--Argument	Description
(none)	--TempDirectory	<p>Specifies a temporary directory for use with Object Level Recovery. Use this argument when the default Windows temp directory does not have enough free disk space for the restore process.</p> <p><b>Note:</b> You can specify the default temp directory using the TempPath parameter in the [LiteSpeed] section of the LiteSpeedSettings.ini file. (Usually, <i>C:\Documents and Settings\All Users\Application Data\Quest Software\LiteSpeed\SQLServer\LiteSpeedSettings.ini</i>.)</p>
-t	--Threads	<p>Determines the number of threads used for the backup. You will achieve the best results by specifying multiple threads, but the exact value depends on several factors including: processors available, affinity setting, compression level, encryption settings, IO device speed, and SQL Server responsiveness. The default is <math>n-1</math> threads, where <math>n</math> is the number of processors.</p>
-h	--Throttle	<p>Specifies the maximum CPU usage allowed. The argument accepts an integer value between 1 and 100. The default value is 100. This is the percentage of the total amount of CPU usage (across all enabled processors) available.</p> <p><b>Tip:</b> Before you start tuning the CPU Throttle or Affinity parameters to adjust backup performance, try limiting the number of threads. If you decide to use an affinity value other than default, it is recommended that you limit the threading as well. You may also want to consider using Adaptive Compression to maintain backup performance. See "Adaptive Compression" (page 55) for more information.</p>
-U	--UserId	<p>User login ID. Required if the connection type (-R) is not a trusted connection.</p> <p>Login IDs are case-sensitive.</p>
(none)	--Verify	<p>Performs a restore verification on the backup file just created (if backup was successful).</p>

-Argument	--Argument	Description
-T	--WindowsAuth	Uses a trusted connection (to the server) instead of requiring a password.
-W	--With	<p>Specifies strings that will be passed directly to SQL Server.</p> <p>Some of the accepted parameters are the following:</p> <ul style="list-style-type: none"> <li>• DIFFERENTIAL—Specifies the differential backup.</li> <li>• CHECKSUM—Causes checksums to be verified when a LiteSpeed backup is created. <b>Note:</b> When you restore a backup containing checksum, it is automatically checked. If you do not want to check the checksums during a restore, supply 'NO_CHECKSUM'.</li> <li>• CONTINUE_AFTER_ERROR—Causes the backup be executed despite encountering an invalid backup checksum.</li> <li>• COPY_ONLY—Specifies the copy-only backup.</li> <li>• KEEP_REPLICATION—Instructs the restore operation to keep the replication settings when restoring a published database to a server other than that on which it was created (used when setting up replication with log shipping).</li> <li>• MOVE—Specifies that the given logical_file_name should be moved to operating_system_file_name.</li> <li>• REPLACE—Specifies that LiteSpeed should create the specified database and its related files even if another database already exists with the same name. The existing database is deleted.</li> <li>• RECOVERY—Instructs the restore operation to roll back any uncommitted transactions. After the recovery process, the database is ready for use.</li> <li>• NORECOVERY—Instructs the restore operation to not roll back any uncommitted transactions.</li> <li>• NO_TRUNCATE—Allows backing up the</li> </ul>

-Argument	--Argument	Description
		<p>log in situations where the database is damaged.</p> <ul style="list-style-type: none"> <li>RESTRICTED_USER—When used in conjunction with recovery (another with param and the default) leaving a usable database, this restricts access for the restored database to members of the db_owner, dbcreator, or sysadmin roles.</li> <li>STATS—Displays a message each time a percentage of the activity completes. The default is 10%.</li> <li>BLOCKSIZE—Specifies the physical block size, in bytes. Supported values are: 512, 1024, 2048, 4096, 8192, 16384, 32768, and 65536 (Default).</li> <li>PASSWORD—Specifies the password for the backup set.</li> </ul>

## TSM-Specific Arguments

TSM arguments work in conjunction with the LiteSpeed arguments. See Syntax (page 91) and Examples (page 105) for more information.

-Argument	--Argument	Description
(none)	--TSMArchive	<p>Specifies to store backup as a TSM archive. This argument accepts one of the following values:</p> <ul style="list-style-type: none"> <li>0—False</li> <li>1—True</li> </ul>
-c	--TSMClientNode	<p>Specifies the TSM server LiteSpeed connects to during backups and restores. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.</p>
-k	--TSMClientOwnerPwd	<p>Specifies the TSM client owner user password. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.</p>
-j	--TSMConfigFile	<p>Specifies the TSM configuration file.</p>

-Argument	--Argument	Description
-i	--TSMFile	<p>Defines the TSM filespace, high level and low level. This argument accepts the following format:</p> <pre>tsm_filespace\tsm_high_level\tsm_low_level</pre> <p>where:</p> <ul style="list-style-type: none"> <li>• <code>tsm_filespace</code> is the logical space on the TSM server that contains a group of files. It can be the drive label name or UNC name.</li> <li>• <code>tsm_high_level</code> specifies the directory path in which the file belongs.</li> <li>• <code>tsm_low_level</code> specifies actual name of the file.</li> </ul> <p><b>Note:</b> You may only store one item the location specified by this argument. It is not possible to append an object to this location. You can use the <code>-I</code> command-line argument or <code>@init</code> to back up to a non-unique location.</p>
-l	--TSMFileSpace	<p>Specifies the TSM file space, the logical space on the TSM server. It can be the drive label name or UNC name. You can supply multiple instances of this argument.</p> <p><b>Note:</b> IBM recommends that an application client should select a unique file space; it is recommended that LiteSpeed users follow this practice with a specific file space reserved for LiteSpeed backups.</p>
-z	--TSMManagementClass	<p>Specifies the TSM management class. If not specified, LiteSpeed uses the default management class.</p>



-Argument	--Argument	Description
(none)	--TSMPointInTime	Specifies the date for restore/to filter results. If it is not passed, LiteSpeed will choose the most recent archived backup. The format is yyyy-mm-dd hh:mm:ss. <b>Note:</b> If the backup was a striped backup and the point-in-times of the various striped files are different (rare but can be different a second or so), then the most recent of the times must be chosen.
-M	--TSMUsername	The TSM username ID.
-J	--TSMUserPwd	The TSM username password. Passwords are case-sensitive.

## Examples

1. Backup the Northwind database using a trusted connection to the backup device c:\temp\Northwind.bak:

```
sqlritespeed.exe -B Database -T -D Northwind -F "C:\temp\Northwind.bak"
```

2. Backup the Northwind database, log errors (if any) to a specified directory:

```
sqlritespeed.exe -B Database -D Northwind -F "C:\temp\Northwind.bak" -L1
--trace logpath="C:\backup_logs"
```

3. Restore the Northwind database from the backup device c:\temp\Northwind.bak using variables:

```
sqlritespeed.exe -R Database -D Northwind -F "C:\temp\%D.bak"
```

4. Restore attached files only:

```
sqlritespeed.exe -R attachedfilesonly -F "C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\Backup\CapacityManagerRepository_Full_200903012353_
20091006124204.bak" -N 1 --attachedfile "'C:\temp\CProg-29-Sep-2009 19-
22-51-233.txt' to 'C:\Documents and
Settings\nphilipp\Desktop\CProg.txt'" -S "spb9771" -T
```

5. Write a database backup where each failure can be retried once after a 30 second wait:

```
sqlritespeed.exe -B database -D foo -F c:\test.bkp -X disk_retry_count=1
-X disk_retry_wait=30
```

6. Back up to TSM with the Passwordaccess Generate option:

```
sqlritespeed.exe -Bdatabase -DNorthwind -L1 -i"fsMH\nw\%D" -  
j"C:\TSM\baclient\dsm_pg.opt" -I -z"SPS_MGTD" -S"MyServer\Inst2" -  
U"sa" -P"****"
```

7. Create a differential backup of the Northwind database:

```
sqlritespeed.exe -Bdatabase -DNorthwind -i"fsMH\nw\%D-%T" -c"10.0.1.200"  
-k"password" -j"c:\program files\Tivoli\TSM\baclient\dsm.opt" -  
WDIFFERENTIAL -I -n"Northwind Diff Backup" -d"Differential Backup of  
Northwind on 4/12/2011 1:50:58 PM"
```

8. Restore the Northwind database:

```
sqlritespeed.exe -RDatabase -j"C:\TSM\baclient\dsm.opt" -  
i"fsMH\nw\Northwind" --TSMPointInTime "2011-04-12 16:57:22" -N1 -  
DNorthwind -WREPLACE -A0 -L1 -S"w2k3-22" -U"sa" -P"****"
```

## Returns

0 (success) or 1 (failure)

## Fast Compression Command-Line Arguments

The LiteSpeed command-line utility (SLSFastCompression.exe) allows you to conduct full and differential backups directly from your operating system command-line.

## Syntax

```
slsFastCompression.exe -? | --ShowHelp | --SDShowSyntax
```

### FastCompression backup options:

```
--SDBackupDirectory <path>  
[--SDForceFull  
|--SDForceDifferential  
]  
(--SDExtentsChgRatioRequireFull (<n>|<n>%)  
|--SDDiffToFullRatioRequireFull (<n>|<n>%)  
)  
[--SDCheckForFullBackup]  
[--SDFullBackupEscalation]  
[--SDElapsedDaysRequireFull <n>]  
[--SDSpecificDaysForbidFull (<day>[-<day>]),...]  
[--SDSearchAlternateBackupDirectory <path>]  
[--SDMirrorDirectory <path>...]  
[--SDAppendDifferential]
```

### FastCompression verify option:

```
--SDVerify (Full|Last|Last,Full|All),...
```

**Accepted LiteSpeed arguments:**

```
[-K encryption_key]
[-t number_of_threads]
[-I]
[-X parameters]
[-N file_number]
[-A affinity ]
[-L logging_level]
[-W with_arguments]
[-n backup_name]
[-d backup_description]
[-p priority ]
[--adaptivecompression option]
[-C compression_level ]
[-e encryption_level]
[-Y comment ]
[--attachedfile path/file]
[-h throttle]
[-o buffer_count]
[-x maximum_transfer_size]
```

**Connection options:**

```
-D <database_name>
-S <server_name>
[-U <username> -P <password> ]
```

**TSM connection options:**

```
[
-j TSM_configuration_file
[-c TSM_client_node ]
[-k TSM_client_owner_password]

[--TSMDeviceTimeoutMinutes minutes]
[-l TSM_filespace]
[-z TSM_management_class]
]
```

## Arguments

**Notes:**

- Single-letter arguments are case-sensitive, and they can be preceded by a figure dash '-' or '/'.
- Verbose multi-letter arguments are not case-sensitive, they must be preceded by double dashes '--'.

Argument	Description
--SDAppendDifferential	Appends data to an existing full backup file.
--SDBackupDirectory	Specifies a directory for the backup file.

Argument	Description
--SDCheckForFullBackup	Checks if the expected full backup exists and returns a failure message if it is not found.
-- SDDiffToFullRatioRequireFull	Specifies the last differential backup size to last full backup size ratio. When exceeding the specified ratio LiteSpeed performs a full backup. The argument accepts one of the following formats: <ul style="list-style-type: none"><li>• ".4"</li><li>• "40%"</li></ul>
--SDElapsedDaysRequireFull	Specifies the minimum number of days since last full backup required to perform full backup.
-- SDExtentsChgRatioRequireFull	Specifies the minimum amount of database changes required for the full backup. The argument accepts one of the following formats: <ul style="list-style-type: none"><li>• ".4"</li><li>• "40%"</li></ul>
--SDForceDifferential	Forces differential backup.
--SDForceFull	Forces full backup.
--SDFullBackupEscalation	<p>Causes LiteSpeed to issue a full backup, if one of the following problems is discovered in the current backup set:</p> <ul style="list-style-type: none"><li>• The full backup is missing.</li><li>• A differential backup is missing from the backup set (excludes backups automatically removed after the specified retention period).</li><li>• LSN verification fails in the backup set.</li><li>• Verify operation fails on full or differential backup.</li></ul> <p><b>Note:</b> If a problem is detected and a full backup is created through escalation, an error will be returned.</p> <p>This argument accepts one of the following values:</p> <ul style="list-style-type: none"><li>• 0—false</li><li>• 1—true</li></ul>
--SDMirrorDirectory	Specifies a directory for a mirror backup. You can supply multiple instances of this argument.
--SDSearchAlternateBackup	Specifies the directory where to search for the backup file.

Argument	Description
--SDShowSyntax	Displays the slsFastCompression syntax.
--SDSpecificDaysForbidFull	Specifies days of the week when a full backup is never performed. The argument accepts one of the following formats: <ul style="list-style-type: none"> <li>• 3—on Tuesday</li> <li>• "tu"—on Tuesday</li> <li>• "5-7"—from Thursday to Saturday</li> <li>• "m, w, su"—on Monday, Wednesday, and Sunday</li> </ul>
--SDVerify	Performs a restore verification on the backup file just created (if backup was successful). The following values direct which part of backup chain to verify: <ul style="list-style-type: none"> <li>• Full—Verifies the last full backup</li> <li>• Last—Verifies last backup performed (can be either a full or differential)</li> <li>• Full,Last—Verifies the last full backup and last differential is available</li> <li>• All—Verifies last full backup and all differentials since</li> </ul>

## Accepted LiteSpeed Arguments

The following parameters work in conjunction with the Fast Compression parameters. See Syntax (page 106) and Examples (page 116) for more information.

--Argument	-Argument	Description
--AdaptiveCompression	(none)	Automatically selects the optimal compression level based on CPU usage. See "Compression Levels" (page 54) for more information. You can tell Adaptive Compression to optimize backups either for size or for speed. This argument accepts one of the following values: <ul style="list-style-type: none"> <li>• Size</li> <li>• Speed</li> </ul>
--Affinity	-A	Processor affinity designates specific processors to run LiteSpeed, while not allowing LiteSpeed to run on the remaining processors. This argument accepts decimal values and

--Argument	-Argument	Description																					
		<p>hexadecimal values. If a value begins with "0x" it is interpreted as hexadecimal. A positive 64-bit integer value translates to a binary mask where a value of 1 designates the corresponding processor to be able to run the LiteSpeed process.</p> <p><b>Note:</b> 32-bit Windows is internally limited to a 32-bit mask.</p> <p>For example, you need to select processors 2, 3, and 6 for use with LiteSpeed. Number the bits from the right to left. The rightmost bit represents the first processor. Set the second, third, and sixth bits to 1 and all other bits to 0. The result is binary 100110, which is decimal 38 or hexadecimal 0x26. Review the following for additional information:</p> <table border="1"> <thead> <tr> <th>Decimal Value</th><th>Binary Bit Mask</th><th>Allow LiteSpeed Threads on Processors</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>All (default)</td></tr> <tr> <td>1</td><td>1</td><td>1</td></tr> <tr> <td>3</td><td>11</td><td>1 and 2</td></tr> <tr> <td>7</td><td>111</td><td>1, 2 and 3</td></tr> <tr> <td>38</td><td>100110</td><td>2, 3, and 6</td></tr> <tr> <td>205</td><td>11001101</td><td>1, 3, 4, 7, and 8</td></tr> </tbody> </table> <p><b>Tip:</b> Before you start tuning the CPU Throttle or Affinity parameters to adjust backup performance, try limiting the number of threads. If you decide to use an affinity value other than default, it is recommended that you limit the threading as well. You may also want to consider using Adaptive Compression to maintain backup performance. See "Adaptive Compression" (page 55) for more information.</p> <p>See "Configure LiteSpeed Defaults" (page 24) for more information.</p>	Decimal Value	Binary Bit Mask	Allow LiteSpeed Threads on Processors	0	0	All (default)	1	1	1	3	11	1 and 2	7	111	1, 2 and 3	38	100110	2, 3, and 6	205	11001101	1, 3, 4, 7, and 8
Decimal Value	Binary Bit Mask	Allow LiteSpeed Threads on Processors																					
0	0	All (default)																					
1	1	1																					
3	11	1 and 2																					
7	111	1, 2 and 3																					
38	100110	2, 3, and 6																					
205	11001101	1, 3, 4, 7, and 8																					
--AttachedFile	(none)	Specifies a file or directory (and all nested files and directories) to attach with this backup																					

--Argument	-Argument	Description
		operation. You can supply multiple instances of this argument..
--BackupDescription	-d	Specifies a description to store with the backup. This argument accepts variables. See "LiteSpeed Variables" (page 271) for more information.
--BackupName	-n	Specifies the name of the backup set. This argument accepts variables. See "LiteSpeed Variables" (page 271) for more information.
--BufferCount	-o	Number of SQL Server buffers available for a LiteSpeed operation. The default value is 20. <b>Note:</b> LiteSpeed defaults typically result in the best performance. You should only modify advanced options after careful planning and testing.
--Comment	-Y	Appends a user comment to the backup. This argument accepts variables. See "LiteSpeed Variables" (page 271) for more information.
--CompressionLevel	-C	Specifies the compression level for the backup. Valid values are 0 through 8. 0 bypasses the compression routines. The remaining values of 1 through 8 specify compression with increasingly aggressive computation. 2 is the default value. When choosing a compression level, it is best to try various options using your equipment and data to determine the best option for your environment. Use the Backup Analyzer to test the performance of different compression levels. See "Test Optimal Backup Settings" (page 56) for more information. <b>Note:</b> If both the compression level and Adaptive Compression option are passed in, LiteSpeed will not error out and will select and use Adaptive Compression.
--Database	-D	Name of database to be backed up or restored.
--EncryptionLevel	-e	Specifies encryption level. Works in conjunction with the Key (K) parameter. This argument accepts one of the following values:

--Argument	-Argument	Description
		<ul style="list-style-type: none"> <li>• 0—40-bit RC2</li> <li>• 1—56 bit RC2</li> <li>• 2—112 bit RC2</li> <li>• 3—128 bit RC2</li> <li>• 4—168 bit 3DES</li> <li>• 5—128 bit RC4</li> <li>• 6—128 bit AES</li> <li>• 7—192 bit AES</li> <li>• 8—256 bit AES</li> </ul>
--Expiration	-y	<p>Specifies the date and time when the backup expires. LiteSpeed will not overwrite this file until expiration datetime is passed. The argument accepts one of the following formats:</p> <ul style="list-style-type: none"> <li>• yyyy-mm-dd</li> <li>• yyyy-mm-dd hh:mm:ss</li> </ul>
--File	-f	<p>Specifies a logical database file used for file or filegroup backups. You can supply multiple instances of this argument.</p>
--FileGroup	-g	<p>Specifies a database filegroup to include in the backup or restore. You can supply multiple instances of this argument.</p> <p>A filegroup backup is a single backup of all files in the filegroup and is equivalent to explicitly listing all files in the filegroup when creating the backup. Files in a filegroup backup can be restored individually or as a group.</p>
--Ioflag	-X	<p>Specifies if LiteSpeed should wait and retry the read or write operation on failure. You can define retry options using the following parameters:</p> <ul style="list-style-type: none"> <li>• DISK_RETRY_COUNT—Specifies the number of times that a specific operation will be retried on failure. The default is 4 retries, the maximum allowed setting is 1000.</li> <li>• DISK_RETRY_WAIT—Specifies the number of seconds to wait immediately</li> </ul>



--Argument	-Argument	Description
		<p>following a failure before retrying. The default is 15 seconds, the maximum allowed setting is 300.</p> <p><b>Note:</b> This functionality is only available for disk operations.</p>
--Key	-K	String used to encrypt database backups and used for restore authorization. Default is No Encryption.
--LogLevel	-L	<p>Creates a log file. This argument accepts one of the following values:</p> <ul style="list-style-type: none"> <li>• 0—Logging off.</li> <li>• 1 or any odd value—Logging on. Log file is removed on success.</li> <li>• 2 or any even value—Logging on.</li> </ul> <p>The default output directory is <i>C:\Documents and Settings\All Users\Application Data\Quest Software\LiteSpeed\SQL Server\Logs</i>. To log to a different directory run this utility with the following argument: <code>--trace logpath = "path"</code>.</p> <p>See "Configure Logging in LiteSpeed" (page 285) for more information.</p>
--MaxTransferSize	-x	Specifies the data size in bytes for each transfer when communicating with SQL Server. The size can be any multiple of 64KB in the range from 64KB to 4MB. The default is 1048576 bytes.
--OLRMap	(none)	<p>Generates a map file during a backup for object level recovery. This argument accepts one of the following values:</p> <ul style="list-style-type: none"> <li>• 0—False</li> <li>• 1—True</li> </ul>
--Password	-P	A user-specified password. Passwords are case-sensitive.
--Priority	-p	Specifies the priority of the LiteSpeed process compared to other processes running on the same server. This argument accepts one of the following values:

--Argument	-Argument	Description
		<ul style="list-style-type: none"> <li>• -1—Below Normal</li> <li>• 0—Normal (Default)</li> <li>• 1—AboveNormal</li> <li>• 2—High</li> </ul>
--Read_Write_Filegroups	(none)	Specifies a partial backup, which includes the primary filegroup and any read-write secondary filegroups.
--RetainDays	-r	Specifies a number of days to retain the backup. LiteSpeed will not overwrite this file for this number of days.
--Server	-S	<p>Specifies the instance of Microsoft SQL Server to connect to. This argument accepts one of the following values:</p> <ul style="list-style-type: none"> <li>• server_name</li> <li>• server_name\instance_name</li> </ul> <p>If no server is specified, the LiteSpeed command-line utility will connect to the default instance of SQL Server on the local computer.</p>
--ShowHelp	-?	Displays the syntax summary of the LiteSpeed command-line utility.
--Threads	-t	Determines the number of threads used for the backup. You will achieve the best results by specifying multiple threads, but the exact value depends on several factors including: processors available, affinity setting, compression level, encryption settings, IO device speed, and SQL Server responsiveness. The default is $n-1$ threads, where $n$ is the number of processors.
--Throttle	-h	<p>Specifies the maximum CPU usage allowed. The argument accepts an integer value between 1 and 100. The default value is 100. This is the percentage of the total amount of CPU usage (across all enabled processors) available.</p> <p><b>Tip:</b> Before you start tuning the CPU Throttle or Affinity parameters to adjust backup performance, try limiting the number of threads. If you decide to use an affinity value other than default, it is recommended that you limit the threading as</p>

--Argument	-Argument	Description
		well. You may also want to consider using Adaptive Compression to maintain backup performance. See "Adaptive Compression" (page 55) for more information.
--UserId	-U	User login ID. Required if the connection type (-R) is not a trusted connection. Login IDs are case-sensitive.
--Verify	(nonr)	Performs a restore verification on the backup file just created (if backup was successful).
--With	-W	Specifies strings that will be passed directly to SQL Server in the backup/restore SQL. Some of the accepted parameters are the following: <ul style="list-style-type: none"> <li>CHECKSUM—Causes checksums to be verified when a LiteSpeed backup is created. <b>Note:</b> When you restore a backup containing checksum, it is automatically checked. If you do not want to check the checksums during a restore, supply 'NO_CHECKSUM'.</li> <li>CONTINUE_AFTER_ERROR—Causes the backup be executed despite encountering an invalid backup checksum.</li> <li>STATS—Displays a message each time a percentage of the activity completes. The default is 10%.</li> </ul>

### Accepted TSM Command-Line Arguments

The following arguments work in conjunction with the Fast Compression arguments and accepted LiteSpeed arguments. See Syntax (page 106) and Examples (page 116) for more information.

--Argument	--Argument	Description
-c	--TSMClientNode	Specifies the TSM server LiteSpeed connects to during backups and restores. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.
-k	--TSMClientOwnerPwd	Specifies the TSM client owner user

-Argument	--Argument	Description
		password. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.
-j	--TSMConfigFile	Specifies the TSM configuration file.
(none)	--TSMDeviceTimeoutMinutes	Specifies how long to wait for a TSM device. <b>Note:</b> This parameter can be set via the LiteSpeed UI Console.
(none)	--TSMdsmi-dir	DSMI_DIR path if needed.
(none)	--TSMdsmi_log	DSMI_LOG path.
-l	--TSMFileSpace	Specifies the TSM file space, the logical space on the TSM server. It can be the drive label name or UNC name. You can supply multiple instances of this argument. <b>Note:</b> IBM recommends that an application client should select a unique file space; it is recommended that LiteSpeed users follow this practice with a specific file space reserved for LiteSpeed backups.
(none)	--TSMLogname	Log name.
-z	--TSMManagementClass	Specifies the TSM management class. If not specified, LiteSpeed uses the default management class.

## Examples

1. Back up the Northwind database. Perform full backup only if the amount of database changes since the last full backup is more than 40%.

```
SLSFastCompression.exe --SDBackupDirectory "C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Backup" --SDExtentsChgRatioRequireFull ".4" -D Northwind -S SPB9771
```

2. Back up the Northwind database to multiple locations. Perform full backup only if more than 10 days have passed since last full backup. Full backup escalation option is on.

```
SLSFastCompression.exe --SDBackupDirectory "C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Backup" --SDMirrorDirectory "D:\SQLServerBackups" --SDElapsedDaysRequireFull 10 --SDFullBackupEscalation -S SPB9771 -D Northwind
```

3. Back up the Northwind database. Force full backup.

```
SLSFastCompression.exe --SDBackupDirectory "C:\Program Files\Microsoft  
SQL Server\MSSQL.1\MSSQL\Backup" --SDForceFull -D Northwind -S SPB9771
```

4. Back up to TSM using the Passwordaccess Generate option specified in the option file.  
Perform full backup only if the amount of database changes since the last full backup is more than 40%.

```
SLSFastCompression.exe -D LiteSpeedLocal -C 2 --  
SDExtentsChgRatioRequireFull ".4" --TSMFileSpace "FC" --TSMConfigFile  
"C:\Program Files\Tivoli\TSM\baclient\dsm_pg.opt" --TSMgmtClass  
"STANDARD" --tsmdevicetimeoutminutes 2 --SDFullBackupEscalation --  
SDElapsedDaysRequireFull 14 -S"W2K5_TSM" -U"sa" -P"****"
```

## Returns

0 (success) or 1 (failure)

## SmartCleanup Command-Line Arguments

The slsSmartCleanup command-line utility (slsSmartCleanUp.exe) allows you to find and remove old full, differential and transaction log backups directly from your operating system command-line.

The backup retention will never delete:

- The backup files, if there are mixed backups in the same backup file. For example, if a user performs a backup of AdventureWorks and Pubs into the same mybackups.bak backup file.
- The full backup, if there are associated differential or t-log backups in the backup set that are not eligible for cleanup.
- File/FileGroup backups
- File/FileGroup differential backups
- Partial backups
- Partial differential backups

## Syntax

```
slsSmartCleanup.exe -? | --ShowHelp | --ShowSyntax | <options>
```

### Cleanup options:

```
(--BkpRetainDays <N>  
|--BkpExpiration <date>  
)  
(--LogRetainDays <N>  
|--LogExpiration <date>  
)  
[--CopyOnlyBackups <option>]  
[--DryRun]
```

### Connection options:

```
--Database <database_name>
--Server <server_name>
[--UserName <username> --Password <password> ]
```

**TSM connection options:**

```
[-j | --TSMConfigFile <path>]
[--TSMClientNode <node>]
[--TSMClientOwnerPwd <password>]
[--TSMdsmi_dir <path>]
[--TSMdsmi_log <path>]
[--TSMLogName <name>]
```

## Arguments

**Notes:**

- Single-letter arguments are case-sensitive, and they can be preceded by a figure dash '-' or '/'.
- Verbose multi-letter arguments are not case-sensitive, they must be preceded by double dashes '--'.

-Argument	--Argument	Description
-b	--BkpRetainDays	Specifies the number of days (N). The full or differential backup must be at least N days old before it is eligible for cleanup.
-c	--BkpExpiration	Specifies the date using one of the following formats: YYYY-MM-DD YYYY-MM-DD HH:MM:SS where <ul style="list-style-type: none"> <li>• YYYY—4-digit year</li> <li>• MM—2-digit month</li> <li>• DD—2-digit day of the month</li> <li>• HH—2-digit hour using the local 24-hour clock</li> <li>• MM—2-digit minute</li> <li>• SS—2-digit second</li> </ul> To be eligible for cleanup, the full or differential backup must be older than this date.
(none)	--CopyOnlyBackups	Controls how LiteSpeed handles copy-only backups. This argument accepts one of the following values:

-Argument	--Argument	Description
		<ul style="list-style-type: none"> <li>• Default—LiteSpeed will ignore copy-only backups except on secondary replicas in AlwaysOn Availability groups, in which case it will allow deletions. This is the default behavior when the parameter is not specified.</li> <li>• Ignore—Copy-only backups are never deleted.</li> <li>• AllowDeletes—Copy-only backups are removed according to the specified retention options.</li> </ul> <p><b>Notes:</b></p> <ul style="list-style-type: none"> <li>• Transaction log backups are not considered dependent on copy-only full or copy-only tlog backups.</li> <li>• Copy-only transaction log backups will not mark other transaction log or full backups as having a dependent.</li> <li>• The values are not case-sensitive.</li> </ul>
-D	--Database	Name of database to be backed up or restored. Only backups of this database are eligible for cleanup.
-d	--DryRun	Displays backups that are to be removed (delete candidates) or kept according to the specified conditions and SmartCleanup logic. SmartCleanup does not remove any backups, if this parameter is specified.
-l	--LogRetainDays	Specifies the number of days (N). The t-log backup must be at least N days old before it is eligible for cleanup.
-k	--LogExpiration	<p>Specifies the date of one of the following formats:</p> <p>YYYY-MM-DD</p> <p>YYYY-MM-DD HH:MM:SS</p> <p>where</p> <ul style="list-style-type: none"> <li>• YYYY—4-digit year</li> <li>• MM—2-digit month</li> </ul>

-Argument	--Argument	Description
		<ul style="list-style-type: none"> <li>• DD—2-digit day of the month</li> <li>• HH—2-digit hour using the local 24-hour clock</li> <li>• MM—2-digit minute</li> <li>• SS—2-digit second</li> </ul> <p>To be eligible for cleanup, the t-log backup must be older than this date.</p>
-S	--Server	<p>Specifies the instance of Microsoft SQL Server to connect to. This argument accepts one of the following values:</p> <ul style="list-style-type: none"> <li>• server_name</li> <li>• server_name\instance_name</li> </ul> <p>If no server is specified, the LiteSpeed command-line utility will connect to the default instance of SQL Server on the local computer.</p>
-c	--TSMClientNode	Specifies the TSM server LiteSpeed connects to during backups and restores. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.
-k	--TSMClientOwnerPwd	Specifies the TSM client owner user password. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.
-j	--TSMConfigFile	Specifies the TSM configuration file.
(none)	--TSMdsmi_dir	DSMI_DIR path if needed.
(none)	--TSMdsmi_log	DSMI_LOG path.
(none)	--TSMLogName	Log name.
-U	--UserName	<p>User login ID. Required if the connection type (-R) is not a trusted connection.</p> <p>Login IDs are case-sensitive.</p>
-P	--Password	A user-specified password. Passwords are case-sensitive.
-?	--ShowHelp	Displays the syntax summary of the LiteSpeed command-line utility.
(none)	--ShowSyntax	Displays the utility syntax.



## Example

Remove the full and differential backups created before 08/21/2009 00:00:00:

```
slsSmartCleanUp.exe -c "2009-08-21 00:00:00" --Database Northwind --Server  
spb9771
```

## Returns

0 (success) or 1 (failure)

## Recast LiteSpeed Backups

The SLSRecast utility (slsrecast.exe) allows you to convert one LiteSpeed backup into another LiteSpeed backup through the command line, optionally changing encryption, compression, retention and other settings. Also, using this utility you can create disk stripe files, append several backups to one file, convert TSM objects to disk backups to restore on another machine. See "Examples" (page 128) for more information.

## Syntax

### Usage:

```
SLSRecast.exe -? | {source_options} {target_options} [options]
```

### Source Options:

```
-E|--SrcBackupFiles path  
[-N|--SrcBackupIndex n ]  
[-P|--SrcKey key ]
```

### Target Options:

```
-F|--TgtBackupFiles path  
[-K|--TgtKey key ]  
[-I|--Overwrite ]  
[-C|--CompressionLevel 0...8 ]  
[-e|--EncryptionLevel 0...8 ]  
[-y|--Expiration time ]  
[-r|--RetainDays n ]  
[ -J|--DoubleClick ]  
[ -M|--OLRMap ]
```

### Other Options:

```
[ -A|--Affinity n ]  
[ -p|--Priority -1...2 ]  
[ -h|--Throttle 1...100 ]
```

### Tape Options:

```
[-m|--TapeFormat 0...3}  
[-w|--TapeRewind ]  
[-u|--TapeUnload ]
```

**TSM Options:**

```
[-c|--TSMClientNode node_name ]  
[-k|--TSMClientOwnerPwd password ]  
[-j|--TSMConfigFile path ]  
[-z|--TSMMGmtClass class ]  
[ --TSMDeviceTimeoutMinutes n ]  
[ --TSMarchive ]  
[ --TSMdsmi_dir path ]  
[ --TSMdsmi_log path ]  
[ --TSMLogname name ]
```

Arguments

Notes:

- Single-letter arguments are case-sensitive, and they can be preceded by a figure dash '-' or '/'.
- Verbose multi-letter arguments are not case-sensitive, they must be preceded by double dashes '--'.

