Quest

LiteSpeed® for SQL Server® 8.8

User Guide

Patents

Quest Software is proud of our advanced technology. Patents and pending patents may apply to this product. For the most current information about applicable patents for this product, please visit our website at https://www.quest.com/legal.

Trademarks

Legend

CAUTION: A CAUTION icon indicates potential damage to hardware or loss of data if instructions are not followed.

IMPORTANT, NOTE, TIP, MOBILE, or VIDEO: An information icon indicates supporting information.

LiteSpeed User Guide
Updated - March 2019
Version - 8.8

Contents

1

About Backing Up/Restoring with LiteSpeed

About LiteSpeed

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.

About Backing Up/Restoring Databases

LiteSpeed has the following key features:

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. For more information, see Test Optimal Backup Settings on page 83.
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: • encrypt (For more information, see Encryption Methods on page 123.) • compress (For more information, see Compression Methods on page 121.) • mirror • stripe • attach files

Use this feature…	To…
	• log activity • and more... Be sure to select **Use LiteSpeed** to leverage many LiteSpeed advanced features. For more information, see Back Up Databases on page 98.
Multi-Database Backup Wizard	Back up several databases with the same options. For more information, see Multi-Database Backup on page 115.
Backup Templates	Automate backing up databases on multiple server instances by deploying a LiteSpeed Backup Template. For more information, see Create Backup Templates on page 87.
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. For more information, see About Automating Maintenance Tasks on page 128.
Fast Compression	Reduce backup size and decrease backup times from hours to minutes by including differential backups in the nightly backup routine. For more information, see Fast Compression on page 115.
Double Click Restore	Restore Double Click Restore executable files on a server instance that does not have LiteSpeed installed. For more information, see Double Click Restore Executables on page 120.
Network Resilience	Control network resilience options. LiteSpeed's read and write resilience can handle various failures on both network and attached storage devices. For more information, see Network Resilience on page 124.
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. For more information, see Compression Methods on page 121.
Restore Wizard	Restore databases and attached files immediately or schedule a restore job. For more information, see Restore Databases Using the Restore Wizard on page 149.
Automated Restore	Automate restoring the most recent database backups. If backing up to unique filenames, you only need to specify a folder where LiteSpeed will look for candidates for restore. For more information, see Restore Databases Using the Restore Wizard on page 149.
Object Level Recovery	List, query, preview and restore specific objects directly from the native SQL Server or LiteSpeed backup files. For more information, see Restore Objects on page 163.
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.

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

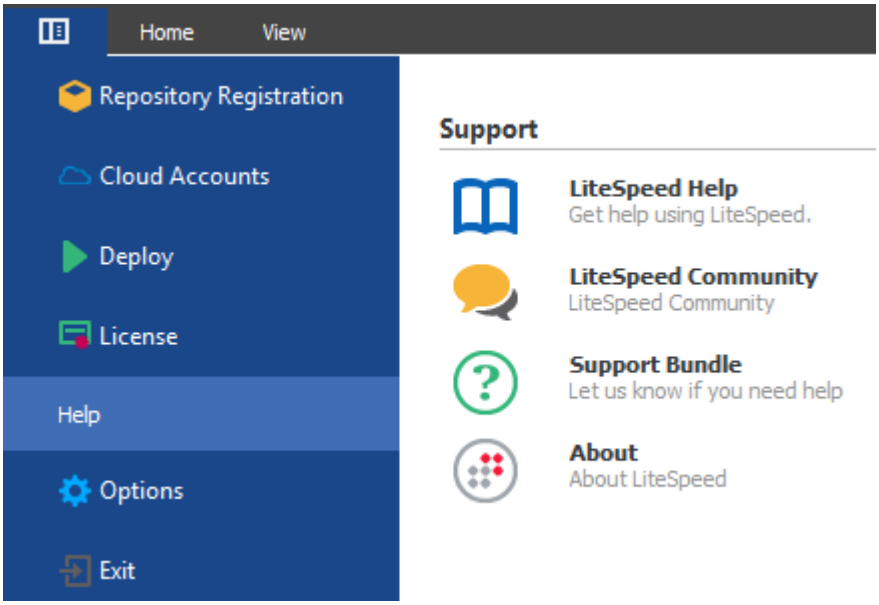
Use the LiteSpeed UI Console to view activity and history for your backups, including processes that fail or succeed, the amount of disk space you save, a list of all of the jobs for a server instance or database. For more information, see View Backup Manager Activity and History on page 174.

LiteSpeed User Interface

Use the LiteSpeed user interface (called the LiteSpeed UI Console) to create backups, restore databases, recover database objects, view activity and history for your backups, design Maintenance Plans, schedule SQL Agent jobs and monitor tasks progress.

Application Menu

The Application menu displays often used LiteSpeed features and replaces the previous File menu.



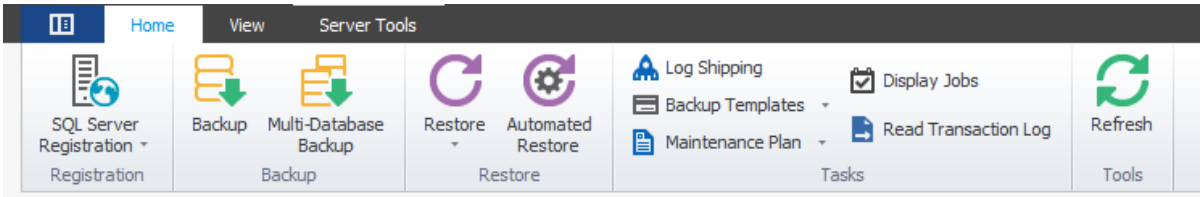
Use the Application menu to access the following features:

Feature	Description
Repository Registration	Display and edit repository registration. You can add repository, edit

Feature	Description
	connection, delete connection, test connection, import, and export repositories. For steps on repository registration, refer to Register Central Repositories.
Cloud Accounts	Display and edit registered cloud account settings. You can add, edit, delete, import, and export cloud accounts.
Deploy	Run the Remote Deploy Configuration wizard. Use this wizard to deploy the LiteSpeed core components and edit LiteSpeed the configuration to remote servers.
License	Display and edit the LiteSpeed license key information.
Help	Display the LiteSpeedSupport information. Displays the LiteSpeed Help, LiteSpeed Community, Support Bundle, and About information. The File ribbon defaults to the Help selection.
Options	Display and edit the following LiteSpeed options: General, Backup Manager Options), Log Shipping Options, Job Manager Options, Log Reader Options, and Object Level Recovery Options.
Exit	Exit the LiteSpeed product.

Home Ribbon

The Home ribbon displays often used features.



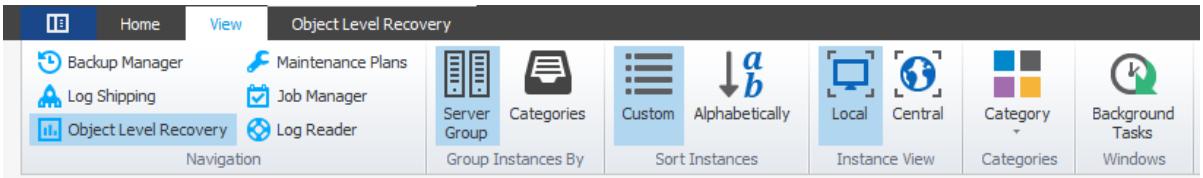
Use the Home ribbon to access the following features:

Feature	Description
SQL Server Registration	Register new SQL Servers and groups. You can also delete and edit existing ones. You can also manage licenses and edit credential information for each registered SQL Server instance. See License Registration.
Backup	Run the Backup Wizard.
Multi-Database Backup	Run the Backup Wizard for several databases.
Restore	Run the Restore Wizard. You can also restores databases, files and file groups, transaction logs, and attached files.

Feature	Description
Automated Restore	Run the Restore Wizard with Automated Restore as the default selection.
Log Shipping	Run the Create LiteSpeed Log Shipping Plan Wizard.
Backup Templates	Run the Create New Backup Template Wizard. You can also create new, edit, deploy or remove existing backup templates.
Maintenance Plans	Create a new maintenance plan by default. You can also create new maintenance plan, paste maintenance plan, import maintenance plan, and upgrade LiteSpeed maintenance plans.
Display Jobs	Run the Job Manager and displays the Job Manager Ribbon.
Read Transaction Log	Run the Read Log Wizard and displays the Log Reader Ribbon.
Refresh	Refreshes the selected items.

View Ribbon

The View ribbon displays often used view features.



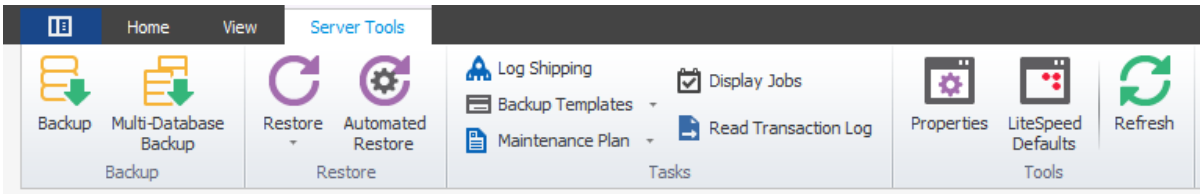
Use the View Ribbon to access the following features:

Feature	Description
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.
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.
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. **NOTE:** Object Level Recovery is only available with the Enterprise

Feature	Description
	license.
Maintenance Plans	Create a new maintenance plan by default. Several other items are available. Automate routine database maintenance tasks, such as backing up databases, updating statistics, and rebuilding indexes to run on a specific day and time.
Job Manager	Schedule, monitor, and manage SQL Agent jobs and Windows tasks.
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.
Server Group	View the list of servers grouped by instances.
Categories	View the list of servers grouped by category.
Custom	View instances in the navigation pane by a defined custom sort.
Alphabetically	View instances in the navigation pane by an alphabetic sort.
Local	View list of local instances in the navigation pane. The SQL Servers local server instance icon is displayed in the navigation pane. Note: The local repository instance icon (and central) are displayed in the View ribbon when a central repository is selected. When a central repository is not used, both the local and central icons are not visible in the View ribbon.
Central	View list of central instances in the navigation pane. The SQL Servers central server instance icon is displayed in the navigation pane. Note: The central repository instance icon (and local) are displayed in the View ribbon when a central repository is selected. When a central repository is not used, both the local and central icons are not visible in the View ribbon.
Category	Select the server category grouping based on location or role, assign categories, and edit categories.
Background Tasks	Display running background tasks.

Backup Manager Server Tools Ribbon

The Backup Manager Server Tools ribbon is displayed when you select a SQL Server from the Navigation Pane.

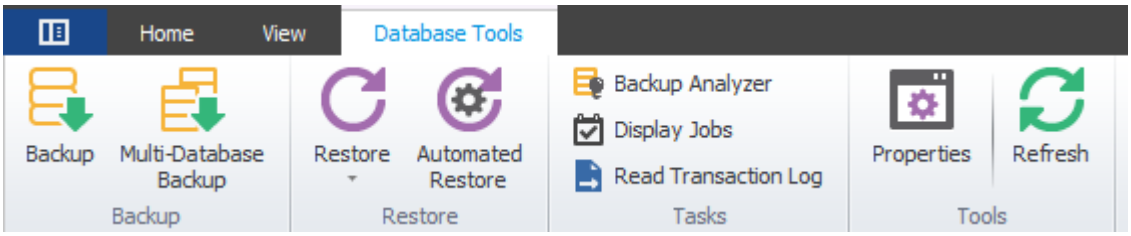


Use the Backup Manager Server Tools ribbon to access the following features:

Feature	Description
Backup	Run the Backup Wizard.
Multi-Database Backup	Run the Backup Wizard for several databases.
Restore	Run the Restore Wizard with database as the default selection. You can also restore files and file groups, transaction logs, and attached files.
Automated Restore	Run the Restore Wizard with Automated Restore as the default selection.
Log Shipping	View the LiteSpeed Log Shipping information in the navigation pane.
Backup Templates	Runs the Create New Backup Template Wizard. You can also create new, edit, deploy, and remove templates.
Maintenance Plans	Create a new maintenance plan by default. Several other items are available. Automate routine database maintenance tasks, such as backing up databases, updating statistics, and rebuilding indexes to run on a specific day and time.
Display Jobs	View the Job Manager and list of jobs in various states. Makes visible the Job Manager ribbon.
Read Transaction Log	Run the Read Log Wizard and LiteSpeed displays the Log Reader Ribbon.
Properties	Displays the general properties of the current server.
LiteSpeed Defaults	Displays the LiteSpeed backup default options. Refer to Configure LiteSpeed Defaults.
Refresh	Refreshes the selected item.

Database Tools Ribbon

The Database Tools ribbon is displayed when you select Backup Manager from the View Ribbon.

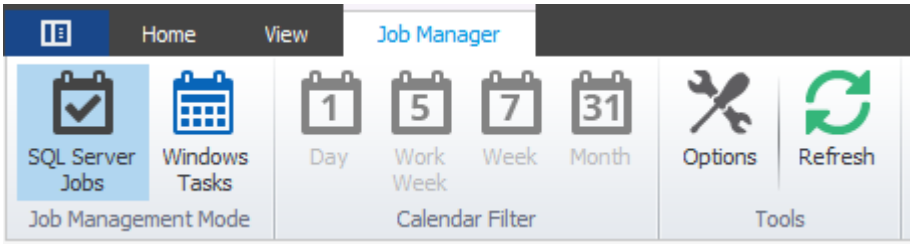


Use the Database Tools ribbon to access the following features:

Feature	Description
Backup	Run the Backup Wizard.
Multi-Database Backup	Run the Backup Wizard for several databases.
Restore	Run the Restore Wizard. You can also restores databases, files and file groups, transaction, and attached files.
Automated Restore	Run the Restore Wizard with Automated Restore as the default selection.
Backup Analyzer	Display analyzer graph of backup jobs.
Display Jobs	Display jobs and tasks according to their schedules and execution history.
Read Transaction Log	Display a listing of recent SQL Server transactions.
Properties	Display general LiteSpeed properties.
Refresh	Refresh the database items.

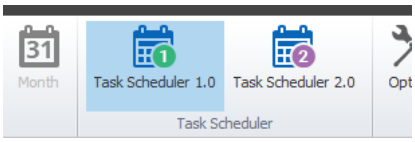
Job Manager Ribbon

The Job Manager ribbon is displayed when you select Job Manager from the View Ribbon.



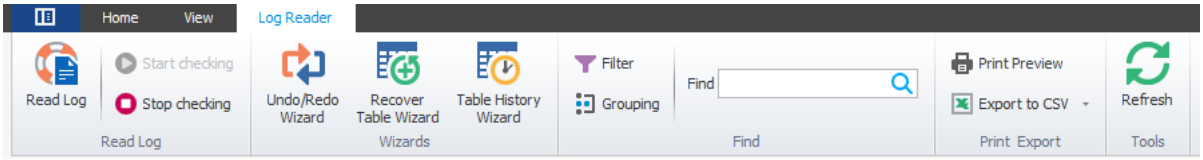
Use the Job Manager ribbon to access the following features:

Feature	Description
SQL Server Jobs	Select a server in the SQL Server Jobs pane.
Windows Tasks	When selected the Task Scheduler displays the Task

Feature	Description
	Scheduler 1.0 pane and the Task Scheduler 2.0 pane. The panes display two distinct separate task schedules that can be alternately viewed. LiteSpeed automatically detects the version of Windows Tasks. The Task Scheduler 2.0 includes support for Task Scheduler 1.0 tasks.  Select a server in the Windows Tasks pane
Day	Display jobs for a particular day.
Work Week	Display jobs for a particular work week.
Week	Display jobs for a particular week.
Month	Display jobs for a particular month.
Options	Select the options for Job Manager.
Refresh	Refreshes the Job Manager items.

Log Reader Ribbon

The Log Reader ribbon is displayed when you select Log Reader from the View Ribbon.



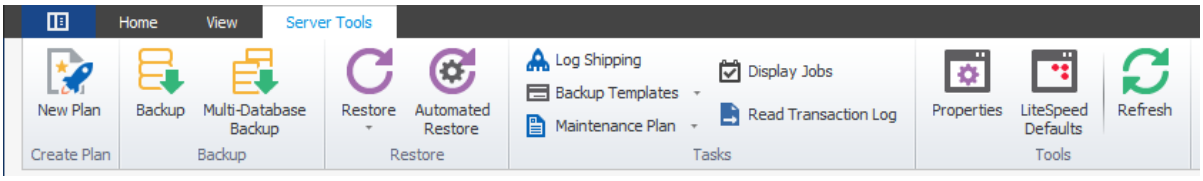
Use the Log Reader ribbon to access the following features:

Feature	Description
Read Log	Run the Read Log Wizard to generate a job transaction log.
Start checking	Begin checking log changes.
Stop checking	End checking log changes.
Undo / Redo Wizard	Run the Undo / Redo Wizard to generate a T-SQL script. that reverses changes previously made or reapplies

Feature	Description
	current changes.
Recover Table Wizard	Run the Recover Table Wizard to recover dropped tables.
Table History Wizard	Run the Table History Wizard to view changes made to specific tables.
Filter	Filter the log entries by general, objects, operations, logins, users, hosts, and applications. When selected the Filter option remains highlighted in the log reader ribbon.
Grouping	Group the log entries by column headers. When selected the Grouping option remains highlighted in the log reader ribbon.
Find	Search and locate log entries.
Print Preview	Displays a preview of currently selected log.
Export to CSV	Export table to a CSV file. Other export options include: HTML, XML, and DB.
Refresh	Refreshes the Log Reader items.

Log Shipping Server Tools Ribbon

The Log Shipping Server Tools ribbon is displayed when you select Maintenance Plans from the View Ribbon.



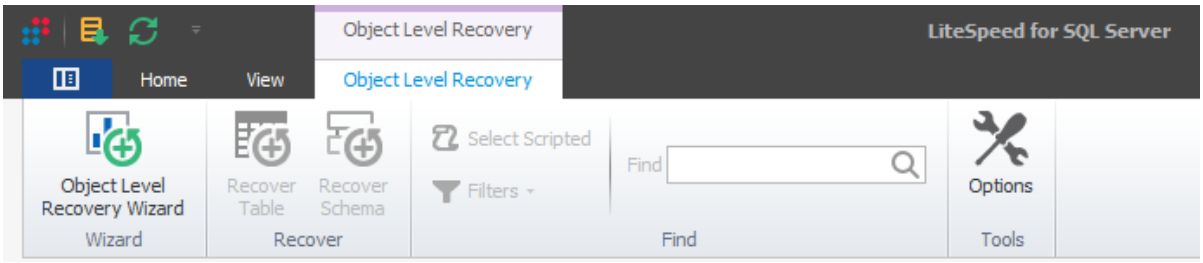
Use the Log Shipping Server Tools ribbon to access the following features:

Feature	Description
New Plan	Run the Create Log Shipping Plan Wizard.
Backup	Run the Backup Wizard.
Multi-Database Backup	Run the Backup Wizard for several databases.
Restore	Run the Restore Wizard with database as the default selection. You can also restore files and file groups, transaction logs, and attached files.

Feature	Description
Automated Restore	Run the Restore Wizard with Automated Restore as the default selection.
Log Shipping	View the LiteSpeed Log Shipping information in the navigation pane.
Backup Templates	Runs the Create New Backup Template Wizard. You can also create new, edit, deploy, and remove templates.
Maintenance Plans	Create a new maintenance plan by default. Several other items are available. Automate routine database maintenance tasks, such as backing up databases, updating statistics, and rebuilding indexes to run on a specific day and time.
Display Jobs	View the Job Manager and list of jobs in various states. Makes visible the Job Manager ribbon.
Read Transaction Log	Run the Read Log Wizard and LiteSpeed displays the Log Reader Ribbon.
Properties	Displays the general properties of the current server.
LiteSpeed Defaults	Displays the LiteSpeed backup default options. Refer to Configure LiteSpeed Defaults.
Refresh	Refreshes the selected item.

Object Level Recovery Ribbon

The Object Level Recoveryribbon is displayed when you select Object Level Recovery from the View Ribbon.



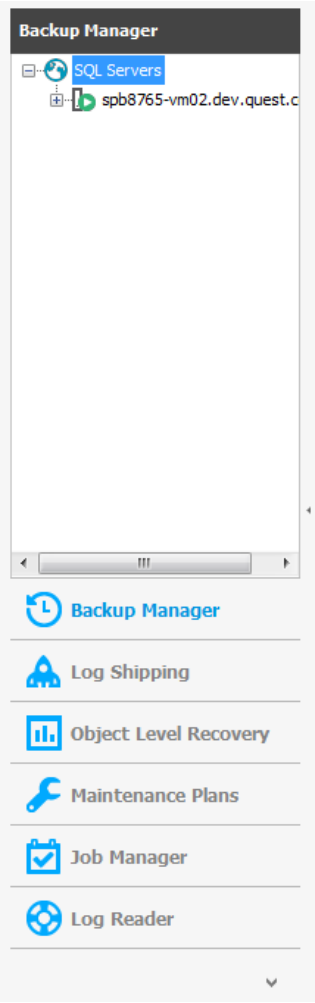
Use the Object Level Recovery ribbon to access the following features:

Feature	Description
Object Level Recovery Wizard	Read native SQL Server or LiteSpeed backups to start working with Object Level Recovery. (version 2: to view database objects, query and restore backup data, generate DDL scripts and more.) **NOTE:** Object Level Recovery is only available with the

Feature	Description
	Enterprise license.
Recover Table	Run the Recover Table Wizard.
Recover Schema	Recover a selected object.
Select Scripted	Select scripted objects.
Filters	Apply filters to the database object search. Optional filters include: All, Defaults, Extended Procedures, Functions, Partition Functions, Partition Schemes, Roles, Rules, Stored Procedures, Tables, Memory Optimized Tables, History Tables, Triggers, Types, Users, Views, Views (indexed). and Xml Schema Collection.
Find	Search for a specified database object.
Options	Open the Object Level Recovery options display.

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. When local servers are selected in the **View** ribbon, the SQL Servers local instance icon is displayed in the navigation pane. When central servers are selected in the **View** ribbon, the SQL Servers central instance icon is displayed in the navigation pane.



Tip: Click to change the features that appear as button selectable in the navigation pane.

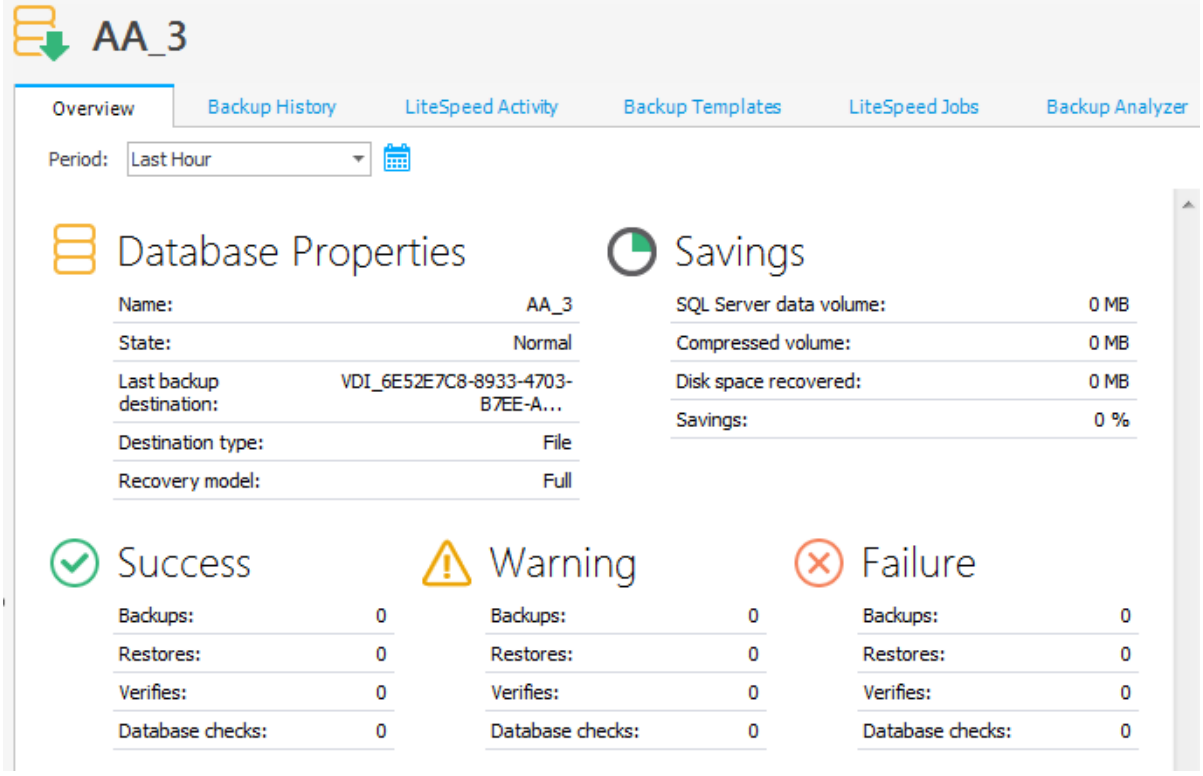
Use the Navigation pane to access the following features:

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	CTRL+3

Feature	Description	Keyboard shortcut
	one or more databases objects. **NOTE:** Object Level Recovery is only available with the Enterprise license.	
Maintenance Plans	Create a new maintenance plan by default. Several other items are available. 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

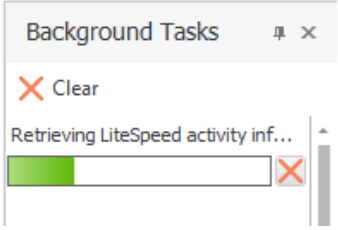
The central pane displays information based upon your selection in the navigation pane. Each feature has a unique home page and set of tabs. For a description of the Backup Manager tabs and central pane options, see View Backup Manager Activity and History.



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 **Auto Hide** or clear a process by clicking .

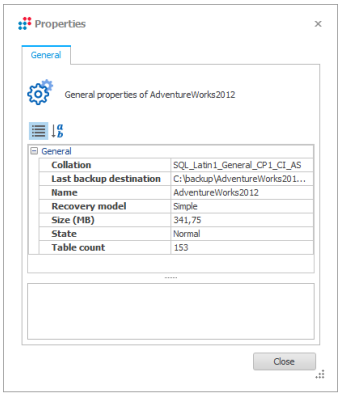


To view the pane, select **View | Background Tasks**.

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.



To view the properties pane, click the Application Button and select **Options** and then **Display properties in dockable window** on the General tab. You can dock the pane by clicking .

Toolbar

The toolbar has the following parts:

Toolbar Part	Description
Navigation	Includes buttons to navigate within the LiteSpeed UI Console, including Minimize the Ribbon, Back, Forward, Home, and Help:

Toolbar Part	Description
	
Quick Access Tool Bar	Includes buttons to navigate within the LiteSpeed UI Console, including Start Backup Wizard, Refresh selected item, and Customize Quick Access Toolbar: 

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

3

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.

Note: If the LiteSpeed install includes a central repository and the console, then LiteSpeed defaults the console to use that central repository automatically without the need to manually register and select it in the LiteSpeed console.

To register a repository

1. Click **Repository Registration** from the **Application Menu**.
   Alternately you can click beside the **Central Repository** field at the bottom of the LiteSpeed UI Console and select **Select Repository**.