-Argument	--Argument	Description
-A	--Affinity	<p>Processor affinity designates specific processors to run LiteSpeed, while not allowing LiteSpeed to run on the remaining processors.</p> <p>This argument accepts decimal values and hexadecimal values. If a value begins with "0x" it is interpreted as hexadecimal. A positive 64-bit integer value translates to a binary mask where a value of 1 designates the corresponding processor to be able to run the LiteSpeed process.</p> <p><b>Note:</b> 32-bit Windows is internally limited to a 32-bit mask.</p> <p>For example, you need to select processors 2, 3, and 6 for use with LiteSpeed. Number the bits from the right to left. The rightmost bit represents the first processor. Set the second, third, and sixth bits to 1 and all other bits to 0. The result is binary 100110, which is decimal 38 or hexadecimal 0x26. Review the following for additional information:</p>

-Argument	--Argument	Description																					
		<table> <tr> <th>Decimal Value</th><th>Binary Bit Mask</th><th>Allow LiteSpeed Threads on Processors</th></tr> <tr> <td>0</td><td>0</td><td>All (default)</td></tr> <tr> <td>1</td><td>1</td><td>1</td></tr> <tr> <td>3</td><td>11</td><td>1 and 2</td></tr> <tr> <td>7</td><td>111</td><td>1, 2 and 3</td></tr> <tr> <td>38</td><td>100110</td><td>2, 3, and 6</td></tr> <tr> <td>205</td><td>11001101</td><td>1, 3, 4, 7, and 8</td></tr> </table> <p><b>Tip:</b> Before you start tuning the CPU Throttle or Affinity parameters to adjust backup performance, try limiting the number of threads. If you decide to use an affinity value other than default, it is recommended that you limit the threading as well. You may also want to consider using Adaptive Compression to maintain backup performance. See "Adaptive Compression" (page 55) for more information.</p>	Decimal Value	Binary Bit Mask	Allow LiteSpeed Threads on Processors	0	0	All (default)	1	1	1	3	11	1 and 2	7	111	1, 2 and 3	38	100110	2, 3, and 6	205	11001101	1, 3, 4, 7, and 8
Decimal Value	Binary Bit Mask	Allow LiteSpeed Threads on Processors																					
0	0	All (default)																					
1	1	1																					
3	11	1 and 2																					
7	111	1, 2 and 3																					
38	100110	2, 3, and 6																					
205	11001101	1, 3, 4, 7, and 8																					
-C	--CompressionLevel	<p>Specifies the compression level for the backup. Valid values are 0 through 8. 0 bypasses the compression routines. The remaining values of 1 through 8 specify compression with increasingly aggressive computation. 2 is the default value.</p> <p>When choosing a compression level, it is best to try various options using your equipment and data to determine the best option for your environment. Use the Backup Analyzer to test the performance of different compression levels. See "Test Optimal Backup Settings" (page 56) for more information.</p> <p><b>Note:</b> If both the compression level and Adaptive Compression option are passed in, LiteSpeed will not error out and will select and use Adaptive Compression.</p>																					
-J	--DoubleClick	Creates a Double Click Restore executable. This argument accepts one of the following values:																					

-Argument	--Argument	Description
		<ul style="list-style-type: none"> <li>• 1—Converts a LiteSpeed backup to a Double Click Restore executable, if the size of the backup is less than 4GB. If a backup is a native SQL Server backup or it is larger than 4 GB, it creates a Double Click Restore loader in the same location.</li> <li>• 2—Creates a Double Click Restore loader in the same location.</li> </ul> <p>See "Double Click Restore Executables" (page 53) for more information.</p>
-e	--EncryptionLevel	<p>Specifies encryption level. Works in conjunction with the Key (K) parameter.</p> <p>This argument accepts one of the following values:</p> <ul style="list-style-type: none"> <li>• 0—40-bit RC2</li> <li>• 1—56 bit RC2</li> <li>• 2—112 bit RC2</li> <li>• 3—128 bit RC2</li> <li>• 4—168 bit 3DES</li> <li>• 5—128 bit RC4</li> <li>• 6—128 bit AES</li> <li>• 7—192 bit AES</li> <li>• 8—256 bit AES</li> </ul>
-y	--Expiration	<p>Specifies the date and time when the backup expires. LiteSpeed will not overwrite this file until expiration datetime is passed. The argument accepts one of the following formats:</p> <ul style="list-style-type: none"> <li>• yyyy-mm-dd</li> <li>• yyyy-mm-dd hh:mm:ss</li> </ul>
-M	--OLRMap	<p>Generates a map file during a backup for object level recovery. This argument accepts one of the following values:</p> <ul style="list-style-type: none"> <li>• 0—False</li> <li>• 1—True</li> </ul>

-Argument	--Argument	Description
-I	--Overwrite	Re-initializes (overwrites and replaces) the target backup files. For TSM backups, this will create the TSM object and version the backup based on the retention policy.
-p	--Priority	Specifies the priority of the LiteSpeed process compared to other processes running on the same server. This argument accepts one of the following values: <ul style="list-style-type: none"> <li>• -1—Below Normal</li> <li>• 0—Normal (Default)</li> <li>• 1—AboveNormal</li> <li>• 2—High</li> </ul>
-r	--RetainDays	Specifies a number of days to retain the backup. LiteSpeed will not overwrite this file for this number of days.
-?	--ShowHelp	Displays the syntax summary of the LiteSpeed command-line utility.
-E	--SrcBackupFiles	Location and name of the source backup/restore file device(s). You can also specify a UNC path. For TSM backups and TSM archives, this argument accepts the following formats: <ul style="list-style-type: none"> <li>• tsmbkp:&lt;filespace&gt;\&lt;high&gt;\&lt;low&gt;</li> <li>• tsmarc:&lt;filespace&gt;\&lt;high&gt;\&lt;low&gt;</li> </ul> See "Examples" (page 128) for more information.
-N	--SrcBackupIndex	Specifies the particular backup to use when recasting, restoring, extracting or reading from files with multiple appended backups. You can run xp_restore_headeronly to query the files contained within the backup set given by backup_file_name.
-P	--SrcKey	Password/key used to decrypt backup. Passwords are case-sensitive.
-m	--TapeFormat	Initializes the media on the device. This argument only applies to tape backups. This argument accepts one of the following values: <ul style="list-style-type: none"> <li>• 0—Do not format (default)</li> <li>• 1—Quick erase</li> </ul>

-Argument	--Argument	Description
		<ul style="list-style-type: none"> <li>• 2—Strict erase (Full tape erase)</li> <li>• 3—Low level format</li> </ul> <p><b>Note:</b> Any successful format operation (values 1, 2, and 3; not all are available to all drive types) lays down a LiteSpeed tape header that will identify this tape as containing LiteSpeed backups. Using @init=1 (or -I in the command line) will not lay down a tape header.</p>
-w	--TapeRewind	<p>Applies only to backing up and restoring tape. This argument accepts one of the following values:</p> <ul style="list-style-type: none"> <li>• 0—Leave the tape unwound (default)</li> <li>• 1—Rewind the tape after writing/reading</li> </ul>
-u	--TapeUnload	<p>Applies to tape backups and restores. This argument accepts one of the following values:</p> <ul style="list-style-type: none"> <li>• 0—Keep tape loaded (default)</li> <li>• 1—Unload and eject tape from the drive after operation</li> </ul>
(none)	--TempDirectory	<p>Specifies a temporary directory for use with Object Level Recovery. Use this argument when the default Windows temp directory does not have enough free disk space for the restore process.</p> <p><b>Note:</b> You can specify the default temp directory using the TempPath parameter in the [LiteSpeed] section of the LiteSpeedSettings.ini file. (Usually, <i>C:\Documents and Settings\All Users\Application Data\Quest Software\LiteSpeed\SQLServer\LiteSpeedSettings.ini</i>.)</p>

-Argument	--Argument	Description
-h	--Throttle	<p>Specifies the maximum CPU usage allowed. The argument accepts an integer value between 1 and 100. The default value is 100. This is the percentage of the total amount of CPU usage (across all enabled processors) available.</p> <p><b>Tip:</b> Before you start tuning the CPU Throttle or Affinity parameters to adjust backup performance, try limiting the number of threads. If you decide to use an affinity value other than default, it is recommended that you limit the threading as well. You may also want to consider using Adaptive Compression to maintain backup performance. See "Adaptive Compression" (page 55) for more information.</p>
-F	--TgtBackupFiles	<p>Location and name of the target backup/restore file device(s). You can supply multiple instances of this argument.</p> <p>Examples:</p> <p>UNC Path: \\servername\share\path\filename</p> <p>Local path: c:\filedirectory\filename</p> <p><b>Note:</b> You cannot use the same location for the source and target files if you want to recast files with the same names.</p>
-K	--TgtKey	Password/key used to encrypt new backup.
(none)	--TSMarchive	Specifies to store backup as a TSM archive. This argument accepts one of the following values:0—False1—True
-c	--TSMClientNode	Specifies the TSM server LiteSpeed connects to during backups and restores. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.
-k	--TSMClientOwnerPwd	Specifies the TSM client owner user password. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.
-j	--TSMConfigFile	Specifies the TSM configuration file.
(none)	--TSMDeviceTimeoutMinutes	Maximum wait time to acquire TSM device.

-Argument	--Argument	Description
(none)	--TSMdsmi_dir	DSMI_DIR path if needed.
(none)	--TSMdsmi_log	DSMI_LOG path.
(none)	--TSMLogname	Log name.
-z	--TSMManagementClass	Specifies the TSM management class. If not specified, LiteSpeed uses the default management class.
(none)	--TSMPointInTime	Specifies the date for restore/to filter results. If it is not passed, LiteSpeed will choose the most recent archived backup. The format is yyyy-mm-dd hh:mm:ss.  <b>Note:</b> If the backup was a striped backup and the point-in-times of the various striped files are different (rare but can be different a second or so), then the most recent of the times must be chosen.

## Examples

1. Convert a backup to a Double Click Restore file:

```
Slsreicast.exe -E 1.bkp --DoubleClick 1 -F new
```

2. Convert a 4-striped backup to a single file:

```
Slsreicast.exe -E 1.bkp 2.bkp 3.bkp 4.bkp -F new.bkp
```

3. Convert a full, diff, and 2 t-log backups to a single appended file:

```
Slsreicast.exe -E full.bkp -F new.bkp
Slsreicast.exe -E diff.bkp -F new.bkp
Slsreicast.exe -E tlog.bkp -F new.bkp
Slsreicast.exe -E tlog.bkp -F new.bkp
```

4. Change compression, remove the encryption, add an OLRMap file:

```
Slsreicast.exe -E full.bkp -P password -F new.bkp -C 5 -M
```

5. Recompress a backup at the highest compression level for archival:

```
Slsreicast.exe -E old.bkp -F new.bkp -C 8
```

6. Encrypt a backup:

```
Slsreicast.exe -E old.bkp -F new.bkp -e 6 -K password
```



7. Convert a TSM backup to a disk backup and convert to a Double Click Restore executable:

```
Slsrecast.exe -j tsmconfig.opt -E tsmbkp:fs\highlevel\lowlevel -F new -J
```

8. Stripe a TSM backup to 3 disk files:

```
Slsrecast.exe -j tsmconfig.opt -E tsmbkp:fs\highlevel\lowlevel -F  
new1.bkp new2.bkp new3.bkp
```

## Returns

0 (success) or 1 (failure)

## Convert LiteSpeed Backups to SQL Server Backups

The extraction utility (extractor.exe) allows you to create MTF compliant SQL Server backup files from LiteSpeed backup files through the command-line. The devices created by the extractor utility can be restored on any SQL Server using the native RESTORE DATABASE or RESTORE LOG commands. The utility must be run on the server where the backup files are located.

To use the utility, run the command line and change the directory until you are in the LiteSpeed installation directory (Usually, *C:\Program Files\Quest Software\LiteSpeed\SQL Server*).

## Syntax

```
extractor.exe  
    [-?]  
    [  
        [-F sqllitespeed_backup_file]  
        [-E base_file_name]  
        [-N file_number]  
        [-K encryption_key]  
        [-L 0 | 1 | 2 ]  
        [-c client_node]  
        [-k owner_password]  
        [-j config_file]  
        [--TSMPointInTime date_time]  
        [-I]  
    ]
```

## Arguments

### Notes:

- Single-letter arguments are case-sensitive, and they can be preceded by a figure dash '-' or '/'.
- Verbose multi-letter arguments are not case-sensitive, they must be preceded by double dashes '--'.

-Argument	--Argument	Description
-?	(none)	Displays the syntax summary of the LiteSpeed command-line utility.
-c	--TSMClientNode	Specifies the TSM server LiteSpeed connects to during backups and restores. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.
-E base_file_name	--MTFFFile	<p>Specify the location and name of the Microsoft Tape Format (MSTF) base file.</p> <p>The extractor utility will create one backup device file for each thread used in a LiteSpeed backup.</p> <p>The extracted files containing the native SQL Server backup will have the following format: base_file_namex.</p> <p>Where:</p> <ul style="list-style-type: none"> <li>base_file_name is the specified Microsoft Tape Format base file.</li> <li>x is a number or letter that represents the sequence of the files. In case there are no additional files, the base file will not have an x appended to its name.</li> </ul> <p><b>Notes:</b></p> <ul style="list-style-type: none"> <li>You can specify a network destination for this parameter.</li> <li>You only need to specify this parameter once. The extraction utility will create all the necessary files automatically.</li> <li>You cannot tell the extraction utility to extract a different number of native SQL Server files. However, you can specify different names for the extracted files by supplying a file name with the -E parameter for each of the native SQL Server files. To see how many files extractor.exe will create, run it without this parameter.</li> <li>If a full path is not specified, the extracted files will be created in the current directory.</li> </ul>
-F	--BackupFile	<p>The name of the LiteSpeed backup device file to be extracted. This argument accepts network destinations.</p> <p>For TSM backups and TSM archives, this argument</p>

-Argument	--Argument	Description
		<p>accepts the following formats:</p> <ul style="list-style-type: none"> <li>• tsmbkp:&lt;filespace&gt;\&lt;high&gt;\&lt;low&gt;</li> <li>• tsmarc:&lt;filespace&gt;\&lt;high&gt;\&lt;low&gt;</li> </ul> <p>You can supply multiple instances of this argument.</p>
-I	--Overwrite	Re-initializes (overwrites and replaces) the target native backup files.
-j	--TSMConfigFile	Specifies the TSM configuration file.
-k	--TSMClientOwnerPwd	Specifies the TSM client owner user password. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.
(none)	--TSMPointInTime	<p>Specifies the date for restore/to filter results. If it is not passed, LiteSpeed will choose the most recent archived backup. The format is yyyy-mm-dd hh:mm:ss.</p> <p><b>Note:</b> If the backup was a striped backup and the point-in-times of the various striped files are different (rare but can be different a second or so), then the most recent of the times must be chosen.</p>
-L	(none)	<p>Creates a log file. This argument accepts one of the following values:</p> <ul style="list-style-type: none"> <li>• 0—Logging off.</li> <li>• 1 or any odd value—Logging on. Log file is removed on success.</li> <li>• 2 or any even value—Logging on.</li> </ul> <p>The default output directory is <i>C:\Documents and Settings\All Users\Application Data\Quest Software\LiteSpeed\SQL Server\Logs</i>. To log to a different directory run this utility with the following argument: <code>--trace logpath = "path"</code>. See "Configure Logging in LiteSpeed" (page 285) for more information.</p>
-K encryption_key	--Key	String used to encrypt database backups and used for restore authorization. Default is No Encryption.
-N file_number	--BackupIndex	Specifies the particular backup to use when recasting, restoring, extracting or reading from files with multiple appended backups. You can run xp_

-Argument	--Argument	Description
		restore_headeronly to query the files contained within the backup set given by backup_file_name.

## Examples

1. Extract a LiteSpeed backup to a Network Share:

```
extractor.exe -F "C:\temp\Northwind.bak" -E \\my_server\my_
share\Native.bak
```

2. Extract only the first backup in a backup set:

```
extractor.exe -F "C:\temp\Northwind.bak" -E
"C:\temp\NorthwindNative.bak" -N 1
```

3. Extract a striped LiteSpeed backup:

```
extractor.exe -F "C:\temp\LS1.bak" -F "C:\temp\LS2.bak" -F
"C:\temp\LS3.bak" -F "C:\temp\LS4.bak" -E "C:\temp\Native.bak"
```

**Note:** The number of extracted files does not have to match the number of files in a LiteSpeed backup. See "Arguments" (page 129) for more information about the -E parameter.

4. Extract a LiteSpeed backup to native SQL Server backup files with custom names.

- a. See how many destination files the extractor utility is going to create:

```
extractor.exe -F"c:\Backup\Northwind.bak"
```

- b. Specify one filename for each destination file. The number of file names you specify must match the number of files the extractor utility has returned for the LiteSpeed backup. The example below has 3 destination files:

```
extractor.exe -F"c:\Backup\Northwind.bak" -E "c:\Data\NW1.bak"
"c:\Data\NW2.bak" "c:\Data\NW3.bak"
```

5. Extract a TSM backup to native SQL Server backup:

```
extractor.exe -c"10.0.1.200" -k"password" -j"C:\Program
Files\Tivoli\TSM\baclient\dsm.opt" -F tsmbkp:fsMH\nw\test -E
"C:\temp\TestNative.bak"
```

## Returns

0 (success) or 1 (failure)

## Restore Objects with the Command-Line Interface

Object Level Recovery utility (olr.exe) allows you to restore objects from the command-line interface (CLI).

### Syntax

```
olr.exe
[-?]
or
View Contents:
    [
        {
            [-V]
            [-F backup_filename]
        }
    ]
Table Recovery:
    [
        {
            [-F backup_filename]
            [-g log_backup_file_name]
            [-h striped_log_backup_file_name]
            [-N file_number]
            [-K encryption_password]
            [-Y object_type]
            [-O object_name]
            [-E destination_server_name]
            [-S destination_database_name]
            [-T destination_table_name]
            [-R connection_type]
            [-U user_id]
            [-P Password]
            [-Q script_file_name]
        }

    [-y]
    ]
```

### Arguments

**Notes:**

- Arguments are case sensitive and can only be preceded with "-".
- Either use -H or -Q but not both.
- Either use -J or -T (with -S and -W) but not both.

Argument	Description
-?	Displays the syntax summary of the LiteSpeed command-line utility.

Argument	Description
-A user_defined_string	Ignore attached LSM if any and force complete rescan.
-B	Indicates Execute SELECT mode of operation.
-b	<p>By default, Object Level Recovery writes all of the recovered data to a file and then connects to SQL Server and issues a bulk insert command for SQL Server to bulk insert from that temp file.</p> <p>Instead you can pass in @backend='SQLNativeClient' or -b 1 from command line. That causes Object Level Recovery to use a bcp api implemented in Microsoft's sqlncli.dll to send the data straight to the sqlservr.exe process, thereby not needing any disk space.</p>
-C user_defined_string	Object name.
-D user_defined_string	Name of backup file to restore. It can be multiply specified for striping. Used for differential backups instead of full backup files.
-E destination_server_name	Name of the destination server.
-F backup_file_name	<p>Location and name of the backup file device containing the object to recover.</p> <p>Examples:</p> <p>UNC Path: \\servername\share\path\filename</p> <p>Local path: c:\filedirectory\filename</p> <p><b>Note:</b> There can be multiple files but they must be listed in the order in which they were backed up.</p>
-G user_defined_string	Destination ON filegroup name.
-g log_file_name	Specifies the name of the log backup file. You can supply multiple instances of this argument.
-H user_defined_string	SELECT Script literal text.
-h striped_log_file_name	<p>Specifies the striped log file name.</p> <p><b>Note:</b> The striped files for a given log backup must be specified before the next log backup set is specified.</p>

Argument	Description
-I user_ defined_ string	Destination TEXTIMAGE_ON filegroup name. Used to restore a BLOB (binary large object).
-J user_ defined_ string	Name of comma separated file (.csv) that is generated instead restoring into a database. This is an ad hoc solution for users want to see the restored data in Excel. You can only use this argument for text data.
-K encryption_ key	String used to encrypt database backups and used for restore authorization. Default is No Encryption.
-L user_ defined_ string	Limit restore to 1 matching row (in column=value format, may be repeated).
-N file_ number	Specifies the particular backup to use when recasting, restoring, extracting or reading from files with multiple appended backups. You can run xp_restore_headeronly to query the files contained within the backup set given by backup_file_name.
-O object_ name	Name of the object to recover. <b>Note:</b> Currently only tables. Table name must be preceded by database owner.
-P password	A user-specified password. Passwords are case-sensitive. Required if the connection type (-R) is not a trusted connection.
-Q script_ file_name	The file name that the script is output into.
-R connection_ type	The connection type to the SQL server when restoring an object to a table. This argument accepts one of the following values: <ul style="list-style-type: none"> <li>0—Use the User ID (-U) and Password (-P) specified</li> <li>1—Use a trusted connection (default)</li> </ul>
-S destination_ database_ name	Specifies the destination database when restoring an object to a table.
-T destination_ table_name	Name of the destination table. Use this when a table with the same name already exists in the destination database. LiteSpeed will not overwrite an existing table. <b>Note:</b> LiteSpeed will not overwrite a table.
-U user_id	User login ID. Required if the connection type (-R) is not a trusted connection.

Argument	Description
	Login IDs are case-sensitive.
-V user_ defined_ string	View contents.
-W user_ defined_ string	Specifies a temporary directory for use with Object Level Recovery. Use this argument when the default Windows temp directory does not have enough free disk space for the restore process.  <b>Note:</b> You can specify the default temp directory using the TempPath parameter in the [LiteSpeed] section of the LiteSpeedSettings.ini file. (Usually, <i>C:\Documents and Settings\All Users\Application Data\Quest Software\LiteSpeed\SQLServer\LiteSpeedSettings.ini</i> .)
-X user_ defined_ string	Ship directory for packaging files for subsequent restore.
-Y object_ type	This argument accepts one of the following values: <ul style="list-style-type: none"> <li>Database</li> <li>Default</li> <li>ExtendedProcedure</li> <li>Function</li> <li>Role</li> <li>Rule</li> <li>StoredProcedure</li> <li>SystemTable</li> <li>Table</li> <li>TablePrimaryKeyClustered</li> <li>TableIndexClustered</li> <li>TableConstraints</li> <li>TableIndexes</li> <li>TableForeignKeys</li> <li>Type</li> <li>User</li> <li>View</li> </ul>
-y	Instructs LiteSpeed to skip all transaction log backups and tail log processing. This argument accepts one of the following values: <ul style="list-style-type: none"> <li>0—False (Default).</li> <li>1—True. LiteSpeed will entirely ignore any transaction log backups specified and will not process the tail log. This option may improve read and recovery times.</li> </ul>
-Z user_ defined_ string	Name of file containing list of objects



## Examples

1. List the objects in the 'Northwind' database in the backup file c:\temp\northwind.bak:

```
olr.exe -V -Fc:\temp\northwind.bak
```

2. Recover the contents of the object 'Products' in the 'Northwind' database in the backup file c:\temp\northwind.bak:

```
olr.exe -Fc:\temp\northwind.bak -Odbo.Products
```

3. Recover the contents of the table 'Products' in the 'Northwind' database in the backup file c:\temp\northwind.bak to the 'Documentation' server, 'Northwind' database, 'ProductsNew' table using a trusted connection:

```
olr.exe -Fc:\temp\northwind.bak -Odbo.Products -R1 -EDocumentation -  
Snorthwind -TProductsNew
```

4. Recover the contents of the table 'Products' in the Northwind database in the backup file c:\temp\northwind.bak to the local 'spb9771' server, 'Northwind1' database, 'Products' table using custom temp directory:

```
olr.exe -Fc:\temp\northwind.bak -Odbo.Products -R1 -Espb9771 -  
SNorthwind1 -Wd:\products
```

5. Recover dbo.employees from a striped backup:

```
olr.exe -F"C:\Program Files\Microsoft SQL  
Server\MSSQL.1\MSSQL\Backup\Nothwind.bak" -g"C:\Program Files\Microsoft  
SQL Server\MSSQL.1\MSSQL\Backup\Northwind_log.bak" -h"C:\Northwind_  
log1.bak" -h"C:\SQLServerBackups\Northwind_log2.bak" -N1 -Odbo.employees  
-R1 -Espb9771 -SDatabase1 -Tdbo.employees
```

## Returns

0 (success) or 1 (failure)

## LicenseInfoCmd Utility

The LicenseInfoCmd utility allows you to license LiteSpeed from the command line.

**Note:** This utility will only register a local copy of LiteSpeed.

To use the utility, run the command line and change the directory until you are in the LiteSpeed installation directory (Usually, *C:\Program Files\Quest Software\LiteSpeed\SQL Server*).

## Syntax

```
LicenseInfoCmd.exe  
[-?]  
[-r]  
[[-s] -l license -m site_message]
```

## Examples

1. View information about the accepted parameters:

```
LicenseInfoCmd.exe -?
```

2. View information about the supplied license key

```
LicenseInfoCmd.exe -l C20TM3Q3K2HD74UDLBMHC6KYV6HZ3MQFNXZFB-123-45678-34  
-m "Trial Version"
```

3. Register the supplied license key

```
LicenseInfoCmd.exe -l C20TM3Q3K2HD74UDLBMHC6KYV6HZ3MQFNXZFB-123-45678-34  
-m "Trial Version" -s
```

**Note:** LicenseInfoCmd needs to be run from an elevated command prompt on Windows Vista/2008/7 to be able to store the license key.

# Use Extended Stored Procedures

## About Using Extended Stored Procedures

You can use extended stored procedures to perform LiteSpeed activities without using the LiteSpeed UI Console.

When you install LiteSpeed, the installer registers the LiteSpeed extended stored procedures with every instance of SQL Server selected at installation. These extended stored procedures contain a series of commands that you can execute in SQL Query Analyzer or other SQL scripting tool, such as Quest's Toad for SQL Server.

### Notes:

- You can only execute these commands on SQL Servers on which the LiteSpeed stored procedures have been registered.
- Extended stored procedures are executed against the master database.

If you want to...	Use...
Back up a database (full, differential, file, or filegroup)	xp_backup_database (page 140)
Back up a transaction log	xp_backup_log (page 156)
Back up using Fast Compression	xp_slsFastCompression (page 242)
Restore a database	xp_restore_database (page 200)
Restore transaction logs	xp_restore_log (page 224)
Restore only files attached to a backup	xp_restore_attachedfilesonly (page 194)
Validate that a file has not been corrupted	xp_restore_checksumonly (page 200)
Verify the backup without restoring it	xp_restore_verifyonly (page 234)
Check the progress of an activity	xp_slsreadprogress (page 254)
Check available memory	xp_memory_size (page 171)
List the logical file names in a backup	xp_restore_filelistonly (page 212)
View information about a stripe set	xp_restore_setinfo (page 222)
List header information for all LiteSpeed backups on a backup device	xp_restore_headeronly (page 215)

If you want to...	Use...
Restore objects other than a table	xp_objectrecovery_createscript (page 178)
List restorable objects in a backup	xp_objectrecovery_viewcontents (page 184)
Create DDL scripts	xp_objectrecovery_createscript (page 178)
Execute a SELECT statement against the backup (can be used for row-level restores)	xp_objectrecovery_executeselect (page 188)
Delete an object from a specified TSM location	xp_delete_tsmfile (page 169)
Retrieve TSM-specific information	xp_view_tsmcontents (page 261)
See available TSM management classes	xp_view_tsmmc (page 268)

#### Tips:

- To see the list of accepted arguments and data types for arguments, execute the following:  
`exec master.dbo.<procedure_name> show_help`
- To convert the script for use with the command-line utilities, execute the following:  
`exec master.dbo.<procedure_name> show_cmd, <xp_arguments>`

LiteSpeed parameters are flexible and usually do not have any hard coded length limits. Review the following for additional information:

- Path type parameters are the MAX\_PATH for any filesystem being employed.
- Backup name, database name, etc. are passed into SQL Server and have the same requirements.
- TSM has a limit on object names.
- Other than that, you are only limited by OS.

## xp\_backup\_database

Performs full, differential, file, filegroup, or transaction log backups.

### Syntax

**Note:** You can replace argument values with variables. See "LiteSpeed Variables" (page 271) for more information.

#### xp\_backup\_database (Disk or TSM)

```
EXEC master.dbo.xp_backup_database  
@database = 'database_name'  
(, @filename = 'backup_file_name') [,...n]
```

```
[, @nowrite = 0 | 1 ]
[, @desc = 'backup_description']
[, @backupname = 'backupset_name']
[, @threads = 1..32]
[, @init = 0 | 1 ]
[, @LSECompatible = 1]
[, @mirror = 'mirror_backup_file_name'] [,...n]
[, @doubleclick = 0 | 1 ]
[, ( @encryptionkey = 'encryption_key'| @jobp = 'encrypted_key' ) ]
[, @cryptlevel = 'encryption_level']
[, @read_write_filegroups = 0 | 1 ]
[, @file = 'logical_file_name'] [,...n]
[, @filegroup = 'logical_filegroup_name'] [,...n]
[, @priority = -1 | 0 | 1 | 2 ]
[, @with = 'additional_with_parameters'] [,...n]
[, ( @retaindays = 0..99999 | @expiration = 'date' ) ]
[, @logging = 0 | 1 | 2 ]
[, @olrmap = 0 | 1 ]
[, @affinity = 0..2147483648]
[, @throttle = 1..100]
[, @ioflag = 'DISK_RETRY_COUNT=n']
[, @ioflag = 'DISK_RETRY_WAIT=n']
[, @comment = 'comment']
[, @buffercount = 'buffer_count']
[, @maxtransfersize = 'maximum_transfer_size']
[, @adaptivecompression = 'speed' | 'size' ]
[, @compressionlevel = 'compression_level']
[, @attachedfile = 'pathname']
[, @tsmclientnode = 'TSM_client_node']
[, @tsmclientownerpwd = 'TSM_client_owner_password']
[, @tsmobject = 'TSM_object']
[, @tsmconfigfile = 'TSM_configuration_file']
[, @tsmmanagementclass = 'TSM_management_class']
[, @tsmarchive = 0 | 1 ]
[, @verify = 0 | 1 ]
[, @returndetails = 0 | 1 ]
```

## **xp\_backup\_database (Tape)**

```
EXEC master.dbo.xp_backup_database
@database = 'database_name'
, @filename = 'tape_device_name'
[, @desc = 'backup_description']
[, @backupname = 'backupset_name']
[, @threads = 1..32]
[, @format = 0..3]
[, @rewind = 0 | 1 ]
[, @unload = 0 | 1 ]
[, @encryptionkey = 'encryption_key']
[, @file = 'logical_file_name'] [,...n]
[, @filegroup = 'logical_filegroup_name'] [,...n]
[, @priority = -1 | 0 | 1 | 2 ]
[, @with = 'additional_with_parameters'] [,...n]
[, ( @retaindays = 0..99999 | @expiration = 'date' ) ]
[, @logging = 0 | 1 | 2 ]
```

```
[, @affinity = 0..2147483648]
[, @throttle = 1..100]
[, @comment = 'comment']
[, @buffercount = 'buffer_count']
[, @maxtransfersize = 'maximum_transfer_size']
[, @adaptivecompression = 'speed' | 'size' ]
[, @compressionlevel = 'compression_level']
[, @attachedfile = 'pathname']
[, @verify = 0 | 1 ]
[, @returndetails = 0 |1 ]
```

## Arguments

### Tips:

- To see the list of accepted arguments and data types for arguments, execute the following:  
`exec master.dbo.<procedure_name> show_help`
- To convert the script for use with the command-line utilities, execute the following:  
`exec master.dbo.<procedure_name> show_cmd, <xp_arguments>`

### @adaptivecompression

Automatically selects the optimal compression level based on CPU usage. See "Compression Levels" (page 54) for more information.

You can tell Adaptive Compression to optimize backups either for size or for speed. This argument accepts one of the following values:

- Size
- Speed

### @affinity

Processor affinity designates specific processors to run LiteSpeed, while not allowing LiteSpeed to run on the remaining processors.

This argument accepts decimal values and hexadecimal values. If a value begins with "0x" it is interpreted as hexadecimal. A positive 64-bit integer value translates to a binary mask where a value of 1 designates the corresponding processor to be able to run the LiteSpeed process.

**Note:** 32-bit Windows is internally limited to a 32-bit mask.

For example, you need to select processors 2, 3, and 6 for use with LiteSpeed. Number the bits from the right to left. The rightmost bit represents the first processor. Set the second, third, and sixth bits to 1 and all other bits to 0. The result is binary 100110, which is decimal 38 or hexadecimal 0x26. Review the following for additional information:

Decimal Value	Binary Bit Mask	Allow LiteSpeed Threads on Processors
0	0	All (default)

Decimal Value	Binary Bit Mask	Allow LiteSpeed Threads on Processors
1	1	1
3	11	1 and 2
7	111	1, 2 and 3
38	100110	2, 3, and 6
205	11001101	1, 3, 4, 7, and 8

**Tip:** Before you start tuning the CPU Throttle or Affinity parameters to adjust backup performance, try limiting the number of threads. If you decide to use an affinity value other than default, it is recommended that you limit the threading as well. You may also want to consider using Adaptive Compression to maintain backup performance. See "Adaptive Compression" (page 55) for more information.

### @attachedfile

Specifies filepaths to include in both backup and restore operations. The filepath can be either a single file or a directory. If it is a directory, then LiteSpeed recursively includes all files and subdirectories. All attached files are encrypted and compressed, with all pertinent backup parameters supported. This feature works for disk, tape, TSM, and Double Click Restore as well. You can supply multiple instances of this argument.

When used within the context of a restore operation, the path parameter can be expanded to include a new destination. This form will take the syntax of <file\_path> to <new\_file\_path>. The new filepath can be used to specify a new location but cannot rename a file.

This argument only restores the attached files. It does not restore the database, just the files that were attached to that backup.

#### Notes:

- The original entire directory path need not be supplied (e.g. c: to c:\testadSattsm is allowed).
- c:\testad to testadr would restore all files in directory c:\testad to c:\testadr.

### @backupname

Specifies the name of the backup set.

This argument accepts variables. See "LiteSpeed Variables" (page 271) for more information.

### @buffercount

Number of SQL Server buffers available for a LiteSpeed operation. The default value is 20.

### **@comment**

Appends a user comment to the backup.

This argument accepts variables. See "LiteSpeed Variables" (page 271) for more information.

### **@compressionlevel**

Specifies the compression level for the backup. Valid values are 0 through 8. 0 bypasses the compression routines. The remaining values of 1 through 8 specify compression with increasingly aggressive computation. 2 is the default value.

When choosing a compression level, it is best to try various options using your equipment and data to determine the best option for your environment. Use the Backup Analyzer to test the performance of different compression levels. See "Test Optimal Backup Settings" (page 56) for more information.

**Note:** If both the compression level and Adaptive Compression option are passed in, LiteSpeed will not error out and will select and use Adaptive Compression.

### **@cryptlevel**

Works in conjunction with the @encryptionkey parameter.

Specify the encryption level. Higher levels improve security, but they require more CPU and take longer. See "Test Optimal Backup Settings" (page 56) for more information on analyzing the best backup settings for your environment.

This argument accepts one of the following values:

- 0—40-bit RC2
- 1—56 bit RC2
- 2—112 bit RC2
- 3—128 bit RC2
- 4—168 bit 3DES
- 5—128 bit RC4
- 6—128 bit AES
- 7—192 bit AES
- 8—256 bit AES

### **@database**

Name of database to be backed up or restored.



This parameter specifies a database:

- to be backed up (xp\_backup\_database and xp\_slsSmartDiff)
- containing the transaction log to be backed up (xp\_backup\_log)
- to be restored (xp\_restore\_database and xp\_restore\_log)
- on which you wish to check the progress of an activity (xp\_slsReadProgress)
- for which you want to delete old backups (xp\_slsSmartCleanup)

If supplied as a variable (@database), this name can be specified either as a string constant (@database = database name) or as a variable of character string data type, except for the ntext or text data types.

### **@desc**

Specifies a description to store with the backup.

This argument accepts variables. See "LiteSpeed Variables" (page 271) for more information.

### **@doubleclick**

Creates a Double Click Restore executable. This argument accepts one of the following values:

- 1—Converts a LiteSpeed backup to a Double Click Restore executable, if the size of the backup is less than 4GB. If a backup is a native SQL Server backup or it is larger than 4 GB, it creates a Double Click Restore loader in the same location.
- 2—Creates a Double Click Restore loader in the same location.

See "Double Click Restore Executables" (page 53) for more information.

### **@encryptionkey**

Value used to generate the encryption key for the encryption algorithm. If you do not supply encryption\_key, then the program will not encrypt the backup.

**Caution:** When encrypting data, take care not to lose the encryption\_key; a backup cannot be restored or recovered without the original encryption key.

Example of key: 'Mypassword'

You can also specify the encryption level with @cryptlevel.

### **@expiration**

Specifies the date and time when the backup expires. LiteSpeed will not overwrite this file until expiration datetime is passed. The argument accepts one of the following formats:

- yyyy-mm-dd
- yyyy-mm-dd hh:mm:ss

### **@file**

Specifies a logical database file used for file or filegroup backups. You can supply multiple instances of this argument.

### **@filegroup**

Specifies a database filegroup to include in the backup or restore. You can supply multiple instances of this argument.

A filegroup backup is a single backup of all files in the filegroup and is equivalent to explicitly listing all files in the filegroup when creating the backup. Files in a filegroup backup can be restored individually or as a group.

### **@filename**

Location and name of the backup file or device to back up or restore. You can supply multiple instances of this argument.

This argument accepts variables. See "LiteSpeed Variables" (page 271) for more information.

### **@format**

Initializes the media on the device. This argument only applies to tape backups. This argument accepts one of the following values:

- 0—Do not format (default)
- 1—Quick erase
- 2—Strict erase (Full tape erase)
- 3—Low level format

**Note:** Any successful format operation (values 1, 2, and 3; not all are available to all drive types) lays down a LiteSpeed tape header that will identify this tape as containing LiteSpeed backups. Using @init=1 (or -I in the command line) will not lay down a tape header.

### **@init**

#### **Disk or TSM backups**

- 0—Appends the backup to an existing backup file set. For TSM backups, it results in an error if the file object already exists.
- 1—Re-initializes (overwrites and replaces) the target backup files. For TSM backups, this will create the TSM object and version the backup based on the retention policy.

### Tape backups

- 0—Appends the backup to tape.
- 1—If the tape was previously formatted by LiteSpeed, it wipes out all the backups by writing at the tape's beginning.

See also `@format`.

**Note:** 0 is the default value if you do not provide this parameter.

### `@ioflag`

Specifies if LiteSpeed should wait and retry the read or write operation on failure. You can define retry options using the following parameters:

- `DISK_RETRY_COUNT`—Specifies the number of times that a specific operation will be retried on failure. The default is 4 retries, the maximum allowed setting is 1000.
- `DISK_RETRY_WAIT`—Specifies the number of seconds to wait immediately following a failure before retrying. The default is 15 seconds, the maximum allowed setting is 300.

**Note:** This functionality is only available for disk operations.

For example, to specify a database backup where each failure can be retried once after a 30-second wait:

```
EXEC master.dbo.xp_backup_database  
@filename='c:\test.bkp'  
, @database='test'  
, @ioflag='DISK_RETRY_COUNT=1'  
, @ioflag='DISK_RETRY_WAIT=30'
```

See "Network Resilience" (page 58) for more information.

### `@jobp`

Specifies the `encrypted_key`. (Similar to `@EncryptionKey`).

You can use `xp_encrypt_backup_key` (page 171) to convert the password (`encryption_key`) for use with `@jobp`. The original password (or `encrypted_key` generated by `xp_encrypt_restore_key` (page 171)) must be used to restore a backup.

### `@logging`

Writes a log file for the operation. This argument accepts one of the following values:

- 0—Logging off.
- 1 or any odd value—Logging on. Log file is removed on success.
- 2 or any even value—Logging on.

The default output directory is *C:\Documents and Settings\All Users\Application Data\Quest Software\LiteSpeed\SQL Server\Logs*. To log to a different directory add `@Trace='logpath=path'`.

See *Configure Logging in LiteSpeed* (page 285) for information about LiteSpeed logging.

### **@LSECompatible**

Produces a backup that is compatible for use with LiteSpeed Engine for SQL Server. The parameter can be used whenever a new backup file is created and should only be set when backups are needed for cross-compatibility between the products. This switch will force modifications to internal settings such as the thread count, striping model, and encryption levels. In some cases, performance may be degraded. The parameter is ignored when appending to a backup file created without the switch.

This argument accepts one of the following values:

- 0—False
- 1—True

### **@maxtransfersize**

Specifies the data size in bytes for each transfer when communicating with SQL Server. The size can be any multiple of 64KB in the range from 64KB to 4MB. The default is *1048576* bytes.

### **@mirror**

Mirrors the backup file (copies the backup to multiple locations). If you back up the primary to a set of striped files, all mirrored backups must match the primary in the number of stripes in each mirror.

This argument accepts variables. See "LiteSpeed Variables" (page 271) for more information.

### **@nowrite**

When the backup is completed, it is not written to disk (similar to the native the SQL Native Backup commands: `backup database xxx to disk = 'NUL'` or `backup log xxx to disk = 'NUL'` command). This argument accepts one of the following values:

- 0—False
- 1—True

The MSDB history tables are updated with the file name specified, but the file will not get created and no IO is performed.

If compression or encryption parameters are specified, then the data will get compressed or encrypted before being thrown away.

### **@olrmap**

Generates a map file during a backup for object level recovery. This argument accepts one of the following values:

- 0—False
- 1—True

### **@priority**

Specifies the priority of the LiteSpeed process compared to other processes running on the same server. This argument accepts one of the following values:

- -1—Below Normal
- 0—Normal (Default)
- 1—AboveNormal
- 2—High

### **@read\_write\_filegroups**

Specifies a partial backup, which includes all the read/write files in a database: the primary filegroup, any read/write secondary filegroups, and any specified read-only files or filegroups. If the database is read-only, @read\_write\_filegroups includes only the primary filegroup.

### **@retaindays**

Specifies a number of days to retain the backup. LiteSpeed will not overwrite this file for this number of days.

### **@returndetails**

Generates a single-row result set.

- 0—False (default)
- 1—True

The result set contains the following details:

Column Name	Data Type	Description
Database	nvarchar (128)	Database name.
Operation	nvarchar (30)	Operation type: Backup or Restore.

Column Name	Data Type	Description
Threads	tinyint	The number of threads used for a LiteSpeed backup.
CompressionLevel	tinyint	Compression level used for compressing the backup. The compression level can be NULL, if backed up with Adaptive Compression.
AdaptiveCompression	nvarchar (max)	Adaptive Compression option used for compressing the backup: 'speed' or 'size'.
MaxTransferSize	int	The data size in bytes for each transfer when communicating with SQL Server. The size can be any multiple of 64 KB in the range from 64 KB to 4 MB. The default is 1048576 bytes.
BaseSize	int	The smallest chunk of memory LiteSpeed attempts to write to disk at any given time.
BufferCount	smallint	The number of SQL Server buffers available for a LiteSpeed operation.
StripeCount	smallint	Number of backup files in the stripe set.
OverlappedBuffers	tinyint	The number of buffers that any single VDI thread can use at a time.
CPUSeconds	numeric (18, 3)	Processor time used by the LiteSpeed operation.
ElapsedSeconds	numeric (18, 3)	Duration of the operation.
NativeSize	bigint	Backup size (in bytes) without LiteSpeed compression.
BackupSize	bigint	Size of the backup (in bytes).

**Tip:** In Toad, you can use Group Execute to produce a single result set for several server instances.

### @rewind

Applies only to backing up and restoring tape. This argument accepts one of the following values:

- 0—Leave the tape unwound (default)
- 1—Rewind the tape after writing/reading

### @skip

Skips normal retention checks before overwriting the backup.

- 0—False (default).
- 1—True, will overwrite the backup that has not expired.

### **@threads**

Determines the number of threads used for the backup. You will achieve the best results by specifying multiple threads, but the exact value depends on several factors including: processors available, affinity setting, compression level, encryption settings, IO device speed, and SQL Server responsiveness. The default is  $n-1$  threads, where  $n$  is the number of processors.

### **@throttle**

Specifies the maximum CPU usage allowed. The argument accepts an integer value between 1 and 100. The default value is 100. This is the percentage of the total amount of CPU usage (across all enabled processors) available.

**Tip:** Before you start tuning the CPU Throttle or Affinity parameters to adjust backup performance, try limiting the number of threads. If you decide to use an affinity value other than default, it is recommended that you limit the threading as well. You may also want to consider using Adaptive Compression to maintain backup performance. See "Adaptive Compression" (page 55) for more information.

### **@tsmarchive**

Specifies to store backup as a TSM archive. This argument accepts one of the following values:

- 0—False
- 1—True

### **@tsmclientnode**

Specifies the TSM server LiteSpeed connects to during backups and restores. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.

### **@tsmclientownerpwd**

Specifies the TSM client owner user password. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.

### **@tsmconfigfile**

Specifies the TSM configuration file.

### **@tsmmanagementclass**

Specifies the TSM management class. If not specified, LiteSpeed uses the default management class.

### **@tsmobject**

Defines the TSM filespace, high level and low level. This argument accepts the following format:

```
tsm_filespace\tsm_high_level\tsm_low_level
```

where:

- `tsm_filespace` is the logical space on the TSM server that contains a group of files. It can be the drive label name or UNC name.
- `tsm_high_level` specifies the directory path in which the file belongs.
- `tsm_low_level` specifies actual name of the file.

**Note:** You may only store one item the location specified by this argument. It is not possible to append an object to this location. You can use the `-I` command-line argument or `@init` to back up to a non-unique location.

### **@unload**

Applies to tape backups and restores. This argument accepts one of the following values:

- 0—Keep tape loaded (default)
- 1—Unload and eject tape from the drive after operation

### **@verify**

Performs a restore verification on the backup file just created (if backup was successful). This argument accepts one of the following values:

- 0—False
- 1—True

`@verify` is similar to an `xp_restore_verifyonly` call following `xp_backup_database` (or log). But if you use variables in the file names, then the caller does not need to determine what file names were chosen. See "xp\_restore\_verifyonly" (page 234) for more information.

### **@with**

Each `@with` argument should be a syntactically complete and correct statement. Please refer to the SQL Server Transact-SQL backup and restore documentation for the syntax and usage.

The supported formats are:

- `@with='PARAMETER'`
- `@with='PARAMETER="accepted_value"'`



#### Notes:

- Extended stored procedure arguments are limited to 255 characters. If you need more than 255 characters, use multiple @with arguments.
- Do not supply the @with parameter if no additional features are required.

This extended stored procedure accepts the following @with parameters:

Parameter	Description
DIFFERENTIAL	Specifies that the database or file backup should consist only of the portions of the database or file changed since the last full backup. A differential backup is usually smaller than a full backup. Use this option so that all individual log backups since the last full backup do not need to be applied.
STATS	Specifies the percentage at which SQL Server returns backup progress. It defaults to 10%.
COPY_ONLY	Specifies the copy-only backup.
CHECKSUM	Causes checksums to be verified when a LiteSpeed backup is created. <b>Note:</b> When you restore a backup containing checksum, it is automatically checked. If you do not want to check the checksums during a restore, supply 'NO_CHECKSUM' .
CONTINUE_ AFTER_ ERROR	Causes the backup be executed despite encountering an invalid backup checksum.
BLOCKSIZE	Specifies the physical block size, in bytes. Supported values are: 512, 1024, 2048, 4096, 8192, 16384, 32768, and 65536 (Default).
PASSWORD	Specifies the password for the backup set.

**Note:** During a full database or differential backup, LiteSpeed backs up enough of the transaction log to produce a consistent database when the database is restored.

## Examples

### Back Up Database with Init

```
EXEC master.dbo.xp_backup_database  
@database='MyDB'  
, @filename='C:\MSSQL\Backup\MyDB_Backup.BAK'  
, @init= 1
```

## Create Differential Backup

```
EXEC master.dbo.xp_backup_database  
@database='MyDB'  
, @filename = 'C:\MSSQL\Backup\MyDB_Backup.BAK'  
, @with = 'DIFFERENTIAL'
```

## Back Up Database with Encryption

```
EXEC master.dbo.xp_backup_database  
@database='MyDB'  
, @filename='C:\MSSQL\Backup\MyDB_Backup.BAK'  
, @init=1  
, @encryptionkey='Password'
```

## Back Up Database with Multiple Threads

```
EXEC master.dbo.xp_backup_database  
@database='MyDB'  
, @filename = 'C:\MSSQL\Backup\MyDB_Backup.BAK'  
, @threads = 3
```

## Multiple Backup Devices (Striped Backup)

```
EXEC master.dbo.xp_backup_database  
@database='MyDB'  
, @filename = 'C:\MSSQL\Backup\MyDB_Backup1.BAK'  
, @filename= 'D:\MSSQL\Backup\MyDB_Backup2.BAK'  
, @filename = 'E:\MSSQL\Backup\MyDB_Backup3.BAK'  
, @init = 1
```

## Create Filegroup Backup

```
EXEC master.dbo.xp_backup_database  
@database='MyDB'  
, @filename= 'C:\MSSQL\Backup\MyDB_Backup.BAK'  
, @filegroup = 'PRIMARY'  
, @filegroup = 'SEC'  
, @file = 'file1'  
, @init= 1
```

## Create Differential Filegroup Backup

```
exec master.dbo.xp_backup_database  
@database = 'MyDB'  
, @backupname = 'MyDB - Differential Filegroup Backup'  
, @compressionlevel = 1  
, @filegroup = 'PRIMARY'  
, @filegroup = 'SEC'  
, @filegroup = 'THRD'  
, @filename = 'C:\Program Files\Microsoft SQL  
Server\MSSQL.1\MSSQL\Backup\MyDB_200909111234_differential.bak'
```

```
, @init = 1  
, @with = N'DIFFERENTIAL'
```

### Create Partial Backup (includes the primary filegroup, all read/write secondary filegroups, and a specified read-only file)

```
EXEC master.dbo.xp_backup_database  
@database = 'MyDB'  
, @backupname = 'MyDB - Partial Backup'  
, @read_write_filegroups = 1  
, @file = 'file3_RO'  
, @filename = 'C:\Program Files\Microsoft SQL  
Server\MSSQL.1\MSSQL\Backup\MyDB.bak'
```

### Back Up Database to Tape

```
EXEC master.dbo.xp_backup_database  
@database='MyDB'  
, @filename='\\.\TAPE0'  
, @desc = 'Daily tape backup'  
, @format = 0
```

### Create TSM Archive

```
EXEC master.dbo.xp_backup_database  
@database= 'MyDB'  
, @tsmclientnode = 'ClusterGroup'  
, @tsmclientownerpwd= 'test16'  
, @tsmobject= 'SLS_Mar\MyDB\ (16) Thursday_14:14'  
, @tsmconfigfile= 'C:\Program Files\Tivoli\tsm\baclient\dsm.opt'  
, @desc='test'  
, @tsmarchive=1  
, @init=1
```

### Returns

0 (success) or non-zero (failure). Return codes represent the native error number returned from SQL Server for any errors encountered during the operation.

*To capture the output message, run the following:*

```
declare @rmsg varchar(999)  
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output  
select @rmsg
```

*To capture the output message and the result code, run the following:*

```
declare @rc int  
declare @rmsg varchar(999)  
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output,  
@resultcode=@rc output  
select @rc, @rmsg
```

**Note:** For tape backups, LiteSpeed returns the size and dataset number of the backup file. This number is used in the restore when multiple backups are sent to the same tape.

## xp\_backup\_log

Backs up a transaction log. You cannot use xp\_backup\_log to back up databases with a simple recovery model. Instead, use xp\_backup\_database.

**Note:** xp\_backup\_log does not accept the @with NO\_LOG | TRUNCATE\_ONLY parameters, and you have to back up with SQL Server to use them.

## Syntax

### Back Up Log (Disk or TSM)

```
EXEC master.dbo.xp_backup_log
@database = 'database_name'
[, @filename = 'backup_file_name'] [,...n]
[, @nowrite = 0 | 1 ]
[, @desc = 'backup_description']
[, @backupname = 'backupset_name']
[, @threads = 1..32]
[, @init = 0 | 1 ]
[, @LSECompatible = 1]
[, @mirror = 'mirror_backup_file_name'] [,...n]
[, @doubleclick = 0 | 1 ]
[, ( @encryptionkey = 'encryption_key' | @jobp = 'encrypted_key' ) ]
[, @cryptlevel = 'encryption_level']
[, @file = 'logical_file_name'] [,...n]
[, @filegroup = 'logical_filegroup_name'] [,...n]
[, @priority = -1 | 0 | 1 | 2 ]
[, @with = 'additional_with_parameters'] [,...n]
[, ( @retaindays = 0..99999 | @expiration = 'date' ) ]
[, @logging = 0 | 1 | 2 ]
[, @ioflag = 'DISK_RETRY_COUNT=n']
[, @ioflag = 'DISK_RETRY_WAIT=n']
[, @affinity = 0..2147483648]
[, @throttle = 1..100]
[, @comment = 'comment']
[, @buffercount = 'buffer_count']
[, @maxtransfersize = 'maximum_transfer_size']
[, @adaptivecompression = 'size' | 'speed' ]
[, @compressionlevel = 'compression_level']
[, @attachedfile = 'pathname']
[, @tsmclientnode = 'TSM_client_node']
[, @tsmclientownerpwd = 'TSM_client_owner_password']
[, @tsmobject = 'TSM_object']
[, @tsmconfigfile = 'TSM_configuration_file']
[, @tsmmanagementclass = 'TSM_management_class']
[, @tsmarchive = 0 | 1 ]
[, @verify = 0 | 1 ]
[, @returndetails = 0 | 1 ]
```

## Back Up Log (Tape)

```
EXEC master.dbo.xp_backup_log
@database = 'database_name'
, @filename = 'tape_device_name'
[, @desc = 'backup_description']
[, @backupname = 'backupset_name']
[, @threads = 1..32]
[, @format = 0..3]
[, @rewind = 0 | 1 ]
[, @unload = 0 | 1 ]
[, @encryptionkey = 'encryption_key']
[, @file = 'logical_file_name' ] [,...n]
[, @filegroup = 'logical_filegroup_name' ] [,...n]
[, @priority = -1 | 0 | 1 | 2 ]
[, @with = 'additional_with_parameters' ] [,...n]
[, ( @retaindays = 0..99999 | @expiration = 'date' ) ]
[, @logging = 0 | 1 | 2 ]
[, @affinity = 0..2147483648]
[, @throttle = 1..100]
[, @comment = 'comment']
[, @buffercount = 'buffer_count']
[, @maxtransfersize = 'maximum_transfer_size']
[, @adaptivecompression = 'size' | 'speed' ]
[, @compressionlevel = 'compression_level']
[, @attachedfile = 'pathname']
[, @verify = 0 | 1 ]
[, @returndetails = 0 | 1 ]
```

## Arguments

### Tips:

- To see the list of accepted arguments and data types for arguments, execute the following:  
`exec master.dbo.<procedure_name> show_help`
- To convert the script for use with the command-line utilities, execute the following:  
`exec master.dbo.<procedure_name> show_cmd, <xp_arguments>`

### @adaptivecompression

Automatically selects the optimal compression level based on CPU usage. See "Compression Levels" (page 54) for more information.

You can tell Adaptive Compression to optimize backups either for size or for speed. This argument accepts one of the following values:

- Size
- Speed

## @affinity

Processor affinity designates specific processors to run LiteSpeed, while not allowing LiteSpeed to run on the remaining processors.

This argument accepts decimal values and hexadecimal values. If a value begins with "0x" it is interpreted as hexadecimal. A positive 64-bit integer value translates to a binary mask where a value of 1 designates the corresponding processor to be able to run the LiteSpeed process.

**Note:** 32-bit Windows is internally limited to a 32-bit mask.

For example, you need to select processors 2, 3, and 6 for use with LiteSpeed. Number the bits from the right to left. The rightmost bit represents the first processor. Set the second, third, and sixth bits to 1 and all other bits to 0. The result is binary 100110, which is decimal 38 or hexadecimal 0x26. Review the following for additional information:

Decimal Value	Binary Bit Mask	Allow LiteSpeed Threads on Processors
0	0	All (default)
1	1	1
3	11	1 and 2
7	111	1, 2 and 3
38	100110	2, 3, and 6
205	11001101	1, 3, 4, 7, and 8

**Tip:** Before you start tuning the CPU Throttle or Affinity parameters to adjust backup performance, try limiting the number of threads. If you decide to use an affinity value other than default, it is recommended that you limit the threading as well. You may also want to consider using Adaptive Compression to maintain backup performance. See "Adaptive Compression" (page 55) for more information.

## @attachedfile

Specifies filepaths to include in both backup and restore operations. The filepath can be either a single file or a directory. If it is a directory, then LiteSpeed recursively includes all files and subdirectories. All attached files are encrypted and compressed, with all pertinent backup parameters supported. This feature works for disk, tape, TSM, and Double Click Restore as well. You can supply multiple instances of this argument.

When used within the context of a restore operation, the path parameter can be expanded to include a new destination. This form will take the syntax of <file\_path> to <new\_file\_path>. The new filepath can be used to specify a new location but cannot rename a file.

This argument only restores the attached files. It does not restore the database, just the files that were attached to that backup.

**Notes:**

- The original entire directory path need not be supplied (e.g. c: to c:\testadSattsm is allowed).
- c:\testad to testadr would restore all files in directory c:\testad to c:\testadr.

**@backupname**

Specifies the name of the backup set.

This argument accepts variables. See "LiteSpeed Variables" (page 271) for more information.

**@buffercount**

Number of SQL Server buffers available for a LiteSpeed operation. The default value is 20.

**@comment**

Appends a user comment to the backup.

This argument accepts variables. See "LiteSpeed Variables" (page 271) for more information.

**@compressionlevel**

Specifies the compression level for the backup. Valid values are 0 through 8. 0 bypasses the compression routines. The remaining values of 1 through 8 specify compression with increasingly aggressive computation. 2 is the default value.

When choosing a compression level, it is best to try various options using your equipment and data to determine the best option for your environment. Use the Backup Analyzer to test the performance of different compression levels. See "Test Optimal Backup Settings" (page 56) for more information.

**Note:** If both the compression level and Adaptive Compression option are passed in, LiteSpeed will not error out and will select and use Adaptive Compression.

**@cryptlevel**

Works in conjunction with the @encryptionkey parameter.

Specify the encryption level. Higher levels improve security, but they require more CPU and take longer. See "Test Optimal Backup Settings" (page 56) for more information on analyzing the best backup settings for your environment.

This argument accepts one of the following values:

- 0—40-bit RC2
- 1—56 bit RC2

- 2—112 bit RC2
- 3—128 bit RC2
- 4—168 bit 3DES
- 5—128 bit RC4
- 6—128 bit AES
- 7—192 bit AES
- 8—256 bit AES

### **@database**

Name of database to be backed up or restored.

This parameter specifies a database:

- to be backed up (xp\_backup\_database and xp\_slSmartDiff)
- containing the transaction log to be backed up (xp\_backup\_log)
- to be restored (xp\_restore\_database and xp\_restore\_log)
- on which you wish to check the progress of an activity (xp\_slReadProgress)
- for which you want to delete old backups (xp\_slSmartCleanup)

If supplied as a variable (@database), this name can be specified either as a string constant (@database = database name) or as a variable of character string data type, except for the ntext or text data types.

### **@desc**

Specifies a description to store with the backup.

This argument accepts variables. See "LiteSpeed Variables" (page 271) for more information.

### **@doubleclick**

Creates a Double Click Restore executable. This argument accepts one of the following values:

- 1—Converts a LiteSpeed backup to a Double Click Restore executable, if the size of the backup is less than 4GB. If a backup is a native SQL Server backup or it is larger than 4 GB, it creates a Double Click Restore loader in the same location.
- 2—Creates a Double Click Restore loader in the same location.

See "Double Click Restore Executables" (page 53) for more information.



### **@encryptionkey**

Value used to generate the encryption key for the encryption algorithm. If you do not supply `encryption_key`, then the program will not encrypt the backup.

**Caution:** When encrypting data, take care not to lose the `encryption_key`; a backup cannot be restored or recovered without the original encryption key.

Example of key: 'Mypassword'

You can also specify the encryption level with `@cryptlevel`.

### **@expiration**

Specifies the date and time when the backup expires. LiteSpeed will not overwrite this file until expiration datetime is passed. The argument accepts one of the following formats:

- yyyy-mm-dd
- yyyy-mm-dd hh:mm:ss

### **@file**

Specifies a logical database file used for file or filegroup backups. You can supply multiple instances of this argument.

### **@filegroup**

Specifies a database filegroup to include in the backup or restore. You can supply multiple instances of this argument.

A filegroup backup is a single backup of all files in the filegroup and is equivalent to explicitly listing all files in the filegroup when creating the backup. Files in a filegroup backup can be restored individually or as a group.

### **@filename**

Location and name of the backup file or device to back up or restore. You can supply multiple instances of this argument.

This argument accepts variables. See "LiteSpeed Variables" (page 271) for more information.

### **@format**

Initializes the media on the device. This argument only applies to tape backups. This argument accepts one of the following values:

- 0—Do not format (default)
- 1—Quick erase

- 2—Strict erase (Full tape erase)
- 3—Low level format

**Note:** Any successful format operation (values 1, 2, and 3; not all are available to all drive types) lays down a LiteSpeed tape header that will identify this tape as containing LiteSpeed backups. Using `@init=1` (or `-I` in the command line) will not lay down a tape header.

## @init

### Disk or TSM backups

- 0—Appends the backup to an existing backup file set. For TSM backups, it results in an error if the file object already exists.
- 1—Re-initializes (overwrites and replaces) the target backup files. For TSM backups, this will create the TSM object and version the backup based on the retention policy.

### Tape backups

- 0—Appends the backup to tape.
- 1—If the tape was previously formatted by LiteSpeed, it wipes out all the backups by writing at the tape's beginning.

See also `@format`.

**Note:** 0 is the default value if you do not provide this parameter.

## @ioflag

Specifies if LiteSpeed should wait and retry the read or write operation on failure. You can define retry options using the following parameters:

- `DISK_RETRY_COUNT`—Specifies the number of times that a specific operation will be retried on failure. The default is 4 retries, the maximum allowed setting is 1000.
- `DISK_RETRY_WAIT`—Specifies the number of seconds to wait immediately following a failure before retrying. The default is 15 seconds, the maximum allowed setting is 300.

**Note:** This functionality is only available for disk operations.

For example, to specify a database backup where each failure can be retried once after a 30-second wait:

```
EXEC master.dbo.xp_backup_database  
@filename='c:\test.bkp'  
, @database='test'  
, @ioflag='DISK_RETRY_COUNT=1'  
, @ioflag='DISK_RETRY_WAIT=30'
```

See "Network Resilience" (page 58) for more information.

### **@jobp**

Specifies the encrypted\_key. (Similar to @EncryptionKey).

You can use xp\_encrypt\_backup\_key (page 171) to convert the password (encryption\_key) for use with @jobp. The original password (or encrypted\_key generated by xp\_encrypt\_restore\_key (page 171)) must be used to restore a backup.

### **@logging**

Writes a log file for the operation. This argument accepts one of the following values:

- 0—Logging off.
- 1 or any odd value—Logging on. Log file is removed on success.
- 2 or any even value—Logging on.

The default output directory is *C:\Documents and Settings\All Users\Application Data\Quest Software\LiteSpeed\SQL Server\Logs*. To log to a different directory add @Trace='logpath=path'.

See Configure Logging in LiteSpeed (page 285) for information about LiteSpeed logging.

### **@LSECompatible**

Produces a backup that is compatible for use with LiteSpeed Engine for SQL Server. The parameter can be used whenever a new backup file is created and should only be set when backups are needed for cross-compatibility between the products. This switch will force modifications to internal settings such as the thread count, striping model, and encryption levels. In some cases, performance may be degraded. The parameter is ignored when appending to a backup file created without the switch.

This argument accepts one of the following values:

- 0—False
- 1—True

### **@maxtransfersize**

Specifies the data size in bytes for each transfer when communicating with SQL Server. The size can be any multiple of 64KB in the range from 64KB to 4MB. The default is 1048576 bytes.

### **@mirror**

Mirrors the backup file (copies the backup to multiple locations). If you back up the primary to a set of striped files, all mirrored backups must match the primary in the number of stripes in each mirror.

This argument accepts variables. See "LiteSpeed Variables" (page 271) for more information.

### **@nowrite**

When the backup is completed, it is not written to disk (similar to the native the SQL Native Backup commands: backup database xxx to disk = 'NUL' or backup log xxx to disk = 'NUL' command). This argument accepts one of the following values:

- 0—False
- 1—True

The MSDB history tables are updated with the file name specified, but the file will not get created and no IO is performed.

If compression or encryption parameters are specified, then the data will get compressed or encrypted before being thrown away.

### **@priority**

Specifies the priority of the LiteSpeed process compared to other processes running on the same server. This argument accepts one of the following values:

- -1—Below Normal
- 0—Normal (Default)
- 1—AboveNormal
- 2—High

### **@retaindays**

Specifies a number of days to retain the backup. LiteSpeed will not overwrite this file for this number of days.

### **@returndetails**

Generates a single-row result set.

- 0—False (default)
- 1—True

The result set contains the following details:

Column Name	Data Type	Description
Database	nvarchar (128)	Database name.

Column Name	Data Type	Description
Operation	nvarchar (30)	Operation type: Backup or Restore.
Threads	tinyint	The number of threads used for a LiteSpeed backup.
CompressionLevel	tinyint	Compression level used for compressing the backup. The compression level can be NULL, if backed up with Adaptive Compression.
AdaptiveCompression	nvarchar (max)	Adaptive Compression option used for compressing the backup: 'speed' or 'size'.
MaxTransferSize	int	The data size in bytes for each transfer when communicating with SQL Server. The size can be any multiple of 64 KB in the range from 64 KB to 4 MB. The default is 1048576 bytes.
BaseSize	int	The smallest chunk of memory LiteSpeed attempts to write to disk at any given time.
BufferCount	smallint	The number of SQL Server buffers available for a LiteSpeed operation.
StripeCount	smallint	Number of backup files in the stripe set.
OverlappedBuffers	tinyint	The number of buffers that any single VDI thread can use at a time.
CPUSeconds	numeric (18, 3)	Processor time used by the LiteSpeed operation.
ElapsedSeconds	numeric (18, 3)	Duration of the operation.
NativeSize	bigint	Backup size (in bytes) without LiteSpeed compression.
BackupSize	bigint	Size of the backup (in bytes).

**Tip:** In Toad, you can use Group Execute to produce a single result set for several server instances.

### @rewind

Applies only to backing up and restoring tape. This argument accepts one of the following values:

- 0—Leave the tape unwound (default)
- 1—Rewind the tape after writing/reading

### **@skip**

Skips normal retention checks before overwriting the backup.

- 0—False (default).
- 1—True, will overwrite the backup that has not expired.

### **@threads**

Determines the number of threads used for the backup. You will achieve the best results by specifying multiple threads, but the exact value depends on several factors including: processors available, affinity setting, compression level, encryption settings, IO device speed, and SQL Server responsiveness. The default is  $n-1$  threads, where  $n$  is the number of processors.

### **@throttle**

Specifies the maximum CPU usage allowed. The argument accepts an integer value between 1 and 100. The default value is 100. This is the percentage of the total amount of CPU usage (across all enabled processors) available.

**Tip:** Before you start tuning the CPU Throttle or Affinity parameters to adjust backup performance, try limiting the number of threads. If you decide to use an affinity value other than default, it is recommended that you limit the threading as well. You may also want to consider using Adaptive Compression to maintain backup performance. See "Adaptive Compression" (page 55) for more information.

### **@tsmarchive**

Specifies to store backup as a TSM archive. This argument accepts one of the following values:

- 0—False
- 1—True

### **@tsmclientnode**

Specifies the TSM server LiteSpeed connects to during backups and restores. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.

### **@tsmclientownerpwd**

Specifies the TSM client owner user password. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.

### **@tsmconfigfile**

Specifies the TSM configuration file.

### **@tsmmanagementclass**

Specifies the TSM management class. If not specified, LiteSpeed uses the default management class.

### **@tsmobject**

Defines the TSM filesystem, high level and low level. This argument accepts the following format:

```
tsm_filespace\tsm_high_level\tsm_low_level
```

where:

- `tsm_filespace` is the logical space on the TSM server that contains a group of files. It can be the drive label name or UNC name.
- `tsm_high_level` specifies the directory path in which the file belongs.
- `tsm_low_level` specifies actual name of the file.

**Note:** You may only store one item the location specified by this argument. It is not possible to append an object to this location. You can use the `-I` command-line argument or `@init` to back up to a non-unique location.

### **@unload**

Applies to tape backups and restores. This argument accepts one of the following values:

- 0—Keep tape loaded (default)
- 1—Unload and eject tape from the drive after operation

### **@verify**

Performs a restore verification on the backup file just created (if backup was successful). This argument accepts one of the following values:

- 0—False
- 1—True

`@verify` is similar to an `xp_restore_verifyonly` call following `xp_backup_database` (or log). But if you use variables in the file names, then the caller does not need to determine what file names were chosen. See "xp\_restore\_verifyonly" (page 234) for more information.

## @with

Each @with argument should be a syntactically complete and correct statement. Please refer to the SQL Server Transact-SQL backup and restore documentation for the syntax and usage.

The supported formats are:

- @with='PARAMETER'
- @with='PARAMETER="accepted\_value"'

### Notes:

- Extended stored procedure arguments are limited to 255 characters. If you need more than 255 characters, use multiple @with arguments.
- Do not supply the @with parameter if no additional features are required.

This extended stored procedure accepts the following @with parameters:

Parameter	Description
NO_ TRUNCATE	Allows backing up the log in situations where the database is damaged.
COPY_ ONLY	Specifies the copy-only backup.
CHECKSUM	Causes checksums to be verified when a LiteSpeed backup is created. <b>Note:</b> When you restore a backup containing checksum, it is automatically checked. If you do not want to check the checksums during a restore, supply 'NO_CHECKSUM'.
CONTINUE_ AFTER_ ERROR	Causes the backup be executed despite encountering an invalid backup checksum.
BLOCKSIZE	Specifies the physical block size, in bytes. Supported values are: 512, 1024, 2048, 4096, 8192, 16384, 32768, and 65536 (Default).
PASSWORD	Specifies the password for the backup set.

## Example

```
EXEC master.dbo.xp_backup_log  
@database = 'MyDB'  
, @filename='C:\MSSQL\Backup\MyDB_Backup.BAK'  
, @init = 1
```



## Returns

0 (success) or non-zero (failure). Return codes represent the native error number returned from SQL Server for any errors encountered during the operation.

*To capture the output message, run the following:*

```
declare @rmsg varchar(999)
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output
select @rmsg
```

*To capture the output message and the result code, run the following:*

```
declare @rc int
declare @rmsg varchar(999)
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output,
@resultcode=@rc output
select @rc, @rmsg
```

**Note:** For tape backups, LiteSpeed returns the size and dataset number of the backup file. This number is used in the restore when multiple backups are sent to the same tape.

## xp\_delete\_tsmfile

Deletes an object from a specified TSM location.

## Syntax

```
EXEC master.dbo.xp_delete_tsmfile
@tsmclientnode = 'TSM_client_node'
, @tsmclientownerpwd = 'TSM_client_owner_password'
, @tsmobject = 'TSM_object'
, @tsmconfigfile = 'TSM_configuration_file'
[, @tsmpointintime = 'date_time' ]
```

## Arguments

### Tips:

- To see the list of accepted arguments and data types for arguments, execute the following:  
`exec master.dbo.<procedure_name> show_help`
- To convert the script for use with the command-line utilities, execute the following:  
`exec master.dbo.<procedure_name> show_cmd, <xp_arguments>`

### @tsmclientnode

Specifies the TSM server LiteSpeed connects to during backups and restores. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.

### **@tsmclientownerpwd**

Specifies the TSM client owner user password. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.

### **@tsmconfigfile**

Specifies the TSM configuration file.

### **@tsmobject**

Defines the TSM filespace, high level and low level. This argument accepts the following format:

`tsm_filespace\tsm_high_level\tsm_low_level`

where:

- `tsm_filespace` is the logical space on the TSM server that contains a group of files. It can be the drive label name or UNC name.
- `tsm_high_level` specifies the directory path in which the file belongs.
- `tsm_low_level` specifies actual name of the file.

**Note:** You may only store one item the location specified by this argument. It is not possible to append an object to this location. You can use the `-I` command-line argument or `@init` to back up to a non-unique location.

### **@tsmpointintime**

Specifies the date for restore/to filter results. If it is not passed, LiteSpeed will choose the most recent archived backup. The format is yyyy-mm-dd hh:mm:ss.

**Note:** If the backup was a striped backup and the point-in-times of the various striped files are different (rare but can be different a second or so), then the most recent of the times must be chosen.

## **Returns**

0 (success) or non-zero (failure).

*To capture the output message, run the following:*

```
declare @rmsg varchar(999)
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output
select @rmsg
```

*To capture the output message and the result code, run the following:*

```
declare @rc int
declare @rmsg varchar(999)
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output,
```

```
@resultcode=@rc output  
select @rc, @rmsg
```

## xp\_encrypt\_backup\_key

The procedure encrypts a password into a value for @jobp, that is suitable for xp\_backup\_database and xp\_backup\_log procedures as an encrypted key.

### Syntax

```
EXEC master.dbo.xp_encrypt_backup_key  
@key = 'Mypassword'
```

### Results

The Results tab displays the encrypted\_key.

## xp\_encrypt\_restore\_key

The procedure encrypts a password into a value for @jobp, that is suitable for the xp\_restore\_database and xp\_restore\_log procedures as an encrypted key.

### Syntax

```
EXEC master.dbo.xp_encrypt_restore_key  
@key = 'Mypassword'
```

### Results

The Results tab displays the encrypted\_key.

## xp\_memory\_size

Extended stored procedure to return the contiguous memory availability of the system within the SQL Server process space. This is the space to be utilized for the transfer buffer for backup and restore operations.

### Syntax

```
EXEC master.dbo.xp_memory_size
```

## Results

Column Name	Data Type	Description
ContiguousSize	Int	Available contiguous memory in the SQL Server process space in bytes.

## xp\_objectrecovery

Restores a table from backup files. There are several ways to restore a table:

- Restore table to a database - Allows you to directly restore the table. If a table with the same name already exists in the destination database, LiteSpeed will not overwrite it. However, you can use @destinationtable to rename the new table and restore it to the database.
- Restore table to a ship directory - Allows you to restore the table later or on a different location.
- Restore table to a .csv file - Allows you to open the file with Excel or any other spreadsheet application recognizing .csv file format.

### Notes:

- You cannot restore objects directly from TSM files or tape backups. See "Object Level Restores from TSM Backups" (page 83) for more information.
- Object Level Recovery does not support SQL Server 2008 Transparent Data Encryption (TDE).
- LiteSpeed may take a long time to read the backup file for large databases, often with little response in the LiteSpeed UI Console. To prevent this, select the **Optimize Object Level Recovery speed** option on the Backup wizard Options page to create the index during the backup process.
- Objects are recovered as they existed at the time they were backed up. You cannot recover data to a random point in time.
- LiteSpeed restores partitioned tables as a non-partitioned table.
- LiteSpeed does not restore a table's primary key, constraints, indexes, and triggers.