2. Click **Add repository** and complete the Register New Repository dialog fields.

Server name:	Enter the name of the server instance, or click to select from a list of available server instances.
Display name:	Enter the name to display in the navigation pane tree.
Connect using:	Select the following options:
Windows authentication	Select this option to connect to the SQL Server instance using Windows authentication.
SQL Server Authentication	Select this option to connect to the SQL Server instance using SQL Server authentication. Complete the **Login name** and **Password** fields.
Additional Connection Parameters	Enter additional connection string parameters. **TIP:** Use this selection as needed to include custom parameters such as encryption, AlwaysOn and others to the SQL server connection string.

Repository database:	Displays the name of the repository database selected.
Repository version:	Displays the version of the repository.

3. Click **Test connection**.

TIPS:

- Select add repository, edit connection, delete connection, test connection, import, and export by clicking the corresponding button in the Edit Repository Registration dialog. Imports can be done from .csv files and from LiteSpeed 4.x.
- 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

SQL Server Instances

You must register server instances before you can manage them in the LiteSpeed UI Console. You can register them one at a time, or import registrations from a CSV file, LiteSpeed central repository, your native SQL Server tool (for example, Management Studio) or a previous version of LiteSpeed.

Before registering instances, review the following information on existing instance registration and centralized instance management.

Existing instance registration

LiteSpeed has the following capabilities for existing instance registration:

- Keeps everything in sync if you are not using a central repository.
- Preserves grouping in the export file.

- The import shows all instances in the import file (like is done for manual imports from the central) and automatically deselects instances that are already registered in the console.
- Grouping information is used in the export file to put the instances where they need to be. Groups are created automatically as needed when importing. If there are duplicate names, they are handled accordingly.

Centralized instance management

LiteSpeed has the following capabilities for centralized instance management:

- Shares a common set of instance registrations and maintain grouping across all consoles.
- The **Backup Manager** View tab is used to switch between local and central repository views (if a central is used).
- The central does not store passwords for SQL Server Authentication. Passwords are still stored locally for the user.
- In the central repository view, all instances are read from the central, including their proper grouping.
- In the central view, any instances that are added, removed, or sorted, automatically get persisted into the central.
- You can change from Windows Authentication to SQL Authentication, or change the account used for SQL Authentication without affecting the central repository.
- Login credentials are not persisted in the central. The actual password, if SQL Authentication is used, is stored locally. The only thing stored centrally is the type of authentication (Windows or SQL).
- Changes to the central, trigger a version change. This helps you know when the central data has changed.
- When LiteSpeed starts, if the central repository view is selected, LiteSpeed checks a timestamp against the central (which is automatically updated each time a change is made to a registration). If the timestamps do not match, LiteSpeed automatically refreshes the view from the central repository. If the timestamp matched, the local central registration cache is loaded.
- When importing or refreshing, any local login information is persisted while the central repository cache is saved locally. You can then import while preserving login information.

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

TIP: From the **Edit SQL Server Registration** window you can add server, edit connection, display overview tab, test connection, import and export. You can also edit credential information for each instance including display name, server name, authentication, login, password, and parent group. You can also see information for each instance including license, license type, and license expire date.

1. Select **Home | SQL Server Registration | Manage**.
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 that is used for the SQL Server connection. It can be a Windows login in the following format: 'Domain\Username', or SQL Server login.
   - Password–Encrypted password for the login. May be blank.
   - LicenseInfo–License key information (Site Message).
   - LicenseType–Examples: Trial, Permanent, Term.
   - LicenseEdition–Style of license. Example: Enterprise.
   - Path–Server group location of the instance.
   - ExpireDate–Date that the LiteSpeed license expires.

   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;LicenseInfo
   LicenseType;LicenseEdition;Path;ExpireDate
   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>.
Connect using	Select either Windows authentication or SQL Server authentication (enter the user name and password). **NOTE:** Microsoft recommends using Windows authentication when possible because it is more secure than SQL Server authentication.
Additional Connection Parameters	Enter additional connection string parameters. **Tip:** Use this selection as needed to include custom parameters such as encryption, AlwaysOn and others to the SQL server connection string.
Advanced>>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 and select **Edit | SQL Server Registration**.
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 **SQL Servers** at the top of the navigation tree and select **New SQL Server Group**.
Create a subgroup of an existing server group	Right-click the group in the navigation pane tree and select **New SQL Server Group**.

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. For more information, see Change Server Instance Grouping Methods on page 58.

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. For more information, see Select a Central Repository on page 54.

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. For more information, see Select a Central Repository on page 54.

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 (Location or Role), and subcategories (Europe, North America, and Asia, and Dev, Test, and Prod).

NOTE: You can use categories, if the central repository is configured and selected for use. For more information, see Select a Central Repository on page 54.

To create categories

1. Select **Category | Edit Categories**.
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. For more information, see Select a Central Repository on page 54.

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:

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 for disk backups and *7* is the default value for cloud backups. For more information, see Compression Methods on page 121.
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 *128-bit AES*. For more information, see Encryption Methods on page 123.
Compression 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.
Max transfer size	Specifies the largest unit of transfer in bytes to be used between SQL Server and LiteSpeed. The possible values are multiples of 65536 bytes (64 KB) ranging up to 4,194,304 bytes (4 MB). The default is 1048576 (1 MB).
Buffer count	Specifies the number of SQL Server buffers available for a LiteSpeed operation. The default value is set by SQL Server.
CPU 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. For more information, see Adaptive Compression on page 122.
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. For more information, see Processor Affinity on page 64.
Processor priority	Select the priority of the backup over other transactions or processes running on the same server. The default is *Normal*.
Init backup set	LiteSpeed will appends the backup to an existing backup file set or tape.
File name	Location and name of a LiteSpeed backup file. LiteSpeed uses the default SQL Server backup directory. The default file name format is *%D_%DATETIME%.bak*. For more information, see LiteSpeed Variables on page 125. **NOTE:** Fast Compression handles the naming of files automatically. For more information, see Backup Files and Folders on page 116.

Comment	User comment written into the backup header. Is blank by default.
Path to TSM .opt file	See the *Integration with TSM* 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. For more information, see About Using the Command-Line Interface on page 178.

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. For more information, see Adaptive Compression on page 122.

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. Click the Application Button and select **Options** (*Keyboard: ALT+F+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. For more information, see Register Central Repositories on page 53.	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
Show background task complete notification	View background task complete notification messages.	Cleared
Log the LiteSpeed UI Console activity	Select a logging level to define what events to log for the console. You can find the log events in the Application Event Log.	Errors only
Show database status	View if a database has been backed up recently using LiteSpeed The instance tree view displays databases icons of the following colors: • GREEN–If the latest database backup was successfully created within the number of days specified for GREEN. • 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. • RED–If the latest database backup was NOT successfully created within the number of days specified for YELLOW. **NOTE:** Database status can only be displayed for LiteSpeed backups and if the local repository is configured.	Cleared

Backup Manager Options

To set Backup Manager options

1. Click the Application Button and select **Options** (*Keyboard: ALT+F+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
Ask for confirmation when making changes to the Central View	Display a confirmation message before you are able to edit central repository instances.	Selected
Ask to import instances when switching from Local to Central View.	Display a confirmation message about importing registered SQL Server instances when switching from the Local to Central View.	Selected
Automatic refresh Overview every.	Refresh the Overview and LiteSpeed Activity screens a configurable amount in increments from 1 to 30 minutes.	Not selected
Specify proxy settings if needed to set up cloud storage connection.	Specify proxy settings for the LiteSpeed console. Select the **Proxy Settings** link and enter the following. • **Use proxy** - Click to enable proxy use. • **Address** - Proxy server address. • **Port** - Proxy server port (default is 80). • **Username** - Proxy server user credential. • **Password** - Proxy server password credential.	Not selected

Log Shipping Options

To set Log Shipping options

1. Click the Application Button and select **Options** (*Keyboard: ALT+F+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. Click the Application Button and select **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
Show jobs that run continuously during the day on top of the schedule	Display the continuously running jobs at the top of the schedule.	–

Use this option…	To…	Default
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 **Show History**. (Selected.) If cleared, history displays automatically when you select a job.	Cleared
Advanced	Click **Advanced** 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
Show disabled jobs	Display the jobs that have been disabled.	Selected
Refresh job status after start/stop	Refresh the job status automatically after the job is either started or stopped.	Selected
Auto refresh windows tasks every:	Specify the auto refresh rate in seconds for windows tasks.	–

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 the Application Button and select **Options** (Keyboard: ALT+F+O).
2. Select the Log Reader tab.

3. Review the following for additional information:

Use this option…	To…	Default
Check log file changes every	Check the log file for changes every so many seconds.	Selected 10 seconds
Show uncommitted transactions	List all the transactions that are not committed.	Selected
Reconstruct DDL commands	Select this option to display both DML and DDL commands	Selected
Export all details of the log	Export all details of the log file. TIP: Select this option to get a full report.	Not Selected
Online log reading timeout	Set the amount of time the log is read before a timeout is reached.	300 seconds
Maximum BLOB size	Set the maximum Binary Large Object (Blob) Data size.	8000 bytes
Reconstruction	Select one or both of the following: • 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. • Enclose Undo/Redo scripts in a transaction - Use this option to turn on or off transactions for undo/redo scripts. NOTE: You need to turn off transactions to undo or redo in-memory objects.	Selected
Search Depth	Select whether to read the entire log file or only a set number of records in the log file. The options are: • Read entire log file - Select this option to read the entire log file. • Read approximately - Select this option to set the number of records to read. NOTE: The minimum number of records to read is 100,000. The Log Reader stores temp data in **%temp%/Quest Software** where %temp% is the first path found of: the path specified by the TMP environment variable, the path specified by the TEMP environment variable, the path specified by the USERPROFILE environment variable, the Windows temp directory.	100000

Configure Replication and Retention Options for Repositories

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.

NOTE: The xp_replicate_activity_statistics command is designed for sync changes between the local and central repositories. LiteSpeed does not contain information about the native backup history because it is a SQL Server native tool. You must use the [LiteSpeedLocal]..[LiteSpeed_ImportNativeHistory] procedure to import the native backup history first into the local repository.

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 you 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 you 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`.

Purge Repository Data

You can remove historical data using any of the following:

- Clean Up History task. For more information, see About Creating Maintenance Plans on page 129.
- SLSSQLMaint utility. For more information, see Script Maintenance Plans Tasks on page 230.
- LiteSpeed_DeleteActivity stored procedure. For more information, see LiteSpeed_DeleteActivity on page 72.
- Repository cleanup.

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.

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–You can use the LiteSpeed Instance Configuration wizard (**Start | All Programs | Quest Software | LiteSpeed for SQL Server | Instance Configuration**) to choose where to store the repository data.

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

This stored procedure accepts the following arguments:

Argument	Description
@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:

Argument	Description
	• 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

Argument	Description
@delStatus	Deletes old data from dbo.DbMaintStatus, dbo.DTSStatus, 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.DTSStatus, dbo.JobStatus

```
USE LiteSpeedLocal
EXEC LiteSpeed_DeleteActivity
@delLocal=1,
@delLogshipping = 1,
@delStatus = 1 ,
@delUnit = 6,
@delUnitType = 'MONTHS'
```

4

Cloud

About the Cloud

LiteSpeed supports backup and restore directly to and from the following.

Note: Cloud backup and restore is only available with the Enterprise license.

Microsoft Azure Blob

Microsoft Azure Blob storage is a cloud object storage that can be used to store any data, including SQL Server database backups. Information about the Microsoft Azure Blob is available at https://azure.microsoft.com/en-us/services/storage/blobs/

Amazon S3 (Simple Storage Service) cloud storage

Amazon S3 provides a fully redundant data storage infrastructure for storing and retrieving any amount of data, at any time, from anywhere on the Web. Information about the Amazon S3 is available at: http://docs.aws.amazon.com/AmazonS3/latest/dev/Introduction.html.

Google Cloud Storage

Google Cloud Storage is unified object storage for developers and enterprises, from live data serving to data analytics/ML to data archiving. LiteSpeed supports Multi-Regional, Regional, Nearline and Coldline storage types. More information on Google Cloud Storage is available at https://cloud.google.com/storage.

Using the Cloud

LiteSpeed is able to:

- backup to the cloud.
- restore databases and specific objects from the cloud.
- read transaction logs from the cloud.

Tip: Set the default compression level to 7 for cloud backups. Using a higher compression level has real savings. Reducing the number of bytes sent to the cloud makes for faster backups and restores and reduces Internet bandwidth.

You can get started using the cloud with LiteSpeed by first setting up your account with the cloud vendor and then registering that account with LiteSpeed.

Note: LiteSpeed contains all the necessary software and components necessary for running LiteSpeed backups and restores to / from the following cloud vendors.

Setting up a cloud account with the cloud vendor

The first step in using the cloud with LiteSpeed is to create your own Cloud account with the vendor.

To create an Amazon S3 account

1. Setup your Amazon account by registering on: http://aws.amazon.com.
2. Click the IAM icon on the Cloud administration console to create a Cloud user.
3. The User Access Key IDs and Secret Access Keys are automatically generated.
4. Write down and save the Access Key IDs and Secret Access Keys. You will use these values later in the LiteSpeed Cloud account registration.
5. Select the Permissions tab in the Cloud admin console and assign user Administrator Access permissions.
6. Use the Cloud admin console to create a Bucket Name and Region. This is the location where your backup files will be stored.
7. Write down and save the Bucket Name and Region. You will use these values later in the LiteSpeed Cloud account registration.
8. Use the following procedure to register your Amazon S3 Cloud account with LiteSpeed.

Note: Amazon changes their Cloud account Web UI on a regular basis. Therefore the actual icons, tabs, and selection buttons might be different than what is indicated in the procedures below.

To create a Microsoft Azure account

1. Log in with your existing Microsoft credentials at: https://azure.microsoft.com/en-us/free/ to setup your initial free Microsoft Azure account.
2. Click **Sign up for a free trial**.
3. Complete the Microsoft Azure information page and click **Sign up**. The **Welcome to Microsoft Azure** web page is displayed.
4. Click **Start managing my service**.

To create a Google Cloud Storage account

LiteSpeed uses Google Storage service accounts access Google Storage. More information about Google storage service accounts can be found here: https://cloud.google.com/compute/docs/access/service-accounts

Registering the cloud account with LiteSpeed

After setting up your vendor cloud account, then register your cloud account with LiteSpeed before attempting LiteSpeed backups and restores.

To register a cloud account with LiteSpeed

1. Click .
2. Click **Cloud Accounts**.
3. Click **Add**.

When the vendor is Amazon S3

1. Select Cloud vendor **Amazon S3** from the drop-down list.
2. Enter the display name for this account.
3. Select the authentication type.

   Select **Access and Secret Keys** to enter the access key and secret key in LiteSpeed.

   a. Enter the Access key saved in Step 4 from the "To create your Amazon S3 account" procedure above.

   b. Enter the Secret key saved in Step 4 from the "To create your Amazon S3 account" procedure above.

   Select **IAM Roles** to grant permission to the Amazon S3 resource without specifying the Access and Secret key to LiteSpeed. The IAM Role supplies temporary permissions that LiteSpeed can use to make calls to other Amazon Web Services applications. For information on creating and using IAM roles, refer to the Amazon Web Services documentation.
4. Enter the Region name saved in Step 7 from the same procedure above.
5. Enter the Storage Class from the drop-down list. The options are standard, standard-infrequent access, and reduced redundancy storage.
6. Enter the Bucket name saved in Step 7 from the same procedure above.
7. Click to deselect or select SSL security.
8. Click to select Use Server Side Encryption (AES-256).
9. Select 'Use Amazon S3 Transfer Acceleration Speed' to use Amazon's S3 Transfer Acceleration feature which allows for increased upload speed to S3 storage up to 200% in some cases by using local CloudFront endpoints.
10. Click to select GovCloud (US) Region.
11. Click to deselect or select Automatic Striping. The automatic striping options are: auto, 10,25, 50, 100, 250, 500, 1000, and 1995 GB.

When the vendor is Microsoft Azure

1. Select Cloud vendor **Azure Blob** from the drop-down list.
2. Enter a display name for this account.
3. Enter the Storage Account name.
4. Enter the Access key.
5. Select Storage type Block Blobs or Page Blobs from the drop-down list.
6. Enter the Container name or select an existing name from the drop-down list.
7. Click to deselect or select SSL security.
8. Click to select or deselect Government account.
9. Click to deselect or select Automatic Striping. The automatic striping options for block blobs are: auto, 10, 25, 50, 100, and 190 GB. The automatic striping options for page blobs are: auto, 10, 25, 50, 100, 250, 500, and 995 GB.

When the vendor is Google Cloud Storage

1. Select Cloud vendor **Google Storage** from the drop-down list.
2. Enter a display name for this account.
3. Enter the Service account ID. This is an e-mail styled account.
4. Enter the private key. This is a very long string.
5. Enter the Project ID.
6. Select the Region.
7. Enter the Storage Class from the drop-down list. The options are Multi-Regional, Regional, Nearline and Coldline.
8. The Storage Class and Region are not used by Google. Leave as is.
9. Enter the Bucket. Google defines rules for buckets naming: https://cloud.google.com/storage/docs/naming.
10. SSL Security is always enabled for Google Cloud Storage.
11. Click to deselect or select Automatic Striping.

Test the connection

1. Click **Test Connection** to verify that all entered information is working. When successful, the **Cloud Accounts** screen is displayed with your new account listed.
2. Click **OK** to save your entries. Your cloud account is now registered with LiteSpeed.
3. Use one of the following procedures and run the LiteSpeed Backup or Restore Wizard to save backups to the cloud or restore files from the same.

Running LiteSpeed Backups to the cloud

After first setting up your account with the cloud vendor, and registering your account with LiteSpeed, you now can run backups to the cloud using the LiteSpeed Backup Wizard.

Tip: There are three ways to back up to the cloud: Backup Wizard, Maintenance Plans, or Backup Templates.

Caution:**Backups are run to new files. Appending to an existing file is not supported. Existing backups can only be overwritten. The option "Overwrite existing object" is available in the Backup Wizard and MP (Back Up Database task). The option is turned off by default.**

To run LiteSpeed backups to the cloud

1. Run the LiteSpeedBackup wizard and make your appropriate selections.
2. Select Cloud from the destination list and complete the Backup Destination page.
3. Select the cloud vendor account that you created from the Accounts list.
4. Conclude making the Backup wizard selections and finish by executing the backup.

   Tip: You can review the LiteSpeed cloud backup scripts at the last wizard step before executing the Cloud backup. You can also view an example cloud backup database script here (scroll to the bottom of the page).

Running LiteSpeed Restores from the cloud

After first setting up your account with the cloud vendor, registering your account with LiteSpeed, and running backups to the cloud, you now can run restores from the cloud using the LiteSpeed Restore Wizard.

To run LiteSpeed restores from the cloud

1. Run the LiteSpeedRestore Wizard and make your appropriate selections.
2. Select Cloud from the Backup Source page and select the Database from which to restore.
3. Conclude making the Restore wizard selections and finish by executing the restore.

   Tip: You can review the LiteSpeed Cloud restore script at the last wizard step before executing the Cloud restore. You can also view an example Cloud restore database script here (scroll to the bottom of the page).

Cloud Account Settings

You can setup the Microsoft Azure Blob Cloud and Amazon S3 Cloud account and proxy settings in the Back Up Databases wizard, Restore Databases Using the Restore Wizard, and Create Backup Templates wizard.

NOTE: LiteSpeed supports the following cloud storage: Amazon S3, Microsoft Azure Blob, Google Cloud Storage.

Cloud account	Select the Cloud account from the drop-down list.
Cloud Accounts...	Select the following items to add or edit the registered cloud account settings.

- Add - (For Amazon S3 only), click to add cloud vendor, display name, authentication, region, storage class (standard, infrequent access, reduced redundancy storage), and bucket. Select: use SSL, use server side encryption (AES-256), GovCloud (US) Region, and automatic striping (auto, 10, 25, 50, 100, 500, 1000, 1995) GB. Select 'Use Amazon S3 Transfer Acceleration Speed' to use Amazon's S3 Transfer Acceleration feature which allows for increased upload speed to S3 storage up to 200% in some cases by using local CloudFront endpoints.
- Add - (For Azure Blob only), click to add cloud vendor, display name, storage account name, access key, storage type (block blobs or page blobs), container, use SSL, government account, and automatic striping . Options for block blobs are: auto, 10, 25, 50, 100, 250, 500, 1000, 2000, 3000, 4000 and 4300 GB. Options for page blobs are: auto, 10, 25, 50, 100, 250, 500, and 995 GB.
- Add - (For Google Storage only), click to add cloud vendor, display name, service account ID, private key, project ID, storage class, region and bucket. Use SSL is always selected as Google always uses it.
- Edit - (For Amazon S3 only), click to edit display name, authentication, region, storage class (standard, infrequent access, reduced redundancy storage), and bucket. Bucket name must conform to DNS naming requirements and must not contain periods ("."). Select: use SSL, use server side encryption (AES-256), GovCloud (US) Region, and automatic striping. Options for automatic striping are: auto, 10, 25, 50, 100, 250, 500, 1000, and 1995 GB.

  Select 'Use Amazon S3 Transfer Acceleration' to use Amazon's S3 Transfer Acceleration feature which allows for increased upload speed to S3 storage up to 200% in some cases by using local CloudFront endpoints. Additional data transfer charges may apply. See Amazon S3 pricing for more details.
- Edit - (For Azure Blob only), click to edit display name, access key, storage type (block blobs or page blobs), container, use SSL, government account, and automatic striping. Options for block blobs are: auto, 10, 25, 50, 100, 250, 500, 1000, 2000, 3000, 4000 and 4300 GB. Options for page blobs are: auto, 10, 25, 50, 100, 250, 500, and 995 GB.
- Edit - (For Google Storage only), click to edit display name, service account ID, private key, project ID, storage class, region and bucket. Use SSL is always selected as Google always uses it.
- Delete - Click to delete the Cloud account from the console.
- Import - Click to import a saved Cloud account in XML format.

	• Export - Click to export and save a Cloud account in XML format.
Proxy Settings	Select the following items to edit the Cloud account proxy settings. • Use LiteSpeed Server proxy settings - Click to use the server proxy setting. This is the default selection. You can also edit the proxy settings from this item. • Use LiteSpeed Console proxy settings - Click to use the console proxy setting. You can also edit the proxy settings from this item. • Specify custom proxy settings - Click to add your own custom proxy address, port, username and password.

Cloud Automatic Striping

Striping is splitting one backup file into multiple files. The advantage of striping is that it provides higher upload performance when backing up to the Cloud and overcomes some Cloud limitations (e.g. maximum object size on Azure Block Blob storage is 4.75TB). Striping provides a significant performance benefit and time saving when backing up large database files. For example, striping three blocks of one database file up provides three times the upload bandwidth over backing up one large database file to a single cloud container.

Setting up Cloud Automatic Striping

LiteSpeed provides an automatic striping option for backing up databases. The option can be set to auto which enables LiteSpeed to manage the striped size. Alternately, you can set the striping size in GB. If the database size is larger than the specified stripe size then the file is striped when backed up.

Tip: Quest Software recommends using LiteSpeed defaults.

To enable cloud automatic striping

1. Click .
2. Click **Cloud Accounts**.
3. Click **Edit**.

4. Click to select Automatic Striping. The automatic striping options:

   - Microsoft Azure blob storage (Block blob): auto, 10, 25, 50, 100, 250, 500, 1000, 2000, 3000, 4000 and 4300 GB.
     Note: Maximum size of block blob supported by Microsoft Azure is 4.77 TB.
   - Microsoft Azure blob storage (Page blob): auto, 10, 25, 50, 100, 250, 500 and 995 GB.
     Note: Maximum size of page blob supported by Microsoft Azure is 1000GB.
   - Amazon S3: auto, 10, 25, 50, 100, 250, 500, 1000 and 1995 GB.
     Note: Maximum size of object supported by Amazon S3 is 5000GB. LiteSpeed is limited by 2000GB.
   - Google Cloud Storage: auto, 10, 25, 50, 100, 250, 500, 1000 and 1995 GB.
     Note: Maximum size of object supported by Google Cloud Storage is 5 TB. LiteSpeed is limited by 2TB.

5. Click **Test Connection** to verify that all entered information is working. When successful, the **Cloud Accounts** screen is displayed with your new account listed.
6. Click **OK**.

5

Back Up Databases

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. Click **Next**.
4. On the **Backup Location** screen, select **Backup to** (Disk, Cloud, or TSM Backup) and **Backup locations**. Add or remove backups from the locations. Additionally select striping options.
5. Click **Next**.
6. On the **Data Sample** screen, enter the amount of data to sample in megabytes or the percentage of data to test.
7. Click **Next**.
8. On the **Compression Level** screen, select the compression levels to test. Any combination of compression levels from 0 to 8 are available. You can also specify additional backup advanced options.
9. Click **Next**.
10. On the **Execute Script** screen, the list of databases to test is displayed.
11. Click **Next**. The **Backup Analyzer Wizard** test scrip runs and displays the test results summary.
12. 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.

Scheduling Backup Analyzer

You can schedule analyzing from within the Backup Analyzer Wizard. An SQL Server job with the name "LiteSpeed Analyzer SERVERNAME.DATABASENAME" is created. View the list of Backup Analyzer jobs directly from the Backup Analyzer tab by clicking "Show Analyzer Jobs." Right-click on the job. You can change schedule, start/stop, disable or even delete the job. To view the script, select the "View in Job Manager" option and open the job properties from the Job Manager module. You can then go to the job step details.

Backup Analyzer Tab

The Backup Analyzer tab consists of the toolbar, graph, and grid. The Backup Analyzer toolbar is used to select the following items:

- **Existing test** - View the existing test or use the drop down to select other tests.
- **Purge All** - Click to remove all Backup Analyzer tests for the database.
- **Show Analyzer Jobs** - Click to view all **Backup Analyzer** jobs.
- **New Test** - Click to run the **Backup Analyzer Wizard** and begin a new test.
- - Click to export the grid to Microsoft Excel.
- - Click to print the grid.

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

- Graph–Displays the backup duration and compression amount for each test in a bar graph so you can easily compare the results between size and duration. If you hover the mouse over the bar graph, test number, size, and duration are displayed. If you select a test in the grid, LiteSpeed indicates the corresponding test in the bar graph with a yellow border around it. You can view previous tests by changing the **Existing test** field.

- Grid—Displays the details of each test in columnar format. LiteSpeed indicates its recommendation with the number indicated first in the **Test Number** column and a yellow box around the test bar graph. Review the following for additional information:

  Tip: .Use the Grid horizontal scroll bar to view other table columns.

Column	Description
Test Number	The number assigned by the Backup Analyzer.
Level	The compression level that is used for a particular database backup test.
Encryption	Indicates enabled encryption for the database backup.
Estimated Duration	The approximate time allotted to run the database backup.
Estimated Backup Size (MB)	The approximate size of the backed up database.
Database Size (MB)	The amount of disk space the database occupies.
Compression (MB)	The amount of disk space you can save with this compression ratio.
Compression (%)	Data compression ratio.
Read Speed (MB/Sec)	Maximum read speed.
Throughput (MB/Sec)	Actual speed of processing data (compression and encryption).
Backup Speed (MB/Sec)	Virtual write speed of the backup. Compare this value to Throughput and Read Speed to determine if you are experiencing a write-bound issue.
Write Speed (MB/Sec)	Actual write speed. It shows how quickly the compressed data stream is written to the destination disk.
Stripes	Number of stripes associated with the backup.
Destination	Backup destination directory location.
Threads	Number of threads associated with the backup.