To workaround this issue, complete the following steps:

1. Restore the table to a ship directory.
2. Use xp\_objectrecovery\_createscript to create scripts for the following pseudo-object types:

```
TableConstraintClustered,<object_name>  
TableIndexClustered,<object_name>  
TableConstraints,<object_name>  
TableIndexes,<object_name>  
TableForeignKeys,<object_name>  
TableTriggers,<object_name>
```

3. Append above generated scripts to the CREATE <object\_name>.sql file in the ship directory.
4. Run the above modified CREATE <object\_name>.sql, then BULK INSERT <object\_name>.sql.

## Syntax

### Restore table to a database

```
EXEC master.dbo.xp_objectrecovery  
(@filename = 'backup_file_name') [,...n]  
[, @filenumber = n]  
[, @encryptionkey = 'encryption_key']  
[( , @logfilename = 'log_file_name'  
  [, @stripedlogfilename = 'striped_log_file_name'] [,...n]  
  [, @logencryptionkey = 'log_encryption_key']  
  [, @logfilenumber = n ] ) [,...n]]  
[, @difffilename = 'diff_file_name'] [,...n]  
[, @difffilenumber = n]  
[, @diffencryptionkey = 'diff_encrypt_key']  
[, @objectname = 'object_name']  
[, @destinationdatabase = 'database_name']  
[, @destinationtable = 'dest_table_name']  
[, @destinationserver = 'dest_server_name']  
[, @tempdirectory = 'recovery_temp_dir']  
[, @onfilegroup = 'table_filegroup_name']  
[, @textimageonfilegroup = 'blob_filegroup_name']  
[, @disablelogprocessing = 0 | 1 ]
```

### Restore table to a ship directory

```
EXEC master.dbo.xp_objectrecovery  
(@filename = 'backup_file_name') [,...n]  
[, @filenumber = n]  
[, @encryptionkey = 'encryption_key']  
[( , @logfilename = 'log_file_name'  
  [, @stripedlogfilename = 'striped_log_file_name'] [,...n]  
  [, @logencryptionkey = 'log_encryption_key']  
  [, @logfilenumber = n ] ) [,...n]]  
[, @difffilename = 'diff_file_name'] [,...n]  
[, @difffilenumber = n]  
[, @diffencryptionkey = 'diff_encrypt_key']  
[, @objectname = 'object_name']  
[, @shipdirectory = 'recovery_ship_dir']  
[, @destinationtable = 'dest_table_name']
```

```
[, @onfilegroup = 'table_filegroup_name']  
[, @textimageonfilegroup = 'blob_filegroup_name']  
[, @disablelogprocessing = 0 | 1 ]
```

## Restore table to a .csv file

```
EXEC master.dbo.xp_objectrecovery  
(@filename = 'backup_file_name') [,...n]  
[, @filenumber = n]  
[, @encryptionkey = 'encryption_key']  
[( , @logfilename = 'log_file_name'  
  [, @stripedlogfilename = 'striped_log_file_name'] [,...n]  
  [, @logencryptionkey = 'log_encryption_key']  
  [, @logfilenumber = n ] ) [,...n]]  
[, @difffilename = 'diff_file_name'] [,...n]  
[, @difffilenumber = n]  
[, @diffencryptionkey = 'diff_encrypt_key']  
, @objectname = 'object_name'  
, @destinationfilename = 'csv_file_name'  
[, @disablelogprocessing = 0 | 1 ]
```

## Arguments

### Tips:

- To see the list of accepted arguments and data types for arguments, execute the following:  
`exec master.dbo.<procedure_name> show_help`
- To convert the script for use with the command-line utilities, execute the following:  
`exec master.dbo.<procedure_name> show_cmd, <xp_arguments>`

**Note:** LiteSpeed does not support the @limiter argument.

### @backend

By default, Object Level Recovery writes all of the recovered data to a file and then connects to SQL Server and issues a bulk insert command for SQL Server to bulk insert from that temp file.

Instead you can pass in @backend='SQLNativeClient' or -b 1 from command line. That causes Object Level Recovery to use a bcp api implemented in Microsoft's sqlncli.dll to send the data straight to the sqlservr.exe process, thereby not needing any disk space.

### @destinationdatabase

Specifies the destination database when restoring an object to a table.

### @destinationfilename

Name of comma separated file (.csv) that is generated instead restoring into a database. This is an ad hoc solution for users want to see the restored data in Excel. You can only use this argument for text data.

### **@destinationserver**

Name of the destination server.

### **@destinationtable**

Name of the destination table. Use this when a table with the same name already exists in the destination database. LiteSpeed will not overwrite an existing table.

### **@diffencryptionkey**

Identifies the restoring user as a valid owner of the backup file.

**Caution:** If you use the wrong encryption key, the restore will fail. If you forget the encryption key, you will not be able to restore or read the backup.

Equivalent to @encryptionkey, but used for differential backups instead of full backup files.

### **@difffilename**

Name of backup file to restore. It can be multiply specified for striping. Used for differential backups instead of full backup files.

### **@difffilenumber**

Identifies the backup file within the backup set. Equivalent to @filenumber, but used for differential backups instead of full backup files.

### **@disablelogprocessing**

Instructs LiteSpeed to skip all transaction log backups and tail log processing. This argument accepts one of the following values:

- 0—False (Default).
- 1—True. LiteSpeed will entirely ignore any transaction log backups specified and will not process the tail log. This option may improve read and recovery times.

### **@encryptionkey**

Identifies the restoring user as a valid owner of the backup file.

**Caution:** If you use the wrong encryption key, the restore will fail. If you forget the encryption key, you will not be able to restore or read the backup.

### **@filename**

Location and name of the backup file or device to back up or restore. You can supply multiple instances of this argument.

This argument accepts variables. See "LiteSpeed Variables" (page 271) for more information.

### **@filenumber**

Specifies the particular backup to use when recasting, restoring, extracting or reading from files with multiple appended backups. You can run `xp_restore_headeronly` to query the files contained within the backup set given by `backup_file_name`.

### **@logencryptionkey**

Identifies the restoring user as a valid owner of the backup file.

**Caution:** If you use the wrong encryption key, the restore will fail. If you forget the encryption key, you will not be able to restore or read the backup.

Equivalent to `@encryptionkey`, but used for transaction log backups.

### **@logfilename**

Specifies the name of the log backup file. You can supply multiple instances of this argument.

See "Syntax" (page 173) for more information.

### **@logfilenumber**

Identifies the log backup file within the backup set. Equivalent to `@filenumber`, but used for log backups.

### **@objectname**

Name of the object to recover.

### **@onfilegroup**

Filegroup with the object to restore.

### **@shipdirectory**

Name of the ship directory. Use this argument when you want to restore the object later or at a different physical location. This argument creates the following files in the ship directory:

- `CREATE <object_name>.sql`
- `BULK INSERT <object_name>.sql`
- `<object_name>.fmt`
- `<object_name>.bcp`



To restore the object, run the CREATE file first, and then run the BULK INSERT file. You will need to slightly modify the BULK INSERT file because of the .fmt and .bcp file path names.

**Tip:** You can zip the files and send them to someone else.

### **@stripedlogfilename**

Specifies the striped log file name.

**Note:** The striped files for a given log backup must be specified before the next log backup set is specified.

### **@tempdirectory**

Specifies a temporary directory for use with Object Level Recovery. Use this argument when the default Windows temp directory does not have enough free disk space for the restore process.

**Note:** You can specify the default temp directory using the TempPath parameter in the [LiteSpeed] section of the LiteSpeedSettings.ini file. (Usually, *C:\Documents and Settings\All Users\Application Data\Quest Software\LiteSpeed\SQLServer\LiteSpeedSettings.ini*.)

### **@textimageonfilegroup**

Destination TEXTIMAGE\_ON filegroup name. Used to restore a BLOB (binary large object).

## **Examples**

### **Restore a table from a full backup file into a database**

```
EXEC master.dbo.xp_objectrecovery
@filename = 'C:\MSSQL\Backup\MyDB_Backup.BAK'
, @objectname = 'dbo.Customers'
, @destinationdatabase = 'tempdb'
```

### **Restore a table from a full backup file to a database using table, server, filegroup and temp directory parameters**

```
EXEC master.dbo.xp_objectrecovery
@filename = 'C:\MSSQL\Backup\MyDB_Backup.BAK'
, @objectname = 'dbo.Customers'
, @destinationdatabase = 'tempdb'
, @destinationtable = 'dbo.Restored_Customers'
, @destinationserver = 'MyMachine\SQL2000'
, @tempdirectory = 'D:\temp'
, @onfilegroup = 'Secondary'
, @textimageonfilegroup = 'Secondary'
```

### **Restore a table from a striped backup**

```
EXEC master.dbo.xp_objectrecovery
@filename='C:\TestScriptBackups\full_Mon'
, @filename='C:\TestScriptBackups\'
```

```
, @filenumber=1
, @encryptionkey = 'key'
, @logfilename='C:\TestScriptBackups\Log_Mon_0900_1'
, @logencryptionkey = 'key'
, @stripedlogfilename='C:\TestScriptBackups\Log_Mon_0900_2'
, @stripedlogfilename='C:\TestScriptBackups\Log_Mon_0900_3'
, @logfilenumber=1
, @destinationtable='dbo.employees_recovered'
, @destinationdatabase='OLRRegressionTest'
, @objectname='dbo.employees'
```

### Restore a table from a full backup file to a ship directory

```
EXEC master.dbo.xp_objectrecovery
@filename = 'C:\MSSQL\Backup\MyDB_Backup.BAK'
, @objectname = 'dbo.Customers'
, @shipdirectory = 'C:\temp\ship'
```

### Restore a table from a full backup file to a .csv file

```
EXEC master.dbo.xp_objectrecovery
@filename = 'C:\MSSQL\Backup\MyDB_Backup.BAK'
, @objectname = 'dbo.Customers'
, @destinationfilename = 'C:\temp\Customers.csv'
```

## Returns

0 (success) or non-zero (failure). Return codes represent the native error number returned from SQL Server for any errors encountered during the operation.

*To capture the output message, run the following:*

```
declare @rmsg varchar(999)
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output
select @rmsg
```

*To capture the output message and the result code, run the following:*

```
declare @rc int
declare @rmsg varchar(999)
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output,
@resultcode=@rc output
select @rc, @rmsg
```

## xp\_objectrecovery\_createscript

Creates DDL scripts. You can use this extended stored procedure to restore objects other than tables by generating the DDL scripts and then running the scripts in your native SQL Server tool (such as Management Studio).

## Syntax

```
EXEC master.dbo.xp_objectrecovery_createscript
(@filename = 'backup_file_name') [,...n]
[, @filenumber = n]
[, @encryptionkey = 'encryption_key']
[( , @logfilename = 'log_file_name'
  [, @stripedlogfilename = 'striped_log_file_name'] [,...n]
  [, @logencryptionkey = 'log_encryption_key']
  [, @logfilenumber = n ] ) [,...n]]
[, @difffilename = 'diff_file_name'] [,...n]
[, @difffilenumber = n]
[, @diffencryptionkey = 'diff_encrypt_key']
[, @objectname = 'object_name']
[, @objectfilename = 'object_file_name']
[, @scriptfilename = 'script_file_name']
[, @type = 'object_type']
[, @onfilegroup = 'table_filegroup_name']
[, @textimageonfilegroup = 'blob_filegroup_name']
[, @disablelogprocessing = 0 | 1 ]
```

## Arguments

### Tips:

- To see the list of accepted arguments and data types for arguments, execute the following:  
`exec master.dbo.<procedure_name> show_help`
- To convert the script for use with the command-line utilities, execute the following:  
`exec master.dbo.<procedure_name> show_cmd, <xp_arguments>`

### @diffencryptionkey

Identifies the restoring user as a valid owner of the backup file.

**Caution:** If you use the wrong encryption key, the restore will fail. If you forget the encryption key, you will not be able to restore or read the backup.

Equivalent to @encryptionkey, but used for differential backups instead of full backup files.

### @difffilename

Name of backup file to restore. It can be multiply specified for striping. Used for differential backups instead of full backup files.

### @difffilenumber

Identifies the backup file within the backup set. Equivalent to @filenumber, but used for differential backups instead of full backup files.

### **@disablelogprocessing**

Instructs LiteSpeed to skip all transaction log backups and tail log processing. This argument accepts one of the following values:

- 0—False (Default).
- 1—True. LiteSpeed will entirely ignore any transaction log backups specified and will not process the tail log. This option may improve read and recovery times.

### **@encryptionkey**

Identifies the restoring user as a valid owner of the backup file.

**Caution:** If you use the wrong encryption key, the restore will fail. If you forget the encryption key, you will not be able to restore or read the backup.

### **@filename**

Location and name of the backup file or device to back up or restore. You can supply multiple instances of this argument.

This argument accepts variables. See "LiteSpeed Variables" (page 271) for more information.

### **@filenumber**

Specifies the particular backup to use when recasting, restoring, extracting or reading from files with multiple appended backups. You can run `xp_restore_headeronly` to query the files contained within the backup set given by `backup_file_name`.

### **@logencryptionkey**

Identifies the restoring user as a valid owner of the backup file.

**Caution:** If you use the wrong encryption key, the restore will fail. If you forget the encryption key, you will not be able to restore or read the backup.

Equivalent to `@encryptionkey`, but used for transaction log backups.

### **@logfilename**

Specifies the name of the log backup file. You can supply multiple instances of this argument.

See "Syntax" (page 179) for more information.

### **@logfilenumber**

Identifies the log backup file within the backup set. Equivalent to `@filenumber`, but used for log backups.

### **@objectfilename**

Identifies a file that contains a list of objects. The format of this file is "ObjectType,ObjectName" per line.

**Tip:** You can use command line with -V -Y switches to generate the objects list. See "Restore Objects with the Command-Line Interface" (page 133) for more information.

### **@objectname**

Name of the object to recover.

### **@onfilegroup**

Filegroup with the object to restore.

### **@scriptfilename**

Name of the script file to save the generated SQL scripts.

### **@stripedlogfilename**

Specifies the striped log file name.

**Note:** The striped files for a given log backup must be specified before the next log backup set is specified.

### **@textimageonfilegroup**

Destination TEXTIMAGE\_ON filegroup name. Used to restore a BLOB (binary large object).

### **@type**

Specifies the type of object. If you omit this parameter the object type defaults to table, so you should use this argument to recover schema objects other than tables. This argument accepts one of the following values:

- |                       |   |                                 |
|-----------------------|---|---------------------------------|
| • All <sup>1, 3</sup> | • StoredProcedure                       | • TableIndexes <sup>2</sup>     |
| • Default             | • SystemTable                           | • TableForeignKeys <sub>2</sub> |
| • ExtendedProcedure   | • Table                                 | • TableTriggers <sup>2</sup>    |
| • Function            | • TableConstraintClustered <sub>2</sub> | • Type                          |
| • Role <sup>1</sup>   | • TableIndexClustered <sup>2</sup>      | • User <sup>1</sup>             |
| • Rule                | • TableConstraints <sup>2</sup>         | • View                          |

#### **Notes:**

<sup>1</sup> These values cannot be used with xp\_objectrecovery\_createscript.

<sup>2</sup> These values are pseudo-object types and are not real schema objects. They are only used to generate SQL scripts to alter the table, and they will be ignored when used with `xp_objectrecovery_viewcontents`.

<sup>3</sup> This value lists all object types, which are prefixed with "object\_type, ". All pseudo-table object types will be listed even though they might not exist for the associated table.

## Examples

### Generate SQL scripts to create a table

```
EXEC master.dbo.xp_objectrecovery_createscript
@filename = 'C:\MSSQL\Backup\MyDB_Backup.BAK'
, @objectname = 'dbo.Customers'
, @scriptfilename = 'C:\sql\CREATE_Customers.sql'
```

### Generate SQL scripts to alter a table

```
EXEC master.dbo.xp_objectrecovery_createscript
@filename = 'C:\MSSQL\Backup\MyDB_Backup.BAK'
, @objectname = 'dbo.Customers'
, @scriptfilename = 'C:\sql\ALTER_Customers.sql'
, @type = 'TableConstraintClustered'
```

### Generate SQL scripts to create a view

```
EXEC master.dbo.xp_objectrecovery_createscript
@filename = 'C:\MSSQL\Backup\MyDB_Backup.BAK'
, @objectname = 'dbo.Invoices'
, @scriptfilename = 'C:\sql\CREATE_VIEW_Invoices.sql'
, @type = 'View'
```

### Generate SQL scripts for objects listed in an object file

```
EXEC master.dbo.xp_objectrecovery_createscript
@filename = 'C:\MSSQL\Backup\MyDB_Backup.BAK'
, @objectfilename = 'C:\temp\MyDB_All.txt'
, @scriptfilename = 'C:\sql\CREATE_VIEW_Invoices.sql'
```

Where the `MyDB_ALL.txt` looks like the following:

```
Table,dbo.Customers

Table,dbo.[Order Details]

Table,dbo.Orders

Table,dbo.Products

TableConstraintClustered,dbo.Customers

TableConstraintClustered,dbo.[Order Details]
```

```
TableConstraintClustered, dbo.Orders
TableConstraintClustered, dbo.Products
TableIndexClustered, dbo.Customers
TableIndexClustered, dbo.[Order Details]
TableIndexClustered, dbo.Orders
TableIndexClustered, dbo.Products
TableConstraints, dbo.Customers
TableConstraints, dbo.[Order Details]
TableConstraints, dbo.Orders
TableConstraints, dbo.Products
TableIndexes, dbo.Customers
TableIndexes, dbo.[Order Details]
TableIndexes, dbo.Orders
TableIndexes, dbo.Products
TableForeignKeys, dbo.Customers
TableForeignKeys, dbo.[Order Details]
TableForeignKeys, dbo.Orders
TableForeignKeys, dbo.Products
TableTriggers, dbo.Customers
TableTriggers, dbo.[Order Details]
TableTriggers, dbo.Orders
TableTriggers, dbo.Products
View, dbo.[Current Product List]
StoredProcedure, dbo.CustOrdersDetail
StoredProcedure, dbo.[Sales by Year]
StoredProcedure, dbo.[Ten Most Expensive Products]
```

## Returns

0 (success) or non-zero (failure). Return codes represent the native error number returned from SQL Server for any errors encountered during the operation.

*To capture the output message, run the following:*

```
declare @rmsg varchar(999)
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output
select @rmsg
```

*To capture the output message and the result code, run the following:*

```
declare @rc int
declare @rmsg varchar(999)
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output,
@resultcode=@rc output
select @rc, @rmsg
```

## xp\_objectrecovery\_viewcontents

Lists the objects within the backup file.

### Syntax

```
EXEC master.dbo.xp_objectrecovery_viewcontents
(@filename = 'backup_file_name') [,...n]
[, @filenumber = n]
[, @encryptionkey = 'encryption_key']
[( , @logfilename = 'log_file_name'
[, @stripedlogfilename = 'striped_log_file_name'] [,...n]
[, @logencryptionkey = 'log_encryption_key']
[, @logfilenumber = n ] ) [,...n]]
[, @difffilename = 'diff_file_name'] [,...n]
[, @difffilenumber = n]
[, @diffencryptionkey = 'diff_encrypt_key']
[, @type = 'object_type']
[, @disablelogprocessing = 0 | 1 ]
```

### Arguments

#### Tips:

- To see the list of accepted arguments and data types for arguments, execute the following:  
`exec master.dbo.<procedure_name> show_help`
- To convert the script for use with the command-line utilities, execute the following:  
`exec master.dbo.<procedure_name> show_cmd, <xp_arguments>`

### @diffencryptionkey

Identifies the restoring user as a valid owner of the backup file.

**Caution:** If you use the wrong encryption key, the restore will fail. If you forget the encryption key, you will not be able to restore or read the backup.

Equivalent to @encryptionkey, but used for differential backups instead of full backup files.



### **@difffilename**

Name of backup file to restore. It can be multiply specified for striping. Used for differential backups instead of full backup files.

### **@difffilenumber**

Identifies the backup file within the backup set. Equivalent to @filenumber, but used for differential backups instead of full backup files.

### **@disablelogprocessing**

Instructs LiteSpeed to skip all transaction log backups and tail log processing. This argument accepts one of the following values:

- 0—False (Default).
- 1—True. LiteSpeed will entirely ignore any transaction log backups specified and will not process the tail log. This option may improve read and recovery times.

### **@encryptionkey**

Identifies the restoring user as a valid owner of the backup file.

**Caution:** If you use the wrong encryption key, the restore will fail. If you forget the encryption key, you will not be able to restore or read the backup.

### **@filename**

Location and name of the backup file or device to back up or restore. You can supply multiple instances of this argument.

This argument accepts variables. See "LiteSpeed Variables" (page 271) for more information.

### **@filenumber**

Specifies the particular backup to use when recasting, restoring, extracting or reading from files with multiple appended backups. You can run xp\_restore\_headeronly to query the files contained within the backup set given by backup\_file\_name.

### **@logencryptionkey**

Identifies the restoring user as a valid owner of the backup file.

**Caution:** If you use the wrong encryption key, the restore will fail. If you forget the encryption key, you will not be able to restore or read the backup.

Equivalent to @encryptionkey, but used for transaction log backups.

### **@logfilename**

Specifies the name of the log backup file. You can supply multiple instances of this argument.

See "Syntax" (page 184) for more information.

### **@logfilename**

Identifies the log backup file within the backup set. Equivalent to @filename, but used for log backups.

### **@stripedlogfilename**

Specifies the striped log file name.

**Note:** The striped files for a given log backup must be specified before the next log backup set is specified.

### **@type**

Specifies the type of object. If you omit this parameter the object type defaults to table, so you should use this argument to recover schema objects other than tables. This argument accepts one of the following values:

- |                       |   |                                 |
|-----------------------|---|---------------------------------|
| • All <sup>1, 3</sup> | • StoredProcedure                       | • TableIndexes <sup>2</sup>     |
| • Default             | • SystemTable                           | • TableForeignKeys <sup>2</sup> |
| • ExtendedProcedure   | • Table                                 | • TableTriggers <sup>2</sup>    |
| • Function            | • TableConstraintClustered <sup>2</sup> | • Type                          |
| • Role <sup>1</sup>   | • TableIndexClustered <sup>2</sup>      | • User <sup>1</sup>             |
| • Rule                | • TableConstraints <sup>2</sup>         | • View                          |

#### **Notes:**

<sup>1</sup> These values cannot be used with xp\_objectrecovery\_createscript.

<sup>2</sup> These values are pseudo-object types and are not real schema objects. They are only used to generate SQL scripts to alter the table, and they will be ignored when used with xp\_objectrecovery\_viewcontents.

<sup>3</sup> This value lists all object types, which are prefixed with "object\_type, ". All pseudo-table object types will be listed even though they might not exist for the associated table.

## **Examples**

### **List table objects for backup set #1 on a full backup file**

```
EXEC master.dbo.xp_objectrecovery_viewcontents  
@filename='C:\MSSQL\Backup\MyDB_Backup.BAK'
```

### **List all objects for backup set #1 on an encrypted SLS full backup file**

```
EXEC master.dbo.xp_objectrecovery_viewcontents  
@filename='C:\MSSQL\Backup\MyDB_Backup.BAK'
```

```
, @filenumber=1  
, @encryptionkey='Password'  
, @type='All'
```

### List view objects for backup set #2 on a full backup file + backup set #3 on a diff backup file

```
EXEC master.dbo.xp_objectrecovery_viewcontents  
@filename='C:\MSSQL\Backup\MyDB_Backup.BAK'  
, @filenumber=2  
, @difffilename='C:\MSSQL\Backup\MyDB_Diff.BAK'  
, @difffilenumber=3  
, @type='View'
```

### List all database objects using the full database backup and several t-log backups

```
EXEC master.dbo.xp_objectrecovery_viewcontents  
@filename = 'C:\temp\8_20101007183923.bak'  
, @filenumber = 1  
, @type = 'All'  
, @logfilename = 'C:\temp\8_20101007183923_20101007184136.bak'  
, @logfilenumber = 1  
, @logfilename = 'C:\temp\8_20101007183923_20101007184235.bak'  
, @logfilenumber = 1
```

### List encrypted contents of a striped backup

```
EXEC master.dbo.xp_objectrecovery_viewcontents  
@filename = 'C:\backups\testdecimal_full_1.bkp'  
, @filename = 'C:\backups\testdecimal_full_2.bkp'  
, @filenumber = 1  
, @encryptionkey='Ysbgdd05'  
, @type = 'All'  
, @logfilename = 'C:\backups\testdecimal_log_1_1.bkp'  
, @logencryptionkey='Ysbgdd06'  
, @stripedlogfilename = 'C:\backups\testdecimal_log_1_2.bkp'  
, @logfilenumber = 1  
, @logfilename = 'C:\backups\testdecimal_log_2_1.bkp'  
, @stripedlogfilename = 'C:\backups\testdecimal_log_2_2.bkp'  
, @logencryptionkey='Ysbgdd07'  
, @logfilenumber = 1
```

## Returns

0 (success) or non-zero (failure). Return codes represent the native error number returned from SQL Server for any errors encountered during the operation.

*To capture the output message, run the following:*

```
declare @rmsg varchar(999)  
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output  
select @rmsg
```

*To capture the output message and the result code, run the following:*

```
declare @rc int
declare @rmsg varchar(999)
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output,
@resultcode=@rc output
select @rc, @rmsg
```

## xp\_objectrecovery\_executeselect

Execute SELECT statement queries against the backup files, which you can use for row-level restores. The SELECT results can be a table in a database, ship directory, or a .csv file.

If a table with the same name already exists in the destination database, LiteSpeed will not overwrite it. However, you can use @destinationtable to rename the new table and restore it to the database.

### Syntax

#### Restore the SELECT results into a database

```
EXEC master.dbo.xp_objectrecovery_executeselect
(@filename = 'backup_file_name') [,...n]
[, @filenumber = n]
[, @encryptionkey = 'encryption_key']
[( , @logfilename = 'log_file_name'
[, @stripedlogfilename = 'striped_log_file_name'] [,...n]
[, @logencryptionkey = 'log_encryption_key']
[, @logfilenumber = n ] ) [,...n]]
[, @difffilename = 'diff_file_name'] [,...n]
[, @difffilenumber = n]
[, @diffencryptionkey = 'diff_encrypt_key']
[, @scripttext = 'script_text' |
, @scriptfilename = 'script_file_name']
[, @destinationtable = 'dest_table_name']
[, @destinationdatabase = 'database_name']
[, @destinationserver = 'dest_server_name']
[, @tempdirectory = 'recovery_temp_dir']
[, @onfilegroup = 'table_filegroup_name']
[, @textimageonfilegroup = 'blob_filegroup_name']
[, @disablelogprocessing = 0 | 1 ]
```

#### Recover the SELECT results into ship directory

```
EXEC master.dbo.xp_objectrecovery_executeselect
(@filename = 'backup_file_name') [,...n]
[, @filenumber = n]
[, @encryptionkey = 'encryption_key']
[( , @logfilename = 'log_file_name'
[, @stripedlogfilename = 'striped_log_file_name'] [,...n]
[, @logencryptionkey = 'log_encryption_key']
[, @logfilenumber = n ] ) [,...n]]
```

```
[, @difffilename = 'diff_file_name'] [,...n]
[, @difffilenumber = n]
[, @diffencryptionkey = 'diff_encrypt_key']
{, @scripttext = 'script_text' |
, @scriptfilename = 'script_file_name'}
[, @destinationtable = 'dest_table_name'
, @shipdirectory = 'recovery_ship_dir'
[, @onfilegroup = 'table_filegroup_name']
[, @textimageonfilegroup = 'blob_filegroup_name']
[, @disablelogprocessing = 0 | 1 ]
```

## Recover the SELECT results into a .csv file

```
EXEC master.dbo.xp_objectrecovery_executeselect
(@filename = 'backup_file_name') [,...n]
[, @filenumber = n]
[, @encryptionkey = 'encryption_key']
[( , @logfilename = 'log_file_name'
[, @stripedlogfilename = 'striped_log_file_name'] [,...n]
[, @logencryptionkey = 'log_encryption_key']
[, @logfilenumber = n ] ) [,...n]]
[, @difffilename = 'diff_file_name'] [,...n]
[, @difffilenumber = n]
[, @diffencryptionkey = 'diff_encrypt_key']
{, @scripttext = 'script_text' |
, @scriptfilename = 'script_file_name'}
{, @scripttext = 'script_text' |
, @scriptfilename = 'script_file_name'}
[, @destinationfilename = 'csv_file_name'
[, @disablelogprocessing = 0 | 1 ]
```

## Arguments

### Tips:

- To see the list of accepted arguments and data types for arguments, execute the following:  
`exec master.dbo.<procedure_name> show_help`
- To convert the script for use with the command-line utilities, execute the following:  
`exec master.dbo.<procedure_name> show_cmd, <xp_arguments>`

### @backend

By default, Object Level Recovery writes all of the recovered data to a file and then connects to SQL Server and issues a bulk insert command for SQL Server to bulk insert from that temp file.

Instead you can pass in @backend='SQLNativeClient' or -b 1 from command line. That causes Object Level Recovery to use a bcp api implemented in Microsoft's sqlncli.dll to send the data straight to the sqlservr.exe process, thereby not needing any disk space.

### @destinationdatabase

Specifies the destination database when restoring an object to a table.

### **@destinationfilename**

Name of comma separated file (.csv) that is generated instead restoring into a database. This is an ad hoc solution for users want to see the restored data in Excel. You can only use this argument for text data.

### **@destinationserver**

Name of the destination server.

### **@destinationtable**

Name of the destination table. Use this when a table with the same name already exists in the destination database. LiteSpeed will not overwrite an existing table.

### **@diffencryptionkey**

Identifies the restoring user as a valid owner of the backup file.

**Caution:** If you use the wrong encryption key, the restore will fail. If you forget the encryption key, you will not be able to restore or read the backup.

Equivalent to @encryptionkey, but used for differential backups instead of full backup files.

### **@difffilename**

Name of backup file to restore. It can be multiply specified for striping. Used for differential backups instead of full backup files.

### **@difffilenumber**

Identifies the backup file within the backup set. Equivalent to @filenumber, but used for differential backups instead of full backup files.

### **@disablelogprocessing**

Instructs LiteSpeed to skip all transaction log backups and tail log processing. This argument accepts one of the following values:

- 0—False (Default).
- 1—True. LiteSpeed will entirely ignore any transaction log backups specified and will not process the tail log. This option may improve read and recovery times.

### **@encryptionkey**

Identifies the restoring user as a valid owner of the backup file.

**Caution:** If you use the wrong encryption key, the restore will fail. If you forget the encryption key, you will not be able to restore or read the backup.

### **@filename**

Location and name of the backup file or device to back up or restore. You can supply multiple instances of this argument.

This argument accepts variables. See "LiteSpeed Variables" (page 271) for more information.

### **@filenumber**

Specifies the particular backup to use when recasting, restoring, extracting or reading from files with multiple appended backups. You can run `xp_restore_headeronly` to query the files contained within the backup set given by `backup_file_name`.

### **@logencryptionkey**

Identifies the restoring user as a valid owner of the backup file.

**Caution:** If you use the wrong encryption key, the restore will fail. If you forget the encryption key, you will not be able to restore or read the backup.

Equivalent to `@encryptionkey`, but used for transaction log backups.

### **@logfilename**

Specifies the name of the log backup file. You can supply multiple instances of this argument.

See "Syntax" (page 188) for more information.

### **@logfilenumber**

Identifies the log backup file within the backup set. Equivalent to `@filenumber`, but used for log backups.

### **@onfilegroup**

Filegroup with the object to restore.

### **@scriptfilename**

Name of the SELECT script file to be executed.

### **@scripttext**

The text of the SELECT script to be executed. The SELECT statement is imbedded in a single-quoted string literal, and all single-quoted string literals in the SELECT statement need to be double single-quoted. For example:

```
@scripttext = 'SELECT * FROM dbo.Customers WHERE City=''London'''
```

The single quoted string literal 'London' is double single-quoted.

You can also use SET QUOTED\_IDENTIFIER OFF to allow double quotes. For example:

```
SET QUOTED_IDENTIFIER OFF
EXEC xp_objectrecovery_executeselect
    @filename = 'C:\MSSQL\Backup\MyDB_Backup.BAK'
    , @scripttext = "SELECT * FROM dbo.Customers WHERE City='London'"
    , @destinationtable = 'dbo.CustomersInLondon'
    , @destinationdatabase = 'MyDB'
```

### @shipdirectory

Name of the ship directory. Use this argument when you want to restore the object later or at a different physical location. This argument creates the following files in the ship directory:

- CREATE <object\_name>.sql
- BULK INSERT <object\_name>.sql
- <object\_name>.fmt
- <object\_name>.bcp

To restore the object, run the CREATE file first, and then run the BULK INSERT file. You will need to slightly modify the BULK INSERT file because of the .fmt and .bcp file path names.

**Tip:** You can zip the files and send them to someone else.

### @stripedlogfilename

Specifies the striped log file name.

**Note:** The striped files for a given log backup must be specified before the next log backup set is specified.

### @tempdirectory

Specifies a temporary directory for use with Object Level Recovery. Use this argument when the default Windows temp directory does not have enough free disk space for the restore process.

**Note:** You can specify the default temp directory using the TempPath parameter in the [LiteSpeed] section of the LiteSpeedSettings.ini file. (Usually, *C:\Documents and Settings\All Users\Application Data\Quest Software\LiteSpeed\SQLServer\LiteSpeedSettings.ini*.)

### @textimageonfilegroup

Destination TEXTIMAGE\_ON filegroup name. Used to restore a BLOB (binary large object).



## Examples

### Restore the SELECT query result into a database using inline script

```
EXEC master.dbo.xp_objectrecovery_executeselect
@filename = 'C:\MSSQL\Backup\MyDB_Backup.BAK'
, @scripttext = 'SELECT * FROM dbo.Customers WHERE City='London''
, @destinationtable = 'dbo.CustomersInLondon'
, @destinationdatabase = 'MyDB'
```

### Restore the SELECT query result into a database using script file

```
EXEC master.dbo.xp_objectrecovery_executeselect
@filename = 'C:\MSSQL\Backup\MyDB_Backup.BAK'
, @scriptfilename = 'C:\temp\SelectCustomerFromLondon.sql'
, @destinationtable = 'dbo.CustomersInLondon'
, @destinationdatabase = 'MyDB'
, @destinationserver = 'MyMachine\SQL2000'
, @tempdirectory = 'D:\temp'
, @onfilegroup = 'Secondary'
, @textimageonfilegroup = 'Secondary'
```

### Restore the SELECT query result into ship directory

```
EXEC master.dbo.xp_objectrecovery_executeselect
@filename = 'C:\MSSQL\Backup\MyDB_Backup.BAK'
, @scriptfilename = 'C:\temp\SelectCustomerFromLondon.sql'
, @destinationtable = 'dbo.CustomersInLondon'
, @shipdirectory = 'C:\temp\London'
```

### Restore the SELECT query result into a .csv file

```
EXEC master.dbo.xp_objectrecovery_executeselect
@filename = 'C:\MSSQL\Backup\MyDB_Backup.BAK'
, @scriptfilename = 'C:\temp\SelectCustomerFromLondon.sql'
, @destinationfilename = 'C:\temp\LondonCustomer.csv'
```

## Returns

0 (success) or non-zero (failure). Return codes represent the native error number returned from SQL Server for any errors encountered during the operation.

*To capture the output message, run the following:*

```
declare @rmsg varchar(999)
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output
select @rmsg
```

*To capture the output message and the result code, run the following:*

```
declare @rc int
declare @rmsg varchar(999)
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output,
```

```
@resultcode=@rc output  
select @rc, @rmsg
```

## xp\_restore\_attachedfilesonly

Restores attached files included within LiteSpeed backup sets. This procedure performs no database restore operation, but only restores the specified files.

**Note:** You can direct restored files to be recreated in an alternate location from their original location.

## Syntax

### xp\_restore\_attachedfilesonly (Disk or TSM)

```
EXEC master.dbo.xp_restore_attachedfilesonly  
( @filename = 'backup_file_name') [,...n]  
(, @attachedfile = 'pathname [ to new_pathname ]')) [,...n]  
[, @encryptionkey = 'encryption_key']  
[, @filenumber = n]  
[, @logging = 0 | 1 | 2 ]  
[, @affinity = 0..2147483648]  
[, @throttle = 1..100]  
[, @tsmclientnode = 'TSM_client_node']  
[, @tsmclientownerpwd = 'TSM_client_owner_password']  
[, @tsmobject = 'TSM_object']  
[, @tsmconfigfile = 'TSM_configuration_file']  
[, @tsmpointintime = 'date_time']  
[, @tsmarchive = 0 | 1]
```

### xp\_restore\_attachedfilesonly (Tape)

```
EXEC master.dbo.xp_restore_attachedfilesonly  
( @filename = 'backup_file_name') [,...n]  
(, @attachedfile = 'pathname [ to new_pathname ]')) [,...n]  
[, @filenumber = n]  
[, @rewind = 0 | 1 ]  
[, @unload = 0 | 1 ]  
[, @encryptionkey = 'encryption_key']  
[, @logging = 0 | 1 | 2 ]  
[, @affinity = 0..2147483648]  
[, @throttle = 1..100]
```

## Arguments

xp\_restore\_attachedfilesonly accepts the following arguments:

## @affinity

Processor affinity designates specific processors to run LiteSpeed, while not allowing LiteSpeed to run on the remaining processors.

This argument accepts decimal values and hexadecimal values. If a value begins with "0x" it is interpreted as hexadecimal. A positive 64-bit integer value translates to a binary mask where a value of 1 designates the corresponding processor to be able to run the LiteSpeed process.

**Note:** 32-bit Windows is internally limited to a 32-bit mask.

For example, you need to select processors 2, 3, and 6 for use with LiteSpeed. Number the bits from the right to left. The rightmost bit represents the first processor. Set the second, third, and sixth bits to 1 and all other bits to 0. The result is binary 100110, which is decimal 38 or hexadecimal 0x26. Review the following for additional information:

Decimal Value	Binary Bit Mask	Allow LiteSpeed Threads on Processors
0	0	All (default)
1	1	1
3	11	1 and 2
7	111	1, 2 and 3
38	100110	2, 3, and 6
205	11001101	1, 3, 4, 7, and 8

**Tip:** Before you start tuning the CPU Throttle or Affinity parameters to adjust backup performance, try limiting the number of threads. If you decide to use an affinity value other than default, it is recommended that you limit the threading as well. You may also want to consider using Adaptive Compression to maintain backup performance. See "Adaptive Compression" (page 55) for more information.

## @attachedfile

Specifies filepaths to include in both backup and restore operations. The filepath can be either a single file or a directory. If it is a directory, then LiteSpeed recursively includes all files and subdirectories. All attached files are encrypted and compressed, with all pertinent backup parameters supported. This feature works for disk, tape, TSM, and Double Click Restore as well. You can supply multiple instances of this argument.

When used within the context of a restore operation, the path parameter can be expanded to include a new destination. This form will take the syntax of <file\_path> to <new\_file\_path>. The new filepath can be used to specify a new location but cannot rename a file.

This argument only restores the attached files. It does not restore the database, just the files that were attached to that backup.

**Notes:**

- The original entire directory path need not be supplied (e.g. c: to c:\testadSattsm is allowed).
- c:\testad to testadr would restore all files in directory c:\testad to c:\testadr.

**@encryptionkey**

Identifies the restoring user as a valid owner of the backup file.

**Caution:** If you use the wrong encryption key, the restore will fail. If you forget the encryption key, you will not be able to restore or read the backup.

**@filename**

Location and name of the backup file or device to back up or restore. You can supply multiple instances of this argument.

This argument accepts variables. See "LiteSpeed Variables" (page 271) for more information.

**@filenumber**

*Disk restores:*

Specifies the particular backup to use when recasting, restoring, extracting or reading from files with multiple appended backups. You can run `xp_restore_headeronly` to query the files contained within the backup set given by `backup_file_name`.

*Tape restores:*

Identifies the backup set to be restored. For example, a file number of 1 indicates the first backup set on the backup medium, and a file number of 2 indicates the second backup set.

**@logging**

Writes a log file for the operation. This argument accepts one of the following values:

- 0—Logging off.
- 1 or any odd value—Logging on. Log file is removed on success.
- 2 or any even value—Logging on.

The default output directory is `C:\Documents and Settings\All Users\Application Data\Quest Software\LiteSpeed\SQL Server\Logs`. To log to a different directory add `@Trace='logpath=path'`.

See Configure Logging in LiteSpeed (page 285) for information about LiteSpeed logging.

**@rewind**

Applies only to backing up and restoring tape. This argument accepts one of the following values:

- 0—Leave the tape unwound (default)
- 1—Rewind the tape after writing/reading

### **@throttle**

Specifies the maximum CPU usage allowed. The argument accepts an integer value between 1 and 100. The default value is 100. This is the percentage of the total amount of CPU usage (across all enabled processors) available.

**Tip:** Before you start tuning the CPU Throttle or Affinity parameters to adjust backup performance, try limiting the number of threads. If you decide to use an affinity value other than default, it is recommended that you limit the threading as well. You may also want to consider using Adaptive Compression to maintain backup performance. See "Adaptive Compression" (page 55) for more information.

### **@tsmarchive**

Specifies to store backup as a TSM archive. This argument accepts one of the following values:

- 0—False
- 1—True

### **@tsmclientnode**

Specifies the TSM server LiteSpeed connects to during backups and restores. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.

### **@tsmclientownerpwd**

Specifies the TSM client owner user password. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.

### **@tsmconfigfile**

Specifies the TSM configuration file.

### **@tsmobject**

Defines the TSM filespace, high level and low level. This argument accepts the following format:

```
tsm_filespace\tsm_high_level\tsm_low_level
```

where:

- `tsm_filespace` is the logical space on the TSM server that contains a group of files. It can be the drive label name or UNC name.

- `tsm_high_level` specifies the directory path in which the file belongs.
- `tsm_low_level` specifies actual name of the file.

**Note:** You may only store one item the location specified by this argument. It is not possible to append an object to this location. You can use the `-I` command-line argument or `@init` to back up to a non-unique location.

### **@tsmpointintime**

Specifies the date for restore/to filter results. If it is not passed, LiteSpeed will choose the most recent archived backup. The format is `yyyy-mm-dd hh:mm:ss`.

**Note:** If the backup was a striped backup and the point-in-times of the various striped files are different (rare but can be different a second or so), then the most recent of the times must be chosen.

### **@unload**

Applies to tape backups and restores. This argument accepts one of the following values:

- 0—Keep tape loaded (default)
- 1—Unload and eject tape from the drive after operation

## **Examples**

1. Restore a complete directory:

```
EXEC master.dbo.xp_restore_attachedfilesonly  
@filename= 'C:\MSSQL\Backup\MyDB_Backup.BAK'  
@attachedfile = 'C:\DATA\Images'
```

2. Restore a directory and file to an alternate location:

```
EXEC master.dbo.xp_restore_attachedfilesonly  
@filename= 'C:\MSSQL\Backup\MyDB_Backup.BAK'  
@attachedfile = 'C:\DATA\Images to c:\DATA\Old_Images'  
@attachedfile = 'C:\DATA\Docs\Invoice.pdf to C:\DATA\Docs\Old_  
Invoice.pdf'
```

3. Restore a file attached to a tsm backup:

```
EXEC master.dbo.xp_restore_attachedfilesonly  
@tsmconfigfile = N'C:\Program Files\Tivoli\TSM\baclient\dsm.opt',  
@tsmobject = N'C\Reports\attachedfiles',  
@tsmpointintime = '2012-05-04 00:54:32',  
@filenumber = 1,  
@affinity = 0,  
@logging = 0,  
@attachedfile = N'''C:\Program Files\Tivoli\TSM\baclient\dsm_pg.opt''' to  
'C:\Program Files\Tivoli\TSM\baclient\dsm_pg.opt'''
```

## Returns

0 (success) or non-zero (failure). Return codes represent the native error number returned from SQL Server for any errors encountered during the operation.

*To capture the output message, run the following:*

```
declare @rmsg varchar(999)
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output
select @rmsg
```

*To capture the output message and the result code, run the following:*

```
declare @rc int
declare @rmsg varchar(999)
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output,
@resultcode=@rc output
select @rc, @rmsg
```

## xp\_restore\_checkpassword

This extended stored procedure checks a provided password or key against a single backup set within a single file.

## Syntax

```
EXEC master.dbo.xp_restore_checkpassword
@filename = 'filename'
[,@filenumber = n]
[,@encryptionkey = 'Mypassword']
```

## Arguments

The procedure accepts the following arguments:

### @filename

Location and name of the backup file or device to back up or restore. You can supply multiple instances of this argument.

This argument accepts variables. See "LiteSpeed Variables" (page 271) for more information.

### @filenumber

Specifies the particular backup to use when recasting, restoring, extracting or reading from files with multiple appended backups. You can run xp\_restore\_headeronly to query the files contained within the backup set given by backup\_file\_name.

If @FileNumber is not specified, the backup set defaults to 1.

## @encryptionkey

Identifies the restoring user as a valid owner of the backup file.

**Caution:** If you use the wrong encryption key, the restore will fail. If you forget the encryption key, you will not be able to restore or read the backup.

If @EncryptionKey is not provided (or is an empty string), the procedure checks if the backup set is encrypted or not.

## xp\_restore\_checksumonly

This procedure generates a checksum of an entire backup file. It can be used to validate that a file has not been corrupted since a previous point in time. This feature does not validate that the backup set is valid, but provides you with a check sum value to validate your files with.

### Syntax

```
EXEC master.dbo.xp_restore_checksumonly  
@filename = 'backup_file_name'
```

## xp\_restore\_database

Restores LiteSpeed backups created with the xp\_backup\_database command. Files and filegroups may also be restored either from a file or filegroup LiteSpeed backup operation, or from a full database backup operation using xp\_backup\_database. When restoring files or filegroups, you must apply a transaction log using xp\_restore\_log. In addition, file differential backups can be restored after a full file restore using LiteSpeed.

**Note:** A database cannot be restored unless the restore process has exclusive access to the database. No user connections can exist when performing a database restore.

### Syntax

#### xp\_restore\_database (Disk or TSM)

```
EXEC master.dbo.xp_restore_database  
@database = 'database_name'  
(, @filename = 'backup_file_name') [,...n]  
[, ( @encryptionkey = 'encryption_key' | @jobp = 'encrypted_key' ) ]  
[, @file = 'logical_file_name'] [,...n]  
[, @filenumber = n]  
[, @filegroup = 'logical_filegroup_name'] [,...n]  
[, @with = 'additional_with_parameters'] [,...n]  
[, @logging = 0 | 1 | 2 ]  
[, @affinity = 0..2147483648]  
[, @throttle = 1..100]  
[, @ioflag = 'DISK_RETRY_COUNT=n']  
[, @ioflag = 'DISK_RETRY_WAIT=n']
```



```
[, @restoreasreadonly = 0 | 1 ]  
[, @buffercount = 'buffer_count']  
[, @maxtransfersize = 'maximum_transfer_size']  
[, @attachedfile = 'pathname'] [,...n]  
[, @tsmclientnode = 'TSM_client_node']  
[, @tsmclientownerpwd = 'TSM_client_owner_password']  
[, @tsmobject = 'TSM_object']  
[, @tsmconfigfile = 'TSM_configuration_file']  
[, @tsmpointintime = 'date_time']  
[, @tsmarchive = 0 | 1 ]  
[, @returndetails = 0 | 1]
```

## xp\_restore\_database (Tape)

```
EXEC master.dbo.xp_restore_database  
@database = 'database_name'  
(, @filename = 'backup_file_name') [,...n]  
[, @filenumber = n]  
[, @rewind = 0 | 1 ]  
[, @unload = 0 | 1 ]  
[, @encryptionkey = 'encryption_key']  
[, @file = 'logical_file_name'] [,...n]  
[, @filegroup = 'logical_filegroup_name'] [,...n]  
[, @with = 'additional_with_parameters'] [,...n]  
[, @restoreasreadonly = 0 | 1 ]  
[, @logging = 0 | 1 | 2 ]  
[, @affinity = 0..2147483648]  
[, @throttle = 1..100]  
[, @buffercount = 'buffer_count']  
[, @maxtransfersize = 'maximum_transfer_size']  
[, @attachedfile = 'pathname'] [,...n]  
[, @returndetails = 0 | 1]
```

## Arguments

### Tips:

- To see the list of accepted arguments and data types for arguments, execute the following:  
`exec master.dbo.<procedure_name> show_help`
- To convert the script for use with the command-line utilities, execute the following:  
`exec master.dbo.<procedure_name> show_cmd, <xp_arguments>`

### @affinity

Processor affinity designates specific processors to run LiteSpeed, while not allowing LiteSpeed to run on the remaining processors.

This argument accepts decimal values and hexadecimal values. If a value begins with "0x" it is interpreted as hexadecimal. A positive 64-bit integer value translates to a binary mask where a value of 1 designates the corresponding processor to be able to run the LiteSpeed process.

**Note:** 32-bit Windows is internally limited to a 32-bit mask.

For example, you need to select processors 2, 3, and 6 for use with LiteSpeed. Number the bits from the right to left. The rightmost bit represents the first processor. Set the second, third, and sixth bits to 1 and all other bits to 0. The result is binary 100110, which is decimal 38 or hexadecimal 0x26. Review the following for additional information:

Decimal Value	Binary Bit Mask	Allow LiteSpeed Threads on Processors
0	0	All (default)
1	1	1
3	11	1 and 2
7	111	1, 2 and 3
38	100110	2, 3, and 6
205	11001101	1, 3, 4, 7, and 8

**Tip:** Before you start tuning the CPU Throttle or Affinity parameters to adjust backup performance, try limiting the number of threads. If you decide to use an affinity value other than default, it is recommended that you limit the threading as well. You may also want to consider using Adaptive Compression to maintain backup performance. See "Adaptive Compression" (page 55) for more information.

### @attachedfile

Specifies filepaths to include in both backup and restore operations. The filepath can be either a single file or a directory. If it is a directory, then LiteSpeed recursively includes all files and subdirectories. All attached files are encrypted and compressed, with all pertinent backup parameters supported. This feature works for disk, tape, TSM, and Double Click Restore as well. You can supply multiple instances of this argument.

When used within the context of a restore operation, the path parameter can be expanded to include a new destination. This form will take the syntax of <file\_path> to <new\_file\_path>. The new filepath can be used to specify a new location but cannot rename a file.

This argument only restores the attached files. It does not restore the database, just the files that were attached to that backup.

### Notes:

- The original entire directory path need not be supplied (e.g. c: to c:\testadSattsm is allowed).
- c:\testad to testadr would restore all files in directory c:\testad to c:\testadr.

### @buffercount

Number of SQL Server buffers available for a LiteSpeed operation. The default value is 20.

## **@database**

Name of database to be backed up or restored.

This parameter specifies a database:

- to be backed up (xp\_backup\_database and xp\_slsSmartDiff)
- containing the transaction log to be backed up (xp\_backup\_log)
- to be restored (xp\_restore\_database and xp\_restore\_log)
- on which you wish to check the progress of an activity (xp\_slsReadProgress)
- for which you want to delete old backups (xp\_slsSmartCleanup)

If supplied as a variable (@database), this name can be specified either as a string constant (@database = database name) or as a variable of character string data type, except for the ntext or text data types.

## **@encryptionkey**

Identifies the restoring user as a valid owner of the backup file.

**Caution:** If you use the wrong encryption key, the restore will fail. If you forget the encryption key, you will not be able to restore or read the backup.

## **@file**

Specifies a logical database file used for file or filegroup backups. You can supply multiple instances of this argument.

## **@filegroup**

Specifies a database filegroup to include in the backup or restore. You can supply multiple instances of this argument.

A filegroup backup is a single backup of all files in the filegroup and is equivalent to explicitly listing all files in the filegroup when creating the backup. Files in a filegroup backup can be restored individually or as a group.

## **@filename**

Location and name of the backup file or device to back up or restore. You can supply multiple instances of this argument.

This argument accepts variables. See "LiteSpeed Variables" (page 271) for more information.

## **@filename**

*Disk restores:*

Specifies the particular backup to use when recasting, restoring, extracting or reading from files with multiple appended backups. You can run `xp_restore_headeronly` to query the files contained within the backup set given by `backup_file_name`.

*Tape restores:*

Identifies the backup set to be restored. For example, a file number of 1 indicates the first backup set on the backup medium, and a file number of 2 indicates the second backup set.

## **@ioflag**

Specifies if LiteSpeed should wait and retry the read or write operation on failure. You can define retry options using the following parameters:

- **DISK\_RETRY\_COUNT**—Specifies the number of times that a specific operation will be retried on failure. The default is 4 retries, the maximum allowed setting is 1000.
- **DISK\_RETRY\_WAIT**—Specifies the number of seconds to wait immediately following a failure before retrying. The default is 15 seconds, the maximum allowed setting is 300.

**Note:** This functionality is only available for disk operations.

For example, to specify a database backup where each failure can be retried once after a 30-second wait:

```
EXEC master.dbo.xp_backup_database  
@filename='c:\test.bkp'  
, @database='test'  
, @ioflag='DISK_RETRY_COUNT=1'  
, @ioflag='DISK_RETRY_WAIT=30'
```

See "Network Resilience" (page 58) for more information.

## **@jobp**

Specifies the `encrypted_key`. (Similar to `@EncryptionKey`).

You can use `xp_encrypt_backup_key` (page 171) to convert the password (`encryption_key`) for use with `@jobp`. The original password (or `encrypted_key` generated by `xp_encrypt_restore_key` (page 171)) must be used to restore a backup.

## **@logging**

Writes a log file for the operation. This argument accepts one of the following values:

- 0—Logging off.
- 1 or any odd value—Logging on. Log file is removed on success.

- 2 or any even value—Logging on.

The default output directory is *C:\Documents and Settings\All Users\Application Data\Quest Software\LiteSpeed\SQL Server\Logs*. To log to a different directory add `@Trace='logpath=path'`.

See *Configure Logging in LiteSpeed* (page 285) for information about LiteSpeed logging.

### **@maxtransfersize**

Specifies the data size in bytes for each transfer when communicating with SQL Server. The size can be any multiple of 64KB in the range from 64KB to 4MB. The default is *1048576* bytes.

### **@restoreasreadonly**

Instructs the restore operation to leave the database in read-only mode. Using this option, you can safely restore a database into a NTFS-compressed folder. This argument accepts one of the following values:

- 0—False (default)
- 1—True

### **@returndetails**

Generates a single-row result set.

- 0—False (default)
- 1—True

The result set contains the following details:

Column Name	Data Type	Description
Database	nvarchar (128)	Database name.
Operation	nvarchar (30)	Operation type: Backup or Restore.
Threads	tinyint	The number of threads used for a LiteSpeed backup.
CompressionLevel	tinyint	Compression level used for compressing the backup. The compression level can be NULL, if backed up with Adaptive Compression.
AdaptiveCompression	nvarchar (max)	Adaptive Compression option used for compressing the backup: 'speed' or 'size'.
MaxTransferSize	int	The data size in bytes for each transfer when communicating with SQL Server. The size can be any

Column Name	Data Type	Description
		multiple of 64 KB in the range from 64 KB to 4 MB. The default is 1048576 bytes.
BaseSize	int	The smallest chunk of memory LiteSpeed attempts to write to disk at any given time.
BufferCount	smallint	The number of SQL Server buffers available for a LiteSpeed operation.
StripeCount	smallint	Number of backup files in the stripe set.
OverlappedBuffers	tinyint	The number of buffers that any single VDI thread can use at a time.
CPUSeconds	numeric (18, 3)	Processor time used by the LiteSpeed operation.
ElapsedSeconds	numeric (18, 3)	Duration of the operation.
NativeSize	bigint	Backup size (in bytes) without LiteSpeed compression.
BackupSize	bigint	Size of the backup (in bytes).

**Tip:** In Toad, you can use Group Execute to produce a single result set for several server instances.

### @rewind

Applies only to backing up and restoring tape. This argument accepts one of the following values:

- 0—Leave the tape unwound (default)
- 1—Rewind the tape after writing/reading

### @throttle

Specifies the maximum CPU usage allowed. The argument accepts an integer value between 1 and 100. The default value is 100. This is the percentage of the total amount of CPU usage (across all enabled processors) available.

**Tip:** Before you start tuning the CPU Throttle or Affinity parameters to adjust backup performance, try limiting the number of threads. If you decide to use an affinity value other than default, it is recommended that you limit the threading as well. You may also want to consider using Adaptive Compression to maintain backup performance. See "Adaptive Compression" (page 55) for more information.

### **@tsmarchive**

Specifies to store backup as a TSM archive. This argument accepts one of the following values:

- 0—False
- 1—True

### **@tsmclientnode**

Specifies the TSM server LiteSpeed connects to during backups and restores. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.

### **@tsmclientownerpwd**

Specifies the TSM client owner user password. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.

### **@tsmconfigfile**

Specifies the TSM configuration file.

### **@tsmobject**

Defines the TSM filespace, high level and low level. This argument accepts the following format:

```
tsm_filespace\tsm_high_level\tsm_low_level
```

where:

- `tsm_filespace` is the logical space on the TSM server that contains a group of files. It can be the drive label name or UNC name.
- `tsm_high_level` specifies the directory path in which the file belongs.
- `tsm_low_level` specifies actual name of the file.

**Note:** You may only store one item the location specified by this argument. It is not possible to append an object to this location. You can use the `-I` command-line argument or `@init` to back up to a non-unique location.

### **@tsmpointintime**

Specifies the date for restore/to filter results. If it is not passed, LiteSpeed will choose the most recent archived backup. The format is `yyyy-mm-dd hh:mm:ss`.

**Note:** If the backup was a striped backup and the point-in-times of the various striped files are different (rare but can be different a second or so), then the most recent of the times must be chosen.

## @unload

Applies to tape backups and restores. This argument accepts one of the following values:

- 0—Keep tape loaded (default)
- 1—Unload and eject tape from the drive after operation

## @with

Each @with argument should be a syntactically complete and correct statement. Please refer to the SQL Server Transact-SQL backup and restore documentation for the syntax and usage.

The supported formats are:

- @with='PARAMETER'
- @with='PARAMETER="accepted\_value"'

## Notes:

- Extended stored procedure arguments are limited to 255 characters. If you need more than 255 characters, use multiple @with arguments.
- Do not supply the @with parameter if no additional features are required.

This extended stored procedure accepts the following @with parameters:

Parameter	Description
KEEP_REPLICATION	Instructs the restore operation to keep the replication settings when restoring a published database to a server other than that on which it was created (used when setting up replication with log shipping). You cannot specify this parameter with NORECOVERY.
MOVE	<p>MOVE = 'logical_file_name' TO 'operating_system_file_name'</p> <p>Specifies that the given logical_file_name should be moved to operating_system_file_name. By default, the logical_file_name is restored to its original location.</p> <p>If you use xp_restore_database to copy a database to the same or different server, the MOVE parameter may be needed to relocate the database files and to avoid collisions with existing files. Each logical file in the database can be specified in different MOVE statements.</p> <p>Example:</p> <pre>EXEC master.dbo.xp_restore_database @database = 'MyDB' , @filename = 'C:\MSSQL\Backup\MyDB_Backup.BAK' , @with = 'MOVE 'MyDB_Data' TO 'C:\MSSQL\Data\MyDB_data.MDF''' , @with = 'MOVE 'MyDB_Data2' TO 'C:\MSSQL\Data\MyDB_data2.NDF'''</pre>



Parameter	Description
	<p>, @with = 'MOVE ''MyDB_Log'' TO ''C:\MSSQL\Data\MyDB_log.LDF'''</p> <p><b>Note:</b> Use <code>xp_restore_filelistonly</code> to obtain a list of the logical files from the backup set. See "xp_restore_filelistonly" (page 212) for more information.</p>
NORECOVERY	<p>Instructs the restore operation to not roll back any uncommitted transactions. Either the NORECOVERY or STANDBY option must be specified if another transaction log has to be applied. If NORECOVERY, RECOVERY, or STANDBY is not specified, RECOVERY is the default. SQL Server requires that the WITH NORECOVERY option is used on all but the final xp_restore_log statement when restoring a database backup and multiple transaction logs using LiteSpeed, or when multiple xp_restore_database or xp_restore_log statements are needed (for example, a full database backup followed by a differential database backup).</p> <p><b>Note:</b> When specifying the NORECOVERY option, the database is not usable in this intermediate, non-recovered state.</p> <p>When used with a file or filegroup restore operation, NORECOVERY forces the database to remain in the restoring state after the restore operation. This is useful in either of these situations:</p> <ul style="list-style-type: none"> <li>• a restore script is being run and the log is always being applied.</li> <li>• a sequence of file restores is used and the database is not intended to be usable between two of the restore operations.</li> </ul>
PARTIAL	<p>Specifies a partial restore operation.</p> <p>The granularity of the partial restore operation is the database filegroup. The primary file and filegroup are always restored, along with the files that you specify and their corresponding filegroups. The result is a subset of the database. Filegroups that are not restored are marked as offline and are not accessible.</p>
RECOVERY	<p>Instructs the restore operation to roll back any uncommitted transactions. After the recovery process, the database is ready for use.</p> <p>If subsequent LiteSpeed restore operations (xp_restore_log or xp_restore_database from differential) are planned, NORECOVERY or STANDBY should be specified instead.</p> <p>If NORECOVERY, RECOVERY, or STANDBY is not specified, RECOVERY is the default.</p> <p>When restoring backup sets from an earlier version of SQL Server, a database upgrade may be required. This upgrade is performed automatically when WITH RECOVERY is specified.</p>
REPLACE	<p>Specifies that LiteSpeed should create the specified database and its related files even if another database already exists with the same name.</p>

Parameter	Description
	<p>The existing database is deleted.</p> <p>When the REPLACE option is not specified, LiteSpeed performs a check to ensure that the xp_restore_database statement will not restore the database to the current server if:</p> <ul style="list-style-type: none"> <li>the database named in the xp_restore_database statement already exists on the current server, and</li> <li>the database name is different from the database name recorded in the LiteSpeed backup set.</li> </ul> <p>REPLACE also allows xp_restore_database to overwrite an existing file which cannot be verified as belonging to the database being restored. Normally, xp_restore_database will refuse to overwrite pre-existing files.</p>
RESTRICTED_USER	<p>When used in conjunction with recovery (another with param and the default) leaving a usable database, this restricts access for the restored database to members of the db_owner, dbcreator, or sysadmin roles.</p>
STANDBY	<p>STANDBY = 'undo_file_name'</p> <p>Specifies the undo file name so the recovery effects can be undone. The size required for the undo file depends on the volume of undo actions resulting from uncommitted transactions. If you do not specify NORECOVERY, RECOVERY, or STANDBY, LiteSpeed defaults to RECOVERY.</p> <p>STANDBY allows a database to be brought up for read-only access between transaction log restores and can be used with either warm standby server situations or special recovery situations in which it is useful to inspect the database between log restores.</p> <p>If the specified undo file name does not exist, LiteSpeed creates it. If the file does exist, LiteSpeed overwrites it.</p> <p>The same undo file can be used for consecutive LiteSpeed restores of the same database.</p> <p><b>Note:</b> If free disk space is exhausted on the drive containing the specified undo file name, the LiteSpeed restore operation stops.</p>
STATS	<p>Displays a message each time a percentage of the activity completes. The default is 10%.</p>
CHECKSUM	<p>Causes checksums to be verified when a LiteSpeed backup is created.</p> <p><b>Note:</b> When you restore a backup containing checksum, it is automatically checked. If you do not want to check the checksums during a restore, supply 'NO_CHECKSUM'.</p>
PASSWORD	<p>Specifies the password for the backup set.</p>

## Examples

### Standard Database Restore

```
EXEC master.dbo.xp_restore_database  
@database = 'MyDB'  
, @filename= 'C:\MSSQL\Backup\MyDB_Backup.BAK'
```

### Restore Database with NoRecovery

```
EXEC master.dbo.xp_restore_database  
@database='MyDB'  
, @filename='C:\MSSQL\Backup\MyDB_Backup.BAK'  
, @with='NORECOVERY'
```

### Restore Database with Encryption

```
EXEC master.dbo.xp_restore_database  
@database='MyDB'  
, @filename='C:\MSSQL\Backup\MyDB_Backup.BAK'  
, @encryptionkey='Password'
```

**Note:** LiteSpeed supports restore of encrypted backups created using LiteSpeed to machines which have LiteSpeed installed.

### Restore Files

```
exec master.dbo.xp_restore_database  
@database = 'MyDB'  
, @filename = 'C:\Program Files\Microsoft SQL  
Server\MSSQL.1\MSSQL\Backup\MyDB_200909111303_file.bak'  
, @filenumber = 1  
, @file = 'file1'  
, @file = 'file2'
```

### Restore a Filegroup and a File

```
exec master.dbo.xp_restore_database  
@database = 'MyDB'  
, @filegroup = 'PRIMARY'  
, @file = 'file1'  
, @filename = 'C:\Program Files\Microsoft SQL  
Server\MSSQL.1\MSSQL\Backup\MyDB_200909111243_filegroup.bak'
```

### Restore Database with Move

```
EXEC master.dbo.xp_restore_database  
@database='MyDB'  
, @filename = 'C:\MSSQL\Backup\MyDB_Backup.BAK'  
, @with = 'MOVE 'MyDB_Data' TO 'C:\MSSQL\Data\MyDB_Data.MDF''  
, @with = 'MOVE 'MyDB_log' TO 'C:\MSSQL\Data\MyDB_Log.LDF''
```

## Restore Database from Tape

```
EXEC master.dbo.xp_restore_database
@database = 'MyDB'
, @filename='\\.\TAPE0'
, @filenumber = 2
, @rewind = 1
, @unload = 0
```

## Restore a TSM archive

```
EXEC master.dbo.xp_restore_database
@database= 'Prod'
, @tsmclientnode = 'ClusterGroup'
, @tsmclientownerpwd= 'test16'
, @tsmobject= 'SLS_Mar\Prod\ (16)Thursday_14:14'
, @tsmconfigfile= 'C:\Program Files\Tivoli\tsm\baclient\dsm.opt'
, @tsmpointintime='2006-03-16 14:49:35'
, @tsmarchive=1
```

## Returns

0 (success) or non-zero (failure). Return codes represent the native error number returned from SQL Server for any errors encountered during the operation.

*To capture the output message, run the following:*

```
declare @rmsg varchar(999)
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output
select @rmsg
```

*To capture the output message and the result code, run the following:*

```
declare @rc int
declare @rmsg varchar(999)
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output,
@resultcode=@rc output
select @rc, @rmsg
```

## xp\_restore\_filelistonly

Returns a result set with a list of the database and log files contained in a LiteSpeed backup set.

## Syntax

### xp\_restore\_filelistonly (Disk or TSM)

```
EXEC master.dbo.xp_restore_filelistonly
( @filename = 'backup_file_name') [,...n]
[, @filenumber = n]
[, @tsmclientnode = 'TSM_client_node']
[, @tsmclientownerpwd = 'TSM_client_owner_password']
```

```
[, @tsmobject = 'TSM_object']  
[, @tsmconfigfile = 'TSM_configuration_file']  
[, @tsmpointintime = 'date_time' ]
```

### **xp\_restore\_filelistonly (Tape)**

```
EXEC master.dbo.xp_restore_filelistonly  
@filename = 'tape_device_name'  
[, @filenumber = n]
```

## **Arguments**

### **Tips:**

- To see the list of accepted arguments and data types for arguments, execute the following:  
`exec master.dbo.<procedure_name> show_help`
- To convert the script for use with the command-line utilities, execute the following:  
`exec master.dbo.<procedure_name> show_cmd, <xp_arguments>`

### **@filename**

Location and name of the backup file or device to back up or restore. You can supply multiple instances of this argument.

This argument accepts variables. See "LiteSpeed Variables" (page 271) for more information.

### **@filenumber**

*Disk restores:*

Specifies the particular backup to use when recasting, restoring, extracting or reading from files with multiple appended backups. You can run `xp_restore_headeronly` to query the files contained within the backup set given by `backup_file_name`.

*Tape restores:*

Identifies the backup set to be restored. For example, a file number of 1 indicates the first backup set on the backup medium, and a file number of 2 indicates the second backup set.

### **@tsmclientnode**

Specifies the TSM server LiteSpeed connects to during backups and restores. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.

### **@tsmclientownerpwd**

Specifies the TSM client owner user password. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.

### **@tsmconfigfile**

Specifies the TSM configuration file.

### **@tsmobject**

Defines the TSM filespace, high level and low level. This argument accepts the following format:

```
tsm_filespace\tsm_high_level\tsm_low_level
```

where:

- `tsm_filespace` is the logical space on the TSM server that contains a group of files. It can be the drive label name or UNC name.
- `tsm_high_level` specifies the directory path in which the file belongs.
- `tsm_low_level` specifies actual name of the file.

**Note:** You may only store one item the location specified by this argument. It is not possible to append an object to this location. You can use the `-I` command-line argument or `@init` to back up to a non-unique location.

### **@tsmpointintime**

Specifies the date for restore/to filter results. If it is not passed, LiteSpeed will choose the most recent archived backup. The format is `yyyy-mm-dd hh:mm:ss`.

**Note:** If the backup was a striped backup and the point-in-times of the various striped files are different (rare but can be different a second or so), then the most recent of the times must be chosen.

## Results

Column Name	Data Type	Description for backup sets
LogicalName	nvarchar(4000)	Logical name of the SQL Server data or log device.
PhysicalName	nvarchar(4000)	Full path of the SQL Server data or log device.
Type	nvarchar(4000)	Device Types: <ul style="list-style-type: none"> <li>• D—Data device</li> <li>• L—Transaction Log Device</li> </ul>
FileGroupName	nvarchar(4000)	Name of the filegroup the device files belong to.
Size	nvarchar(4000)	Size (in bytes) of the device file.
MaxSize	nvarchar(4000)	Value returned from SQL Server.
FileId	nvarchar(4000)	Identifier of the device files.
BackupSizeInBytes	nvarchar(4000)	Size (in bytes) of the backup for the device file.
FileGroupId	nvarchar(4000)	Identifier of the filegroup the device files belong to.

## Returns

0 (success) or non-zero (failure). Return codes represent the native error number returned from SQL Server for any errors encountered during the operation.

*To capture the output message, run the following:*

```
declare @rmsg varchar(999)
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output
select @rmsg
```

*To capture the output message and the result code, run the following:*

```
declare @rc int
declare @rmsg varchar(999)
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output,
@resultcode=@rc output
select @rc, @rmsg
```

## xp\_restore\_headeronly

Retrieves the backup header information for all LiteSpeed backups. The header information is sent as a row by the server for each backup on a given backup device in a table.

**Tip:** To retrieve information from TSM backups, also use `xp_view_tsmcontents`.

## Syntax

### xp\_restore\_headeronly (Disk)

```
EXEC master.dbo.xp_restore_headeronly  
[@filename = 'backup_file_name'] [,...n]  
[, @filenumber = n]  
[, @headerdetails = 'option']  
[, @attachedfiles = 0 | 1 | 2 | 3 ]
```

### xp\_restore\_headeronly (Tape)

```
EXEC master.dbo.xp_restore_headeronly  
@filename = 'tape_device_name'  
[, @filenumber = n]  
[, @headerdetails = 'option']  
[, @attachedfiles = 0 | 1 | 2 | 3 ]
```

### xp\_restore\_headeronly (TSM)

```
EXEC master.dbo.xp_restore_headeronly  
@tsmobject = 'TSM_object'  
, @tsmconfigfile = 'TSM_configuration_file'  
[, @tsmclientnode = 'clientnode_name']  
[, @tsmclientownerpwd = '*****']  
[, @tsmarchive = 0 | 1 ]  
[, @tsmpointintime = 'date_time']  
[, @attachedfiles = 0 | 1 | 2 | 3 ]  
[, @headerdetails = 'option']
```

## Arguments

### Tips:

- To see the list of accepted arguments and data types for arguments, execute the following:  
`exec master.dbo.<procedure_name> show_help`
- To convert the script for use with the command-line utilities, execute the following:  
`exec master.dbo.<procedure_name> show_cmd, <xp_arguments>`

### @attachedfiles

Lists files attached to a backup. This argument accepts one of the following values:

- 0—Backup header information only
- 1—Attached files only
- 2—Backup header information and a list of attached files
- 3—Attached directories and individual files outside of those directories



### **@filename**

Location and name of the backup file or device to back up or restore. You can supply multiple instances of this argument.