You can double-click a row for more information about the test.

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

- To sort and group the records, right-click a column header and select the appropriate options.
- To sort records against multiple columns, click column headers while holding the SHIFT key. For example, to sort by type and then by name, click the Type column header and then SHIFT+click the Name column header.
- To add or remove columns, right-click a column header and select **Column Chooser**. Add a column by dragging it from the list into the column headers. Remove a column by dragging its column header into the list.
- To move a column, drag the column header to the new location.

Create and Deploy Backup Templates

Create 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 supports disk, TSM Backup, TSM Archive and cloud storage destinations.

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.

To create a Backup Template

1. Select the **Backup Manager** pane (CTRL+1).
2. In the Backup Templates tab, click **New** on the toolbar. If you do not use the central repository, click

Backup Templates on the toolbar.

3. Review the following additional information about the backup type and destination:

Backup Type	Select what backups will occur after you deploy this template on a server instance. • Full–Full backups and optionally differential backups and transaction log backups. • Fast Compression–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. For more information, see Fast Compression on page 115. The default backup name and description use the following information: • %D–Database name • %T–Backup type (Full, Diff or Log) • %z–Timestamp • %AG% – AlwaysOn Availability Group Name. This allows you to group databases into folders based on the AlwaysOn Availability Group name. It is ignored for databases that are not in an AlwaysOn Availability Group. You can specify custom backup name and description using both the LiteSpeed variables and text. For more information, see LiteSpeed Variables on page 125.
Select Databases	Select databases you want to back up. NOTES: • You can select the **Custom Database Selection at Deployment** 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. For more information, see Deploy Backup Templates on page 96. • If you select **Databases matching regular expression** then you are limited to deploying as a Maintenance Plan. Selecting **Databases matching wildcard expression** is supported by both Maintenance Plans and Jobs. For more information, see Use Wildcard and Regular Expressions in LiteSpeed on page 111.

	• You can exclude databases. Select to exclude **Offline**, **LogShipping** and **ReadOnly** databases from the backup. Optionally enter a wildcard mask or regular expression; databases matching the wildcard mask or regular expression will be excluded. For more information, see Use Wildcard and Regular Expressions in LiteSpeed on page 111. **TIP**: In addition, databases can be excluded by name using the Deployment Wizard. • You can select whether to back up databases participating in AlwaysOn Availability Groups. For more information, see Back Up SQL Server AlwaysOn Availability Groups on page 111.
Backup Destination	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. 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. **NOTE:** Fast Compression handles the naming of files automatically. For more information, see Backup Files and Folders on page 116. You can either supply the backup destination at deployment time or save the default backup destination in the template. The backup destinations include disk, cloud, TSM Backup and TSM Archive. • Cloud - You can additionally select Cloud Accounts and Proxy Settings. For more information, see Cloud Account Settings on page 79. • TSM - By default LiteSpeed sets the option to "Use PASSWORDACCESS GENERATE from TSM configuration file".. You can additionally set TSM Backup and Archive options: client node, client owner password, configuration file, management class and the amount of time to wait for a device to become available. **NOTE**: The %TSMDEFAULTPATH% variable means that LiteSpeed will automatically detect the default TSM configuration file path. Use the **Overwrite** setting to overwrite existing media when a full backup template is created. Click **Add Mirror** to copy the entire backup file to multiple locations. Backup Templates support backup mirroring to disk or the cloud, same as for the backup of individual databases. For more information, see About Backing Up/Restoring Databases on page 36.

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

Optimize the Object Level Recovery speed	Select to create an index of objects in the backup file. This option is only available for LiteSpeed backups. The default is enabled. **NOTE:** 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. During the backup process the .lsm file is created in the temp directory and attached to the backup file after the backup is completed. 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.
Create Double Click Restore executable	Select to create a Double Click Restore Loader that allows you to restore a backup on a server instance that does not have LiteSpeed installed. If you additionally select to **Create one Double-Click Restore executable file** then note the following warning. The executable may be greater than 4GB for large databases. Windows Server is unable to run executable files larger than 4GB. However, the file will be convertible/restorable by LiteSpeed file. For more information, see Double Click Restore Executables on page 120. **NOTE:** A Double Click Restore can only be created for a disk file.
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 **Continue on error** option, the backup is executed despite encountering an invalid backup checksum.
Continue on error	Select this option to continue running the backup even if an invalid checksum is encountered.

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

Fast Compression Type	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): • **Separate backup files**–(Default) 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).

	• **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. The Self-Contained Backup Sets option automatically verifies the Full backup exists. The Separate Backup Files option performs the same validation by default. Note: For cloud backups only "Separate backup files" type is supported.
Fast Compression Backup Options	You can set the following thresholds to define when to issue a full backup: • Force a full backup every - The amount of time elapsed since the last full backup. The default is *14* days. • Data change threshold - The amount of database changes since the last full backup. The default is *35*%. 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 default option is to query actual data pages. It 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 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. Select the **Extension for backup files** checkbox to enter or change the backup file name extension. The default is set to bkp. NOTE: You can select the backup file extension for Fast Compression and make the new default, bak, for new items. For an existing item that

	does not have an extension defined, bkp is displayed when the item is edited (maintenance plans and templates).
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.
Verification	**TIP:** "Cleanup" means SmartCleanup. For more information, refer to SmartCleanup. 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. The **Verification** options include: • Do not verify backup (default). • Verify last backup. • Verify both the last full and latest differential backup. • Verify the last full and all associated differential backups. The **Cleanup** options include: • Clean up full/differential backups older than 28 (default) days. • Clean up transaction logs older than 7 (default) days. Note: Clean up transaction log options are available if you set up a transaction log within the wizard (or template). • Do not delete if archive bit is set.

5. 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	LiteSpeed automatically selects the optimal compression based on throughput and CPU usage and optimize backups either for size or for speed (default). **NOTE:** Adaptive Compression is only available with LiteSpeed 6.5 or later; Enterprise license.
Compression level	Select 0 for no compression or 1-8 (default 2) to compress the file. For more information, see Compression Methods on page 121. **NOTE:** 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. For more information, see Test Optimal Backup Settings on page 83. Tip: For cloud backups, set the default compression level to 7. Using a higher compression level has real savings. Reducing the number of bytes sent to the cloud makes for faster backups and reduces Internet bandwidth.
Encrypt backup	Select this checkbox to encrypt the backup. Then, select the encryption level and enter the encryption password. For more information, see Encryption Methods on page 123.

Review the following additional information about the advanced options:

Compression 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.
Max transfer size	Enter the maximum backup file size in bytes. . The possible values are multiples of 65536 bytes (64 KB) ranging up to 4,194,304 bytes (4 MB). The default is *1048576*.
Buffer count	Enter the number of SQL Server buffers available for a LiteSpeed operation. The default is set by SQL Server.
CPU throttle	Enter the maximum percentage of CPU that LiteSpeed can use for the process. The default is *100*.
Processor affinity	Click ... to select which processors LiteSpeed can use. The default is *0*, which allows LiteSpeed to use all available system processors.

Processor priority	Select the priority of the backup over other transactions or processes running on the same server. The default is *Normal*.
Comment	User comment written into the backup header. Is blank by default.
Logging level	Select a logging level to define what events to log for the console. You can find the log events in the Application Event Log.
Network resilience	If LiteSpeed fails to write disk backups or reads from disk, it waits and retries the operation. You can enable and disable and control the number of times to retry and the amount of time to wait before retrying. • Number of times to retry any given read/write attempt–The default is *4 retries*. The maximum allowed setting is 1000 retries. • Wait period before each retry attempt (in seconds)–The default period to wait before retry is *15 seconds* The maximum allowed setting is 300 seconds. For more information, see Network Resilience on page 124.

TIPS:

- 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. For more information, see Adaptive Compression on page 122.

NOTE: LiteSpeed defaults typically result in the best performance. You should only modify advanced options after careful planning and testing. For more information, see Configure LiteSpeed Defaults on page 60.

6. Configure notification options. You can select an existing notification profile or specify operators at deployment.

   TIP: Variables defined in the maintenance plan can be used to format the Subject Line. There is also help information inside the task that lists the available Subject Line variables.

   NOTE: The SQL Server Agent must be configured to send email using Database Mail. Review the following for additional information:

   - http://msdn.microsoft.com/en-us/library/ms189635.aspx
   - http://technet.microsoft.com/en-us/sqlserver/dd939167.aspx

7. For a Differential backup ("Full" backup type)

   - On the **Diff Schedule** tab, tick **Schedule differential backup** and fill in the details of the schedule type and frequency. The **Diff Destination** and **Diff Compression** tabs are now added to the Backup Template Wizard.
   - On the **Diff Destination** tab set the filename format and default backup path.
   - On the **Diff Compression** tab use the same compression and encryption settings as for a full backup or customize using the available options.

8. For a Transaction Log backup
    - You can nominate a backup folder independent of the backup folder for Fast Compression / Full backups. Do this at the T-Log Destination step. Further to this, the backup folder can be stipulated for individual instances using the Deployment Wizard.
    - On the T-Log Schedule page you can skip Transaction log backups for databases being configured for Log Shipping. Click the **Exclude LogShipping databases for transaction backups** option. This option is available only when Exclude LogShipping is set to off on the Select Databases page. It allows you to run regular Full/Diff backups for Log Shipping databases and not brake the log shipping transaction log backup sequence.
9. Specify backup cleanup options. For more information, see SmartCleanup on page 124.

    NOTE: The "Do not delete if Archive bit is set" option is available for full and fast compression backups in the Backup Template when it is deployed as a maintenance plan. If a backup template is created for a full backup with the option "Do not delete if Archive bit is set" selected, and the backup template is deployed as a job, the option is passed to the job's script to run the xp_slsSmartCleanup procedure. When the backup template is deployed as a maintenance plan, the option is omitted because the “Remove files older than” option (not the SmartCleanup option) for full backup is run in the maintenance plan.
10. 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.
- Regular expressions are only supported in Maintenance Plans.

To deploy a LiteSpeed Backup Template

1. Select the Backup Manager pane (CTRL+1).
2. 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	Use the **Show selected instances only** option to see server instances and databases where the selected template was deployed. **NOTE:** You can select individual databases only if the backup template is saved with the **Custom Database Selection at Deployment** option.
Backup Options	Enter backup destinations for individual instances or for several instances at a time or use the default backup destination you specified in the template. Use the **TLog Backup Location** column to enter the Transaction Log backup folder for the given instance. Use the **Diff Backup Location** column to enter the Differential backup folder for the given instance. Use the **Exclude Databases** column to exclude individual databases from the backup. Use the **TSM Settings** column to customize TSM settings for the backup. 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: • Default to template–To change the current cell back to the template default. • Default all to template–To change all cells back to the template default. • Stagger by n minutes–To stagger all scheduled times by n minutes, starting at the template time and adding n minutes for an instance. • Spread over n hour(s)–To spread the scheduled times evenly over an n hour range, starting with the template time. **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.
Notifications	If you enabled notification in the template, review the Notification page and complete the fields as necessary. You can use the notification profiles only if they are already configured within a SQL Server instance. **TIP:** Variables defined in the maintenance plan can be used to format the Subject Line. There is also help information inside the task that lists the available Subject Line variables. **NOTE:** The SQL Server Agent must be configured to send email using Database Mail. Review the following for additional information: • http://msdn.microsoft.com/en-us/library/ms189635.aspx • http://technet.microsoft.com/en-us/sqlserver/dd939167.aspx
Deployment Type	**Owner** - Optionally, enter the owner of the SQL job (or Maintenance Plan) that will deploy the template. If the name you enter does not exist as a SQL user then the job will fail. If this field is left blank the job will run with the creator of the template or "running" user as the owner.

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 Databases

The Backup Manager wizard guides you through the process of backing up a single database or multiple databases. For single database backups you can perform file, filegroup, or transaction log backups, and you can also mirror, stripe, compress, encrypt, and add attachments to the backup file. For multiple database backups you can create both the native SQL Server and LiteSpeed backups, mirror, compress and encrypt backups, run all backups immediately or schedule via a SQL Server job.

Tip: Database backups created with LiteSpeed Version 8.x cannot be restored using older versions of LiteSpeed.

NOTES:

- Running individual database backups or multi-database backups both use the same Backup wizard. When backing up multiple databases at the same time For more information, see Multi-Database Backup on page 115.

  Alternately, you can create a maintenance plan. For more information, see About Creating Maintenance Plans on page 129.
- If you select **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. For more information, see Create Backup Templates on page 87.
- 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 product documentation (SQL Server 2012 and above) for information about supported backup types.
- For information about full file backups, see http://msdn.microsoft.com/en-us/library/ms189860.aspx.

To back up multiple databases

1. In the Navigation pane, select a database to backup and click the **Multi-Database Backup** button. Alternately you can right-click a database and select **Multi-Database Backup** from the menu.
2. Select the databases for backup. For more information, see LiteSpeed's Logic for Backing Up Multiple Databases on page 110.
3. Complete the wizard. For more information, see To back up databases using the Backup wizard on page 99. 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.

To back up databases using the Backup wizard

1. Select the **Backup Manager** pane (CTRL+1).
2. In the Navigation pane, select a database to backup and click the **Backup** button. Alternately you can right-click a database and select **Backup** from the menu.
3. On the **Select Backup Type** page, review the following:

Recovery model	Indicates the recovery model for selected database. Possible values: Full, Simple, Bulk-Logged, Mixed (in Multi-Databases Wizard if databases with different recovery models are selected for backup).
Backup type	Select one of the following backup types: • 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. For more information, see Fast Compression on page 115. • Regular backup (Full, Differential or Transaction Log). **Scenario:** Select the Fast Compression backup type.
Backup component	Select one of the following backup components: **NOTE:** This field is only available for databases with a full or bulk-logged recovery model and for a regular (not fast compression) backup. • Database - Select to backup the database. • Files and filegroups - Select to backup specific files and filegroups. When selected, the **Specify Filegroups and Files** window is displayed. In the window you can select the entire database, or any combination of filegroups or individual files to backup.
Use LiteSpeed	Select to create a LiteSpeed backup. If you clear this checkbox, LiteSpeed creates a native backup script. **NOTE:** The LiteSpeed version number is displayed.

TIP: You can specify custom backup names and descriptions using variables. For more information, see LiteSpeed Variables on page 125.

4. On the **Backup Destination** page, review the following:

   Caution: **The Cloud only runs backup to new files. Appending to an existing file is not supported by the Cloud. Existing backups can only be overwritten. The option "Overwrite existing object" is available in the Backup Wizard and MP (Back Up Database task). The option is turned off by default.**

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

Backup to	Select one of the following options: • Disk • Cloud • TSM Backup • TSM Archive • Tape
Add	Click to add multiple backup destinations. The **Select Backup Destination** window is displayed. From here you can enter file names or backup devices.
Remove	Click to remove backup destinations.
Contents	Click to view the contents of the backup destination including general properties of file content and backup sets properties.
Overwrite	Select one of the following options: **TIP:** This option is only available for regular backups (and not fast compression backups). • Append to media–This option appends the backup set to existing media set; previous backups remain. • Overwrite existing media–This option replaces previous backups in the existing media set with the current backup. • Create unique media–This option generates a new media set; all previous backups are erased.
Add Mirror	Click to copy the entire backup file to multiple locations. **TIP:** This option is only available with the Overwrite existing media option above. You can create multiple mirrors by clicking this button more than once. You can add mirrors to cloud platforms and different cloud platforms at the same time. a. After clicking the button once, the **Mirror 1** tab is displayed. b. Click the **Add** button to display the **Select mirror destination** window.

	c. Enter the mirror destination. Choices are Disk and Cloud. • If you selected Disk, click the ellipsis button to display backup locations. You can browse the network, add files, delete files, and rename files. • If you selected Cloud then fill in the Cloud Account Settings. Review the Cloud Account Settings information.

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 **Cloud**, review the Cloud Account Settings information.

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.) **NOTE:** This file contains session options such as the TSM server's TCP address. If you select the **Use PASSWORDACCESS GENERATE from TSM configuration file** 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)	Click **TSM Filespaces** 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. **NOTE:** Fast Compression handles the naming of files automatically. For more information, see Backup Files and Folders on page 116.
TSM Object (Regular backups)	Click **Select TSM Object**. Enter the filespace and the high-level and low-level names and click **Query TSM** 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. **NOTE:** 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.

	For a new object, you can manually enter the full three-part name.
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	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): • **Separate backup files**–(Default) 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). • **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. The Self-Contained Backup Sets option automatically verifies the Full backup exists. The Separate Backup Files option performs the same validation by default. Note: For cloud backups only "Separate backup files" type is supported.

6. On the **Options** page, review the following:

Options	You can set the following thresholds to define when to issue a full backup: • Force a full backup every - The amount of time elapsed since the last full backup. The default is *14* days. • Data change threshold - The amount of database changes since the last full backup. The default is *35*%. 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 default option is to query actual data pages. It 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 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. Select the **Extension for backup files** checkbox to enter or change the backup file name extension. The default is set to bkp. **NOTE:** You can select the backup file extension for Fast Compression and make the new default, bak, for new items. For an existing item that does not have an extension defined, bkp is displayed when the item is edited (maintenance plans and templates). Select the **Enable backup escalation** checkbox to maintain high recoverability. This option is enabled by default. 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. Select the **Optimize Object Level Recovery speed** checkbox to enable speed. The default is enabled. Select the **Perform checksum before writing to media** checkbox to verify backup data checksum before writing the backup data to the media. Select the **Continue on error** checkbox to continue running the backup even if an invalid checksum is encountered.
Optimize the Object Level Recovery speed	Select to create an index of objects in the backup file. This option is only available for LiteSpeed backups. The default is enabled. **NOTE:** 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. During the backup process the .lsm file is created in the temp directory and attached to the backup file after the backup is completed. 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.
Create Double Click Restore executable	Select to create a Double Click Restore Loader that allows you to restore a backup on a server instance that does not have LiteSpeed installed. If you select to **Create one Double-Click Restore executable file** then note the following warning. The executable may be greater than 4GB for large databases. Windows Server is unable to run executable files larger than 4GB. However, the file will be convertible/restorable by LiteSpeed file. For more information, see Double Click Restore Executables on page 120. **NOTE:** A Double Click Restore can only be created for a disk file.
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 **Continue on error** option, the backup is executed despite encountering an invalid backup checksum.
Continue on error	Select this option to continue running the backup even if an invalid checksum is encountered.

Select directories to mirror the backup to (Overwrite must be selected)	Click **Disk** to backup to disk. Click **Cloud** to backup to cloud.
Select files/folders to attach to the backup set	Select **Add** to attach files or folders to the backup set.

NOTE: Some options are only available with certain backup types.

7. On the **Compression** page, review the following:

Adaptive Compression	LiteSpeed automatically selects the optimal compression based on throughput and CPU usage and optimize backups either for size or for speed (default). **NOTE:** Adaptive Compression is only available with LiteSpeed 6.5 or later; Enterprise license.
Compression level	Select 0 for no compression or 1-8 (default 2) to compress the file. For more information, see Compression Methods on page 121. **NOTE:** 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. For more information, see Test Optimal Backup Settings on page 83. Tip: For cloud backups, set the default compression level to 7. Using a higher compression level has real savings. Reducing the number of bytes sent to the cloud makes for faster backups and reduces Internet bandwidth.
Encrypt backup	Select this checkbox to encrypt the backup. Then, select the encryption level and enter the encryption password. For more information, see Encryption Methods on page 123.

Review the following additional information about the advanced options:

Compression 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.
Max transfer size	Enter the maximum backup file size in bytes. . The possible values are multiples of 65536 bytes (64 KB) ranging up to 4,194,304 bytes (4 MB). The default is *1048576*.
Buffer count	Enter the number of SQL Server buffers available for a LiteSpeed operation. The default is set by SQL Server.
CPU throttle	Enter the maximum percentage of CPU that LiteSpeed can use for the process. The default is *100*.
Processor affinity	Click ... to select which processors LiteSpeed can use. The default is *0*, which allows LiteSpeed to use all available system processors.
Processor priority	Select the priority of the backup over other transactions or processes running on the same server. The default is *Normal*.

Comment	User comment written into the backup header. Is blank by default.
Logging level	Select a logging level to define what events to log for the console. You can find the log events in the Application Event Log.
Network resilience	If LiteSpeed fails to write disk backups or reads from disk, it waits and retries the operation. You can enable and disable and control the number of times to retry and the amount of time to wait before retrying. • Number of times to retry any given read/write attempt–The default is *4 retries*. The maximum allowed setting is 1000 retries. • Wait period before each retry attempt (in seconds)–The default period to wait before retry is *15 seconds* The maximum allowed setting is 300 seconds. For more information, see Network Resilience on page 124.

TIPS:

- 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. For more information, see Adaptive Compression on page 122.

NOTE: LiteSpeed defaults typically result in the best performance. You should only modify advanced options after careful planning and testing. For more information, see Configure LiteSpeed Defaults on page 60.

8. On the **Attached Files** page, select **Attach files and directories** 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. For more information, see xp_restore_attachedfilesonly on page 401.

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

9. On the **Backup Schedule** page, select **Run immediately**, **Run in background**, or **Schedule**. Selecting **Schedule** launches a page where you can fill out the schedule name, schedule type, occurs, daily frequency, duration, and description.

   If you selected Fast Compression the backup schedule has options to configure backups to run as a SQL Agent job at a scheduled time. The options are: daily at, custom, and do not run full backup on specified days of the week. If you select custom, you can add schedule name, schedule type, occurs, weekly, daily frequency, duration, and description.

10. On the **Notification** page, if you selected the **Fast Compression** backup type, you can configure the Fast Compression job to send notifications when any type of failure occurs. The failure might be due to a failed backup, a validation issue, like a missing Full or Differential backup, an LSN verification issue, or a failed Verify operation. Select one of the following:

    - Do not use notification - All failure notifications (including operator selections) are disabled.
    - Notify every time - Notify for all successes and failures. An operator can be selected and configured to receive notifications. Click the ellipsis button on the far right to create new or edit existing operators.
    - Failure only - Notify for failure only. An operator can be selected and configured to receive notifications. Click the ellipsis button on the far right to create new or edit existing operators.

    TIP: Variables defined in the maintenance plan can be used to format the Subject Line. There is also help information inside the task that lists the available Subject Line variables.

    NOTE: The SQL Server Agent must be configured to send email using Database Mail. Review the following for additional information:

    - http://msdn.microsoft.com/en-us/library/ms189635.aspx
    - http://technet.microsoft.com/en-us/sqlserver/dd939167.aspx

11. On the **T-Log Schedule** page, specify scheduling for the transaction log backup job execution. Select **Schedule transaction log** and fill out the new job schedule screen. Entries include schedule name, schedule type, occurs, daily frequency, duration and description.

12. On the **Verification/Cleanup** page, specify the verification and cleanup options after each backup.

    TIP: "Cleanup" means SmartCleanup. For more information, refer to SmartCleanup.

    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.

    The **Verification** options include:

    - Do not verify backup (default).
    - Verify last backup.
    - Verify both the last full and latest differential backup.
    - Verify the last full and all associated differential backups.

    The **Cleanup** options include:

    - Clean up full/differential backups older than 28 (default) days.
    - Clean up transaction logs older than 7 (default) days.

Note: Clean up transaction log options are available if you set up a transaction log within the wizard (or template).

- Do not delete if archive bit is set.

Scenario: Specify when full/differential backups are eligible for cleanup, according to your company's retention policy. For more information, see SmartCleanup on page 124.

13. On the **Fast Compression Job** page, create the fast compression backup job script.
14. Complete the wizard.

    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.

About partial backups and restores

LiteSpeed supports partial backups and restores for filegroups.

Partial backups

LiteSpeed can backup partial databases. Partial database backups contain only data from some filegroups in a database. Partial backup data includes that from primary filegroups, read and write filegroups, and other specified read-only files. Partial backups are different from full backups in that full backups contains all the data in a specific database or set of filegroups or files, and enough available transaction log information for recovering that data. Partial backups are run using the CLI with backup commands. Refer to for partial backup example syntax.

Restore partial backups

LiteSpeed can restore partially backed up databases. Piecemeal database restore sequences can restore databases in stages at the filegroup level. Restores begin with primary files, all read and write files, and then secondary filegroups. Piecemeal restores are run using the CLI with restore commands. Refer to xp_restore_database for restore partial backups example syntax.

Restore partial backups with fast compression

LiteSpeed can restore partially backed up databases that have been run with fast compression enabled. This restore is similar to restore partial backups except that it performs a restore where the partial backup was run with the Fast Compression technology enabled. Piecemeal restores with fast compression are run using the CLI with fast compression restore commands. Refer to xp_slsFastCompression for restore partial backups with fast compression example syntax.

Back Up Multiple Databases

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 wildcard or regular expressions. For more information, see Use Wildcard and Regular Expressions in LiteSpeed on page 111.

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. For more information, see Exclude Databases on page 110.

Depending on the LiteSpeed wizard, 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 included existing and new databases, except those you select manually and those excluded by a different Exclude option.
- Deleted
- Log Shipping
- Read Only

NOTE: Mirror databases and database snapshots are automatically excluded.

Back Up SQL Server AlwaysOn Availability Groups

Versions 7.0 and later of LiteSpeed provide enhanced support for databases participating in AlwaysOn availability groups (SQL Server 2012 and above). 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 product documentation (SQL Server 2012 and above) for information about supported backup types.
- In Maintenance Plans if you select the **Differential** backup type, all secondary replicas in an AlwaysOn Availability Group are automatically excluded. In the Multi-Database Backup Wizard they are not excluded and differential backup for secondary replicas will fail.
- In Maintenance Plans and in the Multi-Database Backup Wizard, if you select the Transaction log backup type and turn on the Copy Only option, backups for all secondary replicas will fail because transaction log backups with Copy Only are not supported on secondaries.

Use Wildcard and Regular Expressions in LiteSpeed

You can configure LiteSpeed to only back up databases that match the specified pattern. Using wildcard and regular expressions is particularly helpful when there are databases with similar names on the same instance or on multiple instances.

Tips:

- To back up databases with similar names on multiple instances, create and deploy a backup template, or create a maintenance plan on one instance and then simply copy it to other instances.
- LiteSpeed lets you enter multiple patterns. When selecting databases for your maintenance plan, you can click **Validate selected** or **Validate all** to display a list of databases names that match any of the specified patterns.

Wildcard Expressions

You can use wildcard characters '?' and '*' to select databases you want to back up. For example, "litespeed*" will include all databases that start with "litespeed" or "LiteSpeed".

Regular Expressions

Using regular expressions can provide you with more flexibility when selecting databases. For example, the following expressions will include a database if its name:

1. Begins with "LiteSpeed".
   ^LiteSpeed
2. Ends with "_PUBLISHER".
   _PUBLISHER$
3. Begins with "LiteSpeed" followed by digits.
   ^LiteSpeed\d+
4. Contains any number within the range 828500 to 828549.
   8285[0-4][0-9]
5. Begins with "LiteSpeed" or "Quest", except "QuestSoftwareCMSS".
   ^(?!QuestSoftwareCMSS)(Quest|LiteSpeed)

NOTES:

- Regular expressions are only supported in Maintenance Plans.
- LiteSpeed does not support regular expressions with the IgnoreCase option. To ignore case, use `([Aa][Bb][Cc])` instead of `(?i:abc)`.