This argument accepts variables. See "LiteSpeed Variables" (page 271) for more information.

### **@filenumber**

*Disk restores:*

Specifies the particular backup to use when recasting, restoring, extracting or reading from files with multiple appended backups. You can run `xp_restore_headeronly` to query the files contained within the backup set given by `backup_file_name`.

*Tape restores:*

Identifies the backup set to be restored. For example, a file number of 1 indicates the first backup set on the backup medium, and a file number of 2 indicates the second backup set.

### **@headerdetails**

This argument accepts one of the following values:

- `backup`—Retrieves the backup header information.
- `attachedfiles`—Lists files attached to a backup.
- `attachedfileparams`—Lists attached directories and individual files outside of those directories.
- `all`—Retrieves all the backup header information.

### **@tsmarchive**

Specifies to store backup as a TSM archive. This argument accepts one of the following values:

- `0`—False
- `1`—True

### **@tsmconfigfile**

Specifies the TSM configuration file.

### **@tsmclientnode**

Specifies the TSM server LiteSpeed connects to during backups and restores. Not required, if specified in the options file or if backing up with the `Passwordaccess Generate` option.

### @tsmclientownerpwd

Specifies the TSM client owner user password. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.

### @tsmobject

Defines the TSM filespace, high level and low level. This argument accepts the following format:

tsm\_filespace\tsm\_high\_level\tsm\_low\_level

where:

- tsm\_filespace is the logical space on the TSM server that contains a group of files. It can be the drive label name or UNC name.
- tsm\_high\_level specifies the directory path in which the file belongs.
- tsm\_low\_level specifies actual name of the file.

**Note:** You may only store one item the location specified by this argument. It is not possible to append an object to this location. You can use the `-i` command-line argument or `@init` to back up to a non-unique location.

### @tsmpointintime

Specifies the date for restore/to filter results. If it is not passed, LiteSpeed will choose the most recent archived backup. The format is yyyy-mm-dd hh:mm:ss.

**Note:** If the backup was a striped backup and the point-in-times of the various striped files are different (rare but can be different a second or so), then the most recent of the times must be chosen.

## Examples

1. Display backup set information and attached files:

```
exec master.dbo.xp_restore_headeronly
@filename = N'C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\Backup\FASTCOMP_full12.bak'
, @attachedfiles = 2
```

2. List files attached to a tsm backup:

```
exec master.dbo.xp_restore_headeronly
@tsmconfigfile = N'C:\TSM\baclient\dsm.opt'
, @tsmobject = N'DFSSQLSINTP04\Modx\test'
, @attachedfiles = 1
```

## Results

xp\_restore\_headeronly displays the following information:

Column Name	Data Type	Description
FileNumber	Int	Number of the Backup within the LiteSpeed backup device.
BackupFormat	nvarchar(128)	Reserved field. Returns 1.
Guid	Uniqueidentifier	Backup Guid, uniquely identifies LiteSpeed backup sets.
BackupName	nvarchar(128)	Backup set name.
BackupDescription	nvarchar(128)	Backup set description.
BackupType	nvarchar(128)	Backup type: <ul style="list-style-type: none"> <li>• 1—Database</li> <li>• 2—Transaction Log</li> <li>• 4—File</li> <li>• 5—Differential Database</li> <li>• 6—Differential File</li> <li>• 7—Partial</li> <li>• 8—Partial Differential</li> </ul>
ExpirationDate	Datetime	Expiration date for the backup set.
Compressed	Tinyint	0—No compression. 1—Compressed
Position	Smallint	Position of the backup set in the volume (for use with the FILE = option).
DeviceType	Tinyint	Virtual Device > 7—Logical 107—Physical
UserName	nvarchar(128)	Username that performed the backup operation.
ServerName	nvarchar(128)	Name of the server that wrote the backup set.
DatabaseName	nvarchar(128)	Name of the database that

Column Name	Data Type	Description
		was backed up.
DatabaseVersion	int	Version of the database from which the backup was created.
DatabaseCreationDate	datetime	Date and time the database was created.
BackupSize	numeric (20,0)	Size of the backup, in bytes.
FirstLSN	numeric (25,0)	Log sequence number of the first transaction in the backup set. NULL for file backups.
LastLSN	numeric (25,0)	Log sequence number of the last transaction in the backup set. NULL for file backups.
CheckpointLSN	numeric (25,0)	Log sequence number of the most recent checkpoint at the time the backup was created.
DifferentialBaseLSN	numeric (25,0)	Log sequence number of the most recent full database backup.
BackupStartDate	datetime	Date and time that the backup operation began.
BackupFinishDate	datetime	Date and time that the backup operation finished.
SortOrder	Smallint	Server sort order. This column is valid for database backups only. Provided for backward compatibility.
CodePage	Smallint	Server code page or character set used by the server.
CompatibilityLevel	Tinyint	Compatibility level setting of the database from which the backup was created.
SoftwareVendorId	Int	Software vendor identification number. For

Column Name	Data Type	Description
		SQL Server, this number is 4608 (or hexadecimal 0x1200).
SoftwareVersionMajor	Int	Major version number of the server that created the backup set.
SoftwareVersionMinor	Int	Minor version number of the server that created the backup set.
SoftwareVersionBuild	Int	Build number of the server that created the backup set.
MachineName	nvarchar(128)	Name of the server that wrote the backup set.
BindingID	Uniqueidentifier	Binding ID for the database.
RecoveryForkID	Uniqueidentifier	ID for the current recovery fork for this backup.
Encryption	Int	Indicates if backup is encrypted <ul style="list-style-type: none"> <li>• 0—not encrypted</li> <li>• 1—encrypted</li> </ul>
IsCopyOnly	nvarchar(128)	Indicates if the backup is a copy-only backup.

## Returns

0 (success) or non-zero (failure). Return codes represent the native error number returned from SQL Server for any errors encountered during the operation.

*To capture the output message, run the following:*

```
declare @rmsg varchar(999)
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output
select @rmsg
```

*To capture the output message and the result code, run the following:*

```
declare @rc int
declare @rmsg varchar(999)
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output,
```

```
@resultcode=@rc output  
select @rc, @rmsg
```

## xp\_restore\_setinfo

Returns information about the stripe set to which the backup file belongs.

### Syntax

#### xp\_restore\_setinfo (Disk or Tape)

```
EXEC master.dbo.xp_restore_setinfo  
    @filename = 'file_name'  
[, @filenumber = n]
```

#### xp\_restore\_setinfo (TSM)

```
EXEC master.dbo.xp_restore_setinfo  
@tsmobject = 'TSM_object'  
, @tsmconfigfile = 'TSM_configuration_file'  
[, @tsmclientnode = 'clientnode_name']  
[, @tsmclientownerpwd = '****']  
[, @tsmarchive = 0 | 1 ]  
[, @tsmpointintime = 'date_time']
```

### Arguments

#### @filename

Location and name of the backup file or device to back up or restore. You can supply multiple instances of this argument.

This argument accepts variables. See "LiteSpeed Variables" (page 271) for more information.

#### @filenumber

Specifies the particular backup to use when recasting, restoring, extracting or reading from files with multiple appended backups. You can run `xp_restore_headeronly` to query the files contained within the backup set given by `backup_file_name`.

#### @tsmarchive

Specifies to store backup as a TSM archive. This argument accepts one of the following values:

- 0—False
- 1—True

### **@tsmconfigfile**

Specifies the TSM configuration file.

### **@tsmclientnode**

Specifies the TSM server LiteSpeed connects to during backups and restores. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.

### **@tsmclientownerpwd**

Specifies the TSM client owner user password. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.

### **@tsmobject**

Defines the TSM filesystem, high level and low level. This argument accepts the following format:

`tsm_filespace\tsm_high_level\tsm_low_level`

where:

- `tsm_filespace` is the logical space on the TSM server that contains a group of files. It can be the drive label name or UNC name.
- `tsm_high_level` specifies the directory path in which the file belongs.
- `tsm_low_level` specifies actual name of the file.

**Note:** You may only store one item the location specified by this argument. It is not possible to append an object to this location. You can use the `-I` command-line argument or `@init` to back up to a non-unique location.

### **@tsmpointintime**

Specifies the date for restore/to filter results. If it is not passed, LiteSpeed will choose the most recent archived backup. The format is `yyyy-mm-dd hh:mm:ss`.

**Note:** If the backup was a striped backup and the point-in-times of the various striped files are different (rare but can be different a second or so), then the most recent of the times must be chosen.

## **Example**

```
EXEC master.dbo.xp_restore_setinfo  
@filename = 'C:\SQLServerBackups\CD3.bak'
```

## Results

Column Name	Data Type	Description
FormatVersion	Int	Actual version of LiteSpeed binary format used to create the backup.
StripeGUID	Uniqueidentifier	Unique identifier of LiteSpeed stripe set.
StripeNumber	Int	Backup file number within the stripe set.
StripeCount	Int	Number of backup files in the stripe set.

## Returns

0 (success) or non-zero (failure).

## xp\_restore\_log

Restores LiteSpeed transaction log backups taken using the xp\_backup\_log command. Files and filegroups may also be restored either from a file or filegroup LiteSpeed backup operation, or from a full database backup operation using xp\_backup\_log. When restoring files or filegroups, you must apply a transaction log using xp\_restore\_log. In addition, file differential backups can be restored after a full file restore using LiteSpeed.

### Notes:

- A database cannot be restored unless the restore process has exclusive access to the database. No user connections can exist when performing a database restore.
- You cannot restore filegroups to a point in time.

## Syntax

### xp\_restore\_log (Disk or TSM)

```
EXEC master.dbo.xp_restore_log
@database = 'database_name'
[, @filename = 'backup_file_name'] [,...n]
[, ( @encryptionkey = 'encryption_key' | @jobp = 'encrypted_key' )]
[, @filenumber = n]
[, @with = 'additional_with_parameters'] [,...n]
[, @logging = 0 | 1 | 2 ]
[, @affinity = 0..2147483648]
[, @throttle = 1..100]
[, @ioflag = 'DISK_RETRY_COUNT=n']
[, @ioflag = 'DISK_RETRY_WAIT=n']
[, @buffercount = 'buffer_count']
[, @maxtransfer_size = 'maximum_transfer_size']
[, @attachedfile = 'pathname'] [,...n]
```



```
[, @tsmclientnode = 'TSM_client_node']  
[, @tsmclientownerpwd = 'TSM_client_owner_password']  
[, @tsmobject = 'TSM_object']  
[, @tsmconfigfile = 'TSM_configuration_file']  
[, @tsmpointintime = 'date_time' ]  
[, @returndetails = 0 | 1 ]
```

## xp\_restore\_log (Tape)

```
EXEC master.dbo.xp_restore_log  
@database = 'database_name'  
, @filename = 'tape_device_name'  
[, @rewind = 0 | 1 ]  
[, @unload = 0 | 1 ]  
[, @encryptionkey = 'encryption_key']  
[, @filenumber = n]  
[, @with = 'additional_with_parameters'] [,...n]  
[, @logging = 0 | 1 | 2 ]  
[, @affinity = 0..2147483648]  
[, @throttle = 1..100]  
[, @buffercount = 'buffer_count']  
[, @maxtransfersize = 'maximum_transfer_size']  
[, @attachedfile = 'pathname'] [,...n]  
[, @returndetails = 0 | 1 ]
```

## Arguments

### Tips:

- To see the list of accepted arguments and data types for arguments, execute the following:  
`exec master.dbo.<procedure_name> show_help`
- To convert the script for use with the command-line utilities, execute the following:  
`exec master.dbo.<procedure_name> show_cmd, <xp_arguments>`

### @affinity

Processor affinity designates specific processors to run LiteSpeed, while not allowing LiteSpeed to run on the remaining processors.

This argument accepts decimal values and hexadecimal values. If a value begins with "0x" it is interpreted as hexadecimal. A positive 64-bit integer value translates to a binary mask where a value of 1 designates the corresponding processor to be able to run the LiteSpeed process.

**Note:** 32-bit Windows is internally limited to a 32-bit mask.

For example, you need to select processors 2, 3, and 6 for use with LiteSpeed. Number the bits from the right to left. The rightmost bit represents the first processor. Set the second, third, and sixth bits to 1 and all other bits to 0. The result is binary 100110, which is decimal 38 or hexadecimal 0x26. Review the following for additional information:

Decimal Value	Binary Bit Mask	Allow LiteSpeed Threads on Processors
0	0	All (default)
1	1	1
3	11	1 and 2
7	111	1, 2 and 3
38	100110	2, 3, and 6
205	11001101	1, 3, 4, 7, and 8

**Tip:** Before you start tuning the CPU Throttle or Affinity parameters to adjust backup performance, try limiting the number of threads. If you decide to use an affinity value other than default, it is recommended that you limit the threading as well. You may also want to consider using Adaptive Compression to maintain backup performance. See "Adaptive Compression" (page 55) for more information.

### @attachedfile

Specifies filepaths to include in both backup and restore operations. The filepath can be either a single file or a directory. If it is a directory, then LiteSpeed recursively includes all files and subdirectories. All attached files are encrypted and compressed, with all pertinent backup parameters supported. This feature works for disk, tape, TSM, and Double Click Restore as well. You can supply multiple instances of this argument.

When used within the context of a restore operation, the path parameter can be expanded to include a new destination. This form will take the syntax of <file\_path> to <new\_file\_path>. The new filepath can be used to specify a new location but cannot rename a file.

This argument only restores the attached files. It does not restore the database, just the files that were attached to that backup.

#### Notes:

- The original entire directory path need not be supplied (e.g. c: to c:\testadSattsm is allowed).
- c:\testad to testadr would restore all files in directory c:\testad to c:\testadr.

### @buffercount

Number of SQL Server buffers available for a LiteSpeed operation. The default value is 20.

### @database

Name of database to be backed up or restored.

This parameter specifies a database:

- to be backed up (xp\_backup\_database and xp\_slsSmartDiff)
- containing the transaction log to be backed up (xp\_backup\_log)
- to be restored (xp\_restore\_database and xp\_restore\_log)
- on which you wish to check the progress of an activity (xp\_slsReadProgress)
- for which you want to delete old backups (xp\_slsSmartCleanup)

If supplied as a variable (@database), this name can be specified either as a string constant (@database = database name) or as a variable of character string data type, except for the ntext or text data types.

### **@encryptionkey**

Identifies the restoring user as a valid owner of the backup file.

**Caution:** If you use the wrong encryption key, the restore will fail. If you forget the encryption key, you will not be able to restore or read the backup.

### **@filename**

Location and name of the backup file or device to back up or restore. You can supply multiple instances of this argument.

This argument accepts variables. See "LiteSpeed Variables" (page 271) for more information.

### **@filenumber**

*Disk restores:*

Specifies the particular backup to use when recasting, restoring, extracting or reading from files with multiple appended backups. You can run xp\_restore\_headeronly to query the files contained within the backup set given by backup\_file\_name.

*Tape restores:*

Identifies the backup set to be restored. For example, a file number of 1 indicates the first backup set on the backup medium, and a file number of 2 indicates the second backup set.

### **@ioflag**

Specifies if LiteSpeed should wait and retry the read or write operation on failure. You can define retry options using the following parameters:

- DISK\_RETRY\_COUNT—Specifies the number of times that a specific operation will be retried on failure. The default is 4 retries, the maximum allowed setting is 1000.

- **DISK\_RETRY\_WAIT**—Specifies the number of seconds to wait immediately following a failure before retrying. The default is 15 seconds, the maximum allowed setting is 300.

**Note:** This functionality is only available for disk operations.

For example, to specify a database backup where each failure can be retried once after a 30-second wait:

```
EXEC master.dbo.xp_backup_database  
@filename='c:\test.bkp'  
, @database='test'  
, @ioflag='DISK_RETRY_COUNT=1'  
, @ioflag='DISK_RETRY_WAIT=30'
```

See "Network Resilience" (page 58) for more information.

### **@jobp**

Specifies the `encrypted_key`. (Similar to `@EncryptionKey`).

You can use `xp_encrypt_backup_key` (page 171) to convert the password (`encryption_key`) for use with `@jobp`. The original password (or `encrypted_key` generated by `xp_encrypt_restore_key` (page 171)) must be used to restore a backup.

### **@logging**

Writes a log file for the operation. This argument accepts one of the following values:

- 0—Logging off.
- 1 or any odd value—Logging on. Log file is removed on success.
- 2 or any even value—Logging on.

The default output directory is *C:\Documents and Settings\All Users\Application Data\Quest Software\LiteSpeed\SQL Server\Logs*. To log to a different directory add `@Trace='logpath=path'`.

See Configure Logging in LiteSpeed (page 285) for information about LiteSpeed logging.

### **@maxtransfersize**

Specifies the data size in bytes for each transfer when communicating with SQL Server. The size can be any multiple of 64KB in the range from 64KB to 4MB. The default is 1048576 bytes.

### **@priority**

Specifies the priority of the LiteSpeed process compared to other processes running on the same server. This argument accepts one of the following values:

- -1—Below Normal
- 0—Normal (Default)

- 1—AboveNormal
- 2—High

### @returndetails

Generates a single-row result set.

- 0—False (default)
- 1—True

The result set contains the following details:

Column Name	Data Type	Description
Database	nvarchar (128)	Database name.
Operation	nvarchar (30)	Operation type: Backup or Restore.
Threads	tinyint	The number of threads used for a LiteSpeed backup.
CompressionLevel	tinyint	Compression level used for compressing the backup. The compression level can be NULL, if backed up with Adaptive Compression.
AdaptiveCompression	nvarchar (max)	Adaptive Compression option used for compressing the backup: 'speed' or 'size'.
MaxTransferSize	int	The data size in bytes for each transfer when communicating with SQL Server. The size can be any multiple of 64 KB in the range from 64 KB to 4 MB. The default is 1048576 bytes.
BaseSize	int	The smallest chunk of memory LiteSpeed attempts to write to disk at any given time.
BufferCount	smallint	The number of SQL Server buffers available for a LiteSpeed operation.
StripeCount	smallint	Number of backup files in the stripe set.
OverlappedBuffers	tinyint	The number of buffers that any single VDI thread can use at a time.
CPUSeconds	numeric (18, 3)	Processor time used by the LiteSpeed operation.
ElapsedSeconds	numeric	Duration of the operation.

Column Name	Data Type	Description
	(18, 3)	
NativeSize	bigint	Backup size (in bytes) without LiteSpeed compression.
BackupSize	bigint	Size of the backup (in bytes).

**Tip:** In Toad, you can use Group Execute to produce a single result set for several server instances.

### **@rewind**

Applies only to backing up and restoring tape. This argument accepts one of the following values:

- 0—Leave the tape unwound (default)
- 1—Rewind the tape after writing/reading

### **@throttle**

Specifies the maximum CPU usage allowed. The argument accepts an integer value between 1 and 100. The default value is 100. This is the percentage of the total amount of CPU usage (across all enabled processors) available.

**Tip:** Before you start tuning the CPU Throttle or Affinity parameters to adjust backup performance, try limiting the number of threads. If you decide to use an affinity value other than default, it is recommended that you limit the threading as well. You may also want to consider using Adaptive Compression to maintain backup performance. See "Adaptive Compression" (page 55) for more information.

### **@tsmclientnode**

Specifies the TSM server LiteSpeed connects to during backups and restores. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.

### **@tsmclientownerpwd**

Specifies the TSM client owner user password. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.

### **@tsmconfigfile**

Specifies the TSM configuration file.

### **@tsmmanagementclass**

Specifies the TSM management class. If not specified, LiteSpeed uses the default management class.

### **@tsmobject**

Defines the TSM filespace, high level and low level. This argument accepts the following format:

```
tsm_filespace\tsm_high_level\tsm_low_level
```

where:

- `tsm_filespace` is the logical space on the TSM server that contains a group of files. It can be the drive label name or UNC name.
- `tsm_high_level` specifies the directory path in which the file belongs.
- `tsm_low_level` specifies actual name of the file.

**Note:** You may only store one item the location specified by this argument. It is not possible to append an object to this location. You can use the `-I` command-line argument or `@init` to back up to a non-unique location.

### **@tsmpointintime**

Specifies the date for restore/to filter results. If it is not passed, LiteSpeed will choose the most recent archived backup. The format is `yyyy-mm-dd hh:mm:ss`.

**Note:** If the backup was a striped backup and the point-in-times of the various striped files are different (rare but can be different a second or so), then the most recent of the times must be chosen.

### **@unload**

Applies to tape backups and restores. This argument accepts one of the following values:

- 0—Keep tape loaded (default)
- 1—Unload and eject tape from the drive after operation

### **@with**

Each `@with` argument should be a syntactically complete and correct statement. Please refer to the SQL Server Transact-SQL backup and restore documentation for the syntax and usage.

The supported formats are:

- `@with='PARAMETER'`
- `@with='PARAMETER="accepted_value"'`

### **Notes:**

- Extended stored procedure arguments are limited to 255 characters. If you need more than 255 characters, use multiple `@with` arguments.
- Do not supply the `@with` parameter if no additional features are required.

This extended stored procedure accepts the following @with parameters:

Parameter	Description
NORECOVERY	<p>Instructs the restore operation to not roll back any uncommitted transactions. Either the NORECOVERY or STANDBY option must be specified if another transaction log has to be applied. If NORECOVERY, RECOVERY, or STANDBY is not specified, RECOVERY is the default.</p> <p>SQL Server requires that the WITH NORECOVERY option is used on all but the final xp_restore_log statement when restoring a database backup and multiple transaction logs using LiteSpeed, or when multiple xp_restore_database or xp_restore_log statements are needed (for example, a full database backup followed by a differential database backup).</p> <p><b>Note:</b> When specifying the NORECOVERY option, the database is not usable in this intermediate, non-recovered state.</p> <p>When used with a file or filegroup restore operation, NORECOVERY forces the database to remain in the restoring state after the restore operation. This is useful in either of these situations:</p> <ul style="list-style-type: none"> <li>• a restore script is being run and the log is always being applied.</li> <li>• a sequence of file restores is used and the database is not intended to be usable between two of the restore operations.</li> </ul>
RECOVERY	<p>Instructs the restore operation to roll back any uncommitted transactions. After the recovery process, the database is ready for use.</p> <p>If subsequent LiteSpeed restore operations (xp_restore_log or xp_restore_database from differential) are planned, NORECOVERY or STANDBY should be specified instead.</p> <p>If NORECOVERY, RECOVERY, or STANDBY is not specified, RECOVERY is the default.</p> <p>When restoring backup sets from an earlier version of SQL Server, a database upgrade may be required. This upgrade is performed automatically when WITH RECOVERY is specified.</p>
RESTRICTED_USER	<p>When used in conjunction with recovery (another with param and the default) leaving a usable database, this restricts access for the restored database to members of the db_owner, dbcreator, or sysadmin roles.</p>
STANDBY	<p>STANDBY = 'undo_file_name'</p> <p>Specifies the undo file name so the recovery effects can be undone. The size required for the undo file depends on the volume of undo actions resulting from uncommitted transactions. If you do not specify NORECOVERY, RECOVERY, or STANDBY, LiteSpeed defaults to RECOVERY.</p>



Parameter	Description
	<p>STANDBY allows a database to be brought up for read-only access between transaction log restores and can be used with either warm standby server situations or special recovery situations in which it is useful to inspect the database between log restores.</p> <p>If the specified undo file name does not exist, LiteSpeed creates it. If the file does exist, LiteSpeed overwrites it.</p> <p>The same undo file can be used for consecutive LiteSpeed restores of the same database.</p> <p><b>Note:</b> If free disk space is exhausted on the drive containing the specified undo file name, the LiteSpeed restore operation stops.</p>
STOPAT	<p>STOPAT = date_time   @date_time_var</p> <p>Specifies that the database be restored to the state it was in as of the specified date and time. If a variable is used for STOPAT, the variable must be varchar, char, smalldatetime, or datetime data type.</p> <p>Only transaction log records written before the specified date and time are applied to the database.</p> <p><b>Note:</b> If you specify a STOPAT time that is beyond the end of the xp_restore_log operation, the database is left in an unrecovered state, just as if xp_restore_log had been run with NORECOVERY.</p>
STOPATMARK	<p>STOPATMARK = 'mark_name' [ AFTER Datetime ]</p> <p>Specifies recovery to the specified mark, including the transaction that contains the mark.</p> <p>If AFTER Datetime is omitted, recovery stops at the first mark with the specified name. If AFTER Datetime is specified, recovery stops at the first mark having the specified name exactly at or after Datetime.</p>
STOPBEFOREMARK	<p>STOPBEFOREMARK = 'mark_name' [ AFTER Datetime ]</p> <p>Specifies recovery to the specified mark but does not include the transaction that contains the mark.</p> <p>If AFTER Datetime is omitted, recovery stops at the first mark with the specified name. If AFTER Datetime is specified, recovery stops at the first mark having the specified name exactly at or after Datetime.</p>
CHECKSUM	<p>Causes checksums to be verified when a LiteSpeed backup is created.</p> <p><b>Note:</b> When you restore a backup containing checksum, it is automatically checked. If you do not want to check the checksums during a restore, supply 'NO_CHECKSUM'.</p>
PASSWORD	<p>Specifies the password for the backup set.</p>

## Examples

### 1. Restore Log to Log Mark:

```
EXEC master.dbo.xp_restore_log
@database='MyDB'
, @filename = 'C:\MSSQL\Backup\MyDB_Backup.BAK'
, @with = 'RECOVERY'
, @with = 'STOPBEFOREMARK= ''LogMark'''
```

### 2. Restore Log to Point in Time:

```
EXEC master.dbo.xp_restore_log
@database='MyDB'
, @filename = 'C:\MSSQL\Backup\MyDB_Backup.BAK'
, @with = 'RECOVERY'
, @with = 'STOPAT = ''2003-03-19 11:10:57.000'''
```

## Returns

0 (success) or non-zero (failure). Return codes represent the native error number returned from SQL Server for any errors encountered during the operation.

*To capture the output message, run the following:*

```
declare @rmsg varchar(999)
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output
select @rmsg
```

*To capture the output message and the result code, run the following:*

```
declare @rc int
declare @rmsg varchar(999)
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output,
@resultcode=@rc output
select @rc, @rmsg
```

## xp\_restore\_verifyonly

Verifies the backup, but does not restore the backup. It checks to see that the backup set is complete and that all volumes are readable. If the backup is valid, LiteSpeed returns the message from SQL Server: "The backup set is valid."

## Syntax

### xp\_restore\_verifyonly (Disk or TSM)

```
EXEC master.dbo.xp_restore_verifyonly
@filename = 'backup_file_name' [,...n]
[, @filenumber = n]
[, @encryptionkey = 'encryption_key'| @jobp = 'encrypted_key' ]
[, @logging = 0 | 1 | 2 ]
```

```
[, @affinity = 0..2147483648]
[, @throttle = 1..100]
[, @ioflag = 'DISK_RETRY_COUNT=n']
[, @ioflag = 'DISK_RETRY_WAIT=n']
[, @buffercount = 'buffer_count']
[, @maxtransfersize = 'maximum_transfer_size']
[, @tsmclientnode = 'TSM_client_node']
[, @tsmclientownerpwd = 'TSM_client_owner_password']
[, @tsmobject = 'TSM_object']
[, @tsmconfigfile = 'TSM_configuration_file']
[, @tsmarchive = 1 | 0 ]
[, @tsmpointintime = 'date_time']
[, @returndetails = 0 | 1]
```

### xp\_restore\_verifyonly (Tape)

```
EXEC master.dbo.xp_restore_verifyonly
@filename = 'tape_device_name'
[, @filenumber = n]
[, @encryptionkey = 'encryption_key' ]
[, @logging = 0 | 1 | 2 ]
[, @affinity = 0..2147483648]
[, @throttle = 1..100]
[, @unload = 0 | 1 ]
[, @buffercount = 'buffer_count']
[, @maxtransfersize = 'maximum_transfer_size']
[, @returndetails = 0 | 1]
```

## Arguments

### Tips:

- To see the list of accepted arguments and data types for arguments, execute the following:  
`exec master.dbo.<procedure_name> show_help`
- To convert the script for use with the command-line utilities, execute the following:  
`exec master.dbo.<procedure_name> show_cmd, <xp_arguments>`

### @affinity

Processor affinity designates specific processors to run LiteSpeed, while not allowing LiteSpeed to run on the remaining processors.

This argument accepts decimal values and hexadecimal values. If a value begins with "0x" it is interpreted as hexadecimal. A positive 64-bit integer value translates to a binary mask where a value of 1 designates the corresponding processor to be able to run the LiteSpeed process.

**Note:** 32-bit Windows is internally limited to a 32-bit mask.

For example, you need to select processors 2, 3, and 6 for use with LiteSpeed. Number the bits from the right to left. The rightmost bit represents the first processor. Set the second, third, and

sixth bits to 1 and all other bits to 0. The result is binary 100110, which is decimal 38 or hexadecimal 0x26. Review the following for additional information:

Decimal Value	Binary Bit Mask	Allow LiteSpeed Threads on Processors
0	0	All (default)
1	1	1
3	11	1 and 2
7	111	1, 2 and 3
38	100110	2, 3, and 6
205	11001101	1, 3, 4, 7, and 8

**Tip:** Before you start tuning the CPU Throttle or Affinity parameters to adjust backup performance, try limiting the number of threads. If you decide to use an affinity value other than default, it is recommended that you limit the threading as well. You may also want to consider using Adaptive Compression to maintain backup performance. See "Adaptive Compression" (page 55) for more information.

### **@buffercount**

Number of SQL Server buffers available for a LiteSpeed operation. The default value is 20.

### **@encryptionkey**

Identifies the restoring user as a valid owner of the backup file.

**Caution:** If you use the wrong encryption key, the restore will fail. If you forget the encryption key, you will not be able to restore or read the backup.

### **@filename**

Location and name of the backup file or device to back up or restore. You can supply multiple instances of this argument.

This argument accepts variables. See "LiteSpeed Variables" (page 271) for more information.

### **@filenumber**

*Disk restores:*

Specifies the particular backup to use when recasting, restoring, extracting or reading from files with multiple appended backups. You can run `xp_restore_headeronly` to query the files contained within the backup set given by `backup_file_name`.

*Tape restores:*

Identifies the backup set to be restored. For example, a file number of 1 indicates the first backup set on the backup medium, and a file number of 2 indicates the second backup set.

### **@ioflag**

Specifies if LiteSpeed should wait and retry the read or write operation on failure. You can define retry options using the following parameters:

- **DISK\_RETRY\_COUNT**—Specifies the number of times that a specific operation will be retried on failure. The default is 4 retries, the maximum allowed setting is 1000.
- **DISK\_RETRY\_WAIT**—Specifies the number of seconds to wait immediately following a failure before retrying. The default is 15 seconds, the maximum allowed setting is 300.

**Note:** This functionality is only available for disk operations.

For example, to specify a database backup where each failure can be retried once after a 30-second wait:

```
EXEC master.dbo.xp_backup_database  
@filename='c:\test.bkp'  
, @database='test'  
, @ioflag='DISK_RETRY_COUNT=1'  
, @ioflag='DISK_RETRY_WAIT=30'
```

See "Network Resilience" (page 58) for more information.

### **@jobp**

Specifies the encrypted\_key. (Similar to @EncryptionKey).

You can use xp\_encrypt\_backup\_key (page 171) to convert the password (encryption\_key) for use with @jobp. The original password (or encrypted\_key generated by xp\_encrypt\_restore\_key (page 171)) must be used to restore a backup.

### **@logging**

Writes a log file for the operation. This argument accepts one of the following values:

- 0—Logging off.
- 1 or any odd value—Logging on. Log file is removed on success.
- 2 or any even value—Logging on.

The default output directory is *C:\Documents and Settings\All Users\Application Data\Quest Software\LiteSpeed\SQL Server\Logs*. To log to a different directory add @Trace='logpath=path'.

See Configure Logging in LiteSpeed (page 285) for information about LiteSpeed logging.

## @maxtransfersize

Specifies the data size in bytes for each transfer when communicating with SQL Server. The size can be any multiple of 64KB in the range from 64KB to 4MB. The default is *1048576* bytes.

## @returndetails

Generates a single-row result set.

- 0—False (default)
- 1—True

The result set contains the following details:

Column Name	Data Type	Description
Database	nvarchar (128)	Database name.
Operation	nvarchar (30)	Operation type: Backup or Restore.
Threads	tinyint	The number of threads used for a LiteSpeed backup.
CompressionLevel	tinyint	Compression level used for compressing the backup. The compression level can be NULL, if backed up with Adaptive Compression.
AdaptiveCompression	nvarchar (max)	Adaptive Compression option used for compressing the backup: 'speed' or 'size'.
MaxTransferSize	int	The data size in bytes for each transfer when communicating with SQL Server. The size can be any multiple of 64 KB in the range from 64 KB to 4 MB. The default is 1048576 bytes.
BaseSize	int	The smallest chunk of memory LiteSpeed attempts to write to disk at any given time.
BufferCount	smallint	The number of SQL Server buffers available for a LiteSpeed operation.
StripeCount	smallint	Number of backup files in the stripe set.
OverlappedBuffers	tinyint	The number of buffers that any single VDI thread can use at a time.

Column Name	Data Type	Description
CPUSeconds	numeric (18, 3)	Processor time used by the LiteSpeed operation.
ElapsedSeconds	numeric (18, 3)	Duration of the operation.
NativeSize	bigint	Backup size (in bytes) without LiteSpeed compression.
BackupSize	bigint	Size of the backup (in bytes).

**Tip:** In Toad, you can use Group Execute to produce a single result set for several server instances.

### **@throttle**

Specifies the maximum CPU usage allowed. The argument accepts an integer value between 1 and 100. The default value is 100. This is the percentage of the total amount of CPU usage (across all enabled processors) available.

**Tip:** Before you start tuning the CPU Throttle or Affinity parameters to adjust backup performance, try limiting the number of threads. If you decide to use an affinity value other than default, it is recommended that you limit the threading as well. You may also want to consider using Adaptive Compression to maintain backup performance. See "Adaptive Compression" (page 55) for more information.

### **@tsmarchive**

Specifies to store backup as a TSM archive. This argument accepts one of the following values:

- 0—False
- 1—True

### **@tsmclientnode**

Specifies the TSM server LiteSpeed connects to during backups and restores. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.

### **@tsmclientownerpwd**

Specifies the TSM client owner user password. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.

### **@tsmconfigfile**

Specifies the TSM configuration file.

### **@tsmobject**

Defines the TSM filespace, high level and low level. This argument accepts the following format:

```
tsm_filespace\tsm_high_level\tsm_low_level
```

where:

- `tsm_filespace` is the logical space on the TSM server that contains a group of files. It can be the drive label name or UNC name.
- `tsm_high_level` specifies the directory path in which the file belongs.
- `tsm_low_level` specifies actual name of the file.

**Note:** You may only store one item the location specified by this argument. It is not possible to append an object to this location. You can use the `-I` command-line argument or `@init` to back up to a non-unique location.

### **@tsmpointintime**

Specifies the date for restore/to filter results. If it is not passed, LiteSpeed will choose the most recent archived backup. The format is `yyyy-mm-dd hh:mm:ss`.

**Note:** If the backup was a striped backup and the point-in-times of the various striped files are different (rare but can be different a second or so), then the most recent of the times must be chosen.

### **@unload**

Applies to tape backups and restores. This argument accepts one of the following values:

- 0—Keep tape loaded (default)
- 1—Unload and eject tape from the drive after operation

### **@with**

Each `@with` argument should be a syntactically complete and correct statement. Please refer to the SQL Server Transact-SQL backup and restore documentation for the syntax and usage.

The supported formats are:

- `@with='PARAMETER'`
- `@with='PARAMETER="accepted_value"`

#### **Notes:**

- Extended stored procedure arguments are limited to 255 characters. If you need more than 255 characters, use multiple `@with` arguments.
- Do not supply the `@with` parameter if no additional features are required.



This extended stored procedure accepts the following @with parameters:

Parameter	Description
CHECKSUM	Causes checksums to be verified when a LiteSpeed backup is created. <b>Note:</b> When you restore a backup containing checksum, it is automatically checked. If you do not want to check the checksums during a restore, supply 'NO_CHECKSUM'.
PASSWORD	Specifies the password for the backup set.

## Example

```
EXEC master.dbo.xp_restore_verifyonly  
  
@filename='C:\MSSQL\Backup\MyDB_Backup.BAK'
```

## Returns

0 (success) or non-zero (failure). Return codes represent the native error number returned from SQL Server for any errors encountered during the operation.