Review the following for additional information:

Construct	Description
^	The character following this construct is at the beginning of the document or new line. **Text:** no cat can catch a crow **Expression:** ^\w+ **Matches:** no
$	The character followed by this construct is at the end of the string. **Text:** no cat can catch a crow **Expression:** \w+$ **Matches:** crow
(normal characters)	Characters match themselves, except for the following: ^ . * + ? { [(|) \ $
\	The character immediately following it, when that character is not an escaped character.

Construct	Description
	Expression: * **Matches:** *
\w	Any word character (A - Z and a - z), digits, and underscores. **Text:** no cat can catch a crow **Expression:** ca\w+ **Matches:** cat, can, catch
\W	Any non-word character. **Text:** 9pm @ RockCafe **Expression:** \W **Matches:** @
\d	Any digit. **Text:** APR-2012 **Expression:** \d\d\d\d **Matches:** 2012
\D	Any non-digit. **Text:** APR-2012 **Expression:** \D\D\D **Matches:** APR
\s	Any whitespace character. **Text:** no cat can catch a crow **Expression:** c\w\w\s **Matches:** "cat " and "can "(both matches include the space)
\S	Any non-whitespace character. **Text:** no cat can catch a crow **Expression:** c\S+ **Matches:** cat, can, catch, crow
[ ]	Any character within the brackets. **Expression:** p[ai]ck **Matches:** pick, pack
[-]	Any character within the specified range. [0-9a-fA-F] matches a single hexadecimal digit, without case sensitivity. **Expression:** b[a-z]ttle **Matches:** battle, bottle **NOTE:** The - character is considered a literal character if it used as the first or last character within the brackets, or if it is escaped with a backslash.
[^]	Any character that is not in the specified set of characters.

Construct	Description
	Expression: d[^u]g **Matches:** dog and dig but not dug
.	Any single character. When preceded with the escape character (\), it is equivalent to a period character. **Expression:** d.g **Matches:** dog, dig, and dug
*	Zero or more matches of the preceding character. **Expression:** a*ha **Matches:** ha, aha, aaha, aaaha, etc.
?	Zero or one matches of the preceding character. **Expression:** colou?r **Matches:** color and colour
+	One or more matches of the preceding character. **Expression:** goo+gle **Matches:** google, gooogle, etc.
{n}	The exact number of matches for the preceding character. **Text:** he had to go too far **Expression:** o{2} **Matches:** oo in "too"
{n,m}	The preceding character must match at least "n" times but no more than "m" times. **Expression:** mo{2,3}n **Matches:** moon and mooon, but not mon or moooon
( )	Matches subexpression.
|	Matches any of the alternate expressions. **Expression:** theat(er|re) **Matches:** theater and theatre
(?#)	A comment within a regular expression. The comment ends after the first closing parenthesis. **Expression:** theat(er|re)(?# using this example a second time) **Matches:** theater and theatre
(?=)	A zero-width positive lookahead assertion. **Expression:** ^(?=.{32}$)(\d+) **Matches:** numbers at the beginning of any line which is exactly 32 characters long
(?!)	A zero-width negative lookahead assertion. **Expression:** ^(?!test).+

Construct	Description
	Matches: any line that does not begin with "test"

For more information about regular expressions, see

- msdn.microsoft.com/en-us/library/az24scfc.aspx
- msdn.microsoft.com/en-us/magazine/cc163473.aspx

Multi-Database Backup

Multi-Database Backup launches the Backup wizard which 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 now available for multi-database backup in the UI.

To back up multiple databases

1. Right-click the server instance and select **Multi-Database Backup…**. Alternately you can select the **Multi-Database Backup**button at the top.
2. Select databases for backup. For more information, see LiteSpeed's Logic for Backing Up Multiple Databases on page 110.
3. Complete the wizard. For more information, see To back up databases using the Backup wizard on page 99. on backup options.

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

Fast Compression

Fast Compression 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 LiteSpeed 5.1 or later; 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. For more information, see Back Up Databases on page 98.
- Create a maintenance plan with the Fast Compression Backup task. For more information, see Back Up Databases Using Maintenance Plans on page 133.

Tip: To create Fast Compression jobs or tasks on several instances, create and deploy a LiteSpeed backup template. For more information, see Create Backup Templates on page 87.

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):

- **Separate backup files**–(Default) 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).
- **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.

The Self-Contained Backup Sets option automatically verifies the Full backup exists. The Separate Backup Files option performs the same validation by default.

Note: For cloud backups only "Separate backup files" type is supported.

Full Backup Conditions

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

- Force a full backup every - The amount of time elapsed since the last full backup. The default is *14* days.
- Data change threshold - The amount of database changes since the last full backup. The default is *35*%.

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 default option is to query actual data pages. It 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 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.

Select the **Extension for backup files** checkbox to enter or change the backup file name extension. The default is set to bkp.

NOTE: You can select the backup file extension for Fast Compression and make the new default, bak, for new items. For an existing item that does not have an extension defined, bkp is displayed when the item is edited (maintenance plans and templates).

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. xp_slsFastCompression 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

TIP: "Cleanup" means SmartCleanup. For more information, refer to SmartCleanup.

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.

The **Verification** options include:

- Do not verify backup (default).
- Verify last backup.
- Verify both the last full and latest differential backup.
- Verify the last full and all associated differential backups.

The **Cleanup** options include:

- Clean up full/differential backups older than 28 (default) days.
- Clean up transaction logs older than 7 (default) days.

  Note: Clean up transaction log options are available if you set up a transaction log within the wizard (or template).

- Do not delete if archive bit is set.

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.

For more information, see SmartCleanup on page 124.

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

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 whether you create a double click restore loader. By default you will create a double click restore loader. 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. If you do not create a double click restore loader then the backup file name will have the .exe extension.

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
Create one Double-Click Restore executable file. The executable may be greater than 4GB for large databases. Windows Server is unable to run executable files larger than 4GB. However, the file will be convertible/restorable by LiteSpeed file. `@doubleclick = 1`	X.exe	X.exe (No name changes)
	X	X.exe
Create a Double Click Restore Loader (Default). `@doubleclick = 2`	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.

Create Double Click Restore Executables

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 = 2` when creating a backup using procedures. For more information, see xp_backup_database on page 280.
- Supply `--doubleclick 2` when creating a backup using command-line interface. For more information, see LiteSpeed Command-Line Arguments on page 179.

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 Methods

You can specify a compression level manually, or you can use Adaptive Compression to let LiteSpeed automatically select the optimal compression level. For more information, see Adaptive Compression on page 122.

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 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	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. **NOTE:**Levels 9, 10 and 11 were deprecated in version 6.0 and are now automatically mapped to the new compression level 8.

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. For more information, see Test Optimal Backup Settings on page 83.

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.

NOTE: LiteSpeed supports backing up, restoring and shipping transaction logs of the databases encrypted with transparent data encryption (TDE). If you want to compress the backup, you should choose compression level 1 to minimize CPU, since using a higher level of compression will only cause CPU to increase without any real benefit on the backup file size. If you choose compression level 0, LiteSpeed will not attempt to compress the backup. Review the following for additional information:

- Its important to back up the database encryption key, because there is no way to recover the data without the key. LiteSpeed does not automatically export the encryption key. If needed, you can include the encrypted key file in the backup. For more information, see Back Up Databases on page 98.
- To further protect the backup, you can use LiteSpeed Encryption with TDE databases to add a secondary layer or protection to the backup. For more information, see Encryption Methods on page 123.

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 Methods

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
- 128-bit AES Microsoft implementation (MS_AES_128)
- 192-bit AES Microsoft implementation (MS_AES_192)
- 256-bit AES Microsoft implementation (MS_AES_256)

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.

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.

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. For more information, see About Using Extended Stored Procedures on page 275.

To configure retry options from the command line

Use -X or --IOFlags parameter. For more information, see LiteSpeed Command-Line Arguments on page 179.

NOTE: LiteSpeed makes 3 attempts for creation of the backup folder at 0.5 second intervals. This is not configurable.

SmartCleanup

SmartCleanup 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.

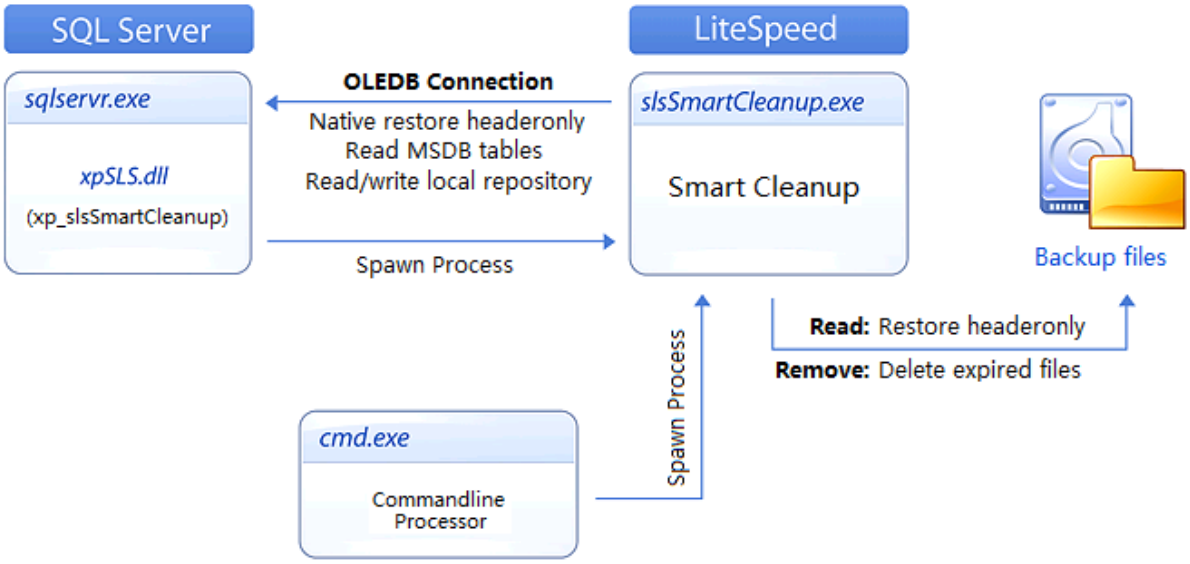
SmartCleanup is backup set aware and will never remove a full backup that is needed by a differential backup that is not being deleted.

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
- Files that have the filesystem archive bit set (if that option is selected)

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



To configure SmartCleanup in Wizards

- Open existing or create a new backup template. For more information, see Create Backup Templates on page 87.
- Open or create the Fast Compression Database task in a maintenance plan. For more information, see About Creating Maintenance Plans on page 129.
- Run the Backup wizard and select the Fast Compression backup type. For more information, see Back Up Databases on page 98.

To run SmartCleanup manually, do one of the following:

- Use the xp_slsSmartCleanup extended stored procedure. For more information, see xp_slsSmartCleanup on page 550.
- Run slsSmartCleanup.exe with the appropriate arguments from the command-line. For more information, see SmartCleanup Command-Line Arguments on page 222.

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.

ALL variables are supported for both files and folders.

Accepted Variables

LiteSpeed accepts the following variables:

Variable	Description
%DATABASENAME% or %D	Database name
%TYPE% or %T	Backup type (full, diff, log, fast compression folder)
%SERVER%	Server name
%INSTANCE%	Server instance name
%DEFAULTDIR%	Default backup directory
%AG%	AlwaysOn Availability Group name
%z	Timestamp, the number of seconds elapsed since 00:00:00 January 1, 1970, UCT. See Note[1].
Date and time variables	
%DATE%	Date. See Note[1].
%TIME%	Time (hhmm). See Note[1].
%DATETIME%	Date and time. See Note[1].
%a	Abbreviated weekday name. See Note[1].
%A	Full weekday name. See Note[1].
%b	Abbreviated month name. See Note[1].
%B	Full month name. See Note[1].
%d	Day of the month (01-31). See Note[1].
%H	Hour in 24h format (00-23). See Note[1].
%I	Hour in 12h format (01-12). See Note[1].
%j	Day of the year (001-366). See Note[1].
%m	Month as a decimal number. See Note[1].
%M	Minute (00-59). See Note[1].
%p	AM or PM designation. See Note[1].
%S	Second (00-59). See Note[1].
%U	Week number with the first Sunday as the first day of week one (00-53). See Note[1].
%w	Weekday as a decimal number with Sunday as 0 (0-6). See Note[1].

Variable	Description
%W	Week number with the first Monday as the first day of week one (00-53). See Note[1].
%y	Year, last two digits (00-99). See Note[1].
%Y	Year. See Note[1].
%Z	Time zone name or abbreviation.
	Note[1]: Not supported for Fast Compression.

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.

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

6

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

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 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 in 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.

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 **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. Copy Maintenance Plans

   Tip: 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…
Fast Compression Backup	Perform full or differential backups using the Fast Compression technology. **NOTE:** Fast Compression is only available with LiteSpeed 5.1 or later; Enterprise license. Back Up Databases Using Maintenance Plans
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. Back Up Databases Using Maintenance Plans
Check Database Integrity	Validate the following: • Disk space allocation • Page and structural integrity for tables and indexed views • Catalog consistency • Contents of indexed views • Service Broker data
Reorganize Index	Defragment and compact existing indexes to improve performance.
Execute Job	Execute an existing SQL Server Agent job.
Clean Up Maintenance Plans	Remove obsolete backup files and reports created by a maintenance plan. Clean Up Maintenance Plans

Task	Add this task to…
Clean Up History	Remove historical data from the **msdb** database and LiteSpeed Local and Central repositories for the following: • Back up and restore history • SQL Server Agent jobs • Maintenance plans • LiteSpeed activity **NOTE:** LiteSpeed activity is removed from the Central repository, only if it is located on the same server, where the Local repository resides. • Any information for deleted databases • Log shipping history • Status history (Job, DTS, Maint Plans)
Notify Operator	Send an email notification to one or more existing operators. You can use the notification profiles only if they are already configured within a SQL Server instance. **TIP:** Variables defined in the maintenance plan can be used to format the Subject Line. There is also help information inside the task that lists the available Subject Line variables. **NOTE:** The SQL Server Agent must be configured to send email using Database Mail. Review the following for additional information: • http://msdn.microsoft.com/en-us/library/ms189635.aspx • http://technet.microsoft.com/en-us/sqlserver/dd939167.aspx
Rebuild Index	Drop and recreate an index to improve performance.
Shrink Database	Reduce the size of data and log files in a database that grow beyond a specified size.
Execute T-SQL	Execute statements or batches on one or more databases. **NOTE:** SSIS maintenance plans support only the Transact-SQL command type.
Update Statistics	Update column and index statistics.

4. (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:
   1. Clean Up History
   2. Check Database Integrity
   3. 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.

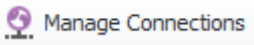
6. Click and repeat steps 2, 3 and 4 for additional subplans you want to add.

7. (Optional) Set reporting options.

Reporting and Logging	Click to set the reporting and logging options. Reporting and Logging in Maintenance Plans
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 Designpane 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). For more information, please refer to the "Privilege and Grant Requirements" section of the LiteSpeed *Installation Guide*.
- 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 

. 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). About Automating Maintenance Tasks

Scenario
You need to create a maintenance plan to only back up databases which names start with "C", "DB1" and "LiteSpeed", except "SoftwareCMSS".

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. LiteSpeed's Logic for Backing Up Multiple Databases **Scenario:** Select the Databases matching regular expression option, enter the following in the Mask field: ^(?!SoftwareCMSS)(C|Quest Software|LiteSpeed) and click **Add**. Use Wildcard and Regular Expressions in LiteSpeed
Back up database across one or more files/destination	This option creates one backup file for all selected databases. If you need to create striped backups, provide multiple backup destinations. Tip: You can backup databases to disk, cloud, TSM backup, TSM archive, or tape. Fast compression backups can be run to disk, cloud, or TSM backups.
Create backup file for every database	Select this option, if you want to create separate disk backups for databases.
LiteSpeed file format	The default backup file format uses the following information: • %D–Database name • %T–Backup type (Full, Diff or Log) • %Y-%m-%d-%H%M%S–Date and time • %EXT%–File extension You can specify a custom backup file format using both the LiteSpeed variables and text. LiteSpeed Variables If you want LiteSpeed to remember a custom file format and use it for new backup tasks on this instance, modify the format in the LiteSpeed file format field as needed and click **Set to Instance**. **NOTE:** Fast Compression handles the naming of files automatically. For more information, see Backup Files and Folders on page 116.

If you selected **Cloud** review the following:

Cloud vendor	Select the cloud vendor, Amazon S3, Azure Blob or Google Storage, from the drop-down list, or use ... to choose an existing cloud account. You can also specify a custom cloud account directly in the task.
Storage Account name (Azure blob only)	Enter the name of your Azure blob storage account.
Access key	Enter the name of the unique Web service alphanumeric access key that identifies each user.

Secret key (Amazon S3 only)	Enter the name of the Web service secret key that was assigned when you initially setup your cloud account.
Service account ID (Google Cloud Storage only)	Enter the name of your Google Cloud Storage Service Account ID.
Private key (Google Cloud Storage only)	Enter the name of your Google Cloud Storage Private Key.
Project ID (Google Cloud Storage only)	Enter the name of your Google Cloud Storage Project ID.
Storage type (Azure blob only)	Select the Azure storage type: block blobs or page blobs from the drop-down list.
Container (Azure blob only)	Select the Azure blob storage container from the drop-down list.
Region (Amazon S3 and Google Cloud Storage only)	Select a Web service region to use for a bucket with the drop-down list.
Bucket (Amazon S3 and Google Cloud Storage only)	Enter the name of the container for objects. Bucket names must be at least 3 and no more than 63 characters long. Alternately use the drop-down list to select an existing bucket.
Folder name	Enter the name of your Cloud folder.
Overwrite existing fields	Click to overwrite existing fields.
Advanced options	Click to specify cloud advanced options. • Azure blob advanced options include: use SSL, Government account, automatic striping, and proxy settings. • Amazon S3 advanced options include: use SSL, GovCloud (US) region, automatic striping, storage class, use server side encryption (AES-256), and Use Amazon S3 Transfer Acceleration. • Google Cloud Storage advanced options include Automatic Striping, Storage Class and Proxy Settings.

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.) **NOTE:** This file contains session options such as the TSM server's TCP

	address. If you select the **Use PASSWORDACCESS GENERATE from TSM configuration file** 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 **TSM Filespaces** to select the existing or enter new file space name(s). **NOTE:** Fast Compression handles the naming of files automatically. For more information, see Backup Files and Folders on page 116.
TSM Objects (Regular backups)	Click **TSM Objects** and specify the filespace, the high-level name and the format of the low-level name. If the option Stripe on TSM object and the number of stripes are selected, the mask %@ is added by default to the end of the defined file format. If the option Stripe on TSM object and the number of stripes are selected, the mask %@ can be added by the user to any place of file format.

3. Select the Options page.

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

Backup set expires	Select to set the transaction log expiration. The expiration can be set to expire after a selected amount of days or on a particular day. Select one of the following: • After - The backup set expires after a selected number of days. • On - The backup set expires on a specifically selected day.
Remove files older than	This option only removes files that match the file format from the specified **destination folder**. • Delete empty subfolder - Select this option to delete the empty subfolder. **TIP**: To remove obsolete files from different locations, use the Clean Up task. Clean Up Maintenance Plans
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 to manage whether to use SQL server native compression (available for native maintenance plans only): • Use the default server setting • Compress backup • Do not compress backup
Copy Only Backup	Select this option to enable copy-only backups.
Force a full backup if one has not been created	Select this option to run a force full backup if one has not been created. This is a useful option for differential and transaction log backups that require an initial full backup at first.

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: • Force a full backup every - The amount of time elapsed since the last full backup. The default is *14* days. • Data change threshold - The amount of database changes since the last full backup. The default is *35*%. 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 default option is to query actual data pages. It

	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 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. Select the **Extension for backup files** checkbox to enter or change the backup file name extension. The default is set to bkp. NOTE: You can select the backup file extension for Fast Compression and make the new default, bak, for new items. For an existing item that does not have an extension defined, bkp is displayed when the item is edited (maintenance plans and templates).
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.
Verification and Cleanup	TIP: "Cleanup" means SmartCleanup. For more information, refer to SmartCleanup. 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. The **Verification** options include: • Do not verify backup (default). • Verify last backup. • Verify both the last full and latest differential backup. • Verify the last full and all associated differential backups. The **Cleanup** options include: • Clean up full/differential backups older than 28 (default) days. • Clean up transaction logs older than 7 (default) days. Note: Clean up transaction log options are available if you set up a transaction log within the wizard (or template). • Do not delete if archive bit is set. Specify the cleanup options. SmartCleanup **NOTE:** While transaction log backups can be automated within a separate Back Up Database task, you can configure cleanup of transaction log backups in the Fast Compression Backup task.
Notification	Send an email notification to one or more existing operators. You can use the notification profiles only if they are already configured within a SQL Server instance. **TIP:** Variables defined in the maintenance plan can be used to format the Subject Line. There is also help information inside the task that lists the available Subject Line variables. **NOTE:** The SQL Server Agent must be configured to send email using Database Mail. Review the following for additional information: • http://msdn.microsoft.com/en-us/library/ms189635.aspx • http://technet.microsoft.com/en-us/sqlserver/dd939167.aspx

4. Select the LiteSpeed page. Review the following additional information about the encryption and compression options:

Adaptive Compression	LiteSpeed automatically selects the optimal compression based on throughput and CPU usage and optimize backups either for size or for speed (default). **NOTE:** Adaptive Compression is only available with LiteSpeed 6.5 or later; Enterprise license.
Compression level	Select 0 for no compression or 1-8 (default 2) to compress the file. For more information, see Compression Methods on page 121. **NOTE:** 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. For more information, see Test Optimal Backup Settings on page 83. Tip: For cloud backups, set the default compression level to 7. Using a higher compression level has real savings. Reducing the number of bytes sent to the cloud makes for faster backups and reduces Internet bandwidth.
Encrypt backup	Select this checkbox to encrypt the backup. Then, select the encryption level and enter the encryption password. For more information, see Encryption Methods on page 123.

Review the following additional information about the advanced options:

Compression 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.
Max transfer size	Enter the maximum backup file size in bytes. . The possible values are multiples of 65536 bytes (64 KB) ranging up to 4,194,304 bytes (4 MB). The default is *1048576*.
Buffer count	Enter the number of SQL Server buffers available for a LiteSpeed operation. The default is set by SQL Server.
CPU throttle	Enter the maximum percentage of CPU that LiteSpeed can use for the process. The default is *100*.
Processor affinity	Click ... to select which processors LiteSpeed can use. The default is *0*, which allows LiteSpeed to use all available system processors.
Processor priority	Select the priority of the backup over other transactions or processes running on

	the same server. The default is *Normal*.
Comment	User comment written into the backup header. Is blank by default.
Logging level	Select a logging level to define what events to log for the console. You can find the log events in the Application Event Log.
Network resilience	If LiteSpeed fails to write disk backups or reads from disk, it waits and retries the operation. You can enable and disable and control the number of times to retry and the amount of time to wait before retrying. • Number of times to retry any given read/write attempt–The default is *4 retries*. The maximum allowed setting is 1000 retries. • Wait period before each retry attempt (in seconds)–The default period to wait before retry is *15 seconds* The maximum allowed setting is 300 seconds. For more information, see Network Resilience on page 124.

TIPS:

- 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. For more information, see Adaptive Compression on page 122.

NOTE: LiteSpeed defaults typically result in the best performance. You should only modify advanced options after careful planning and testing. For more information, see Configure LiteSpeed Defaults on page 60.

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

Optimize the Object Level Recovery speed	Select to create an index of objects in the backup file. This option is only available for LiteSpeed backups. The default is enabled. **NOTE:** 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. During the backup process the .lsm file is created in the temp directory and attached to the backup file after the backup is completed. 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.
Create Double Click Restore executable	Select to create a Double Click Restore Loader that allows you to restore a backup on a server instance that does not have LiteSpeed installed. If you select to **Create one Double-Click Restore executable file** then note the following warning. The executable may be greater than 4GB for large databases. Windows Server is unable to run executable files larger than 4GB. However, the file will be convertible/restorable by LiteSpeed

	file. For more information, see Double Click Restore Executables on page 120. **NOTE:** A Double Click Restore can only be created for a disk file.
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 **Continue on error** option, the backup is executed despite encountering an invalid backup checksum.
Continue on error	Select this option to continue running the backup even if an invalid checksum is encountered.
Select directories to mirror the backup to (Overwrite must be selected)	Click **Disk** to backup to disk. Click **Cloud** to backup to cloud.
Select files/folders to attach to the backup set	Select **Add** to attach files or folders to the backup set.

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: • Backup files (disk and cloud) • Maintenance plan text reports • Any files
Skip files in use by another process or access is denied	Select this option to skip files that are in use by another process or application and/or cannot be accessed. When this option is not selected locked files are forcibly deleted.
Delete specific file	Select this option to delete a specific file. Enter the file name or select ... to browse for the file.
Search folder and delete files based on extension	Select this option to remove any files with a specific file extension. Enter the folder name or select ... to browse for the folder.
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. • Delete empty subfolder - Select this option to delete the empty subfolder.
Delete files based on the age of the file at task run time. Delete files older than the following:	Select this option to automatically delete files based on their age in number of minutes, hours, days, weeks, months, and years.

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. Back Up Databases Using Maintenance Plans
- Different LiteSpeed versions–Some options may be lost if they are not supported by the previous LiteSpeed version.

- Different LiteSpeed versions and different SQL Server versions–Back Up Databases Using Maintenance Plans

  - LiteSpeed versions on the source and the target are different, but database lists are the same.

  - LiteSpeed version is the same but the database list is different (warning indicates that some databases do not exist).

  - LiteSpeed versions and database lists are different.

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 **Paste Maintenance Plan**.

To copy a subplan

1. In the Design pane, select a subplan and click to 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 to paste.

To export or import a maintenance plan

1. To export a plan, select a maintenance plan you want to export and click the **Export Plan** button on the toolbar. **NOTE:** The file extension is **.mpp**.
2. To import a plan, right-click the SQL Server instance from where you want to import the plan and select **Import Maintenance Plan**.
3. Select the **.mpp** file that relates to the maintenance plan you want to import.

To export or import a subplan

1. To export a sublan, in the Design pane select a subplan you want to export and click Export. **NOTE:** The file extension is **.mps**.
2. To import a subplan, open a new or existing maintenance plan in the Design pane.
3. This step is optional. Click New on the Design pane toolbar to add a new subplan.
4. Select a subplan and click Import.
5. Select the **.mps** file that relates to the subplan you want to import.

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

7

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.

Tip: Database backups created with LiteSpeed Version 8.x cannot be restored using older versions of LiteSpeed.

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. In the Navigation pane, select a database to restore and click the **Restore** or **Automated Restore** button. Alternately you can right-click a database and select **Restore** or **Automated Restore** from the menu. If you clicked **Restore**, click one of the following. If you clicked **Automated Restore**, review the automated restore information below.

   - **Database (full or differential)**–Restore database only.
   - **Files and Filegroups**–Restore files and filegroups only.
   - **Transaction Log**–Restore transaction log only.

     Tip: The database (full or differential), files and filegroups, and transaction log restore types use the restore wizard pages listed in the steps below: restore destination, backup source, backup content, recovery options, data files, scheduled restore, execute script, and finish.

   - **Attached Files**–Restore attached files only.

     Tip: The attached files type uses the restore wizard pages listed in the steps below: restore destination, backup source, backup content, data files, attached files, scheduled restore, execute script, and finish.

   - **Automated Restore**–Restore the most recent full backup and optionally differential and transaction log backups. If you select this option, the Restore wizard creates an Automated Restore job. Backup files can be restored immediately or restored in the background.

     Tip: The automated restore type uses the restore wizard pages listed in the steps below: restore destination, backup source, backup files, restore options, database integrity database files, scheduled restore, notification, execute script, and finish.

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. Additionally, you can select the **View current activity** link to view the database connection activity table. 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.

   Caution: **Automated Restore for multiple databases - When running restores for multiple databases, we recommend that you use the %DATABASENAME% variable to automatically generate an original database name for database(s) going to be restored. When running restores from multiple servers, different databases with the same server name can be overwritten. To prevent this, the variable %SOURCESERVER% adds the source server name (server+instance) to the target database name when running multiple automated restores. For example: %SOURCESERVER%_%DATABASENAME%.**

   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 select **Device** to manually select files to restore.

Database

If you select **Database**, the database you are restoring is displayed. You can select another database using the **Database** drop down selector. Use the **SQL Server** drop down selector if the database is on a different SQL Server instance.

TIP: You can do point in time restores, restoring from a specific date and time or a time period (for example four hours prior to recovery).

TIP: The restore type option **Restore Verify Only** in combination with the backup type **Verify the latest full or differential backup** is less time consuming for full and differential backups or fast compression backups where you only want to verify the latest backup, rather than the full and the latest differential. For backups that are run every day this will serve to verify all backups.

Device

If you select **Device**, the options are **Disk**, **Cloud**, **TSM Backup**, **TSM Archive**, and **Tape**.

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

If you selected **Cloud**, review the Cloud Account Settings information below:

Cloud account	Select the Cloud account from the drop-down list.
Cloud Accounts...	Select the following items to add or edit the registered cloud account settings. • Add - (For Amazon S3 only), click to add cloud vendor, display name, authentication, region, storage class (standard, infrequent access, reduced redundancy storage), and bucket. Select: use SSL, use server side encryption (AES-256), GovCloud (US) Region, and automatic striping (auto, 10, 25, 50, 100, 500, 1000, 1995) GB. Select 'Use Amazon S3 Transfer Acceleration Speed' to use Amazon's S3 Transfer Acceleration feature which allows for increased upload speed to S3 storage up to 200% in some cases by using local CloudFront endpoints. • Add - (For Azure Blob only), click to add cloud vendor, display name, storage account name, access key, storage type (block blobs or page blobs), container, use SSL, government account, and automatic striping . Options for block blobs are: auto, 10, 25, 50, 100, 250, 500, 1000, 2000, 3000, 4000 and 4300 GB. Options for page blobs are: auto, 10, 25, 50, 100, 250, 500, and 995 GB.

	• Add - (For Google Storage only), click to add cloud vendor, display name, service account ID, private key, project ID, storage class, region and bucket. Use SSL is always selected as Google always uses it. • Edit - (For Amazon S3 only), click to edit display name, authentication, region, storage class (standard, infrequent access, reduced redundancy storage), and bucket. Bucket name must conform to DNS naming requirements and must not contain periods ("."). Select: use SSL, use server side encryption (AES-256), GovCloud (US) Region, and automatic striping. Options for automatic striping are: auto, 10, 25, 50, 100, 250, 500, 1000, and 1995 GB. Select 'Use Amazon S3 Transfer Acceleration' to use Amazon's S3 Transfer Acceleration feature which allows for increased upload speed to S3 storage up to 200% in some cases by using local CloudFront endpoints. Additional data transfer charges may apply. See Amazon S3 pricing for more details. • Edit - (For Azure Blob only), click to edit display name, access key, storage type (block blobs or page blobs), container, use SSL, government account, and automatic striping. Options for block blobs are: auto, 10, 25, 50, 100, 250, 500, 1000, 2000, 3000, 4000 and 4300 GB. Options for page blobs are: auto, 10, 25, 50, 100, 250, 500, and 995 GB. • Edit - (For Google Storage only), click to edit display name, service account ID, private key, project ID, storage class, region and bucket. Use SSL is always selected as Google always uses it. • Delete - Click to delete the Cloud account from the console. • Import - Click to import a saved Cloud account in XML format. • Export - Click to export and save a Cloud account in XML format.
Proxy Settings	Select the following items to edit the Cloud account proxy settings. • Use LiteSpeed Server proxy settings - Click to use the server proxy setting. This is the default selection. You can also edit the proxy settings from this item. • Use LiteSpeed Console proxy settings - Click to use the console proxy setting. You can also edit the proxy settings from this item.

	• Specify custom proxy settings - Click to add your own custom proxy address, port, username and password.

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.) **NOTE:** This file contains session options such as the TSM server's TCP address. If you select the **Use PASSWORDACCESS GENERATE from TSM configuration file** checkbox and your options file is configured to support this option, you do not need to specify the client node and client owner password.
TSM Object	Click **Select TSM Object**. Enter the filespace and the high-level and low-level names and click **Query TSM** 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. **NOTE:** 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.

If you selected **Automated Restore**, review the following for additional information.

Restore from	Select SQL Server and databases to specify the source from where to search the backups for automated restore. **NOTE:** The restore from parameters help LiteSpeed to narrow down its search for the required backup files in the source folders. Use the drop-down treeview to add multiple sources from the System and User parent nodes. Select User databases and all subordinate user databases are automatically selected. Individually select and deselect databases using the checkbox next to it. For AlwaysOn availability groups, you may need to specify both SQL Server (primary and secondary) to allow LiteSpeed to search backups among all replicas. To specify a secondary SQL Server, use the semicolumn as a separator between SQL Servers (e.g. SQL-PROD\AG1; SQL-PROD\AG2) in the Source list (Server column).
Latest backup search method	Select one of the following options:

	• Folder scan–LiteSpeed will search the specified folder for the most recent database backups. You can configure LiteSpeed to search subfolders and filter backups using the specified file extensions. • Specific backup file name–You can use this option to automate restore operations if the same file gets overwritten during the backup or if new backups are always appended to the same file. NOTES: • If restoring a striped backup, you can specify multiple locations/filenames. • You can enter several backup extensions per path. Separate them with commas or semicolons.
Backup type	Specify backup types to use for the restore. Select one of the following options: • Full–The most recent full database backup. • Full and differential–The most recent full database backup and any existing differential backups based on this full. • Full, differential and transaction logs–The most recent full database backup and any existing differential and/or transaction log backups created after the most recent full backup. • Include copy only backups–Select the **Include copy only backups** check box to add copy only backups in the restore. • Specify a point in time to restore to. By default LiteSpeed restores to the most recent state possible. Alternatively, specify a specific date and time or a date and time relative to the restore time. For example, specify a time measured in days, hours, minutes and seconds from the restore time.

5. If you selected Automated Restore, on the **Backup Files** page, select the backup file locations to search.

Restore from	Select the restore from location (disk or cloud) using the drop down menu.
Add	Add the backup file location to restore to the list by selecting and using the Backup File Location wizard.
Remove	Select and remove the backup file location from the list.

6. If you selected Automated Restore, on the **Database Integrity** page, define the options to check database integrity after restore.

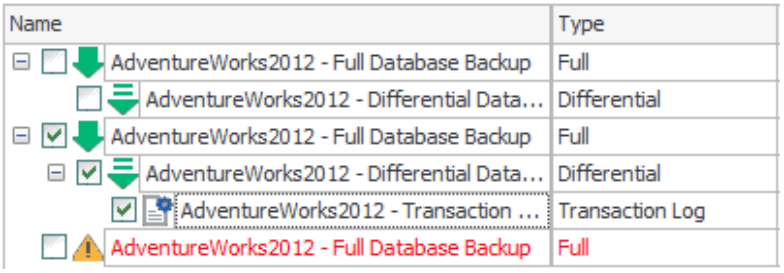
Check database integrity after restore (DBCC_CHECKDB)	Use this option to run a CHECKDB on the restored database and

	report the results to the repository for review. This option is selected by default. Select a combination of the following database integrity options: • Check physical structure only (PHYSICAL_ONLY).This option is selected by default. • Check the database for column values that are not valid or out-of-range (DATA_PURITY). • Perform logical consistency checks on indexed views. XML indexes and spatial indexes (EXTENDED_LOGICAL_ CHECKS). • Do not perform intensive checks of nonclustered indexes for user tables. This option is selected by default. • Use locks instead of using an internal database snapshot. • Do not include informational messages in notification report (NO_INFOMSGS). This option is selected by default.

7. If you selected Automated Restore, on the **Restore Options** page, specify options for automated restore.

Drop databases after restore	Use this option if you no longer need the restored database. For example, if you are only restoring the latest backup for testing purposes. This option contains two additional options to select. One or both options can be selected. • On success restore and check database integrity operations - The database is dropped after a successful restore and database integrity check. • On failure any of restore or check databases integrity operations - The database is dropped after failing the restore or database integrity check.
Overwrite the existing database	Use this option if you want to overwrite the existing database with the restored database.
Include databases that are part of a replication plan	Use this option to include databases that are part of a replication plan.
Recovery state	The options are as follows: • Select to leave the database in an operational state (RESTORE WITH RECOVERY). The default is selected. • Select to leave the database in a non-operational state and allow restoration of additional transaction logs (RESTORE WITH NORECOVERY). The default is not selected.
Password	Provide a password for encrypted backups. **NOTE:** Automated Restore requires that you use the same password for all encrypted backups.

8. Review the following additional information about the **Backup Content** page. Skip this page for Automated Restore.

Select server instance	Select the server that contains the backup you want to restore.
Select database backup	Select the database that you want to restore.
First backup to recover	a. Click the ellipsis ... to launch a window containing a list of backups to restore. b. Scroll down the list and select a backup. The backup to recover table is populated with a list of backups. The table includes backup name, type, destination, encrypted, server, database, position, begin date, finish date, size, user, expiration, and copy only. See the partial table below. • indicates a successful full backup. • indicates a successful differential backup. • indicates a successful transaction log backup. • indicates a warning. Possible reasons: encrypted backup that requires entering an encryption key, backup no longer exists, or corrupted backup. • Red text indicates the backup is not available. This could be because the backup was local to the source database and not available from the target or because the backup does not exist. • indicates a selected backup. 
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.

9. On the **Recovery Options** page, select the database recovery state following the restore. Skip this page for Automated Restore.

Overwrite the existing database	Select to have the current database overwritten with the restored database. The default is not selected.
Preserve the replication settings	Select to preserve the replication settings for the restored database. The default is not selected.
Restrict access to the restored database	Select to restrict access to the database after it is restored. The default is not selected.
Recovery state	The options are as follows: • Select to leave the database in an operational state (RESTORE WITH RECOVERY). The default is selected. • Select to leave the database in a non-operational state and allow restoration of additional transaction logs (RESTORE WITH NORECOVERY). The default is not selected. • Select to leave the database in read-only mode. Undo committed transactions and save the undo actions in a standby file (RECOVER WITH STANDBY). The default is not selected. When selected, the default listed standby file can be used. Or select the ellipsis button to browse and select another standby file.

10. Review the following additional information about the **Data Files** page:

Prompt before restoring each backup	Select this option if you would like to receive a prompt notification before restoring each backup. The default is not selected.
Eject tapes (if any) after restoring each backup	Select this option if you would like to eject any tapes after restoring each backup. The default is not selected. This option is available only for tapes.
Restore as compressed, read-only database	Using this option, you can restore a user database into an NTFS compressed folder or restore a tlog to a read-only database in a compressed folder. **NOTES:** • When using an NTFS-compressed folder for a database, it can only be restored as read-only. • You can only use this feature on Windows NTFS file systems. Specify a compressed folder for the data files by editing the Restore As paths. If a folder does not exist, LiteSpeed will create it as NTFS compressed.
Restore the database files as	Although you can manually enter DATA and LOG locations, including secondary data files locations, it is recommended that you use locations generated by LiteSpeed. If restoring database (full or differential), the following links can be selected (not available for Automated Restore): • Keep original database (full or differential)–Click to display in the table below the original database locations. • Use SQL Server instance default locations–Click to display in the table below the SQL Server instance default locations. • Select new location–Click to launch the **Database Files Destination** window and select a new database output file location. You can browse the network, add, delete, and rename files. • Restore to locations from backup set–Click to get locations from the backup set. If the source and target locations do not match or if they are set to other than the default, LiteSpeed you can select one of the following options in Automated Restore: • Use SQL Server instance default locations–To use DATA and LOG directories of the existing database you are restoring the backup to. The default is selected. • Custom locations for data and log files–To use DATA and LOG directories of the database which backup you are restoring. The default is not selected.

	• Folder for data files–To enter a new location for all DATA files. Click the ellipsis button to browse and select other folders for data files. • Folders for log files–To enter a new location for all LOG files. Click the ellipsis button to browse and select other folders for log files.
Processor affinity	Click ... to select which processors LiteSpeed can use. The default is *0*, which allows LiteSpeed to use all available system processors.
Logging level	Select one of the following options: • None–LiteSpeed does not write a log file for the backup or restore operation. • Verbose–LiteSpeed writes a log file for the backup or restore operation. • Verbose. Log file is removed on success–LiteSpeed only saves log files if the backup or restore operation fails. If it succeeds, LiteSpeed does not save the log.
Network resilience	If LiteSpeed fails to write disk backups or reads from disk, it waits and retries the operation. You can enable and disable and control the number of times to retry and the amount of time to wait before retrying. • Number of times to retry any given read/write attempt–The default is *4 retries*. The maximum allowed setting is 1000 retries. • Wait period before each retry attempt (in seconds)–The default period to wait before retry is *15 seconds* The maximum allowed setting is 300 seconds. For more information, see Network Resilience on page 124.

11. If you added an attachment to the backup file, select the **Restore Attached Files and Directories** on the **Attached Files** page.
12. Review the following information about the **Schedule Restore** page. Select Weekly on (for Automated Restore only), Run immediately, Run in background, or Schedule (Custom Schedule for Automated Restore. Selecting Schedule launches a page for adding the schedule name, schedule type, occurs, weekly, daily frequency, duration, and description.
13. Review the following information about the **Notification** page (for Automated Restore only). You can specify the notification of failure options that are sent after each restore. Select one of the following:
    - Do not use notification–All failure notifications (including operator selections) are disabled.
    - Notify every time–Notify for all successes and failures. An operator can be selected and configured to receive notifications. Click the ellipsis button on the far right to create new or edit existing operators.
    - Failure only–Notify for failure only. An operator can be selected and configured to receive notifications. Click the ellipsis button on the far right to create new or edit existing operators.

NOTE: The SQL Server Agent must be configured to send email using Database Mail. Review the following for additional information:

- http://msdn.microsoft.com/en-us/library/ms189635.aspx
- http://technet.microsoft.com/en-us/sqlserver/dd939167.aspx

14. 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 sqllitespeed.exe. For more information, see LiteSpeed Command-Line Arguments on page 179.

- 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–For more information, see Configure Logging in LiteSpeed on page 576.
- 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, start the server instance in single-user mode and execute the LiteSpeed restore statement.

To manually restore a master database

1. From a command prompt, run as administrator, 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 to start the server instance in single-user mode:

   ```
   sqlservr.exe -c -m"LiteSpeed"
   ```

   For named instance:

   ```
   sqlservr.exe -c -m"LiteSpeed" -s"<instance name>"
   ```

   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.

3. 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
   ```

4. Execute the LiteSpeed restore statement to restore a full database backup of master.

   ```
   SQLLiteSpeed.exe -R"Database" -D"master" -F "<path to backup file\backup file
   name>" -W"REPLACE" -S "<server name>\<instance name>" -T
   ```

Example	Script
Restore a master database to the default instance	`sqllitespeed.exe -R"Database" -D"master" -F"c:\backup\master.bak" -W"REPLACE" -S "<server name>" -T`
Restore a master to the named instance	`sqllitespeed.exe -R"Database" -D"master" -F"c:\backup\master.bak" -W"REPLACE" -S "<server name>\<instance name>" -T`
Restore with encryption	`sqllitespeed.exe -R"Database" -D"master" -F"c:\backup\master.bak" -Kpassword -W"REPLACE" -S "<server name>\<instance name>" -T`
Restore with replace	`sqllitespeed.exe -R"Database" -D"master" -F"c:\backup\master.bak" -Kpassword -W"REPLACE" -S "<server name>\<instance name>" -T -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'"`
Restore using the Tivoli Storage Manager	`sqllitespeed.exe -R"Database" -D"master" -W"REPLACE -i"filespace\highlevel\lowlevel" -c"nodename" -k"password" -j"c:\program files\Tivoli\TSM\baclient\dsm.opt"`

5. Restart SQL Server and LiteSpeed. If the process hangs, stop the following services and retry them:
    - Alerter
    - 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

NOTE: To restore the master database from a native full backup, refer to msdn.microsoft.com. For example, SQL Server 2012: http://technet.microsoft.com/en-us/library/ms190679(v=sql.110).aspx.

8

Restore Objects

LiteSpeed helps you restore specific objects from a native or LiteSpeed backup file stored locally or in Cloud storage. You can:

- View backup contents and preview table data.
- Query backups. For more information, see Execute SELECT Statements on page 171.
- Restore objects from the backup files.

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

You can also restore objects in the command-line or using extended stored procedures. See the following for more information:

- Restore Objects with the Command-Line Interface
- Recover Objects from Backups using the extended stored procedures

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. During the backup process the .lsm file is created in the temp directory and attached to the backup file after the backup is completed.

Notes:

- You can restore objects directly from the Cloud. It is recommended to use this in cases where there is a fast connection between OLR and the Cloud.
- You cannot restore objects directly from TSM files or tape backups. For more information, see Object Level Restores from TSM Backups on page 170.
- 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, the **Optimize Object Level Recovery speed** option on the Backup wizard Options page is selected by default to create the index during the backup.
- Objects are recovered as they existed at the time they were backed up. You cannot recover data to a random point in time.
- **Direct mode** - In scenarios where you want the application to work with SQL Server directly using a TCP/IP connection without involving the SQL Server client, you can enable direct mode which significantly improves deployment and configuration of your applications. You can enable and disable the use of direct mode from the the Recover Table Wizard.
- **Tail log processing** - In scenarios when you do not require any transaction log backups and the tail log, you can select to bypass tail log processing. Object Level Recovery operations may work much faster in this case. You can enable and disable bypass tail log processing from the toolbar, and when running the Object Level Recovery Wizard and the Recover Table Wizard.

To read the backup files

1. Select the Object Level Recovery pane (CTRL+3).
2. Select **Object Level Recovery Wizard**.
3. On the **Welcome** page click **Next**.
4. On the **Specify Recovery Destination** page, select the server instance.
5. On the **Specify Backup Source**:
   - Select **Database** to restore from a specific database's backup history.
   - Select **Device** and **Disk** to manually select files on disk to restore. Click **Add**. The **Backup File Location** window is displayed. You can also click **Remove** or **Content** to remove the file or view file contents. Selecting **Content** displays a window showing file general properties and file backup sets properties.
   - Select **Device** and **Cloud** to manually select files from the Cloud to restore.
6. On the **Backup File Location** window, locate and select the backup file to read. Once selected, the file path and file name are displayed. Click **OK**. You can browse the network, add, delete, and rename folders.
7. On the **Select Backup Files** page, click **Next**.

   note: LiteSpeed must be installed on the server instance you select on the **Specify Recover Destination** page.

8. On the **Backup Content** page, select a backup to recover and click **Next**.
   - **Point in time restore** - Use the slider to indicate the required point in time. This supports table level recovery.
   - **Bypass tail log processing** - Select this option to bypass tail log processing. Object Level Recovery operations may work faster.

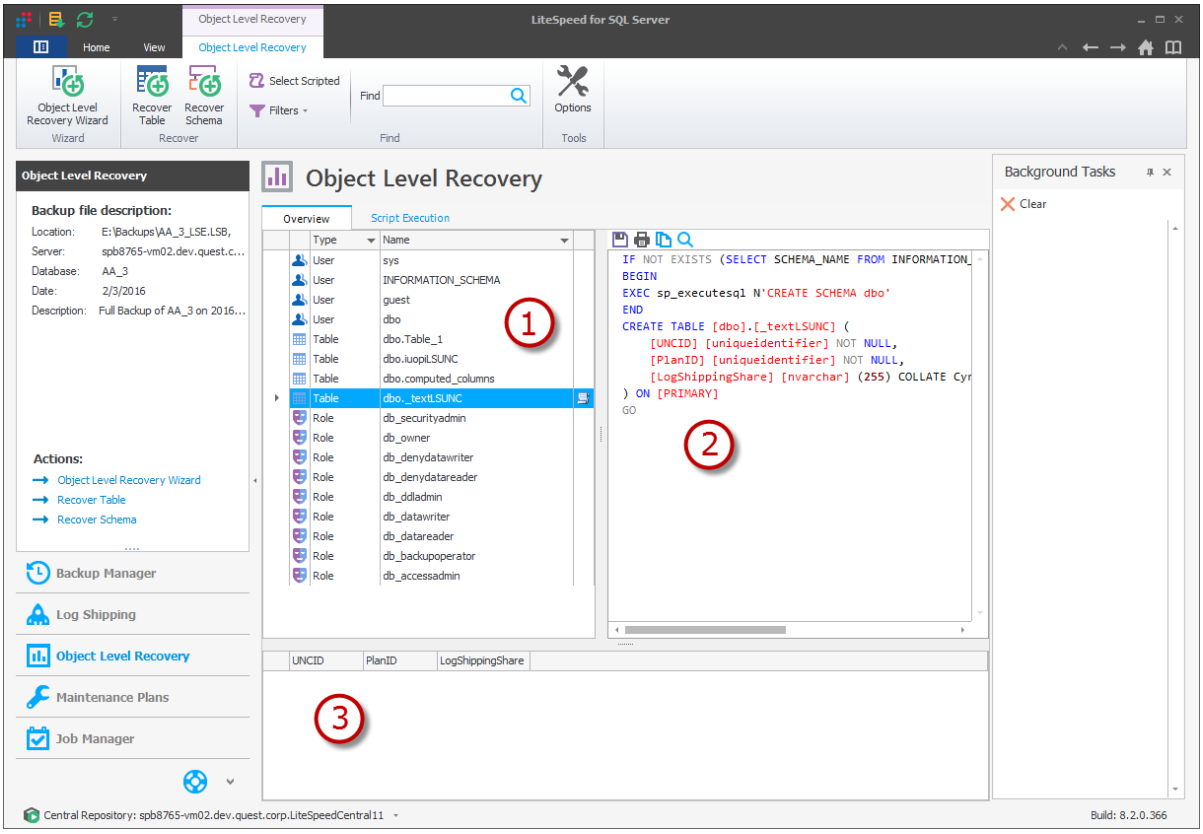
   tip: Only disk backups are supported for OLR.

9. On the **Preview Script** page, view the script that will run to retrieve the backup content. Click **Next**.

10. Complete the wizard.

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 on the Object Level Recovery tab in Options. You can script more than one object, and scripted objects have a small scroll icon beside them in the objects grid. To view all of the scripts in the script preview, click the **Select Scripted** button in the **Object Level Recovery** ribbon menu.

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 a small chart icon beside it in the objects grid. A table that you preview and script has a small chart and scroll icon beside it.

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

- To sort and group the records, right-click a column header and select the appropriate options.
- To sort records against multiple columns, click column headers while holding the SHIFT key. For example, to sort by type and then by name, click the Type column header and then SHIFT+click the Name column header.
- To add or remove columns, right-click a column header and select **Column Chooser**. Add a column by dragging it from the list into the column headers. Remove a column by dragging its column header into the list.
- To move a column, drag the column header to the new location.

Restore Tables and Schemas

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

Tip: CTRL-click objects in the grid to select multiple objects for recovery.

To restore tables

1. Select a table in the objects grid and click **Recover Table**.

2. Complete the wizard. Review the following for additional information:

Database	Select the database. LiteSpeed will not overwrite an existing table. If you select the same server instance and database as the original table, you must use a different table name.
Ship directory	Select the ship directory. Enter the path or click ... to navigate to it. This option stores the object at the directory but does not restore it.
Use table, constraint and index name prefix	Enter the prefix naming convention in the field provided. Select this option when choosing to use prefix naming conventions with tables, constraints, and index names.
Use table, constraint and index name suffix	Enter the suffix naming convention in the field provided. Select this option when choosing to use suffix naming conventions with tables, constraints, and index names.
Bulk insert	Select this option to import data into a table using BCP for data recovery. It requires to use a temporary directory for temporary files. The temporary directory is a Windows temp folder by default or you can specify your custom directory. 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. 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\SQL Server\LiteSpeedSettings.ini*.)
Direct Mode	Select this option to import data directly into a table. This alternate data recovery mechanism eliminates the need for temp file space and uses BULK INSERT operations to recover the data.
Recovery in-memory tables as regular tables	Select to recover or restore memory-optimized tables as regular tables.
Filegroup	Select the filegroup. This option associates the restored object with the filegroup.
Script options	Select these options to generate scripts for table-related objects and constraints:

	• Script constraints except foreign keys - generate scripts for all constraints except for foreign keys • Script indexes - generate scripts for indexes • Script foreign keys - generate scripts for foreign keys • Script triggers - generate script for triggers • Script statistics - generate script for statistics
Advanced options	Bypass tail log processing is not selected by default. Tip: In scenarios when you do not require any transaction log backups and the tail log, you can select to bypass tail log processing. Object Level Recovery operations may work much faster.

To restore schemas

Restore schemas can recover database objects - extended procedures, functions, partition functions, partition schemas, roles, rules, stored procedures, tables, memory optimized tables, triggers, types, users, views, indexed views, and XML schema collections.

1. Click and run the **Object Level Recovery Wizard**.
2. Select an object in the grid and click **Recover Schema**.
3. Complete the wizard. Review the following for additional information:

SQL Server	Select a server instance or click ... to navigate to it.
Database	Select a database. LiteSpeed will not overwrite an existing object.
Preview script	Select to display a preview of the selected script.
OK	Click **OK** to initiate the schema restore.

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. For more information, see Recast LiteSpeed Backups on page 248.
   - Extract a TSM backup to disk as a native uncompressed backup, using the extractor tool. For more information, see Convert LiteSpeed Backups to SQL Server Backups on page 258.

   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

The SQL Server SELECT statement is used to retrieve records from tables in a SQL Server database.

tip: Refer to Restore Objects in the LiteSpeed UI Console for further help with object restore.

To execute a SELECT statement

1. Select the Script Execution tab. The available commands are displayed.

	Click to save script.
	Click to print script.
	Click to copy script.
	Click to find or replace a script.
Execute	Click to execute the script.
Bypass tail log processing	Click to bypass tail log processing.
Save results in Database	Click to save script execution results in a database.

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
[ TOP <expression> ]
<select_list>
FROM <select_source>
[ WHERE <search_condition> ]
[ <offset_fetch> ]

<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 [ NOT ] IN (expression [,...n])
| expression IS [ NOT ] NULL

<expression> ::=
constant
| column

<offset_fetch> ::=
{ OFFSET { integer_constant | offset_row_count_expression } { ROW | ROWS }
```

```
[ FETCH { FIRST | NEXT } {integer_constant | fetch_row_count_expression } { ROW | ROWS
} ONLY ] }
```

Examples

```
SELECT * FROM LiteSpeedActivity OFFSET 0 ROWS FETCH NEXT 10 ROWS ONLY
```

```
SELECT TOP 10 * FROM LiteSpeedActivity WHERE DatabaseID = 6
```

9

View Activity and History

View Backup Manager Activity and History

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 the backup activity and history by selecting the Backup Manager Server Tools Ribbon. 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. Under Database Properties, see the last backup date. You can view information about different dates by changing the **Period** field or select the dates from a calendar.
Databases	Server instance	Lists all of the server instance's databases with their state, recovery model, device, backup destination and last backup date. Print this table or export to Excel. You can right click on any instance and select backup, multi-database backup, restore, automated restore, backup analyzer, display backup/restore jobs, assign categories (available in some cases), refresh, properties, and activity.
LiteSpeed Activity	All	Displays all activity for the selected parameters, including activity type and status, duration, compression ratio, backup throughput (uncompressed backup size to backup duration ratio), Windows domain or SQL Server account that was used to initiate the backup or restore operation, and more. You can change the parameters in the following fields:

Tab Name	Available Level	Description
		• Period–Select the time span, or select dates from a calendar. The options are all, last hour, last 4 hours, last 8 hours, last 24 hours, last 48 hours, last 72 hours, last week, last 2 weeks, and last month. • Report type–Select the type of activity to display. Select as a group: backup, restore, verify, and check database. Individually select: backup, restore, verify, check database, smart cleanup, maintenance plans, DTS packages, LiteSpeed jobs, log shipping plans, shrink database, update statistics, rebuild index, and reorganize index. Select Restore by Source to view restore and verify activity with restore destination for the selected instance/database. • Source–Select the type of backups to display, such as all, LiteSpeed or native SQL Server backups. • Status–View total instance status including all, success, warning, failure, and in progress. You can right click on any instance and select: view in timeline, view details, and re-execute backup. **Tips:** • To re-execute any successful or failed (and fast compression) backup, right-click the activity and select **Re-execute Backup…** You can edit the script if needed and run it immediately as a SQL Agent job or run it in the background. Note that the **Re-execute Backup…** menu item is only visible for LiteSpeed backups, and is enabled only for backups that were executed by using t-sql. If the LiteSpeed backup was executed using the command-line then the menu is disabled. • To verify the backup, right-click the successful backup activity and select **Verify Backup…** You can edit the script if needed and run it immediately as a SQL Agent job or run it in the background. Note that the Verify Backup... menu item is only visible for LiteSpeed backups (Full, Diff, Log), and is enabled only for backups that were executed by using t-sql. If the LiteSpeed backup was executed using the command-line then the menu is disabled. Verify Backup is supported for Disk, Cloud and TSM Backup/Archive backups. • To view information about the native SQL Server backups and restores in the LiteSpeed UI Console, run the Instance Configuration wizard from the start menu and configure the local repository to be updated via a SQL Server job. 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. You can hover over an item for additional information about it. Right-click anywhere in the timeline to navigate to a date, change the time scale, or rotate the name of the

Tab Name	Available Level	Description
		databases.
Backup Templates	All	Displays the LiteSpeed Backup templates stored in the central repository. For more information, see Create Backup Templates on page 87. 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.
Backup Browser	Server instance	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. After expending a directory and making a backup visible, you can right click on the backup and select restore, view content, convert to double click restore backup, and convert to native backup.
LiteSpeed Jobs	Server instance and Database	Lists all scheduled jobs. You can select different job types in the **Jobs filter** field. Right click a job to select the options: change schedule, start, stop, enable jobs, disable jobs, delete, and view in job manager.