*To capture the output message, run the following:*

```
declare @rmsg varchar(999)  
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output  
select @rmsg
```

*To capture the output message and the result code, run the following:*

```
declare @rc int  
declare @rmsg varchar(999)  
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output,  
@resultcode=@rc output  
select @rc, @rmsg
```

## xp\_slsCreateDCR

This stored procedure converts a Native or LiteSpeed backup file to a Double Click Restore backup, a self-executing and self-extracting backup that can be restored on a server instance that does not have LiteSpeed installed. It also performs a rename on the file if applicable. See "Double Click Restore Executables" (page 53) for more information.

**Note:** A Double Click Restore can only be created for a disk file.

## Syntax

```
exec xp_slsCreateDCR  
@FileName='<path>'  
[, @doubleclick = 1 | 2 ]
```

## Agruments

### @filename

Specifies the path to the backup.

### @doubleclick

Creates a Double Click Restore executable. This argument accepts one of the following values:

- 1—Converts a LiteSpeed backup to a Double Click Restore executable, if the size of the backup is less than 4GB. If a backup is a native SQL Server backup or it is larger than 4 GB, it creates a Double Click Restore loader in the same location.
- 2—Creates a Double Click Restore loader in the same location.

See "Double Click Restore Executables" (page 53) for more information.

## Example

```
exec xp_slscreatedcr  
@filename = 'c:\mybackups\accounting.bak'
```

## Returns

0 (success) or non-zero (failure).

*To capture the output message, run the following:*

```
declare @rmsg varchar(999)  
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output  
select @rmsg
```

*To capture the output message and the result code, run the following:*

```
declare @rc int  
declare @rmsg varchar(999)  
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output,  
@resultcode=@rc output  
select @rc, @rmsg
```

## xp\_slFastCompression

xp\_slFastCompression (previously known as xp\_slSmartDiff) performs a full or differential database backup using Fast Compression technology. See "Fast Compression" (page 48) for more information.

## Syntax

```
EXEC master.dbo.xp_slFastCompression  
@database = 'database_name'
```

```
, @BackupDirectory = 'backup_directory'
{ @ForceFull = 0 | 1
| @ForceDifferential = 0 | 1
| ( @ExtentsChgRatioRequireFull = '1%'..'100%'
| @DiffToFullRatioRequireFull = '1%'..'100%')
[, @CheckForFullBackup = 0 | 1 ]
[, @ElapsedDaysRequireFull = 1..n]
([, @SpecificDaysForbidFull = '1'...'7' ] [,...n])}
[, @FullBackupEscalation = 0 | 1 ]
[, @SearchAlternateBackup = 'backup_directory' ]
[, @MirrorDirectory = 'mirror_directory' ] [,...n]
[, @AppendDifferential = 0 | 1 ]
[, @Verify = 'Last' | 'Full' | 'Last,Full' | 'All']
```

#### Other LiteSpeed parameters

```
[, @retaindays = 0..2147483647 ]
[, @expiration = 'date_time']
[, @desc = 'backup_description']
[, @backupname = 'backupset_name']
[, @threads = 1..32]
[, @encryptionkey = 'encryption_key']
[, @cryptlevel = 'encryption_level']
[, @file = 'logical_file_name' ] [,...n]
[, @filegroup = 'logical_filegroup_name' ] [,...n]
[, @priority = -1 | 0 | 1 | 2 ]
[, @with = 'additional_with_parameters' ] [,...n]
[, @logging = 0 | 1 | 2 ]
[, @ioflag = 'DISK_RETRY_COUNT=n']
[, @ioflag = 'DISK_RETRY_WAIT=n']
[, @affinity = 0..2147483647]
[, @throttle = 1..100]
[, @comment = 'comment']
[, @buffercount = 'buffer_count']
[, @maxtransfersize = 'maximum_transfer_size']
[, @adaptivecompression = 'size' | 'speed' ]
[, @compressionlevel = 'compression_level']
[, @attachedfile = 'pathname']
[, @tsmclientnode = 'TSM_client_node']
[, @tsmclientownerpwd = 'TSM_client_owner_password']
[, @tsmfilespace = 'TSM_file_space' ] [,...n]
[, @tsmconfigfile = 'TSM_configuration_file']
[, @tsmmanagementclass = 'TSM_management_class']
[, @tsmdevicetimeoutminutes = n]
[, @tsmdsmi_dir = 'path']
[, @tsmdsmi_log = 'path']
[, @tsmlogname = 'log_name']
[, @with = 'option_name']
```

## Arguments

### Tips:

- To see the list of accepted arguments and data types for arguments, execute the following:  
`exec master.dbo.<procedure_name> show_help`
- To convert the script for use with the command-line utilities, execute the following:  
`exec master.dbo.<procedure_name> show_cmd, <xp_arguments>`

### @affinity

Processor affinity designates specific processors to run LiteSpeed, while not allowing LiteSpeed to run on the remaining processors.

This argument accepts decimal values and hexadecimal values. If a value begins with "0x" it is interpreted as hexadecimal. A positive 64-bit integer value translates to a binary mask where a value of 1 designates the corresponding processor to be able to run the LiteSpeed process.

**Note:** 32-bit Windows is internally limited to a 32-bit mask.

For example, you need to select processors 2, 3, and 6 for use with LiteSpeed. Number the bits from the right to left. The rightmost bit represents the first processor. Set the second, third, and sixth bits to 1 and all other bits to 0. The result is binary 100110, which is decimal 38 or hexadecimal 0x26. Review the following for additional information:

Decimal Value	Binary Bit Mask	Allow LiteSpeed Threads on Processors
0	0	All (default)
1	1	1
3	11	1 and 2
7	111	1, 2 and 3
38	100110	2, 3, and 6
205	11001101	1, 3, 4, 7, and 8

**Tip:** Before you start tuning the CPU Throttle or Affinity parameters to adjust backup performance, try limiting the number of threads. If you decide to use an affinity value other than default, it is recommended that you limit the threading as well. You may also want to consider using Adaptive Compression to maintain backup performance. See "Adaptive Compression" (page 55) for more information.

### @AppendDifferential

Appends data to an existing full backup file. This argument accepts one of the following values:

- 0—false
- 1—true

### **@attachedfile**

Specifies filepaths to include in both backup and restore operations. The filepath can be either a single file or a directory. If it is a directory, then LiteSpeed recursively includes all files and subdirectories. All attached files are encrypted and compressed, with all pertinent backup parameters supported. This feature works for disk, tape, TSM, and Double Click Restore as well. You can supply multiple instances of this argument.

When used within the context of a restore operation, the path parameter can be expanded to include a new destination. This form will take the syntax of <file\_path> to <new\_file\_path>. The new filepath can be used to specify a new location but cannot rename a file.

This argument only restores the attached files. It does not restore the database, just the files that were attached to that backup.

#### **Notes:**

- The original entire directory path need not be supplied (e.g. c: to c:\testadSattsm is allowed).
- c:\testad to testadr would restore all files in directory c:\testad to c:\testadr.

### **@BackupDirectory**

Specifies a directory for the backup file.

### **@backupname**

Specifies the name of the backup set.

This argument accepts variables. See "LiteSpeed Variables" (page 271) for more information.

### **@buffercount**

Number of SQL Server buffers available for a LiteSpeed operation. The default value is 20.

### **@CheckForFullBackup**

Checks if the expected full backup exists and returns a failure message if it is not found. Accepts the following values:

- 0—false
- 1—true

### **@comment**

Appends a user comment to the backup.

This argument accepts variables. See "LiteSpeed Variables" (page 271) for more information.

### **@compressionlevel**

Specifies the compression level for the backup. Valid values are 0 through 8. 0 bypasses the compression routines. The remaining values of 1 through 8 specify compression with increasingly aggressive computation. 2 is the default value.

When choosing a compression level, it is best to try various options using your equipment and data to determine the best option for your environment. Use the Backup Analyzer to test the performance of different compression levels. See "Test Optimal Backup Settings" (page 56) for more information.

**Note:** If both the compression level and Adaptive Compression option are passed in, LiteSpeed will not error out and will select and use Adaptive Compression.

### **@cryptlevel**

Works in conjunction with the @encryptionkey parameter.

Specify the encryption level. Higher levels improve security, but they require more CPU and take longer. See "Test Optimal Backup Settings" (page 56) for more information on analyzing the best backup settings for your environment.

This argument accepts one of the following values:

- 0—40-bit RC2
- 1—56 bit RC2
- 2—112 bit RC2
- 3—128 bit RC2
- 4—168 bit 3DES
- 5—128 bit RC4
- 6—128 bit AES
- 7—192 bit AES
- 8—256 bit AES

### **@database**

Name of database to be backed up or restored.

This parameter specifies a database:

- to be backed up (xp\_backup\_database and xp\_slsSmartDiff)
- containing the transaction log to be backed up (xp\_backup\_log)
- to be restored (xp\_restore\_database and xp\_restore\_log)
- on which you wish to check the progress of an activity (xp\_slsReadProgress)
- for which you want to delete old backups (xp\_slsSmartCleanup)

If supplied as a variable (@database), this name can be specified either as a string constant (@database = database name) or as a variable of character string data type, except for the ntext or text data types.

### **@desc**

Specifies a description to store with the backup.

This argument accepts variables. See "LiteSpeed Variables" (page 271) for more information.

### **@DiffToFullRatioRequireFull**

Specifies the last differential backup size to last full backup size ratio. When exceeding the specified ratio LiteSpeed performs a full backup. The argument accepts one of the following formats:

- '.4'
- '40%'

### **@ElapsedDaysRequireFull**

Specifies the minimum number of days since last full backup required to perform full backup.

### **@encryptionkey**

Value used to generate the encryption key for the encryption algorithm. If you do not supply encryption\_key, then the program will not encrypt the backup.

**Caution:** When encrypting data, take care not to lose the encryption\_key; a backup cannot be restored or recovered without the original encryption key.

Example of key: 'Mypassword'

You can also specify the encryption level with @cryptlevel.

### **@ExtentsChgRatioRequireFull**

Specifies the minimum amount of database changes required for the full backup. The argument accepts one of the following formats:

- '.4'
- '40%'

### **@expiration**

Specifies the date and time when the backup expires. LiteSpeed will not overwrite this file until expiration datetime is passed. The argument accepts one of the following formats:

- yyyy-mm-dd
- yyyy-mm-dd hh:mm:ss

### **@file**

Specifies a logical database file used for file or filegroup backups. You can supply multiple instances of this argument.

### **@filegroup**

Specifies a database filegroup to include in the backup or restore. You can supply multiple instances of this argument.

A filegroup backup is a single backup of all files in the filegroup and is equivalent to explicitly listing all files in the filegroup when creating the backup. Files in a filegroup backup can be restored individually or as a group.

### **@ForceDifferential**

Forces differential backup. It accepts the following values:

- 0—false
- 1—true

### **@ForceFull**

Forces full backup. This argument accepts one of the following values:

- 0—false
- 1—true



### @FullBackupEscalation

Causes LiteSpeed to issue a full backup, if one of the following problems is discovered in the current backup set:

- The full backup is missing.
- A differential backup is missing from the backup set (excludes backups automatically removed after the specified retention period).
- LSN verification fails in the backup set.
- Verify operation fails on full or differential backup.

**Note:** If a problem is detected and a full backup is created through escalation, an error will be returned.

This argument accepts one of the following values:

- 0—false
- 1—true

### @ioflag

Specifies if LiteSpeed should wait and retry the read or write operation on failure. You can define retry options using the following parameters:

- DISK\_RETRY\_COUNT—Specifies the number of times that a specific operation will be retried on failure. The default is 4 retries, the maximum allowed setting is 1000.
- DISK\_RETRY\_WAIT—Specifies the number of seconds to wait immediately following a failure before retrying. The default is 15 seconds, the maximum allowed setting is 300.

**Note:** This functionality is only available for disk operations.

For example, to specify a database backup where each failure can be retried once after a 30-second wait:

```
EXEC master.dbo.xp_backup_database  
@filename='c:\test.bkp'  
, @database='test'  
, @ioflag='DISK_RETRY_COUNT=1'  
, @ioflag='DISK_RETRY_WAIT=30'
```

See "Network Resilience" (page 58) for more information.

### @logging

Writes a log file for the operation. This argument accepts one of the following values:

- 0—Logging off.
- 1 or any odd value—Logging on. Log file is removed on success.

- 2 or any even value—Logging on.

The default output directory is *C:\Documents and Settings\All Users\Application Data\Quest Software\LiteSpeed\SQL Server\Logs*. To log to a different directory add `@Trace='logpath=path'`.

See Configure Logging in LiteSpeed (page 285) for information about LiteSpeed logging.

### **@maxtransfersize**

Specifies the data size in bytes for each transfer when communicating with SQL Server. The size can be any multiple of 64KB in the range from 64KB to 4MB. The default is *1048576* bytes.

### **@MirrorDirectory**

Specifies a directory for a mirror backup. You can supply multiple instances of this argument.

### **@olrmap**

Generates a map file during a backup for object level recovery. This argument accepts one of the following values:

- 0—False
- 1—True

### **@priority**

Specifies the priority of the LiteSpeed process compared to other processes running on the same server. This argument accepts one of the following values:

- -1—Below Normal
- 0—Normal (Default)
- 1—AboveNormal
- 2—High

### **@retaindays**

Specifies a number of days to retain the backup. LiteSpeed will not overwrite this file for this number of days.

### **@SearchAlternateBackupDirectory**

Specifies the directory where to search for the backup file.

### **@SpecificDaysForbidFull**

Specifies days of the week when a full backup is never performed. It accepts the following formats:

- 3—on Tuesday
- 'tu'—on Tuesday
- '5-7'—from Thursday to Saturday
- 'm, w, su'—on Monday, Wednesday, and Sunday

### **@threads**

Determines the number of threads used for the backup. You will achieve the best results by specifying multiple threads, but the exact value depends on several factors including: processors available, affinity setting, compression level, encryption settings, IO device speed, and SQL Server responsiveness. The default is  $n-1$  threads, where  $n$  is the number of processors.

### **@throttle**

Specifies the maximum CPU usage allowed. The argument accepts an integer value between 1 and 100. The default value is 100. This is the percentage of the total amount of CPU usage (across all enabled processors) available.

**Tip:** Before you start tuning the CPU Throttle or Affinity parameters to adjust backup performance, try limiting the number of threads. If you decide to use an affinity value other than default, it is recommended that you limit the threading as well. You may also want to consider using Adaptive Compression to maintain backup performance. See "Adaptive Compression" (page 55) for more information.

### **@tsmclientnode**

Specifies the TSM server LiteSpeed connects to during backups and restores. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.

### **@tsmclientownerpwd**

Specifies the TSM client owner user password. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.

### **@tsmconfigfile**

Specifies the TSM configuration file.

### **@tsmdevicetimeoutminutes**

Specifies how long to wait for a TSM device.

**Note:** This parameter can be set via the LiteSpeed UI Console.

### **@tsmdsmi\_dir**

DSMI\_DIR path if needed.

### **@tsmdsmi\_log**

DSMI\_LOG path.

### **@tsmfilespace**

Specifies the TSM file space, the logical space on the TSM server. It can be the drive label name or UNC name. You can supply multiple instances of this argument.

**Note:** IBM recommends that an application client should select a unique file space; it is recommended that LiteSpeed users follow this practice with a specific file space reserved for LiteSpeed backups.

### **@tsmlogname**

Log name.

### **@tsmmanagementclass**

Specifies the TSM management class. If not specified, LiteSpeed uses the default management class.

### **@tsmpassword**

The TSM username password. Passwords are case-sensitive.

### **@tsmusername**

The TSM username ID.

### **@Verify**

Performs a restore verification on the backup file just created (if backup was successful). The following values direct which part of backup chain to verify:

- Full—Verifies the last full backup
- Last—Verifies last backup performed (can be either a full or differential)
- Full,Last—Verifies the last full backup and last differential is available
- All—Verifies last full backup and all differentials since

### **@with**

Each @with argument should be a syntactically complete and correct statement. Please refer to the SQL Server Transact-SQL backup and restore documentation for the syntax and usage.

The supported formats are:

- @with='PARAMETER'
- @with='PARAMETER="accepted\_value"'

**Notes:**

- Extended stored procedure arguments are limited to 255 characters. If you need more than 255 characters, use multiple @with arguments.
- Do not supply the @with parameter if no additional features are required.

This extended stored procedure accepts the following @with parameters:

Parameter	Description
CHECKSUM	Causes checksums to be verified when a LiteSpeed backup is created. <b>Note:</b> When you restore a backup containing checksum, it is automatically checked. If you do not want to check the checksums during a restore, supply 'NO_CHECKSUM' .
CONTINUE_ AFTER_ ERROR	Causes the backup be executed despite encountering an invalid backup checksum.

## Examples

1. Back up the Northwind database. Perform full backup only if the amount of database changes since the last full backup is more than 40%.

```
EXEC master.dbo.xp_slsFastCompression  
@database = 'Northwind'  
, @BackupDirectory = 'C:\Program Files\Microsoft SQL  
Server\MSSQL.1\MSSQL\Backup'  
, @ExtentsChgRatioRequireFull = '.4'
```

2. Back up the Northwind database to multiple locations. Perform full backup only if more than 10 days have passed since last full backup.

```
EXEC master.dbo.xp_slsFastCompression  
@database = 'Northwind'  
, @BackupDirectory = 'C:\Program Files\Microsoft SQL  
Server\MSSQL.1\MSSQL\Backup'  
, @MirrorDirectory = 'D:\SQLServerBackups'  
, @ElapsedDaysRequireFull = 10
```

3. Back up the Northwind database. Force full backup.

```
EXEC master.dbo.xp_slsFastCompression  
@database = 'Northwind'  
, @BackupDirectory = 'C:\Program Files\Microsoft SQL
```

```
Server\MSSQL.1\MSSQL\Backup'  
, @ForceFull = 1
```

4. Backup to TSM. Perform full backup only if the amount of database changes since the last full backup is more than 40%.

```
exec master.dbo.xp_slsFastCompression  
@database = N'userdb3',  
@backupname = N'userdb3 - Fast Compression Backup',  
@desc = N'Fast Compression Backup of userdb3',  
@AdaptiveCompression = 'Speed',  
@ExtentsChgRatioRequireFull = N'40%',  
@tsmfilespace = N'FC',  
@tsmconfigfile = N'C:\Program Files\Tivoli\TSM\baclient\dsm.opt',  
@tsmclientnode = N'w2k3_TSM',  
@tsmclientownerpwd = N'***',  
@tsmmanagementclass = N'STANDARD',  
@tsmdevicetimeoutminutes = 2,  
@FullBackupEscalation = 1
```

## Returns

0 (success) or non-zero (failure).

*To capture the output message, run the following:*

```
declare @rmsg varchar(999)  
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output  
select @rmsg
```

*To capture the output message and the result code, run the following:*

```
declare @rc int  
declare @rmsg varchar(999)  
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output,  
@resultcode=@rc output  
select @rc, @rmsg
```

## xp\_slsreadprogress

Reads the progress of the current activity on a specified database and returns an integer (0-100) indicating the percentage completed of the current activity on the specified database.

## Syntax

```
EXEC master.dbo.xp_slsreadprogress  
@database = 'database_name'
```

## Examples

1. Read the progress of the current activity at a specified database:

```
exec xp_slsreadprogress @database='Northwind'
```

This command returns the percentage complete, for example:

(1 row(s) affected)

96 percent completed

2. Query the progress of an activity based on the result returned (in this case print 'DONE' when the progress is 100 percent):

```
Declare @Result int
exec xp_slsreadprogress @database='Northwind',
@Percent = @Result output
if @Result = 100
begin
print 'Done'
end
```

## xp\_slsSmartCleanup

Removes full and differential backup files and transaction log backups based on a user-defined period.

The backup retention will never delete:

- The backup files, if there are mixed backups in the same backup file. For example, if a user performs a backup of AdventureWorks and Pubs into the same mybackups.bak backup file.
- The full backup, if there are associated differential or t-log backups in the backup set that are not eligible for cleanup.
- File/FileGroup backups
- File/FileGroup differential backups
- Partial backups
- Partial differential backups

## Syntax

```
exec master.dbo.xp_slsSmartCleanup
@database = 'database_name'
, (@BackupRetainDays = 1...365 | @BackupExpiration = 'date_time')
, (@LogRetainDays = 1...365 | @LogExpiration = 'date_time')
[, @CopyOnlyBackups = 'option']
[, @DryRun = 0 | 1 ]
[, @TSMClientNode = 'TSM_client_node']
[, @TSMUserName = 'TSM_username_ID']
[, @TSMPassword = 'TSM_username_password']
```

```
[, @TSMConfigFile = 'TSM_configuration_file']  
[, @TSMClientOwnerPwd = 'TSM_client_owner_password']  
[, @TSMDSMI_DIR = 'path']  
[, @TSMDSMI_LOG = 'path']  
[, @TSMLogName = 'log_name']
```

## Arguments

### Tips:

- To see the list of accepted arguments and data types for arguments, execute the following:  
`exec master.dbo.<procedure_name> show_help`
- To convert the script for use with the command-line utilities, execute the following:  
`exec master.dbo.<procedure_name> show_cmd, <xp_arguments>`

### @BackupExpiration

Specifies the date using one of the following formats:

YYYY-MM-DD

YYYY-MM-DD HH:MM:SS

where

- YYYY—4-digit year
- MM—2-digit month
- DD—2-digit day of the month
- HH—2-digit hour using the local 24-hour clock
- MM—2-digit minute
- SS—2-digit second

To be eligible for cleanup, the full or differential backup must be older than this date.

### @BackupRetainDays

Specifies the number of days (N). The full or differential backup must be at least N days old before it is eligible for cleanup.

### @database

Name of database to be backed up or restored.

This parameter specifies a database:

- to be backed up (xp\_backup\_database and xp\_slsSmartDiff)
- containing the transaction log to be backed up (xp\_backup\_log)



- to be restored (xp\_restore\_database and xp\_restore\_log)
- on which you wish to check the progress of an activity (xp\_slsReadProgress)
- for which you want to delete old backups (xp\_slsSmartCleanup)

If supplied as a variable (@database), this name can be specified either as a string constant (@database = database name) or as a variable of character string data type, except for the ntext or text data types.

### **@CopyOnlyBackups**

Controls how LiteSpeed handles copy-only backups. This argument accepts one of the following values:

- Default—LiteSpeed will ignore copy-only backups except on secondary replicas in AlwaysOn Availability groups, in which case it will allow deletions. This is the default behavior when the parameter is not specified.
- Ignore—Copy-only backups are never deleted.
- AllowDeletes—Copy-only backups are removed according to the specified retention options.

#### **Notes:**

- Transaction log backups are not considered dependent on copy-only full or copy-only tlog backups.
- Copy-only transaction log backups will not mark other transaction log or full backups as having a dependent.
- The values are not case-sensitive.

### **@DryRun**

Displays backups that are to be removed (delete candidates) or kept according to the specified conditions and SmartCleanup logic. SmartCleanup does not remove any backups, if this parameter is specified.

### **@LogExpiration**

Specifies the date of one of the following formats:

YYYY-MM-DD

YYYY-MM-DD HH:MM:SS

where

- YYYY—4-digit year
- MM—2-digit month
- DD—2-digit day of the month

- HH—2-digit hour using the local 24-hour clock
- MM—2-digit minute
- SS—2-digit second

To be eligible for cleanup, the t-log backup must be older than this date.

### **@LogRetainDays**

Specifies the number of days (N). The t-log backup must be at least N days old before it is eligible for cleanup.

### **@tsmclientnode**

Specifies the TSM server LiteSpeed connects to during backups and restores. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.

### **@tsmclientownerpwd**

Specifies the TSM client owner user password. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.

### **@tsmconfigfile**

Specifies the TSM configuration file.

### **@tsmdsmi\_dir**

DSMI\_DIR path if needed.

### **@tsmdsmi\_log**

DSMI\_LOG path.

### **@tsmlogname**

Log name.

### **@tsmpassword**

The TSM username password. Passwords are case-sensitive.

### **@tsmusername**

The TSM username ID.

## **Example**

```
exec master.dbo.xp_slsSmartCleanup  
@database = 'test2'
```

```
, @BackupRetainDays = 28
, @LogRetainDays = 28
```

Returns

0 (success) or non-zero (failure).

*To capture the output message, run the following:*

```
declare @rmsg varchar(999)
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output
select @rmsg
```

*To capture the output message and the result code, run the following:*

```
declare @rc int
declare @rmsg varchar(999)
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output,
@resultcode=@rc output
select @rc, @rmsg
```

xp\_sqlitespeed\_licenseinfo

Using xp\_sqlitespeed\_licenseinfo you can register a new LiteSpeed license key or remove licensing information (unlicense LiteSpeed). Without the arguments, this extended stored procedure returns a result set showing the currently installed LiteSpeed license.

Syntax

```
EXEC master.dbo.xp_sqlitespeed_licenseinfo
[( @licensekey = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX-111-22222-YY'
, @sitemessage = 'Site_Message'
[, @store = 1])
| @remove = 1]
```

Arguments

@licensekey	Passes license key value.
@sitemessage	Passes license site message.
@store = 1	Overwrites any currently stored licenses with the valid license supplied.
@remove = 1	Removes the currently stored license.

## Examples

1. View information about the supplied license key:

```
EXEC master.dbo.xp_sqlitespeed_licenseinfo  
@licensekey = 'C20TM3Q3K2HD74UDLBMHC6KYV6HZ3MQFNXZFB-123-45678-34'  
, @sitemessage = 'Trial Version'
```

2. Register a license key:

```
EXEC master.dbo.xp_sqlitespeed_licenseinfo  
@licensekey = 'C20TM3Q3K2HD74UDLBMHC6KYV6HZ3MQFNXZFB-123-45678-34'  
, @sitemessage = 'Trial Version'  
, @store=1
```

3. Remove license key:

```
EXEC master.dbo.xp_sqlitespeed_licenseinfo  
@remove= 1
```

## Result Set

Column Name	Data Type	Description
Product Name	nvarchar(128)	NetVault LiteSpeed for SQL Server.
License Key	nvarchar(128)	Key license value.
License Site Message	nvarchar(128)	Site message.
License Type	nvarchar(128)	One of the following: <ul style="list-style-type: none"><li>• Trial—a dated trial license key that expires on a specific date</li><li>• Permanent—a permanent license key</li><li>• Term—similar to trial keys, except it comes with support</li></ul>
Edition	nvarchar(128)	Enterprise or Standard.
Site License	bit	Whether the key is a Site license key that is used company-wide.
Gigabyte Limit	int	Database size limit in GB. If there is a gigabyte limit set in the license (any value larger than 0), LiteSpeed will fail any backup of databases larger than the limit. 0 indicates unlimited database size.
Trial Length	int	Number of trial days. The default value is 15. <b>Note:</b> Some types of trials may have 0, if the expiration date is fixed and not based on the install date.
Expire Date	datetime	Date value (Trial and Term only).

## xp\_sqlitespeed\_version

This extended stored procedure returns the name and version of the installed LiteSpeed utilities.

## xp\_view\_tsmcontents

Retrieves TSM specific information and backup header information for the given LiteSpeed backup.

## Syntax

```
EXEC master.dbo.xp_view_tsmcontents
@tsmclientnode = 'TSM_client_node'
, @tsmclientownerpwd = 'TSM_client_owner_password'
, @tsmfilespace = 'TSM_filespace'
, @tsmconfigfile = 'TSM_configuration_file'
[, @tsmhighlevel = 'TSM_high_level']
[, @tsmlowlevel = 'TSM_low_level']
[, @tsmarchive= 0 | 1 ]
[, @desc='description']
[, @tsminsdatelower='date_time']
[, @tsminsdateupper='date_time']
[, @tsmexpdatelower='date_time']
[, @tsmexpdateupper='date_time']
[, @tsmbrieflist = 0 | 1 ]
[, @tsmsortbypit = 0 | 1 | @tsmsortbylowlevel = 0 | 1 ]
[, @tsmpointintime = 'date_time' ]
```

## Arguments

### Tips:

- To see the list of accepted arguments and data types for arguments, execute the following:  
`exec master.dbo.<procedure_name> show_help`
- To convert the script for use with the command-line utilities, execute the following:  
`exec master.dbo.<procedure_name> show_cmd, <xp_arguments>`

### @desc

Specifies a description to filter the returned results to those that match the pattern.

### @tsmarchive

Specifies to store backup as a TSM archive. This argument accepts one of the following values:

- 0—False
- 1—True

### @tsmbrieflist

Returns a brief list. This argument accepts one of the following values:

- 0—False
- 1—True

**Note:** It is not needed for archives as they are only returned as a brief list.

### **@tsmclientnode**

Specifies the TSM server LiteSpeed connects to during backups and restores. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.

### **@tsmclientownerpwd**

Specifies the TSM client owner user password. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.

### **@tsmconfigfile**

Specifies the TSM configuration file.

### **@tsmexpdatelower**

For TSM archives, it specifies the oldest expiration date and time of where to start the list. The format is yyyy-mm-dd hh:mm:ss.

### **@tsmexpdateupper**

For TSM archives, it specifies the most recent expiration date and time of where to stop the list. The format is yyyy-mm-dd hh:mm:ss.

### **@tsmfilespace**

Specifies the space on the TSM server that contains a group of files. It can be the drive label name or UNC name.

### **@tsmhighlevel**

Specifies the TSM high-level name. If you do not specify this parameter, LiteSpeed will retrieve all high levels from the TSM server.

#### **Notes:**

- This parameter supports wild cards (e.g. MyLowLevelName\*, MyHighLevelName\*).
- You do not necessarily have to have both the @tsmhighlevel and @tsmlowlevel parameters in one query.

### **@tsminsdatelower**

For TSM archives, it specifies the oldest insertion date and time of where to start the list. The format is yyyy-mm-dd hh:mm:ss.

### **@tsminsdateupper**

For TSM archives, it specifies the most recent insertion date and time of where to stop the list. The format is yyyy-mm-dd hh:mm:ss.

### **@tsmlowlevel**

Specifies the TSM low-level name. If you do not specify this parameter, LiteSpeed will retrieve all low levels from the TSM server.

#### **Notes:**

- This parameter supports wild cards (e.g. MyLowLevelName\*, MyHighLevelName\*).
- You do not necessarily have to have both the @tsmhighlevel and @tsmlowlevel parameters in one query.

### **@tsmsortbylowlevel**

Sorts the results by the low level name. This argument accepts one of the following values:

- 0—False
- 1—True

### **@tsmsortbypit**

Sorts the results by the point-in-time date. It accepts one of the following:

- 0—False
- 1—True

### **@tsmpointintime**

Specifies the date for restore/to filter results. If it is not passed, LiteSpeed will choose the most recent archived backup. The format is yyyy-mm-dd hh:mm:ss.

**Note:** If the backup was a striped backup and the point-in-times of the various striped files are different (rare but can be different a second or so), then the most recent of the times must be chosen.

## **Example**

```
exec master.dbo.xp_view_tsmcontents
@tsmclientnode = 'ClusterGroup'
, @tsmclientownerpwd= 'test16'
, @tsmfilespace= 'SLS_Mar'
, @tsmconfigfile= 'C:\Program Files\Tivoli\tsm\baclient\dsm.opt'
, @tsmarchive=1
, @tsminsdatelower='2006-03-15 13:00:00'
, @tsminsdateupper='2006-03-16 18:00:00'
```



```
, @tsmexpdatelower='2007-02-14 09:00:00'  
, @tsmexpdateupper='2007-03-17 18:00:00'
```

## Result Set

xp\_view\_tsmcontents displays the following information:

Column Name	Data Type	Description
File Space	Nvarchar(128)	TSM File Space.
High Level	Nvarchar(128)	TSM High Level.
Low Level	Nvarchar(128)	TSM Low Level.
Management Class	Nvarchar(128)	TSM Management Class.
TsmPointInTime	Nvarchar(128)	The TSM retention point-in-time date for restore.
FileNumber	Int	Number of the Backup within the LiteSpeed Backup device.
BackupFormat	Nvarchar(128)	Reserved field. Returns 1.
Guid	Uniqueidentifier	Backup GUID, uniquely identifies LiteSpeed backup sets.
BackupName	Nvarchar(128)	Backup set name.
BackupDescription	Nvarchar(128)	(For archives only) The description (if any) that the user passed in on the @desc parameter on the original backup.
BackupType	Nvarchar(128)	Backup type: <ul style="list-style-type: none"> <li>• 1—Database</li> <li>• 2—Transaction Log</li> <li>• 4—File</li> <li>• 5—Differential Database</li> <li>• 6—Differential File</li> <li>• 7—Partial</li> <li>• 8—Partial Differential</li> </ul>
ExpirationDate	Datetime	(For archives only) The expiration date and time that TSM assigned the archived object based on the management class policy of the management class assigned to the archived object.

Column Name	Data Type	Description
Compressed	Tinyint	0 = No compression. 1 = Compressed
Position	Smallint	Position of the backup set in the volume (for use with the FILE = option).
DeviceType	Tinyint	Virtual Device > 7 = Logical 107 = Physical
UserName	Nvarchar(128)	Username that performed the backup operation.
ServerName	Nvarchar(128)	Name of the server that wrote the backup set.
DatabaseName	Nvarchar(128)	Name of the database that was backed up.
DatabaseVersion	Int	Version of the database from which the backup was created.
DatabaseCreationDate	Datetime	Date and time the database was created.
BackupSize	Numeric (20,0)	Size of the backup, in bytes.
FirstLsn	Numeric (25,0)	Log sequence number of the first transaction in the backup set. NULL for file backups.
LastLsn	Numeric (25,0)	Log sequence number of the last transaction in the backup set. NULL for file backups.
CheckpointLsn	Numeric (25,0)	Log sequence number of the most recent checkpoint at the time the backup was created.
DifferentialBaseLsn	Numeric (25,0)	Log sequence number of the most recent full database backup.
BackupStartDate	Datetime	Date and time that the backup operation began.
BackupFinishDate	Datetime	Date and time that the backup operation finished.
SortOrder	Smallint	Server sort order. This column is valid for database backups only. Provided for backward compatibility.
CodePage	Smallint	Server code page or character set used by the server.

Column Name	Data Type	Description
CompatibilityLevel	Tinyint	Compatibility level setting of the database from which the backup was created.
SoftwareVendorId	Int	Software vendor identification number. For SQL Server, this number is 4608 (or hexadecimal 0x1200).
SoftwareVersionMajor	Int	Major version number of the server that created the backup set.
SoftwareVersionMinor	Int	Minor version number of the server that created the backup set.
SoftwareVersionBuild	Int	Build number of the server that created the backup set.
MachineName	Nvarchar(128)	Name of the server that wrote the backup set.
BindingId	Uniqueidentifier	Binding ID for the database.
RecoveryForkId	Uniqueidentifier	ID for the current recovery fork for this backup.
Encryption	Int	Indicates if backup is encrypted <ul style="list-style-type: none"><li>• 0—not encrypted</li><li>• 1—encrypted</li></ul>

## Returns

0 (success) or non-zero (failure).

*To capture the output message, run the following:*

```
declare @rmsg varchar(999)
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output
select @rmsg
```

*To capture the output message and the result code, run the following:*

```
declare @rc int
declare @rmsg varchar(999)
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output,
@resultcode=@rc output
select @rc, @rmsg
```

## **xp\_view\_tsmmc**

Displays available TSM management classes with detailed information. If you specify a management class, the information returns only to the specified management class.

### **Syntax**

```
EXEC master.dbo.xp_view_tsmmc  
@tsmclientnode = 'TSM_client_node'  
, @tsmclientownerpwd = 'TSM_client_owner_password'  
, @tsmconfigfile = 'TSM_configuration_file'  
[, @tsmmanagementclass = 'TSM_management_class']
```

### **Arguments**

#### **@tsmclientnode**

Specifies the TSM server LiteSpeed connects to during backups and restores. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.

#### **@tsmclientownerpwd**

Specifies the TSM client owner user password. Not required, if specified in the options file or if backing up with the Passwordaccess Generate option.

#### **@tsmconfigfile**

Specifies the TSM configuration file.

#### **@tsmmanagementclass**

Specifies the TSM management class. If not specified, LiteSpeed uses the default management class.