Further list, backup template, backup history, and backup analyzer information is provided in the following table.

Tab Name	Available Level	Description
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.
Backup History	Database	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 **Period** field or select the dates from a calendar.
Backup Analyzer	Database	Analyzes different settings, such as compression level, striping, and backup destinations, to determine which settings have the best compression and duration values. For more information, see Test Optimal Backup Settings on page 83.

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. For more information, see Change Server Instance Grouping Methods on page 58.

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

- To sort and group the records, right-click a column header and select the appropriate options.
- To sort records against multiple columns, click column headers while holding the SHIFT key. For example, to sort by type and then by name, click the Type column header and then SHIFT+click the Name column header.
- To add or remove columns, right-click a column header and select **Column Chooser**. Add a column by dragging it from the list into the column headers. Remove a column by dragging its column header into the list.
- To move a column, drag the column header 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	• At a group and instance level–Displays information about the backup volume savings, successful jobs, and failed jobs. • At a maintenance plan level–Displays maintenance plan latest status, name, owner, creation date, and last run date. You can view information about different dates by changing the **Period** field or clicking to select the dates from a calendar.
Maintenance Plans	Lists all maintenance plans for the server instance.
History	Displays execution history of the maintenance plans for the server instance. **NOTE:** To view execution history of every task in a subplan, configure extended logging. Reporting and Logging in Maintenance Plans
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.

10

Use Command-Line Interface

About Using the Command-Line Interface

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

NOTES:

- You must run commands on the server instance on which you want to perform activity.
- Review the Syntax sections to see which arguments are mandatory, which are optional, and which are mutually exclusive. Mutually exclusive arguments are separated by a vertical bar. Optional arguments are enclosed in square brackets. Round brackets are used to group arguments.
- Review the Arguments sections for more information about the arguments and accepted values.

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.

Use…	If you want to…
SQLLiteSpeed	Perform backup/restore tasks. For more information, see LiteSpeed Command-Line Arguments on page 179.
SLSFastCompression	Back up a database using Fast Compression technology. For more information, see Fast Compression Command-Line Arguments on page 205.
SLSSmartCleanup	Delete old backups. For more information, see SmartCleanup Command-Line Arguments on page 222.
SLSRecast	Change backup options for the existing LiteSpeed backups. For more information, see Recast LiteSpeed Backups on page 248.
Extractor	Convert LiteSpeed backups to the native SQL Server backups. For more information, see Convert LiteSpeed Backups to SQL Server Backups on page 258.
OLR	Restore database objects. Restore Objects with the Command-Line Interface
SLSSQLMaint	Perform various database maintenance tasks. For more information, see Script Maintenance Plans Tasks on page 230.
LicenseInfoCmd	View currently installed license or to register a new key. For more information, see LicenseInfoCmd Utility on page 273.

LiteSpeed Command-Line Arguments

The LiteSpeed command-line utility (sqllitespeed.exe or sqllitespeedx32.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 sqllitespeedx32.exe if you have a 32-bit SQL Server on a 64-bit operating system.

- Arguments
- TSM-Specific Arguments
- Cloud-Specific Arguments
- Proxy-Specific Arguments

Syntax

```
sqllitespeed.exe ( -? | <options> )
```

Connection Options:

```
-S <server_name\instance_name>
(-U <login_id> -P <password> ) | -T
```

Backup Options:

```
-B <option>
[--nowrite]
[-D <database_name>]
[-f <file_name>]
[-g <filegroup_name>]
-F <backup_device_name>
[-n <backup_set_name>]
[-d <backup_description>]
[-Y <comment> ]
[-E <mirror_path>]
[-I]
[-N <file_number>]
[-W <options>]
[--AdaptiveCompression (Speed|Size) | -C <compression_level> ]
[-e <encryption_level> (-K <password>|--JobP <key>)]
[-y <date_time> | -r <number_of_days>]
[-s ]
[--doubleclick]
[--OLRMap
[--TempDirectory <path>]]
[--attachedfile <path_or_file>]
[-X <options>]
[-h 1...100]
[-o <buffer_count>]
[-x < maximum_transfer_size>]
[-t <number_of_threads>]
[-p (-1|0|1|2)]
[--LSECompatible ]
[-L (0|1|2) [--Trace logpath = "path"]]
[-V (0|1)]
```

Restore Options:

```
-R <restore_option>
-F <backup_device_name>
-D <destination_database_name>
[-N <file_number>]
[-f <file_name>]
[-g <filegroup_name>]
[--Read_Write_Filegroups]
[-K <password>| --JobP <key>]
[-W STATS = <number>]
[-W PASSWORD = "<media_password>"]
[-A <affinity_mask>]
[-h 1...100]
[-o <buffer_count>]
[-x <maximum_transfer_size>]
[-X <options>]
[--RestoreAsReadOnly
[--RestoreAsCompressed ]]
```

```
[--attachedfile <path_or_file>]
[-L (0|1|2) [--Trace logpath = "path"]]
```

Automated Restore Options:

```
-R Automated
[-D <destination_database_name>
    [--DataFilePath <path> ]
    [--LogFilePath <path> ] ]
-F <backup_filename> | (--BackupPath <path>
    --BackupExtension <extensions>
    --CheckSubfolders (0|1) )
[--BackupType <option> ]
--SourceServer <server_name>
--SourceDatabase <database_name>
[-K <password> | --JobP <key>]
[-W "STATS = <number>"]
[-W "PASSWORD = '<my_password>'"]
[--WithReplace (0|1)]
[-A <affinity_mask>]
[-h 1...100]
[-o <buffer_count>]
[-x <maximum_transfer_size>]
[-X <options>]
[--RestoreAsReadOnly
[--RestoreAsCompressed ]]
[-L (0|1|2) [--Trace logpath = "path"]]
[--DryRun (0|1)]
[--DropDatabaseOnFailure (0|1)]
[--DropDatabaseOnSuccess (0|1)]
```

TSM Connection Options:

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

Tape Arguments:

```
[-m (0|1|2|3)]
[-w]
[-u]
```

Cloud connection options:

```
[--CloudVendor <vendor name>]
[--CloudAccessKey <key name>]
[--CloudAccessKeyEnc <encrypted key name>]
[--CloudSecretKey <key name>]
[--CloudSecretKeyEnc <encrypted key name>]
[--CloudBucketName <bucket name>]
[--CloudRegionName <cloud region name>]
[--CloudGovRegion <government region number>]
[--CloudStorageClass <standard, standard-ia, standard-rrs>]
[--AWSUseServerSideEncryption <1, 0>]
[--AzureBlobType <block, page>]
[--CloudAutoStriping <1, 0>]
[--CloudAutoStripingThreshold <param size in GB>]
[--UseSSL <1, 0>]
```

Proxy connection options:

```
[--ProxyHost <proxy host name>]
[--ProxyLogin <proxy server login credential>]
[--ProxyPassword <proxy server password credential>]
[--ProxyPasswordEnc <encrypted proxy server password credential>]
[--ProxyPort <proxy server port number>]
```

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 or Disk IO. For more information, see Compression Methods on page 121. You can tell Adaptive Compression to optimize backups either for size or for speed. This argument accepts one of the following values: • Size • Speed
-A	--Affinity	Processor affinity designates specific processors to run LiteSpeed, while not allowing LiteSpeed to run

-Argument	--Argument	Description
		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. For more information, see Adaptive Compression on page 122.
	--ARPeriod	Specifies a point in time to restore from where the time is measured in days, hours, minutes and seconds from the restore time. Set 0's for periods not used. @ARPeriod = N'DD.HH:MM:SS'
	--ARPointInTime	Specifies a point in time to restore from: year, month,

-Argument	--Argument	Description
		day, hours, minutes, seconds. @ARPointInTime = N'YYYY-MM-DD HH:MM:SS'
(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 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. You can supply multiple instances of this argument.
-B	--Backup	Backup operation. This argument accepts one of the following values: • Database–Back up database • Log–Back up transaction log
(none)	--BackupExtension	When looking for database backups, LiteSpeed will only consider backup files that have the extensions you specify. The value of this parameter is a list of extensions, separated with commas. No value or asterisk (*) specifies any file extension.
-d	--BackupDescription	Specifies a description to store with the backup. This argument accepts variables. For more information, see LiteSpeed Variables on page 125.

-Argument	--Argument	Description
-F	--BackupFiles	Specifies a backup location (e.g. C:\backups\AdventureWorks.bak). This argument accepts network destinations. You can supply multiple instances of this argument to use stripe backups. Examples: UNC Path: \\servername\share\path\filename Local path: c:\filedirectory\filename For TSM backups and TSM archives, this argument accepts the following formats: • tsmbkp:<filespace>\<high>\<low> • tsmarc:<filespace>\<high>\<low> This argument accepts variables. For more information, see LiteSpeed Variables on page 125.
-N	--BackupIndex	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.
-n	--BackupName	Specifies the name of the backup set. This argument accepts variables. For more information, see LiteSpeed Variables on page 125.
(none)	--BackupPath	Specifies the directory where to search for the backup files. You can supply multiple instances of this argument. Each instance of this parameter must be followed by --BackupExtension and --CheckSubfolders arguments.
(none)	--BackupType	Specifies backup types to use for the restore. This argument accepts one of the following values: • full–LiteSpeed will only restore the most recent full database backup. • diff–LiteSpeed will restore the most recent full database backup and any existing differential backups based on this full. • tlog–LiteSpeed will restore the most recent full database backup and any existing differential and/or transaction log backups created after the most recent full backup.

-Argument	--Argument	Description
-o	--BufferCount	Specifies the number of SQL Server buffers available for a LiteSpeed operation. The default value is set by SQL Server. **NOTE:** LiteSpeed defaults typically result in the best performance. You should only modify advanced options after careful planning and testing.
(none)	--CheckDB	Specifies checking the database integrity after running restores. Options include: --CheckDBPhysicalOnly, --CheckDBDataPurity, --CheckDBExtendedLogicalChecks, --CheckDBNoIndex, --CheckDBTableLocks, and --CheckDBNoInfoMessages.
(none)	--CheckDBPhysicalOnly	Specifies the checking database physical structure option. Note: Using this argument can significantly decrease the execution time when using large databases.
(none)	--CheckDBDataPurity	Specifies the checking database column values option for validity or out of range.
(none)	--CheckDBExtendedLogicalChecks	Specifies the performing consistency checks on indexes option. It checks XML indexes and spacial indexes.
(none)	--CheckDBNoIndex	Specifies the performing intensive checks of non-clustered indexes for user tables option.
(none)	--CheckDBTableLocks	Specifies the using table locks instead of using an internal database snapshot option.
(none)	--CheckDBNoInfoMsgs	Specifies the including informational messages in the notification report option.
(none)	--CheckSubfolders	Specifies whether to use subfolders to look for database backups. This argument accepts one of the following values: • 0–False. LiteSpeed will only use backups located in the specified folder. • 1–True. LiteSpeed will look for backups in the specified folder and in its subfolders.
-Y	--Comment	Appends a user comment to the backup. This argument accepts variables. For more information, see LiteSpeed Variables on page 125.
-C	--CompressionLevel	Specifies the compression level for the backup. Valid values are 0 through 8. 0 bypasses the

-Argument	--Argument	Description
		compression routines. The remaining values of 1 through 8 specify compression with increasingly aggressive computation. *2* is the default value for disk backups and *7* is the default value for cloud backups. 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. For more information, see Test Optimal Backup Settings on page 83. **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.
-D	--Database	Name of database to be backed up or restored.
(none)	--DataFilePath	Specifies a location for data files.
(none)	--DisconnectUsers	Disconnect users on executing restore (in standby mode only). This argument accepts one of the following values: • 0–Do not disconnect users (default). • 1–Disconnect users.
(none)	--DontUseCopyOnly	Specifies that LiteSpeed is not to use copy-only backups when running restores.
(none)	--DontUseReplication	Specifies that LiteSpeed is not to include databases that are part of a replication plan when running restores.
-J	--DoubleClick	Creates a Double Click Restore executable. This argument accepts one of the following values: • 1–Creates one Double-Click Restore executable file. Note the following warning: The executable may be greater than 4GB for large databases. Windows Server is unable to run executable files larger than 4GB. However, the file will be convertible/restorable by LiteSpeed file. • 2–Creates a Double Click Restore loader in the same location. (Default) For more information, see Double Click Restore Executables on page 120.

-Argument	--Argument	Description
(none)	--DropDatabaseOnFailure	Drops the restored database only if the restore fails. Use this option if you no longer need the restored database. For example, if you are only restoring the latest backup for testing purposes. This option contains two additional options to select. One or both options can be selected.On success restore and check database integrity operations - The database is dropped after a successful restore and database integrity check.On failure any of restore or check databases integrity operations - The database is dropped after failing the restore or database integrity check. This argument accepts one of the following values: • 0–False (default) • 1–True
(none)	--DropDatabaseOnSuccess	Drops database on success only. Use this option if you no longer need the restored database. For example, if you are only restoring the latest backup for testing purposes. This option contains two additional options to select. One or both options can be selected.On success restore and check database integrity operations - The database is dropped after a successful restore and database integrity check.On failure any of restore or check databases integrity operations - The database is dropped after failing the restore or database integrity check. This argument accepts one of the following values: • 0–False (default) • 1–True
(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	Specifies encryption level. Works in conjunction with the Key (K) parameter. This argument accepts one of the following values: • 0–40-bit RC2 • 1–56 bit RC2 • 2–112 bit RC2 • 3–128 bit RC2

-Argument	--Argument	Description
		• 4–168 bit 3DES • 5–128 bit RC4 • 6–128 bit AES • 7–192 bit AES • 8–256 bit AES • 9–MS_AES_128 • 10–MS_AES_192 • 11–MS_AES_256
(none)	--ExcludeDatabase	Name of database(s) to exclude from this backup / restore. Example: --ExcludeDatabase Northwind Tip: The @ExcludeDatabase argument can be applied together with @MultiDatabaseType to exclude several databases from the process.
-y	--Expiration	Specifies the date and time when the backup expires. LiteSpeed will not overwrite this file until expiration datetime is passed. This argument accepts one of the following formats: • yyyy-mm-dd • yyyy-mm-dd hh:mm:ss
-f	--File	Specifies a logical database file used for file or filegroup backups. You can supply multiple instances of this argument.
-g	--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.
-X	--IOFlags	Specifies if LiteSpeed should wait and retry the read or write operation on failure. You can define retry options using the following parameters:

-Argument	--Argument	Description
		• 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 and cloud operations. For more information, see Network Resilience on page 124.
(none)	--JobP	Specifies an encrypted key. (Similar to -K). **NOTE:** Automated Restore requires that you use the same password for all encrypted backups.
-K	--Key	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. If you use the wrong encryption key, the restore will fail. **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' **NOTE:** Automated Restore requires that you use the same password for all encrypted backups.
(none)	--LogFilePath	Specifies a location for log files.
-L	--LogLevel	Creates a log file. 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 (or C:\ProgramData\Quest Software\LiteSpeed\SQL*

-Argument	--Argument	Description
		Server\Logs) (or *C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs*). To log to a different directory run this utility with the following argument: `--trace logpath = "path"`. For more information, see Configure Logging in LiteSpeed on page 576.
(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 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 (default) • 1–True
-x	--MaxTransferSize	Specifies the largest unit of transfer in bytes to be used between SQL Server and LiteSpeed. The possible values are multiples of 65536 bytes (64 KB) ranging up to 4,194,304 bytes (4 MB). The default is 1048576 (1 MB).
-E	--MirrorFiles	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. For more information, see LiteSpeed Variables on page 125.
(none)	--MultiDatabaseType	Produces a backup that includes several types of databases. Types can include: all, system, user, or selected databases. This argument accepts one of the following values: • All - Backup all system and user databases. • System - Backup only system databases. • User - Backup only user databases. • Selected - Backup specifically selected databases.

-Argument	--Argument	Description
-Z	--NoWrite	The argument is similar to backup log xxx to disk = 'NUL'. When the backup is completed, it is not written to disk. NOTES: • 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. • If compression or encryption parameters are specified, then the data will get compressed or encrypted before being thrown away.
-M	--OLRMap	Generates a map file during a backup for Object Level Recovery. This argument accepts one of the following values: • 0–False (default) • 1–True
-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	Specifies the user password. Passwords are case-sensitive. Required if the connection type is not a trusted connection.
-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: • -1–Below Normal • 0–Normal (Default) • 1–AboveNormal • 2–High
(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.

-Argument	--Argument	Description
-R	--Restore	Restore operation.This argument accepts one of the following values: • AttachedFilesOnly –Restore attached files without restoring the database. • Database–Restore database backup. • Log–Restore log backup. • VerifyOnly–Verify backup. • HeaderOnly–Provide backup details. • FileListOnly–Provide database file details. • CheckPassword–Check password/key. • CheckSumOnly–Checksum a backup file. • AttachedFileNamesOnly–List names of all attached files. • Automated–Restore the most recent full backup and optionally differential and transaction log backups.
(none)	--RestoreAsCompressed	Works in conjunction with --RestoreAsReadOnly, creates a folder if it does not exist, and then compresses it. This argument accepts one of the following values: • 0–False (default) • 1–True
(none)	--RestoreAsReadOnly	Instructs the restore operation to leave the database in read-only mode. This argument accepts one of the following values: • 0–False (default) • 1–True Using this option, you can restore a user database into an NTFS compressed folder or restore a tlog to a read-only database in a compressed folder. **NOTES:** • When using an NTFS-compressed folder for a database, it can only be restored as read-only.

-Argument	--Argument	Description
		• You can only use this feature on Windows NTFS file systems.
-S	--Server	Specifies the instance of Microsoft SQL Server to connect to. This argument accepts one of the following values: • server_name • server_name\instance_name If no server is specified, the LiteSpeed command-line utility will connect to the default instance of SQL Server on the local computer.
-?	--ShowHelp	Displays the syntax summary of the LiteSpeed command-line utility.
-s	--Skip	Skips normal retention checks and overwrites the backup that has not expired. • 0–False (default) • 1–True
(none)	--SourceDatabase	Backups of this database are the source for restore.
(none)	--SourceServer	Backups created on this instance of SQL Server are the source for restore.
-m	--TapeFormat	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–Write new header • 2–Long erase / write new header • 3–Low level controller format / write new header **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.
-w	--TapeRewind	Applies only to backing up and restoring tape. This argument accepts one of the following values:

-Argument	--Argument	Description
		• 0–Leave the tape unwound (default) • 1–Rewind the tape after writing/reading
-u	--TapeUnload	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
(none)	--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\SQL Server\LiteSpeedSettings.ini*.)
-t	--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.
-h	--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. For more information, see Adaptive Compression on page 122.

-Argument	--Argument	Description
	--UDT	Create table script: • 0–Off. Create table with native types, if possible; otherwise (CLR UDT) create with UDT. (Default). • 1–On. Create table with UDT.
-U	--UserId	Specifies user login ID. Required if the connection type is not a trusted connection. Login IDs are case-sensitive.
-V	--Verify	Performs a restore verification on the backup file just created (if backup was successful). • 0–False (default) • 1–True
-T	--WindowsAuth	Uses a trusted connection (to the server) instead of requiring a password.
-W	--With	Specifies strings that will be passed directly to SQL Server. You can supply multiple instances of this argument. Some of the accepted parameters are the following: • 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. • CHECKSUM–Causes checksums to be verified when a LiteSpeed backup is created. **NOTE:** 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. • COPY_ONLY–Specifies the copy-only backup.

-Argument	--Argument	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). • MOVE–Specifies that the given logical_file_name should be moved to operating_system_file_name. • REPLACE–Instructs LiteSpeed to create the specified database and its related files even if another database already exists with the same name. The existing database is deleted. • RECOVERY–Instructs the restore operation to roll back any uncommitted transactions. After the recovery process, the database is ready for use. • NORECOVERY–Instructs the restore operation to not roll back any uncommitted transactions. • NO_TRUNCATE–Allows backing up the log in situations where the database is damaged. • 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. • STATS–Displays a message each time a percentage of the activity completes. The default is 10%. • 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.
(none)	--WithReplace	Instructs LiteSpeed to create the specified database and its related files even if another database already exists with the same name. The existing database is deleted. This argument accepts one of the following values:

-Argument	--Argument	Description
		• 0–False (default) • 1–True

TSM-Specific Arguments

TSM-specific arguments work in conjunction with the LiteSpeed arguments. See Syntax and Examples for more information.

-Argument	--Argument	Description
(none)	--TSMAdminName	Specifies the TSM administrative user name that has client authority for the TSM node. Some operations may require an administrative user with client owner authority to be specified in order to open a TSM session. The correct username and password may be obtained from the TSM administrator.
(none)	--TSMAdminPwd	Specifies the plain text password of the administrative user which is used to log in to the TSM server and start the TSM session.
(none)	--TSMArchive	Specifies to store the backup as a TSM archive. This argument accepts one of the following values: • 0–False (default) • 1–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.
(none)	--TSMDeviceTimeoutMinutes	Specifies how long to wait for a TSM device.
(none)	--TSMDSMI_Dir	DSMI_DIR path if needed.
(none)	--TSMDSMI_Log	DSMI_LOG path.
-j	--TSMConfigFile	Specifies the TSM configuration file.
-i	--TSMFile	Defines the TSM filespace, high level and low level. This argument accepts the following format: `tsm_filespace\tsm_high_level\tsm_low_level`

-Argument	--Argument	Description
		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.
-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. **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.
(none)	--TSMLogname	Log name.
-z	--TSMMgmtClass	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. **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.

Cloud-Specific Arguments

Cloud-specific arguments work in conjunction with the LiteSpeed arguments. See Syntax and Examples for more information.

-Argument	--Argument	Description
(none)	--AWSUseServerSideEncryption	The @AWSUseServerSideEncryption argument enables the encryption of data stored at rest in Amazon S3. This argument accepts one of the following values:

-Argument	--Argument	Description
		• 0–Do not use Server Side Encryption • 1–Use Server Side Encryption
(none)	--AzureBlobType	The @AzureBlobType argument specifies the types of blobs that can be stored in the Microsoft Azure cloud storage. This argument accepts one of the following values: "Block", "Page". note: The LiteSpeed auto striping logic used in the @CloudAutoStriping and @CloudAutoStripingThreshold parameters can override the Azure blob limit for LiteSpeed backups.
(none)	--CloudAccessKey	The @CloudAccessKey argument specifies the name of the unique Cloud Web Service alphanumeric access key that identifies each user. The selections include Amazon Access Key, Azure Account Name, Google e-mail styled account.
(none)	--CloudAccessKeyEnc	The @CloudAccessKeyEnc argument specifies the name of the encrypted unique Cloud Web Service alphanumeric access key that identifies each user.
(none)	--CloudAutoStriping	This parameter enables automatic file striping for LiteSpeed cloud backups.
(none)	--CloudAutoStripingThreshold	This parameter contains the stripe size in GBs. LiteSpeed logic uses the database size to make a decision about the number of stripes needed for LiteSpeed cloud backups. For example, if you have a database with a size of 200GB and set @CloudAutoStripingThreshold = 50, then LiteSpeed uses 200/50 = 4 stripes.
(none)	--CloudBucketName	The @CloudBucketName argument specifies the name of the container for cloud objects. Bucket names must be at least 3 and no more than 63 characters long. The selections are Amazon Bucket Name, Azure Container Name, Google Bucket Name. Google Bucket Name requirements are described at https://cloud.google.com/storage/docs/naming.
(none)	--CloudGovRegion	The @CloudGovRegion argument enables a special restricted region for the US Government use in Amazon S3 and Azure Clouds. This argument accepts one of the following values: • 0–Do not use government cloud (default) • 1–Use government cloud

-Argument	--Argument	Description
(none)	--CloudRegionName	The @CloudRegionName argument specifies the name of the Cloud Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-east-2, us-west-1, us-west-2, ca-central-1, eu-central-1, eu-west-1, eu-west-2, ap-south-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, ap-northeast-2, sa-east-1, N'Germany' and N'China'.
(none)	--CloudSecretKey	The @CloudSecretKey argument specifies the name of the Cloud Web Service secret key that is assigned when you initially get a Cloud account.
(none)	--CloudSecretKeyEnc	The @CloudSecretKeyEnc argument specifies the name of the encrypted Cloud Web Service secret key that is assigned when you initially get a Cloud account.
(none)	--CloudStorageClass	The @CloudStorageClass argument specifies a range of storage classes established for different use cases including: For Amazon S3: • Standard: Standard storage - for general-purpose storage of frequently accessed data. • Standard-IA: Standard Infrequent Access - for long-lived, but less frequently accessed data. • RRS: Reduced Redundancy Storage - for non-critical data considering lower level of redundancy rather than Standard storage. Important: : In versions less than 8.5 you should use --AWSStorageClass. The @AWSStorageClass argument is no longer valid in subsequent LiteSpeed versions after 8.5. For Google Storage: • Multi_regional - for frequently accessed data around the world as per serving website content, streaming videos, or gaming and mobile applications. • Regional - for frequently accessed data in the same region as your Google Cloud DataProc or the Google Compute Engine instances that use it, as per data analytics. • Nearline - for infrequently accessed data (data you expect to access no more than once per month).

-Argument	--Argument	Description
		• Coldline - for infrequently accessed data (data you expect to access no more than once per year).
(none)	--CloudVendor	The @CloudVendor argument specifies the name of the cloud service provider. The argument accepts one of the following values: "AmazonS3", "AzureBlob" or "GoogleStorage".
(none)	--GSProject	DEPRECATED LiteSpeed 8.8: Was used to store for the Google Cloud Storage project ID; the project ID is now obtained from login. This parameter is retained for compatibility with old backup/restore scripts.
(none)	--UseSSL	The @UseSSL argument specifies that the connection uses SSL security. This argument accepts one of the following values: • 0–Do not use SSL • 1–Use SSL (default)

Proxy-Specific Arguments

Proxy-specific arguments work in conjunction with the LiteSpeed arguments. See Syntax and Examples for more information.

-Argument	--Argument	Description
(none)	--ProxyHost	The @ProxyHost argument is optional and specifies the name of the proxy host name that is running the proxy server. note: If the @ProxyHost argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.
(none)	--ProxyLogin	The @ProxyLogin argument is optional and specifies the proxy server login credential. note: If not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.
(none)	--ProxyPassword	The @ProxyPassword argument is optional and

-Argument	--Argument	Description
		specifies the proxy server password credential. note: If the @ProxyPassword argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.
(none)	--ProxyPasswordEnc	The @ProxyPasswordEnc argument is optional and specifies the encrypted proxy server password credential. note: If the @ProxyPasswordEnc argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.
(none)	--ProxyPort	The @ProxyPort argument is optional and contains the port number of the proxy server. The TCP/IP port values can be 1-65535. note: If the @ProxyPort argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

Examples

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

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

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

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

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

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

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

   ```
   sqllitespeed.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"***"
   ```

5. Create a differential backup of the Northwind database:

```
sqllitespeed.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"
```

6. Restore the Northwind database:

```
sqllitespeed.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"***"
```

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

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

8. Restore the most recent full and drop database:

```
sqllitespeed.exe -R Automated -D LiteSpeedLocal_TestRestore --DataFilePath
"D:\DATA" --LogFilePath "D:\DATA" --DropDatabase 3 --BackupPath "D:\temp" --
BackupExtension "" --CheckSubfolders 0 --SourceServer LITESPEED\SQL2005 --
SourceDatabase LiteSpeedLocal --BackupType full -K ****** --WithReplace 1 --
DropDatabaseOnFailure 1 --DropDatabaseOnSuccess 1 -S"LITESPEED\SQL2005" -T
```

9. Restore the most recent full backup and related differential and transaction log backups following this full to a new database.

```
sqllitespeed.exe -R Automated --DataFilePath "D:\DATA" --LogFilePath "D:\DATA" -
-BackupPath "D:\temp" --BackupExtension "" --CheckSubfolders 1 -D
LiteSpeedLocal_Copy --SourceServer LITESPEED\SQL2005 --SourceDatabase
LiteSpeedLocal --BackupType tlog -K ****** -S"LITESPEED\SQL2005" -T
```

10. Restore the most recent Fast Compression backups.

```
sqlLiteSpeed.exe -R Automated -D "LiteSpeedLocal_AutomatedRestore" --
BackupPath "D:\temp\FC\" --BackupExtension "" --CheckSubfolders 0 -K
"*******" --SourceServer "LITESPEED\SQL2005" --SourceDatabase "LiteSpeedLocal"
--BackupType "diff" --JobP "5jzOEztgLxQ=" --WithReplace 1 -S
"LITESPEED\SQL2005" -U "sa" -P "******"
```

11. Restore the most recent striped backup.

```
sqllitespeed.exe -R Automated -D "NEWDB" --DataFilePath "D:\DATA" --LogFilePath
"D:\DATA" --BackupPath "D:\temp" --BackupExtension "stripe1" --CheckSubfolders 0
--BackupPath "E:\temp" --BackupExtension "stripe2" --CheckSubfolders 0 --
SourceServer "LITESPEED\SQL2005" --SourceDatabase "FOX" --BackupType "full" -S
"LITESPEED\SQL2005" -T
```

12. View candidates for Automated Restore.

```
sqllitespeed.exe -R Automated --BackupPath d:\temp\TEST --BackupExtension
"bak,bkp" --CheckSubfolders 1 --SourceServer LITESPEED\SQL2005 --SourceDatabase
"LiteSpeedLocal" --BackupType "diff" --DryRun "1" -S "LITESPEED\SQL2005" -T
```

13. Restore attached files only:

```
sqllitespeed.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:\CProg.txt'" -S "spb9771" -T
```

14. Encrypt a password for the restore operations.

```
sqllitespeed.exe --EncRestoreKey -K "RF^t%7MF"
```

15. Restore Objects using the UDT argument. Create table with native types, if possible; otherwise (CLR UDT) create with UDT.

```
olr.exe -F "C:\temp\FOX_full.bak" -K****** -N3 -C -Y Database -Q d:\temp\create_
database_FOX.sql --UDT 0
```

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.

- Accepted LiteSpeed Arguments
- Accepted TSM Command-Line Arguments
- Cloud-Specific Arguments
- Proxy-Specific Arguments

Syntax

```
slsFastCompression.exe ( -? | --ShowHelp | --SDShowSyntax |
<FastCompression_backup_options> <Connection_options> |
<FastCompression_verify_option> <Connection_options> |
<FastCompression_backup_options> <FastCompression_verify_option> <Connection_
options> )
```

FastCompression backup options:

```
--SDBackupDirectory <path>
[--SDForceFull |--SDForceDifferential ]
--SDExtentsChgRatioRequireFull <value> |--SDDiffToFullRatioRequireFull <value>
[--SDCheckForFullBackup]
```

```
[--SDFullBackupEscalation]
[--SDElapsedDaysRequireFull <n>]
[--SDSpecificDaysForbidFull (<day>,...)]
[--AltDir <path>]
[--SDMirrorDirectory <path>...]
[--SDAppendDifferential]     ]

[-F --FastCompressionExtension]
```

FastCompression verify option:

```
[--SDVerify (Last|Full,Last|All)]
```

Accepted LiteSpeed arguments:
[-t <number_of_threads>]
[-X <options>]
[-A <affinity_mask>]
[-L (0|1|2)]
[-W <with_arguments>]
[-n < backup_name>]
[-d <backup_description>]
[-p <priority_level>]
[--AdaptiveCompression (Speed|Size) | -C < compression_level>]
[-e <encryption_level> (-K <encryption_key>| --JobP <encrypted_key>)]
[-Y <comment>]
[--attachedfile <path_or_file>]
[-h 1...100]
[-o <buffer_count>]
[-x <maximum_transfer_size>]
[-M --OLRMap]
[--MultiDatabaseType <all, system, user, selected>]

Connection options:

```
-D <database_name>
-S <server_name>
(-U <username> -P <password>) |-T]
```

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>]
]
```

Cloud connection options:

```
[--CloudVendor <vendor name>]
[--CloudAccessKey <key name>]

[--CloudAccessKeyEnc <encrypted key name>]

[--CloudAutoStriping <1 or 0>]

[--CloudAutoStripingThreshold <auto, number GB>]

[--CloudSecretKey <key name>]

[--CloudSecretKeyEnc <encrypted key name>]

[--CloudBucketName <bucket name>]

[--CloudRegionName <cloud region name>]

[--CloudGovRegion <government region number>]
```

Proxy connection options:

```
[--ProxyHost <proxy host name>]

[--ProxyLogin <proxy server login credential>]

[--ProxyPassword <proxy server password credential>]

[--ProxyPasswordEnc <encrypted proxy server password credential>]

[--ProxyPort <proxy server port number>]
```

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
--AlterDir	Specifies the directory where to search for the backup file. **Note**: AlterDIr replaced SDSearchAlternateBackupDirectory in LiteSpeed 8.6. Support for the old SDSearchAlternateBackupDirectory parameter will be gradually phased out.
--SDAppendDifferential	Appends data to an existing full backup file.

Argument	Description
--SDBackupDirectory	Specifies a directory for the backup file.
--SDCheckForFullBackup	Checks if the expected full backup exists when backing up to separate files and returns a failure message if it is not found. If the full backup file does not exist, it performs a full backup. If the full backup file does exist, the decision to perform a new full or a differential backup will depend on other conditions specified.
--SDDiffToFullRatioRequireFull	Specifies the last differential backup size to last full backup size ratio. When exceeding the specified ratio LiteSpeed performs a full backup. This argument accepts one of the following formats: • ".4" • "40%"
--SDElapsedDaysRequireFull	Specifies the minimum number of days since last full backup required to perform full backup. The default value is *14*.
--SDExtentsChgRatioRequireFull	Specifies the minimum amount of database changes required for the full backup. This argument accepts one of the following formats: • ".4" • "40%"
--SDForceDifferential	Forces differential backup.
--SDForceFull	Forces full backup.
--SDFullBackupEscalation	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.

Argument	Description
	• 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 (default) • 1–True
--SDMirrorDirectory	Specifies a directory for a mirror backup. You can supply multiple instances of this argument.
--SDShowSyntax	Displays the slsFastCompression syntax.
--SDSpecificDaysForbidFull	Specifies days of the week when a full backup is never performed. This argument accepts one of the following formats: • 3–on Tuesday • "tu"–on Tuesday • "5-7"–from Thursday to Saturday • "m, w, su"–on Monday, Wednesday, and Sunday
--SDVerify	Performs a restore verification on the backup file just created (if backup was successful). This argument accepts one of the following values: • Last–Verifies last backup performed (can be either a full or differential)

Argument	Description
	• Full, Last–Verifies the last full backup and last differential is available • All–Verifies last full backup and all differentials since

Accepted LiteSpeed Arguments

The following parameters work in conjunction with the Fast Compression parameters. See Syntax and Examples for more information.

-Argument	--Argument	Description
(none)	--AdaptiveCompression	Automatically selects the optimal compression level based on CPU usage or Disk IO. For more information, see Compression Methods on page 121. You can tell Adaptive Compression to optimize backups either for size or for speed. This argument accepts one of the following values: • Size • Speed
-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 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:

-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> **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. For more information, see Adaptive Compression on page 122. For more information, see Configure LiteSpeed Defaults on page 60.
(none)	--AttachedFile	Specifies a file or directory (and all nested files and directories) to attach with this backup operation. You can supply multiple instances of this argument..
-d	--BackupDescription	Specifies a description to store with the backup. This argument accepts variables. For more information, see LiteSpeed Variables on page 125.
-n	--BackupName	Specifies the name of the backup set. This argument accepts variables. For more information, see LiteSpeed Variables on page 125.
-o	--BufferCount	Specifies the number of SQL Server buffers available for a LiteSpeed operation. The default value is set by SQL Server. **NOTE:** 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. For more information, see LiteSpeed Variables on page 125.

-Argument	--Argument	Description
-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 for disk backups and *7* is the default value for cloud backups. 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. For more information, see Test Optimal Backup Settings on page 83. **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.
-D	--Database	Name of database to be backed up or restored.
-e	--EncryptionLevel	Specifies encryption level. Works in conjunction with the Key (K) parameter. 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 • 9–MS_AES_128 • 10–MS_AES_192 • 11–MS_AES_256
-y	--Expiration	Specifies the date and time when the backup expires. LiteSpeed will not overwrite this file until expiration datetime is passed. This argument accepts one of the following formats: • yyyy-mm-dd

-Argument	--Argument	Description
		• yyyy-mm-dd hh:mm:ss
(none)	--FastCompressionExtension	Specifies the fast compression file extension. This argument accepts one of the following formats: • bak - the default for new items. • bkp - for an existing item that does not have an extension defined.
-f	--File	Specifies a logical database file used for file or filegroup backups. You can supply multiple instances of this argument.
-g	--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.
-X	--loflag	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 and cloud operations.
-K	--Key	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. If you use the wrong encryption key, the restore will fail. **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'
-L	--LogLevel	Creates a log file. This argument accepts one of the following values: • 0–Logging off.

-Argument	--Argument	Description
		• 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 (or C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs)* (or *C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs*). To log to a different directory run this utility with the following argument: `--trace logpath = "path"`. For more information, see Configure Logging in LiteSpeed on page 576.
-x	--MaxTransferSize	Specifies the largest unit of transfer in bytes to be used between SQL Server and LiteSpeed. The possible values are multiples of 65536 bytes (64 KB) ranging up to 4,194,304 bytes (4 MB). The default is 1048576 (1 MB).
(none)	--MultiDatabaseType	Produces a backup that includes several types of databases. Types can include: all, system, user, or selected databases. This argument accepts one of the following values: • All - Backup all system and user databases. • System - Backup only system databases. • User - Backup only user databases. • Selected - Backup specifically selected databases.
-M	--OLRMap	Generates a map file during a backup for Object Level Recovery. This argument accepts one of the following values: • 0–False (default) • 1–True
(none)	--Password	Specifies the user password. Passwords are case-sensitive. Required if the connection type is not a trusted connection.
-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: • -1–Below Normal • 0–Normal (Default)

-Argument	--Argument	Description
		• 1–AboveNormal • 2–High
(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.
(none)	--SDDryRun	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.
-S	--Server	Specifies the instance of Microsoft SQL Server to connect to. This argument accepts one of the following values: • server_name • server_name\instance_name If no server is specified, the LiteSpeed command-line utility will connect to the default instance of SQL Server on the local computer.
-?	--ShowHelp	Displays the syntax summary of the LiteSpeed command-line utility.
-t	--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.
-h	--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. For more information, see Adaptive Compression on page 122.
-U	--UserId	Specifies user login ID. Required if the connection type is

-Argument	--Argument	Description
		not a trusted connection. Login IDs are case-sensitive.
(none)	--Verify 1	Performs a restore verification on the backup file just created (if backup was successful).
-W	--With	Specifies strings that will be passed directly to SQL Server in the backup/restore SQL. Some of the accepted parameters are the following: • CHECKSUM–Causes checksums to be verified when a LiteSpeed backup is created. NOTE: 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. • STATS–Displays a message each time a percentage of the activity completes. The default is 10%.

Accepted TSM Command-Line Arguments

The following arguments work in conjunction with the Fast Compression arguments and accepted LiteSpeed arguments. See Syntax and Examples 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 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.
(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

-Argument	--Argument	Description
		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.
(none)	--TSMLogname	Log name.
-z	--TSMMgmtClass	Specifies the TSM management class. If not specified, LiteSpeed uses the default management class.

Cloud-Specific Arguments

Cloud arguments work in conjunction with the LiteSpeed arguments. See Syntax and Examples for more information.

-Argument	--Argument	Description
(none)	--AWSUseServerSideEncryption	The @AWSUseServerSideEncryption argument enables the encryption of data stored at rest in Amazon S3. This argument accepts one of the following values: • 0–Do not use Server Side Encryption • 1–Use Server Side Encryption
(none)	--AzureBlobType	The @AzureBlobType argument specifies the types of blobs that can be stored in the Microsoft Azure cloud storage. This argument accepts one of the following values: "Block", "Page". note: The LiteSpeed auto striping logic used in the @CloudAutoStriping and @CloudAutoStripingThreshold parameters can override the Azure blob limit for LiteSpeed backups.
(none)	--CloudAccessKey	The @CloudAccessKey argument specifies the name of the unique Cloud Web Service alphanumeric access key that identifies each user. The selections include Amazon Access Key, Azure Account Name, Google e-mail styled account.
(none)	--CloudAccessKeyEnc	The @CloudAccessKeyEnc argument specifies the name of the encrypted unique Cloud Web Service alphanumeric access key that identifies each user.

-Argument	--Argument	Description
(none)	--CloudAutoStriping	This parameter enables automatic file striping for LiteSpeed cloud backups.
(none)	--CloudAutoStripingThreshold	This parameter contains the stripe size in GBs. LiteSpeed logic uses the database size to make a decision about the number of stripes needed for LiteSpeed cloud backups. For example, if you have a database with a size of 200GB and set @CloudAutoStripingThreshold = 50, then LiteSpeed uses 200/50 = 4 stripes.
(none)	--CloudBucketName	The @CloudBucketName argument specifies the name of the container for cloud objects. Bucket names must be at least 3 and no more than 63 characters long. The selections are Amazon Bucket Name, Azure Container Name, Google Bucket Name. Google Bucket Name requirements are described at https://cloud.google.com/storage/docs/naming.
(none)	--CloudGovRegion	The @CloudGovRegion argument enables a special restricted region for the US Government use in Amazon S3 and Azure Clouds. This argument accepts one of the following values: • 0–Do not use government cloud (default) • 1–Use government cloud
(none)	--CloudRegionName	The @CloudRegionName argument specifies the name of the Cloud Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-east-2, us-west-1, us-west-2, ca-central-1, eu-central-1, eu-west-1, eu-west-2, ap-south-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, ap-northeast-2, sa-east-1, N'Germany' and N'China'.
(none)	--CloudSecretKey	The @CloudSecretKey argument specifies the name of the Cloud Web Service secret key that is assigned when you initially get a Cloud account.
(none)	--CloudSecretKeyEnc	The @CloudSecretKeyEnc argument specifies the name of the encrypted Cloud Web Service secret key that is assigned when you initially get a Cloud account.
(none)	--CloudStorageClass	The @CloudStorageClass argument specifies a range of storage classes established for different use cases including: For Amazon S3:

-Argument	--Argument	Description
		• Standard: Standard storage - for general-purpose storage of frequently accessed data. • Standard-IA: Standard Infrequent Access - for long-lived, but less frequently accessed data. • RRS: Reduced Redundancy Storage - for non-critical data considering lower level of redundancy rather than Standard storage. Important: : In versions less than 8.5 you should use --AWSStorageClass. The @AWSStorageClass argument is no longer valid in subsequent LiteSpeed versions after 8.5. For Google Storage: • Multi_regional - for frequently accessed data around the world as per serving website content, streaming videos, or gaming and mobile applications. • Regional - for frequently accessed data in the same region as your Google Cloud DataProc or the Google Compute Engine instances that use it, as per data analytics. • Nearline - for infrequently accessed data (data you expect to access no more than once per month). • Coldline - for infrequently accessed data (data you expect to access no more than once per year).
(none)	--CloudVendor	The @CloudVendor argument specifies the name of the cloud service provider. The argument accepts one of the following values: "AmazonS3", "AzureBlob" or "GoogleStorage".
(none)	--GSProject	DEPRECATED LiteSpeed 8.8: Was used to store for the Google Cloud Storage project ID; the project ID is now obtained from login. This parameter is retained for compatibility with old backup/restore scripts.
(none)	--UseSSL	The @UseSSL argument specifies that the connection uses SSL security. This argument accepts one of the following values: • 0–Do not use SSL • 1–Use SSL (default)

Proxy-Specific Arguments

Proxy arguments work in conjunction with the LiteSpeed arguments. See Syntax and Examples for more information.

-Argument	--Argument	Description
(none)	--ProxyHost	The @ProxyHost argument is optional and specifies the name of the proxy host name that is running the proxy server. note: If the @ProxyHost argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.
(none)	--ProxyLogin	The @ProxyLogin argument is optional and specifies the proxy server login credential. note: If not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.
(none)	--ProxyPassword	The @ProxyPassword argument is optional and specifies the proxy server password credential. note: If the @ProxyPassword argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.
(none)	--ProxyPasswordEnc	The @ProxyPasswordEnc argument is optional and specifies the encrypted proxy server password credential. note: If the @ProxyPasswordEnc argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.
(none)	--ProxyPort	The @ProxyPort argument is optional and contains the port number of the proxy server. The TCP/IP port values can be 1-65535.

-Argument	--Argument	Description
		note: If the @ProxyPort argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

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" --TSMMgmtClass "STANDARD" --tsmdevicetimeoutminutes 2 --SDFullBackupEscalation --SDElapsedDaysRequireFull 14 -S"W2K5_TSM" -U"sa" -P"***"
```

5. Back up to Amazon S3 cloud.

   TIP: Command line fast compression backups only work with the encrypted access information. They will fail if the access info is not encrypted.

```
SLSFastCompression.exe --SDBackupDirectory "fc" --SDExtentsChgRatioRequireFull ".4" -D model --CloudVendor "AmazonS3" --CloudBucketName "california" --CloudAccessKey "*****" --CloudSecretKey "*****" --CloudRegionName "us-west-1" --UseSSL -S"servername" -T
```

6. Back up to Google Storage

```
SLSFastCompression.exe -S "LSSQL17" -T -D "proddb" --compressionlevel 7 --OLRMAP
1 --SDExtentsChgRatioRequireFull ".4" --SDBackupDirectory "fc" --CloudVendor
"GoogleStorage" --CloudAccessKey "name@lsnewproject.iam.gserviceaccount.com" --
CloudSecretKey "***" --GSProject "yourProjectID" --CloudStorageClass "nearline"
--CloudRegionName "us" --CloudBucketName "mybucket" --UseSSL 1 --
SDFullBackupEscalation --SDElapsedDaysRequireFull 14
```

Note: --CloudSecretKey is a full text key that starts with “-----BEGIN PRIVATE KEY-----” and ends with “-----END PRIVATE KEY-----”

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 based on a user-defined period (either the file age or the date).

- Arguments
- Cloud-Specific Arguments
- Proxy-Specific Arguments

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
- Files that have the filesystem archive bit set (if that option is selected)

Syntax

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

Connection options:

```
--Database <database_name>
--Server <server_name>
--WindowsAuth | (--UserName <username> --Password <password>)
```

Cleanup options:

```
--BackupRetainDays <number> | --BackupExpiration <date>
--LogRetainDays <number> | --LogExpiration <date>

--KeepArchiveFiles

--MultiDatabaseType

[--CopyOnlyBackups <option>]
```

TSM connection options:

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

[Cloud connection options:]

```
[--CloudVendor <vendor name>]
[--CloudAccessKey <key name>]

[--CloudAccessKeyEnc <encrypted key name>]

[--CloudAutoStriping <1 or 0>]

[--CloudAutoStripingThreshold <auto, number GB>]

[--CloudSecretKey <key name>]

[--CloudSecretKeyEnc <encrypted key name>]

[--CloudBucketName <bucket name>]

[--CloudRegionName <cloud region name>]

[--CloudGovRegion <government region number>]
```

[Proxy connection options:]

```
[--ProxyHost <proxy host name>]

[--ProxyLogin <proxy server login credential>]

[--ProxyPassword <proxy server password credential>]

[--ProxyPasswordEnc <encrypted proxy server password credential>]
```

```
[--ProxyPort <proxy server port number>]
```

Other options:

```
[--DryRun]
[--LogLevel (0|1|2) [--trace logpath = "path"]]
```

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	--KeepArchiveFiles	Turns on monitoring and refuses to delete files that have the archive filesystem bit set. When enabled dependent files are not deleted.
-b	--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.
-c	--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.
-C	--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.

-Argument	--Argument	Description
		• 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.
-D	--Database	Name of database to be backed up or restored. Only backups of this database are eligible for cleanup.
(none)	--Destination	Specifies the instance of the destination server to 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.
-k	--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.
-L	--LogLevel	Creates a log file. 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.

-Argument	--Argument	Description
		The default output directory is *C:\Documents and Settings\All Users\Application Data\Quest Software\LiteSpeed\SQL Server\Logs (or C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs)* (or *C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs*). To log to a different directory run this utility with the following argument: `--trace logpath = "path"`. For more information, see Configure Logging in LiteSpeed on page 576.
-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.
(none)	--MultiDatabaseType	Produces a cleanup for several types of databases. Types can include: all, system, user, or selected databases. This argument accepts one of the following values: • All - Cleanup backups for all system and user databases. • System - Cleanup backups for only system databases. • User - Cleanup backups for only user databases. • Selected - Cleanup backups for specifically selected databases.
-S	--Server	Specifies the instance of Microsoft SQL Server to connect to. This argument accepts one of the following values: • server_name • server_name\instance_name If no server is specified, the LiteSpeed command-line utility will connect to the default instance of SQL Server on the local computer.
(none)	--ServerToDelete	Specifies the instance of the destination server to delete.
(none)	--TSMAdminName	Specifies the TSM administrative user name that has client authority for the TSM node. Some operations may require an administrative user with client owner authority to be specified in order to open a TSM session. The correct username and password may be obtained from the TSM administrator.
(none)	--TSMAdminPwd	Specifies the plain text password of the administrative user which is used to log in to the TSM server and start the TSM session.

-Argument	--Argument	Description
(none)	--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.
(none)	--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	Specifies user login ID. Required if the connection type is not a trusted connection. Login IDs are case-sensitive.
-P	--Password	Specifies the user password. Passwords are case-sensitive. Required if the connection type is not a trusted connection.
-T	--WindowsAuth	Specifies Windows authentication. Uses a trusted connection (to the server) instead of requiring a password.
-?	--ShowHelp	Displays the syntax summary of the LiteSpeed command-line utility.
(none)	--ShowSyntax	Displays the utility syntax.

Cloud-Specific Arguments

Cloud-specific arguments work in conjunction with the LiteSpeed arguments. See Syntax and SmartCleanup Command-Line Arguments for more information.

-Argument	--Argument	Description
(none)	--CloudAccessKey	The @CloudAccessKey argument specifies the name of the unique Cloud Web Service alphanumeric access key that identifies each user. The selections include Amazon Access Key, Azure Account Name, Google e-mail styled account.
(none)	--CloudAccessKeyEnc	The @CloudAccessKeyEnc argument specifies the name of the encrypted unique Cloud Web Service alphanumeric access key that identifies each user.

-Argument	--Argument	Description
(none)	--CloudBucketName	The @CloudBucketName argument specifies the name of the container for cloud objects. Bucket names must be at least 3 and no more than 63 characters long. The selections are Amazon Bucket Name, Azure Container Name, Google Bucket Name. Google Bucket Name requirements are described at https://cloud.google.com/storage/docs/naming.
(none)	--CloudGovRegion	The @CloudGovRegion argument enables a special restricted region for the US Government use in Amazon S3 and Azure Clouds. This argument accepts one of the following values: • 0–Do not use government cloud (default) • 1–Use government cloud
(none)	--CloudRegionName	The @CloudRegionName argument specifies the name of the Cloud Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-east-2, us-west-1, us-west-2, ca-central-1, eu-central-1, eu-west-1, eu-west-2, ap-south-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, ap-northeast-2, sa-east-1, N'Germany' and N'China'.
(none)	--CloudSecretKey	The @CloudSecretKey argument specifies the name of the Cloud Web Service secret key that is assigned when you initially get a Cloud account.
(none)	--CloudSecretKeyEnc	The @CloudSecretKeyEnc argument specifies the name of the encrypted Cloud Web Service secret key that is assigned when you initially get a Cloud account.
(none)	--CloudVendor	The @CloudVendor argument specifies the name of the cloud service provider. The argument accepts one of the following values: "AmazonS3", "AzureBlob" or "GoogleStorage".
(none)	--UseSSL	The @UseSSL argument specifies that the connection uses SSL security. This argument accepts one of the following values: • 0–Do not use SSL • 1–Use SSL (default)

Proxy-Specific Arguments

Proxy-specific arguments work in conjunction with the LiteSpeed arguments. See Syntax and SmartCleanup Command-Line Arguments for more information.