## Result Set

Column Name	Data Type	Description
Name	Nvarchar(128)	Management class name.
Backup copy group name	Nvarchar(128)	The name of the backup copy group.
Backup copy group destination	Nvarchar(128)	Names the destination where backups are stored. The destination can be either a storage pool of disk devices or a storage pool of devices that support removable media, such as tape.
Versions data exists	Nvarchar(128)	Specifies the maximum number of different backup versions retained. If you select a management class that permits more than one backup version, the most recent version is called the active version. All other versions are called inactive versions. If the maximum number of versions permitted is five, and you run a backup that creates a sixth version, the oldest version is deleted from server storage.
Versions data deleted	Nvarchar(128)	Specifies the maximum number of different backup versions retained for files and directories that you erased from your drive. This parameter is ignored as long as the file or directory remains on your drive. If you erase the file or directory, the next time you run an incremental backup, the active backup version is changed to inactive and the oldest versions are erased that exceed the number specified by this parameter.
Retain extra versions	Nvarchar(128)	Specifies how many days all but the most recent backup version is retained. The most recent version is the active version, and active versions are never erased. If Nolimit is specified, then extra versions are kept until the number of backup versions exceeds the versions data exists or versions data deleted parameter settings. In this case, the oldest extra version is deleted immediately.

Column Name	Data Type	Description
Retain only versions	Nvarchar(128)	Specifies the number of days the last remaining inactive version of a file or directory is retained. If Nolimit is specified, the last version is retained indefinitely. This parameter goes into effect during the next incremental backup after a file is deleted from the client machine. Any subsequent updates to this parameter will not affect files that are already inactive. For example: If this parameter is set to 10 days when a file is inactivated during an incremental backup, the file will be expired in 10 days.
Archive copy group name	Nvarchar(128)	The name of the backup copy group.
Archive copy group destination	Nvarchar(128)	Names the destination where archives are stored. The destination can be either a storage pool of disk devices or a storage pool of devices that support removable media, such as tape.

## Returns

0 (success) or non-zero (failure).

*To capture the output message, run the following:*

```
declare @rmsg varchar(999)
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output
select @rmsg
```

*To capture the output message and the result code, run the following:*

```
declare @rc int
declare @rmsg varchar(999)
exec master..<procedure_name> <arguments>, @resultmsg=@rmsg output,
@resultcode=@rc output
select @rc, @rmsg
```

## LiteSpeed Variables

LiteSpeed automatically substitutes variables anywhere you need to specify a backup file name, comment, or description in the LiteSpeed UI Console, from the command-line and when using extended stored procedures. See "Examples" (page 272) for more information.

Review the following for information about the command-line and extended stored procedure arguments that accept LiteSpeed variables.

XP argument	CMD argument	Description
@filename	-F	File name or stripe file name
@mirror	-E	Mirror file name
@backupname	-n	Backup name
@desc	-d	Backup description
@comment	-Y	Comment

## Accepted Variables

You can use the following variables:

Variable	Description
%DATABASENAME% or %D	Database name
%TYPE% or %T	Backup type (full, diff, log)
%SERVER%	Server name
%INSTANCE%	Server instance name
%DEFAULTDIR%	Default backup directory
%z	Timestamp, the number of seconds elapsed since 00:00:00 January 1, 1970, UCT
Date and time variables	
%DATE%	Date
%TIME%	Time (hhmm)

Variable	Description
%DATETIME%	Date and time
%a	Abbreviated weekday name
%A	Full weekday name
%b	Abbreviated month name
%B	Full month name
%d	Day of the month (01-31)
%H	Hour in 24h format (00-23)
%I	Hour in 12h format (01-12)
%j	Day of the year (001-366)
%m	Month as a decimal number
%M	Minute (00-59)
%p	AM or PM designation
%S	Second (00-59)
%U	Week number with the first Sunday as the first day of week one (00-53)
%w	Weekday as a decimal number with Sunday as 0 (0-6)
%W	Week number with the first Monday as the first day of week one (00-53)
%y	Year, last two digits (00-99)
%Y	Year
%Z	Time zone name or abbreviation

## Examples

1. Specify backup destination:

```
\\Storage\Backup\%SERVER%\%DATABASENAME%_%TYPE%_%DATETIME%.bak
```



2. Back up the Northwind database with the specified backup set name and description.

```
EXEC master.dbo.xp_backup_database  
@database='MyDB'  
, @filename='C:\MSSQL\Backup\%D.BAK'  
, @init= 1  
, @backupname = '%D_%w'  
, @desc = '%T Backup of %D'
```

3. Restore the Northwind database from the backup device c:\temp\Northwind.bak from the command-line interface.

```
sqlitespeed.exe -R Database -D Northwind -F "C:\temp\%D.bak" -W  
REPLACE
```

## Privilege and Grant Requirements

LiteSpeed requires the following privileges and grants:

Account	Requirement
User account that performs the LiteSpeed installation	Windows administrator rights and DBA privileges required to <ul style="list-style-type: none"><li>• Install stored procedures and extended stored procedures</li><li>• Create databases, tables, and indexes</li><li>• Insert, update, and delete to/from the MSDB, LiteSpeedLocal and LiteSpeedCentral databases</li></ul>
User account that runs the LiteSpeed UI Console	Power User privileges (or some equivalent). <b>Note:</b> When executing a Double Click Restore, Windows Vista, Windows Server 2003 and Windows Server 2008 require that the Windows user account is an administrator.
NT AUTHORITY\SYSTEM	This account must be added to the sysadmin server role on each instance of SQL Server 2012 in the following cases: <ul style="list-style-type: none"><li>• If you plan to use Windows authentication during Instance Configuration.</li><li>• If you plan to complete installation using the Instance Configuration wizard that is launched automatically during installation.</li></ul> <b>Workaround:</b> Do one of the following: <ul style="list-style-type: none"><li>• Use SQL Server authentication during Instance Configuration.</li><li>• Run the Instance Configuration wizard separate from the installer by selecting Instance Configuration from the Start Menu.</li></ul>

Account	Requirement
SQL Server service account	<p>Sysadmin server role privileges required to</p> <ul style="list-style-type: none"> <li>• Push stats to local repository</li> <li>• Back up or restore using extended stored procedures</li> <li>• Execute a scheduled job</li> <li>• Start a job manually</li> </ul> <p><b>Note:</b> SQL Server Configuration Manager should always be used when changing SQL Server service accounts.</p>
SQL Agent service account (or a proxy account)	
Windows or SQL Server accounts specified during instance configuration ( <b>Start   All Programs   Quest Software   LiteSpeed for SQL Server   Instance Configuration</b> )	<p>Sysadmin server role privileges required to retrieve data from local repository and to push to central repository.</p> <p><b>Note:</b> If you select Windows authentication method for both the local and central repository, LiteSpeed uses the Activity Manager service account specified on the Service Account page to push data to central repository. If the central repository is located on another server, the LiteSpeed Activity Manager must have domain rights for two-way communication between the local and central repositories.</p>
Windows or SQL Server account ( <b>File   Repository Registration...</b> )	<p>Db_owner of the central repository (required to retrieve data from the central repository)</p>
Windows or SQL Server account ( <b>File   SQL Server Registration...</b> )	<p>Sysadmin server role privileges required to</p> <ul style="list-style-type: none"> <li>• Connect to SQL Server and retrieve SQL Server properties</li> <li>• Back up or restore</li> <li>• Execute procedures, retrieve MSDB history</li> <li>• Access repositories</li> </ul>

## Repository Maintenance

### LiteSpeed\_DeleteActivity

This stored procedure removes LiteSpeed activity and information about LiteSpeed backups based on the date and time specified. Additionally, if appropriate parameters are specified, it removes log shipping history, jobs and maintenance plans history, DST status, and information about deleted databases.

For better control over the repository cleanup options, configure a Clean Up History task using maintenance plans. See "About Creating Maintenance Plans" (page 62) for more information.

**Note:** You must run the LiteSpeed\_DeleteActivity procedure against the database that has the LiteSpeed procedures installed and can be the one of the following:

- LiteSpeedLocal
- LiteSpeedCentral
- custom database, for example, msdb or any other specified

Use the LiteSpeed Instance Configuration wizard (**Start | All Programs | Quest Software | LiteSpeed for SQL Server | Instance Configuration**) to choose where to store the repository tables.

### Syntax

```
USE {LiteSpeedLocal|LiteSpeedCentral|custom_database_name}
EXEC LiteSpeed_DeleteActivity
{ @deleteDate = 'date_time' | ( @delUnit = n,@delUnitType = 'time' ) }
, { @delLocal = 0 | 1 | @delCentral = 0 | 1 | ( @delLocal = 0 | 1 , @delCentral
= 0 | 1 ) }
[, @purgeDeleted = 0 | 1 ]
[, @delLogshipping = 0 | 1 ]
[, @delStatus = 0 | 1 ]
```

### Arguments

#### @deleteDate

Deletes data older than the date and time specified. The argument accepts the following format:

YYYYMMDD HH:MM:SS

where

- YYYY— 4-digit year
- MM— 2-digit month
- DD— 2-digit day of the month

- HH— 2-digit hour using the local 24-hour clock
- MM— 2-digit minute
- SS— 2-digit second

### **@delLocal**

Deletes old data from the LitespeedActivity and LitespeedBackupFile tables in the Local repository or in the custom database (if the Local repository tables were created in the custom database). This argument accepts one of the following values:

- 0—false
- 1—true

### **@delCentral**

Deletes old data from the LitespeedActivity and LitespeedBackupFile tables in the Central repository or in the custom database (if the Central repository tables were created in the custom database). This argument accepts one of the following values:

- 0—false
- 1—true

### **@delUnit**

Deletes the rows older than the age specified.

### **@delUnitType**

Specifies a unit of measurement of time for @delUnit. This argument accepts one of the following values:

- MINUTES
- HOURS
- DAYS
- WEEKS
- MONTHS
- YEARS

### **@purgeDeleted**

Deletes information about deleted databases from dbo.LitespeedDatabase in the following databases:

- Local repository or custom database, if @delLocal was supplied
- Central repository or custom database, if @delCentral was supplied
- Both, if both @delLocal and @delCentral were supplied

This argument accepts one of the following values:

- 0—false
- 1—true

### **@delLogshipping**

Deletes old log shipping history entries from dbo.LogShippingHistory in the following databases:

- Local repository or custom database, if @delLocal was supplied
- Central repository or custom database, if @delCentral was supplied
- Both, if both @delLocal and @delCentral were supplied

This argument accepts one of the following values:

- 0—false
- 1—true

### **@delStatus**

Deletes old data from dbo.DbMaintStatus, dbo.DTSSStatus, dbo.JobStatus in the following databases:

- Local repository or custom database, if @delLocal was supplied
- Central repository or custom database, if @delCentral was supplied
- Both, if both @delLocal and @delCentral were supplied

This argument accepts one of the following values:

- 0—false
- 1—true

## **Examples**

1. Delete the following data older than 08/21/2009 from the Local and Central repositories:

- LiteSpeed activity
- Information about deleted databases

```
exec LiteSpeedLocal.dbo.LiteSpeed_DeleteActivity
@delLocal=1
, @delCentral=1
, @purgeDeleted = 1
, @deleteDate = '20090821 00:00:00'
```

2. Delete the following data older than 6 months in the Local repository:

- LiteSpeed activity
- Log shipping history
- Data from dbo.DbMaintStatus, dbo.DTSSStatus, dbo.JobStatus

```
USE LiteSpeedLocal_1683
EXEC LiteSpeed_DeleteActivity
@delLocal=1,
@delLogshipping = 1,
@delStatus = 1 ,
@delUnit = 6,
@delUnitType = 'MONTHS'
```

## Upgrade Repositories

LiteSpeed automatically upgrades LiteSpeed repositories during local install. If you install LiteSpeed remotely, the central repository is not installed or updated, you need to run the Instance Configuration wizard from the Start menu and select to configure the central repository. If you install the new central repository, you need to register it in the LiteSpeed UI Console.

If you have a LiteSpeed central repository that needs to communicate with a repository for a previous version of LiteSpeed, do one of the following:

- Upgrade the statistic agent that transfers information between them on the outdated server instance. To do this, run the Remote Deploy Configuration wizard and select **Statistics Agent Only** on the Installer Path page.
- Upgrade both the server where the LiteSpeedLocal and LiteSpeedCentral to the same version of LiteSpeed.
- Rename the LiteSpeedLocal and LiteSpeedCentral databases and recreate them by running the Instance Configuration wizard.

## Push Statistics to Central Repository

There are two options for replicating statistics to the central repository. The first option replicates the LiteSpeed activity. The second option only replicates job status, maintenance plans and log shipping information.

If you did not select to populate the central repository automatically or via a scheduled job during Instance Configuration, you can manually force a replication at any time by calling the ActivityManager.exe file from the command line or using the extended stored procedures.

### *To replicate database and LiteSpeed activity information*

- » On the local repository server, do one of the following:
  - Execute `dbo.xp_replicate_activity_statistics` against the master database.

- From the command line, change the directory until are in the directory containing the LiteSpeed command-line utilities (Usually, *C:\Program Files\Quest Software\LiteSpeed\SQL Server\Engine*) and run *ActivityManager.exe*.

*To replicate job, maintenance plans and log shipping information*

- » On the local repository server, do one of the following:
  - Execute *dbo.xp\_replicate\_job\_statistics* against the master database.
  - From the command line, change the directory until are in the directory containing the LiteSpeed command-line utilities (Usually, *C:\Program Files\Quest Software\LiteSpeed\SQL Server\Engine*) and run *ActivityManager.exe -GatherJobStats*.



## Additional Resources

### LiteSpeed Community

Get the latest product information, find helpful resources, and join a discussion with the LiteSpeed Quest team and other community members. Join the LiteSpeed community at:

<http://communities.quest.com/community/litespeed>.

Use this site to:

- Share your knowledge
- Participate in forums
- Learn about new features and enhancements in Quest products
- Download the latest product releases
- Find expert tips and tricks
- Communicate with product teams
- Participate in beta programs

### Video Tutorials

If you want to learn more about using NetVault LiteSpeed for SQL Server, check out <http://communities.quest.com/blogs/dbmanagement/> the Quest Database Management blog where you can find video tutorials, user tips and usage examples.

### Useful Web Resources

Share and find information about Quest products and SQL Server related technologies at:

Database Management Blog	<a href="http://communities.quest.com/blogs/dbmanagement">http://communities.quest.com/blogs/dbmanagement</a>
SQLServerPedia Community	<a href="http://sqlserverpedia.com">http://sqlserverpedia.com</a>
Quest SQL Server Products Community	<a href="http://communities.quest.com">http://communities.quest.com</a>
Microsoft SQL Server World Wide Users Group	<a href="http://www.sswug.org">http://www.sswug.org</a>
SQL Server Central	<a href="http://www.sqlservercentral.com">http://www.sqlservercentral.com</a>

SQL Server Performance.com	<a href="http://www.sql-server-performance.com">http://www.sql-server-performance.com</a>
MSSQLServer.com	<a href="http://www.mssqlserver.com">http://www.mssqlserver.com</a>
Solid Quality Learning	<a href="http://www.solidq.com">http://www.solidq.com</a>
Microsoft Product Support	<a href="http://support.microsoft.com">http://support.microsoft.com</a>
Microsoft Windows Compatibility Center	<a href="http://www.microsoft.com/windows/compatibility">http://www.microsoft.com/windows/compatibility</a>
Microsoft Developer Network (MSDN)	<a href="http://msdn.microsoft.com">http://msdn.microsoft.com</a>
Professional Association for SQL Server	<a href="http://www.sqlpass.org">http://www.sqlpass.org</a>
Microsoft SQL Server Developer Center	<a href="http://msdn.microsoft.com/en-us/sqlserver">http://msdn.microsoft.com/en-us/sqlserver</a>
TechNet Site	<a href="http://technet.microsoft.com">http://technet.microsoft.com</a>
Microsoft Accessibility Web Site	<a href="http://www.microsoft.com/enable">http://www.microsoft.com/enable</a>
Microsoft SQL Server Web Site	<a href="http://www.microsoft.com/sql">http://www.microsoft.com/sql</a>
IBM Tivoli Storage Manager Information Center	<a href="http://publib.boulder.ibm.com/infocenter/tsminfo/v6r3/index.jsp">http://publib.boulder.ibm.com/infocenter/tsminfo/v6r3/index.jsp</a>

## Review Known Issues

The following is a list of issues known to exist in this version of LiteSpeed.

Feature	Known Issue	Issue ID
Installation	To install LiteSpeed on SQL Server 2012, you may need to launch Instance Configuration manually from the Start Menu or use SQL Server authentication when configuring server instances. Otherwise, you will need to add the NT AUTHORITY\SYSTEM account to the sysadmin server role. See "Privilege and Grant Requirements" (page 274) for more information.	ST#100257
	On some machines, the following error may occur during the remote deploy: "Instance Configuration failed (204): Could not install service", though the service is installed and configured properly. You can check the services to ensure that the LiteSpeed Activity Service has been installed.	ST#100267
Backup and Restore	Mirroring is not supported for TSM backups.	ST#49305
Backup Analyzer	After the required number of bytes is received for analysis, the process is intentionally aborted. This generates the VDI error messages in the LiteSpeed log files and the SQL Server error log. Please ignore them.	ST#79697
Log Shipping	The publisher and subscriber servers must have the same version of LiteSpeed.	ST#52230
	In the native SQL Server 2000 log shipping plans, if the monitoring server is not the same as the subscriber server, information about the restore time is based on the restore job history and may be different from the actual restore time.	ST#98644
Object Level Recovery	Object Level Recovery does not support UDT SQL Server data types.	ST#46004
	Object Level Recovery is not able to read the backup of a SQL Server 2008 table that has FILESTREAM data.	ST#63900
	Object Level Recovery does not support databases encrypted using SQL Server 2008 Transparent Data Encryption (TDE).	ST#66780

Feature	Known Issue	Issue ID
Maintenance Plans	Legacy native plans on SQL Server 2008 may fail due to Microsoft backward compatibility issue. See <a href="http://support.microsoft.com/kb/955626">http://support.microsoft.com/kb/955626</a> .	ST#60281
	Maintenance Plans do not support regular expressions with the IgnoreCase option. To ignore case, use ( [Aa] [Bb] [Cc] ) instead of (?i:abc) .	ST#94513
Log Reader	User information displays for the operations in the sessions that are currently opened. Backups do not contain such information, there is only SPID information available for them.	ST#37112
	Log Reader does not support row-level and page-level compression.	n/a

## Configure Logging in LiteSpeed

Logging is available for the following areas in LiteSpeed:

- Core Engine
- Installation and Remote Deploy
- Maintenance Plans
- Log Shipping
- Object Level Recovery
- LiteSpeed UI Console Activity

**Note:** Log shipping plans activity is also logged and displayed in the History tab of the Log Shipping pane (CTRL+2). See the Configure Log Shipping guide for more information.

### Installer Logging

Installer log files are created in the default output directory. When you use the Remote Deploy Configuration wizard to deploy LiteSpeed on SQL Server instances, this creates a 'RemoteDeploy'-prefixed log file on the machine where you run the remote deploy from and 'SLSInstall'-prefixed log files on all target servers. See "Log File Naming and Location" (page 287) for more information.

### Backup/Restore Logging

Using the following instructions you can enable logging for a particular backup or restore activity. To log all backup/restore activity on the server, see Instance-Wide LiteSpeed Logging (page 286).

#### *To enable backup/restore logging in wizards*

- » Access options or advanced options and set the logging level.

#### *To enable backup/restore logging in procedures*

- » Supply the @logging parameter with one of the following values:
  - 0—Logging off.
  - 1 —Logging on. Log file is removed on success.
  - 2 —Logging on.

LiteSpeed will log the backup and restore operations in the default output directory. To log to a different directory add @Trace='logpath=path'.

Example:

```
EXEC master.dbo.xp_backup_database  
@database='CD'  
, @filename='C:\CD.BAK'  
, @logging=1  
, @Trace='logpath=C:\Documents and Settings\CD_backup_logs'
```

See "LiteSpeed Command-Line Arguments" (page 90) for more information about the command-line logging parameters and examples.

## Instance-Wide LiteSpeed Logging

LiteSpeed's advanced tracing facility allows for even more granular control over the major LiteSpeed features at the server level.

### *To enable/disable advanced logging*

1. Right-click an instance in the server tree, select **Support** and then **LiteSpeed Logging**.
2. Select one or more of the following options:
  - Core Engine—To log all backup/restore activity on the server, including log shipping and Fast Compression backups, but not Maintenance Plans backup tasks. See "Reporting and Logging in Maintenance Plans" (page 288) for more information about logging options in Maintenance Plans.
  - Object Level Recovery (OLR)—To log read operations and object-level recovery.
  - Log Ship—To log backup/copy/restore Log Shipping jobs, but not transaction log backups and restores.
  - Fast Compression—To log Fast Compression jobs and Fast Compression backups.
  - Smart Cleanup—To log Smart Cleanup activity.

Clear one or more checkboxes to disable advanced logging for those options.

3. Complete the dialog. Review the following for additional information:

Option	Description
Log file path	This parameter specifies a path to a default output directory into which the log files are written. See "Log File Naming and Location" (page 287) for more information.
Flush output to log file after every	LiteSpeed will perform a disk flush after each record. Select this parameter if you experience program exceptions resulting in a process abort.

Option	Description
write (slower)	
Delete log file on success	A log file is only saved if an error has occurred during execution.
Log rollover size	This will limit the size of the log file, so that only the last records that fit within the specified size will be kept. Use this option in the case of a long running application, when the log file becomes overly large, or when an error happens near the end of execution. Otherwise, set the rollover size value to zero.

**Notes:**

- Enabling Core Engine advanced logging will not override the logging level specified in wizards and procedures.
- Be sure to disable advanced logging after it is no longer needed.

## Log File Naming and Location

By default, log files are written to *C:\Documents and Settings\All Users\Application Data\Quest Software\LiteSpeed\SQL Server\Logs*. This location is specified by the *logpath* value in the *LiteSpeedSettings.ini* file. (Usually, *C:\Documents and Settings\All Users\Application Data\Quest Software\LiteSpeed\SQLServer\LiteSpeedSettings.ini*.)

*To change the default output directory*

1. Right-click an instance in the server tree, select **Support** and then **LiteSpeed Logging**.
2. Enter a new location in the Log file path field.

Log files will have one of the following formats:

- Source YYYY-MM-DD hh-mm-ss PPPP.Log
- Source YYYY-MM-DD hh-mm-ss PPPP TTTT.Log
- SLSInstaller - YYYYMMDD hh-mm-ss-xxx.txt

Where source components are the following:

- SLS (Backup or restore operations)
- OLR
- LogShip

- FastComp
- SCleanup

Other components of the log file name are as follows:

- YYYY—4-digit year
- MM—2-digit month
- DD— 2-digit day of the month
- hh— 2-digit hour using the local 24-hour clock
- mm—2-digit minute
- ss— 2-digit second
- xxx—3-digit millisecond
- PPPP—4-digit process id
- TTTT— 4-digit thread id

When you use the Remote Deploy Configuration wizard to deploy LiteSpeed on SQL Server instances, this creates a 'RemoteDeploy'-prefixed log file on the machine where you run the remote deploy from and 'SLInstall'-prefixed log files on all target servers.


## LiteSpeed UI Console Activity Logging

The LiteSpeed UI Console activity is logged in the Windows Application event log.

*To set the LiteSpeed UI Console logging level*

1. Select **Tools | Options**.
2. On the General tab, select logging level.

## Reporting and Logging in Maintenance Plans

With your maintenance plan open in the Design tab, click  to configure reports about execution of the entire maintenance plan and set the reporting and logging options. A report contains information about each task within one subplan: task name, duration, results.

Review the following for information about logging in Maintenance Plans:

Option	Description
Log extended information	<p>Logs plan details in msdb.dbo.sysmaintplan_logdetail for both LiteSpeed and native SQL Server SSIS maintenance plans, for all tasks. This additional information is task-specific and is displayed in the Extended Log Information pane of the History tab.</p> <p>For the native SQL Server plans, this information also becomes a part of the native reporting and may increase the size of the stored maintenance plan</p>



Option	Description
	history.
Log to remote Server	Writes the plan history to the remote server.
Debug Logging	<p>Logs LiteSpeed maintenance plan utility activity.</p> <p>Quest Support may ask you to enable debug logging in maintenance plans to help troubleshoot product issues. Make sure LiteSpeed is installed on the server where you have a problem with maintenance plans.</p> <p>When a subplan is executed, the Maintenance Plan engine saves the archived &lt;plan_name&gt;_&lt;subplan_name&gt;[&lt;GUID&gt;].zip file in the default output directory. See "Configure Logging in LiteSpeed" (page 285) for more information about the default output directory.</p> <p>Each archive has:</p> <ul style="list-style-type: none"><li>• one or more files for all LiteSpeed tasks in a subplan, including the Reporting task</li><li>• one file containing <code>-Gather</code> command-line command to collect statistics</li><li>• one file containing <code>-CollectStats</code> command-line command for the process that starts collecting statistics to repositories (only created for SSIS plans)</li></ul> <p>LiteSpeed's debug logging is not supported for the native SQL Server tasks. If the <b>Use LiteSpeed</b> option is cleared for a task, then that task is handled by the native SQL Server components.</p>

## Troubleshoot LiteSpeed Activity

### Local repository is not populated

If the computer name is changed, the local repository settings are not found, because LiteSpeed refers to the old computer name. As a result, the Local Repository database is not being populated and you cannot view LiteSpeed activity.

You need to update @@SERVERNAME for all instances on that computer and re-run the Instance Configuration wizard.

#### *To update @@SERVERNAME value*

» Run the following:

```
exec sp_dropserver 'old_name'  
exec sp_addserver 'new_name', 'local'
```

#### *To run Instance Configuration*

1. Select **Start | All Programs | Quest Software | LiteSpeed for SQL Server | Instance Configuration**.
2. Review the names of the instances on the SQL Instances step.
3. Complete the wizard.

### 4.x jobs not displayed

The timeline does not automatically show 4.x backup/restore jobs. However, you can manually rename the jobs and they will appear in the timeline.

#### *To view 4.x jobs*

- Rename backup jobs to:

```
LiteSpeed Backup SERVER_NAME.database_name
```

- Rename restore jobs to:

```
LiteSpeed Restore SERVER_NAME.database_name
```

Be sure to use upper case for *SERVER\_NAME*. Additionally, you need to add a comment to maintenance plans to provide information about the databases involved in the plan:

- Comment should begin with `/**` and end with `**/`
- Specify the databases inside `{}`. Use a semicolon as a separator. For example:  
`{northwind;pubs;litespeedlocal}`
- `{*system}` means all system databases.

- `{*user}` means all user databases.
- `{*}` means all databases
- It is possible to use combinations. For example:  
`{northwind;pubs;litespeedlocal}{*system}` means all system databases plus specified. `{master}{*user}` means all user databases plus master.

## Troubleshoot Maintenance Plans

### Leverage SSIS and LiteSpeed advanced options

Some tasks and advanced options can be unavailable if you do not select **Use LiteSpeed** when configuring tasks. See "About Automating Maintenance Tasks" (page 61) for more information.

### Install Backward Compatibility components

If you receive one of the following error messages, you probably need the Backward Compatibility components installed:

- "Executed as user: ... sqlmaint.exe failed. [SQLSTATE 42000] (Error 22029). The step failed."- double-click a plan in the History page to view the event message.
- "The SQLDMO 'Application' object failed to initialize." - run the maintenance plan task script in SSMS 2008.

For more information, see <http://support.microsoft.com/kb/955626>.

### Resolve upgrade issues

LiteSpeed plans may fail after the in-place upgrade of SQL Server. See [SOL69809](#) for more information.

### Analyze log information

Enable advanced logging and review log files for errors or work with quest support to identify and resolve plan problems. See "Reporting and Logging in Maintenance Plans" (page 288) for more information.

**Note:** For all scenarios, the best is to upgrade all of your servers to the latest LiteSpeed version.

## Troubleshoot Performance-Related Issues

LiteSpeed uses Virtual Device Interface (VDI) to generate its backups. This operates in a single contiguous region of addresses within the SQL Server process space known as MemToLeave memory area. It is set aside at startup and is left unallocated by SQL Server for the use by components in the SQL Server process space, such as extended procedures, COM/OLE Automation objects, and linked server queries. Memory allocations by SQL Server larger than 8 KB are also made from the MemToLeave area.

SQL Server's MemToLeave area becoming fragmented so that there is insufficient contiguous space to allocate the buffers required for the backups. Rebooting SQL Server will free the memory, but the underlying cause of memory fragmentation should be addressed for long-term resolution.

Factors that can drain this memory area:

- Number of databases on each server
- Number of Servers on physical machine
- Number of concurrent users
- Amount of data
- Number of data/log files
- Running 3rd party software

*To check the available contiguous memory, do one of the following:*

- In the Server tree, right-click an instance and select **Properties**.
- Run the xp\_memory\_size extended stored procedure. See "xp\_memory\_size" (page 171) for more information.

## Troubleshoot Previous Versions of LiteSpeed

If you experience issues with the previous versions of LiteSpeed and they are not discussed in the documentation set delivered with your LiteSpeed version, please visit <http://support.quest.com> to access support documentation and search the extensive Knowledgebase for published solutions and case data.

Note that for most cases the best is to upgrade all of your servers to the latest LiteSpeed version.

## Create Support Bundles

If you have not found an answer to your question, you can create a support bundle and send it to customer support. The support bundle contains information about your database, system configuration, and settings and can help troubleshoot problems.

### *To create a LiteSpeed UI Console support bundle*

- » Select **Help | Support Bundle**.

### *To create a server support bundle*

1. Right-click a server in the tree and select **Support Bundle**.
2. Send the support bundle to Quest customer support or click **Clipboard** to save the generated content to file.

# Appendix: Contact Quest

## About Quest Software

Established in 1987, Quest Software (Nasdaq: QSFT) provides simple and innovative IT management solutions that enable more than 100,000 global customers to save time and money across physical and virtual environments. Quest products solve complex IT challenges ranging from database management, data protection, identity and access management, monitoring, user workspace management to Windows management. For more information, visit [www.quest.com](http://www.quest.com).

## Contact Quest Support

Quest Support is available to customers who have a trial version of a Quest product or who have purchased a Quest product and have a valid maintenance contract. Quest Support provides unlimited 24x7 access to our Support Portal. Visit our Support Portal at: [www.quest.com/support](http://www.quest.com/support).

From our Support Portal, you can do the following:

- Retrieve thousands of solutions from our Knowledge Base
- Download the latest releases and service packs
- Create, update and review Support cases

View the *Global Support Guide* for a detailed explanation of support programs, online services, contact information, policies and procedures. The guide is available at: [www.quest.com/support](http://www.quest.com/support).

## Contact Quest Software

Email [info@quest.com](mailto:info@quest.com)

Quest Software, Inc.

World Headquarters

Mail 5 Polaris Way

Aliso Viejo, CA 92656

USA

Web site [www.quest.com](http://www.quest.com)

See our Web site for regional and international office information.



# Index

	%		multi-database	48
			primary replicas	47
			secondary replicas	47
% variables		271	t-log	38
	.		template	31
.lsm		80	test optimal settings	56
	<b>A</b>		view activity	86
activity logging	See also view activity		wizard	38
activitymanager.exe		279	Backup Analyzer	56
Adaptive Compression		54	backup cleanup	See Smart Cleanup
advanced logging		286	backup escalation	51
AlwaysOn Availability Group		47	backup files	
argument data type		139	attach files	45
attachments			convert LiteSpeed backup to SQL Server backup	129
add to backup file		45	read to restore objects	80
list files and directories		216	verify integrity	75
restore from backup file		76	backup history	86
availability group support	See AlwaysOn Availability Group		backup statistics	86
			backup template	
	<b>B</b>		select databases	32
background processes		17	backup templates	31
backup			export	36
cleanup		58	import from file	36
database		38	view history	36
differential		38	backup wizard	38
Fast Compression		48		<b>C</b>
file and filegroup		38	categories	22
full		38	assign servers and databases to	23

create	23	defaults	24
edit	23	deploy backup templates	37
group by in LiteSpeed Console	22	Double Click Restore	53
central pane	16	requirements	274
central repository	19	restore	77
change log directory	See default log output directory	<b>E</b>	
clone backup templates	36	encryption	56
command-line (LiteSpeed)	90	exe backup files	See Double Click Restore
Fast Compression	48	export backup templates	36
general commands	90	extended stored procedures	139
Object Level Recovery commands	133	extraction utility	129, See also LiteSpeed utilities
Smart Cleanup	117	extractor.exe	See extraction utility
compression	54	<b>F</b>	
contact Quest support	296	Fast Compression	48
convert		backup escalation	51
backups to Double Click Restore	54	<b>I</b>	
LiteSpeed backups	121	import backup templates	36
Maintenance Plans	61	IntelliRestore	75
copyright	2	IO failure	See Network Resilience
custom database selection	32	<b>K</b>	
<b>D</b>		known issues	283
databases		<b>L</b>	
categorize	22	LicenseInfoCmd utility	137
group	22	LiteSpeed Console	15
restore	74	LiteSpeed defaults	24
DCR	See Double Click Restore	LiteSpeed logging	286
DDL scripts, create	178	LiteSpeed utilities	90
default log output directory	286	LiteSpeed variables	271

LiteSpeed_DeleteActivity	276	read backup file	80
log files	See logging	restore from TSM	83
Log Reader		restore tables	80
options	30	wizard	80
logging	285	OLR	See Object Level Recovery
<b>M</b>		options (Job Manager)	29
Maintenance Plans	61	options (LiteSpeed)	26
back up database	66	Backup Manager	28
change plan owner	65	Log Shipping	28
clean up maintenance data	71	options (Log Reader)	30
copy plans and subplans	71	owner of a maintenance plan	65
create	62	<b>P</b>	
debug logging	288	permissions	See also privileges
logging	288	preferred replica	47
prerequisites	61	primary replicas	47
view activity and history	88	privileges	274
MemToLeave	293	processor affinity	25
mirror backup files	39	properties	17
monitor history		<b>R</b>	
backup	86	register repositories	19
Maintenance Plans	88	register server instances	20
multi-database backup	48	remove deployed templates	38
<b>N</b>		reports	See view activity
navigation pane	15	reset LiteSpeed defaults	24
Network Resilience	58	resilience	See Network Resilience
<b>O</b>		resiliency	See Network Resilience
Object Level Recovery	80	restore	74
command-line interface	133	databases	74
execute SELECT statements	83	objects	80

wizard	74	timeline	86
<b>S</b>		toolbars	15
secondary replicas	47	troubleshoot TSM	See also logging
SELECT statements	84	tutorials	281
execute	83	<b>U</b>	
server groups		update central repository	279
assign servers to	22	upgrade repository	279
create	21	user interface	15
group by in LiteSpeed Console	22	utilities	See LiteSpeed utilities
server instances	19	<b>V</b>	
assign to server group	22	variables	271
categorize	22	video tutorials	281
group in LiteSpeed Console	22	view activity	86
set affinity mask	25	view argument data type	See argument data type
set default output directory	See default log output directory	<b>X</b>	
SLSFastCompression	106	xp_backup_database	140
Smart Cleanup	58	xp_backup_log	156
procedure	255	xp_delete_tsmfile	169
utility	117	xp_encrypt_backup_key	171
support bundle	295	xp_encrypt_restore_key	171
<b>T</b>		xp_memory_size	171
T-SQL arguments	90	xp_objectrecovery	172
Object Level Recovery	133	xp_objectrecovery_createscript	178
tables, restore	80	xp_objectrecovery_executeselect	188
technical support	296	xp_objectrecovery_viewcontents	184
template	31	xp_replicate_activity_statistics	279
threads	24	xp_replicate_job_statistics	279
throttle	24	xp_restore_attachedfilesonly	194

xp_restore_checkpassword	199
xp_restore_checksumonly	200
xp_restore_database	200
xp_restore_filelistonly	212
xp_restore_headeronly	215
xp_restore_log	224
xp_restore_setinfo	222
xp_restore_verifyonly	234
xp_slsCreateDCR	241
xp_slsFastCompression	242
xp_slsReadProgress	254
xp_slsSmartCleanup	255
xp_sqlitespeed_licenseinfo	259
xp_sqlitespeed_version	261
xp_view_tsmcontents	261
xp_view_tsmmc	268