-Argument	--Argument	Description
(none)	--ProxyHost	The @ProxyHost argument is optional and specifies the name of the proxy host name that is running the proxy server. note: If the @ProxyHost argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.
(none)	--ProxyLogin	The @ProxyLogin argument is optional and specifies the proxy server login credential. note: If not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.
(none)	--ProxyPassword	The @ProxyPassword argument is optional and specifies the proxy server password credential. note: If the @ProxyPassword argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.
(none)	--ProxyPasswordEnc	The @ProxyPasswordEnc argument is optional and specifies the encrypted proxy server password credential. note: If the @ProxyPasswordEnc argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.
(none)	--ProxyPort	The @ProxyPort argument is optional and contains the port number of the proxy server. The TCP/IP port values can be 1-65535. note: If the @ProxyPort argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

Example

1. Delete disk full and differential backups older than 28 days, delete log backups older than 2 days, allow deletions of the copy-only backups:

   ```
   SLSSmartCleanup.exe --Database test2 --BackupRetainDays 28 --LogRetainDays 2 --
   CopyOnlyBackups AllowDeletes --Server LITESPEED\SQL2005 -U sa -P *******
   ```

2. Delete disk full and differential backups created before 11/15/2012:

   ```
   SLSSmartCleanup.exe --Database test2 --BackupExpiration "2012-11-15" --Server
   LITESPEED\SQL2005 --WindowsAuth
   ```

3. Delete tsm log backups older than 2 days:

   ```
   SLSSmartCleanup.exe --Database test_tsm --TSMConfigFile "C:\Program
   Files\Tivoli\TSM\baclient\dsm.opt" --TSMClientNode w2k3_TSM2 --TSMClientOwnerPwd
   ***** --LogRetainDays 2 --WindowsAuth
   ```

4. Delete full, differential and log TSM backups created before 06/15/2012, using the PASSWORDAccess generate option to connect to the TSM Server:

   ```
   SLSSmartCleanup.exe --Database test_tsm --TSMConfigFile "C:\Program
   Files\Tivoli\TSM\baclient\dsm_gp.opt" --BackupExpiration 2012-06-15 --
   LogExpiration 2012-06-15 --WindowsAuth
   ```

5. Delete Amazon S3 cloud backup.

   TIP: The parameter “-CSecretKey” must be generated by the Maintenance plan wizard and cut\paste into the command line script.

   ```
   SLSsmartcleanup.exe -D model --BackupExpiration "2014-03-02 00:00:00" --
   CloudVendor "AmazonS3" --CloudBucketName "california" --CloudAccessKey "***** "
   --CloudSecretKey "******" --CloudRegionName "us-west-1" --UseSSL -S"servername"
   ```

Returns

0 (success) or 1 (failure)

Script Maintenance Plans Tasks

You can perform various database maintenance tasks from the command line. To use the SLSSQLMaint utility, change the directory until you are in the directory containing slssqlmaint.exe. (Usually, *C:\Program Files\Microsoft SQL Server\MSSQL10.SQL2008\MSSQL\Binn*).

NOTE: You can generate scripts by opening tasks in the LiteSpeed UI Console and clicking **View T-SQL**. About Creating Maintenance Plans

Syntax

```
slssqlmaint.exe
-? | (<connection_options> <task_options>)
```

Connection options:

```
-S <server_name\instance_name>
-T | (-U <login_ID> -P <password>)
```

Backup Database task options (Disk):

```
-D <names_group_or_patterns>
-BkUpMedia DISK
-BkUpDB|-BkUpLog
( (<path_and_filename>
-BkFileName
[-NOINIT] )
| ( (-UseDefDir|<path>)
[-CrBkSubDir]
-Default <file_format> ) )
[-BkExt <extension>]
[-Differential]
[-FileGroups <filegroups>]
[-Expiration <date_time> | -Retaindays <number_of_days> ]
[-DelBkUps <number>(HOURS|DAYS|WEEKS|MONTHS|YEARS)]
[-VrfyBackup]
[-With "COPY_ONLY"]
[-AdaptiveCompression (speed|size) | -CompressionLevel <value> ]
[-Logging (0|1|2)]
[-Comment <comment>]
[-Threads <number>]
[-Throttle <value>]
[-BufferCount <number>]
[-MaxTransferSize <size>]
[-Priority (0|1|2)]
[-Affinity <mask>]
[-X
    [DISK_RETRY_COUNT <value>]
    [DISK_RETRY_WAIT <value>]]
[-CryptLevel <value>
(-jobp <encrypted_key> | -BackupKey <encryption_password>) ]
[-DblClick (1|2)]
[-OPTOLR|-NOOPTOLR]
[-Reliability (1|2|3)]
[-Mirror <path>]
[-Attached <file_or_directory>]
[-Exclude
    [Offline]
    [LogShippng]
    [ReadOnly]
    [Selected]
    [Deleted]
```

```
    [<wildcard_or_regex_pattern>]
    [IgnoreReplica
        [Primary]
        [Secondary]]
```

Backup Database task options (TSM or TSM Archive):

```
-D <names_group_or_patterns>
-BkUpMedia (TSM|TSMARCHIVE)
(-BkUpDB|-BkUpLog)
-BkFileName
[-CrBkSubDir]
-Default <file_format>
[-Differential]
[-FileGroups <filegroups>]
[-TSMClientNode <client_node>]
[-TSMClientOwnerPWD <password>]
-TSMConfigFile <path>
-TSMObjectPath <TSM_path>
[-TSMManagementClass <name>]
[-TSMDeviceTimeoutMinutes <timeout_period>]
[-VrfyBackup]
[-With "COPY_ONLY"]
[-AdaptiveCompression (speed|size) | -CompressionLevel <value> ]
[-Logging (0|1|2)]
[-Comment <comment>]
[-Threads <number>]
[-Throttle <value>]
[-BufferCount <number>]
[-MaxTransferSize <size>]
[-Priority (0|1|2)]
[-Affinity <mask>]
[-CryptLevel <value>
(-jobp <encrypted_key> | -BackupKey <encryption_password>) ]
[-Reliability (1|2|3)]
[-Attached <file_or_directory>]
[-Exclude
    [Offline]
    [LogShippng]
    [ReadOnly]
    [Selected]
    [Deleted]
    [<wildcard_or_regex_pattern>]
    [IgnoreReplica
        [Primary]
        [Secondary]]
```

Backup Database task options (Tape):

```
-D <names_group_or_patterns>
-BkUpMedia TAPE
(-BkUpDB|-BkUpLog)
<path>
-BkFileName
```

```
[-NOINIT]
[-Differential]
[-FileGroups <filegroups>]
[-Expiration <date_time> | -Retaindays <number_of_days> ]
[-VrfyBackup]
[-With "COPY_ONLY"]
[-AdaptiveCompression (speed|size) | -CompressionLevel <value> ]
[-Logging (0|1|2)]
[-Comment <comment>]
[-Threads <number>]
[-Throttle <value>]
[-BufferCount <number>]
[-MaxTransferSize <size>]
[-Priority (0|1|2)]
[-Affinity <mask>]
[-CryptLevel <value>
(-jobp <encrypted_key> | -BackupKey <encryption_password>) ]
[-Reliability (1|2|3)]
[-Attached <file_or_directory>]
[-Exclude
    [Offline]
    [LogShippng]
    [ReadOnly]
    [Selected]
    [Deleted]
    [<wildcard_or_regex_pattern>]
    [IgnoreReplica
        [Primary]
        [Secondary]]
```

Fast Compression task options (Disk):

```
-D <names_group_or_patterns>
-BkUpMedia DISK
-BkUpDB <path>
-BkFileName
[-CrBkSubDir]
[-NOINIT]
-SmartDiff <number>DAYS
(-DataDelta <value> | -SizeDelta <value>)
[-ExcludeDays <values>]
-SingleFile (0|1)
[-RequireFull
[-AltBkpDir <path>]]
[-BackupEsc]
[-VerifyOpt (Last|FullDiff|All)]
[-NotifyOpt (Failure|Every)]
[-NotifyAgent <name>]
[-FullDiffRtn 28 ]
[-LogRtn 28]
[-AdaptiveCompression (speed|size) | -CompressionLevel <value> ]
[-Logging (0|1|2)]
[-Comment <comment>]
```

```
[-Threads <number>]
[-Throttle <value>]
[-BufferCount <number>]
[-MaxTransferSize <size>]
[-Priority (0|1|2)]
[-Affinity <mask>]
[-X
    [DISK_RETRY_COUNT <value>]
    [DISK_RETRY_WAIT <value>]]
[-CryptLevel <value>
(-jobp <encrypted_key> | -BackupKey <encryption_password>) ]
[-OPTOLR|-NOOPTOLR]
[-Reliability (1|2|3)]
[-Mirror <path>]
[-Attached <file_or_directory>]
[-Exclude
    [Offline]
    [LogShippng]
    [ReadOnly]
    [Selected]
    [Deleted]
    [<wildcard_or_regex_pattern>]
    [IgnoreReplica
        [Primary]
        [Secondary]]
```

Fast Compression task options (TSM):

```
-D <names_group_or_patterns>
-BkUpMedia DISK
-BkUpDB <path>
-BkFileName
[-CrBkSubDir]
-Default <file_format>
[-TSMClientNode <client_node>]
[-TSMClientOwnerPWD <password>]
-TSMConfigFile <path>
[-TSMObjectPath <filespace_name>]
[-TSMManagementClass <name>]
[-TSMDeviceTimeoutMinutes <timeout>]
-SmartDiff <number>DAYS
(-DataDelta <value> | -SizeDelta <value>)
[-ExcludeDays <values>]
-SingleFile (0|1)
[-RequireFull
[-AltBkpDir <path>]]
[-BackupEsc]
[-VerifyOpt (Last|FullDiff|All)]
[-NotifyOpt (Failure|Every)]
[-NotifyAgent <name>]
[-FullDiffRtn 28 ]
[-LogRtn 28]
[-AdaptiveCompression (speed|size) | -CompressionLevel <value> ]
```

```
[-Logging (0|1|2)]
[-Comment <comment>]
[-Threads <number>]
[-Throttle <value>]
[-BufferCount <number>]
[-MaxTransferSize <size>]
[-Priority (0|1|2)]
[-Affinity <mask>]
[-CryptLevel <value>
(-jobp <encrypted_key> | -BackupKey <encryption_password>) ]
[-Reliability (1|2|3)]
[-Attached <file_or_directory>]
[-Exclude
    [Offline]
    [LogShippng]
    [ReadOnly]
    [Selected]
    [Deleted]
    [<wildcard_or_regex_pattern>]
    [IgnoreReplica
        [Primary]
        [Secondary]]
```

Cleanup History task options:

```
-CleanHis
    <number>(HOURS|DAYS|WEEKS|MONTHS|YEARS)
    [BKUP]
    [JBS]
    [MAINTPLN]
    [Activity]
    [CentralActivity]
    [PurgeInfo]
    [Logshipping]
    [Status]
    [ShrinkLog]
```

Cleanup Maintenance Plans task options:

```
-MAINTDEL
    -DelDeviceType DISK
    -DELTYPE (FileBkup|FileRpt|FileAny)
    [-DELUSEAGE
        -DELUNIT <number>
        -DELUNITTYPE (MINUTES|HOURS|DAYS|WEEKS|MONTHS|YEARS)]
    (-DELFOLDER <directory>
    -DELEXTENSION <extension>
    [-DELSUBFOLDERS]
    [-DelEmptyFolder])
    |(-DELFILE
        -DELFILEPATH <path_and_filename>)
```

Legacy Task options:

```
(-PlanID <plan_ID>|-PlanName <plan_name>)
[<backup_task_options>]
[-CkDB|-CkDBNoIdx]
[-ReorgIDX]
[<cleanup_maint_plans_task_options>]
[<clean_up_history_task_options>]
[-OpNotify <names>
    -OpMessage <message>
    -OpSubject <subject>
    -OpImportance (0|1|2)]
[-RebldIdx <1-100> ]
[-RmUnusedSpace <max_size> <free_space_to_remain>
    [NoTrunc]]
[-UpdOptiStats <number> -UpdStatsTarget (Columns|Index|All) -UpdStatsScan
(FullScan|Rows|Percent)]
[-Rpt <path>.txt
[-DelTxtRpt <number><time_unit>]]
[-HtmlRpt <path>.html
[-DelHtmlRpt <number><time_unit>]]
[-To <operator_name>]
[-Writehistory]
[-LogEngine (0|1|2)]
```

Arguments

-Argument	Description
-AdaptiveCompression <value>	Automatically selects the optimal compression level based on CPU usage or Disk IO. For more information, see Compression Methods on page 121. 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 <mask>	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. For more information, see Processor Affinity on page 64.
-AltBkpDir <path>	Specifies the directory where to search for the backup file.
-Attached <file_or_ directory>	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_`

-Argument	Description
	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.
-BackupEsc	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.
-BackupKey <encryption_ password>	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. If you use the wrong encryption key, the restore will fail. **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'
-BkExt <extension>	Specifies the extension of the backup files.
-BkFileName	Specifies that each path specified includes the filename.
-BkUpDB	Specifies the database backup type.
-BkUpLog	Specifies the tlog backup type.
-BkUpMedia <value>	Specifies the backup destination. This argument accepts one of the following values: • DISK • TAPE • TSM • TSMArchive
-BUFFERCOUNT	Specifies the number of SQL Server buffers available for a LiteSpeed operation. The default value is set by SQL Server.

-Argument	Description
-CkDB	Checks the allocation and structural integrity of tables and indexes.
-CkDBNoIdx	Checks the allocation and structural integrity of tables.

-CleanHis <values>	Enables the Cleanup History task. This argument accepts one or more of the following:
	• <number><time_unit>–Removes historical data older than specified time period. You can use one of the following time units:
	  • HOURS
	  • DAYS
	  • WEEKS
	  • MONTHS
	  • YEARS
	• BKUP–Deletes back up and restore history.
	• JBS–Deletes SQL Server Agent jobs history.
	• MAINTPLN–Deletes Maintenance plans history.
	• Activity–Deletes LiteSpeed activity from the Local Repository.
	• CentralActivity–Deletes LiteSpeed activity from the Central Repository.
	• PurgeInfo–Deletes any information for deleted databases.
	• LogShipping–Deletes log shipping history.
	• Status–Deletes status history (Job, DTS, Maint Plans).
	• ShrinkLog–Shrinks the repository databases log files.

-COMMENT <comment>	Appends a user comment to the backup. This argument accepts variables. For more information, see LiteSpeed Variables on page 125.
-COMPRESSIONLEVEL <value>	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 for disk backups and *7* is the default value for cloud backups. 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. For more information, see Test Optimal Backup Settings on page 83. **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.
-CrBkSubDir	Creates a subdirectory for each database.
-CRYPTLEVEL <value>	Specifies the encryption level. This argument accepts one of the following

-Argument	Description
	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 • 9–MS_AES_128 • 10–MS_AES_192 • 11–MS_AES_256
-D <names_group_or_ patterns>	Specifies the database names or the following built-in database groups to include in the backup: • "ALL DATABASES" • "ALL SYSTEM DATABASES" • "ALL USER DATABASES" Alternately, you can specify wildcard or regular expression patterns using one of the following formats: • Mask:"<wildcard_expression>" • Regex:"<regular_expression>" For more information, see Use Wildcard and Regular Expressions in LiteSpeed on page 111.
-DataDelta <value>	Specifies the minimum amount of database changes required for the full backup. Accepted values are 1-99, the default value is *35*.
-DblClick	Creates a Double Click Restore executable. This argument accepts one of the following values:

-Argument	Description
	• 1–Creates one Double-Click Restore executable file. Note the following warning: The executable may be greater than 4GB for large databases. Windows Server is unable to run executable files larger than 4GB. However, the file will be convertible/restorable by LiteSpeed file. • 2–Creates a Double Click Restore loader in the same location. (Default) For more information, see Double Click Restore Executables on page 120.
-Default <format>	Specifies the default file format, usually %D_%T_%Y-%m-%d-%H%M%S.%EXT%, where: • %D–Database name • %T–Backup type (Full, Diff or Log) • %Y-%m-%d-%H%M%S–Date and time • %EXT%–File extension For more information, see LiteSpeed Variables on page 125.
-DelBkUps <number><time_unit>	Deletes old backups based on the time period specified in the format: <number><time_unit>, where the time unit is one of the following: • HOURS • DAYS • WEEKS • MONTHS • YEARS
-DelDeviceType DISK	Specifies the device type. Currently DISK is the only supported device type.
-DelEmptyFolder	Deletes empty folders.
-DelExtension <extension>	Specifies the extension of the files eligible for deletion.
-DelFile	Deletes a specific file.
-DelFilePath <filename>	Specifies the location and name of the files.
-DelFolder <path>	Deletes a specific folder.
-DelHtmlRpt <number><time_unit>	Deletes .html reports older than the given time period. This argument accepts one of the following time units: • HOURS • DAYS • WEEKS • MONTHS • YEARS
-DelSubFolders	Deletes the first-level subfolders.

-Argument	Description
-DelTxtRpt <number><time_unit>	Deletes .txt reports older than the given time period. This argument accepts one of the following time units: • HOURS • DAYS • WEEKS • MONTHS • YEARS
-DelType <value>	Deletes file by type. This argument accepts one of the following values: • FileBkupbackup–Backup files • FileRptreport–Maintenance plan text reports • FileAny–Any file
-DelUnit	Deletes files based on the file age.
-DelUnitType <value>	Specifies the unit type for file age. This argument accepts one of the following values: • MINUTES • HOURS • DAYS • WEEKS • MONTHS • YEARS
-DelUseAge	Specifies the file age.
-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.
-Exclude <values>	Excludes databases. This argument accepts one or more of the following: • Selected–This option will exclude databases supplied with the -D argument. • LogShippng • ReadOnly • Offline • Deleted • Mask:"<wildcard_expression>" • Regex:"<regular_expression>"

-Argument	Description
	• IgnoreReplica–This option will ignore the preferred replica setting and will back up all replicas, unless they are excluded by the Primary and Secondary options. • Primary • Secondary
-ExcludeDays <values>	Specifies days of the week when a full backup is never performed. This argument accepts one or more of the following: • Mn • Tu • Wd • Th • Fr • St • Sn
-Expiration <date_time>	Specifies the date and time when the backup expires. LiteSpeed will not overwrite this file until expiration datetime is passed. This argument accepts one of the following formats: • yyyy-mm-dd • yyyy-mm-dd hh:mm:ss
-FileGroups <filegroup_names>	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.
-FullDiffRtn <number_of_days>	Specifies the number of days (N). The full or differential backup must be at least N days old before it is eligible for cleanup.
-HtmlRpt <report_filename>.HTML	Writes an HTML report.
-JobP <encrypted_key>	Specifies an encrypted key.
-LogEngine <value>	Controls the debug logging options for a plan. 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. For more information, see Reporting and Logging in Maintenance Plans on

-Argument	Description
	page 579.
-LOGGING <value>	Controls the logging options for backups. 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. For more information, see Configure Logging in LiteSpeed on page 576.
-LogRtn <number_of_days>	Specifies the number of days (N). The t-log backup must be at least N days old before it is eligible for cleanup.
-MAINTDEL	Enables the Cleanup Maintenance Plans task.
-MAXTRANSFERSIZE <size>	Specifies the largest unit of transfer in bytes to be used between SQL Server and LiteSpeed. The possible values are multiples of 65536 bytes (64 KB) ranging up to 4,194,304 bytes (4 MB). The default is 1048576 (1 MB).
-Mirror <path>	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.
-NOINIT	Does not overwrite the existing backup files.
-NOOPTOLR	Does not generate a map file during a backup for Object Level Recovery.
-NO_OUTPUT	Does not display the output.
-NotifyAgent <operator>	Specifies the operator.
-NotifyOpt <value>	Specifies the notification options after backup. This argument accepts one of the following values: • Every–Notify every time • Failure–Notify on failure only
-OpImportance <value>	Specifies whether the message is of low, high or normal importance. This argument accepts one of the following values: • 0–Low • 1–Normal • 2–High
-OpMessage <message>	Specifies the message for the Notify Operator task.
-OpNotify <names>	Specifies the operator names for the Notify Operator task.
-OpProfile <profile>	Specifies operators profile.
-OpSubject <subject>	Specifies the message subject for the Notify Operator task.subject

-Argument	Description
-OPTOLR	Generates a map file during a backup for Object Level Recovery.
-P <password>	Specifies the user password. Passwords are case-sensitive. Required if the connection type is not a trusted connection.
-PlanID <plan_id>	Specifies GUID identifying the maintenance plan in 36 character hex format.
-PlanName <name>	Specifies the maintenance plan name.
-PRIORITY <value>	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
-RebldIdx <%_free_space>	Drops and recreates indexes to improve performance, creates the specified amount of free space on each data page (1-100).
-Reliability <value>	Specifies whether LiteSpeed should perform checksums before writing to media and whether it should continue on error. This argument accepts one of the following values: • 1–Causes checksums to be verified when a LiteSpeed backup is created, LiteSpeed does not continue on error. • 2–Causes the backup be executed on error, checksums are not verified. • 3–Causes the backup be executed despite encountering an invalid backup checksum.
-ReorgIDX	Defragments and compacts existing indexes to improve performance.
-RequireFull	Checks if the expected full backup exists when backing up to separate files and returns a failure message if it is not found. If the full backup file does not exist, it performs a full backup. If the full backup file does exist, the decision to perform a new full or a differential backup will depend on other conditions specified.
-Retaindays <number_of_days>	Specifies a number of days to retain the backup. LiteSpeed will not overwrite this file for this number of days.
-RmUnusedSpace <max_size> <free_space_to_remain>	Reduces the size of data and log files in a database that grow beyond a specified size. This arguments accepts the following values: • Database size, in MB, when the database grows beyond this amount , it becomes eligible for shrinking. The default is *50*. • The percentage of free space to remain after shrinking. The default is *10*. • NoTrunc–(Optional) Returns freed space to operating system.

-Argument	Description
-Rpt <report_filename>.txt	Writes a .TXT report.
-S <server_name>	Specifies the instance of Microsoft SQL Server to connect to. This argument accepts one of the following values: • server_name • server_name\instance_name If no server is specified, the LiteSpeed command-line utility will connect to the default instance of SQL Server on the local computer.
-SingleFile <value>	This argument accepts one of the following values: • 0–Specifies self-contained backup set. • 1–Appends data to an existing full backup file. For more information, see Fast Compression on page 115.
-SizeDelta <value>	Specifies the last differential backup size to last full backup size ratio. When exceeding the specified ratio LiteSpeed performs a full backup. Accepted values are 1-99, the default value is *35*.
-SmartDiff <number>DAYS	Specifies the minimum number of days since last full backup required to perform full backup. The default value is *14*.
-T	Uses a trusted connection (to the server) instead of requiring a password.
-THREADS <number_of_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 <value>	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. For more information, see Adaptive Compression on page 122.
-To <names>	Specifies operator's name.
-TSMClientNode <node_name>	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 <password>	Specifies the TSM client owner user password. Not required, if specified in the

-Argument	Description
	options file or if backing up with the Passwordaccess Generate option.
-TSMConfigFile <path>	Specifies the TSM configuration file.
-TSMDeviceTimeoutMinutes <timeout>	Specifies how long to wait for a TSM device.
-TSMManagementClass <class>	Specifies the TSM management class. If not specified, LiteSpeed uses the default management class.
-TSMObjectPath <path>	Specifies path to the TSM backup. For Fast Compression backups, it only specifies the filespace.
-U <username>	Specifies user login ID. Required if the connection type is not a trusted connection. Login IDs are case-sensitive.
-UpdOptiStats <number>	Enables the Update Statistics task. The number following this argument is the amount of database to sample (in rows or percent). The default is *50* percent.
-UpdStatsScan <value>	Specifies how to measure the amount of database to sample. This argument accepts one of the following values: • Rows • Percent • FullScan
-UpdStatsTarget <value>	Updates index and/or column statistics. This argument accepts one of the following values: • Columns • Index • All
-UseDefDir	Instructs LiteSpeed to use the default backup directory.
-VerifyOpt <value>	Performs a restore verification on the backup file just created (if backup was successful). This argument accepts one of the following values: • Last–Verifies last backup performed (can be either a full or differential). • FullDiff–Verifies the last full backup and last differential is available. • All–Verifies last full backup and all differentials since.
-VrfyBackup	Performs a restore verification on the backup file just created (if backup was successful).
-With "COPY_ONLY"	Specifies the copy-only backup.
-WriteHistory	Writes history to msdb.dbo.sysdbmaintplan_history.

-Argument	Description
-X <option> <option_value> <option> <option_value>	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 and cloud operations. For more information, see Network Resilience on page 124.

Examples

1. Perform full backups of all user databases, except those starting with "temp".

```
slssqlmaint.exe -S LITESPEED\SQL2005 -U sa -P ****** -D "ALL USER DATABASES" -
BkUpMedia DISK -BkUpDB -UseDefDir -AdaptiveCompression speed -Default
"%D_%z.bak" -CryptLevel 8 -BackupKey ****** -Exclude Offline LogShippng ReadOnly
Mask:"temp*"
```

2. For LiteSpeedLocal and LiteSpeedCentral databases, perform full Fast Compression backups if the amount of database changes since the last full backup is 40% or more.

```
slssqlmaint.exe -S LITESPEED\SQL2005 -U sa -P ****** -D "LiteSpeedLocal"
"LiteSpeedCentral" -BkUpMedia DISK -BkUpDB "C:\temp" -BkFileName -
AdaptiveCompression speed -SmartDiff 14DAYS -DataDelta 40 -SingleFile 0 -
RequireFull -BackupEsc -Exclude Offline LogShippng ReadOnly
```

3. Delete backup files older than 14 days from the d:\temp folder and its subfolders, if the files have "bkp" extension.

```
slssqlmaint.exe -S LITESPEED\SQL2005 -U sa -P ****** -MAINTDEL -DelDeviceType
DISK -DELTYPE FileBkup -DELUSEAGE -DELUNIT "14" -DELUNITTYPE "DAYS" -
DELSUBFOLDERS -DELEXTENSION "bkp" -DELFOLDER "d:\temp"
```

4. Remove backup and restore history older than 3 months from the LiteSpeed Local repository, shrink LiteSpeed Local repository.

```
slssqlmaint.exe -S LITESPEED\SQL2005 -U sa -P ****** -CleanHis 3MONTHS BKUP
Activity ShrinkLog
```

5. Script a legacy plan.

```
slssqlmaint.exe -S LITESPEED\SQL2005 -U sa -P ****** -PlanID "00638AA2-C493-
4F90-9C3B-275B55091D60" -BkUpMedia DISK -BkUpDB -UseDefDir -CrBkSubDir -BkExt
"bak" -NOOPTOLR -Logging 2 -AdaptiveCompression speed -Default
"%D_%T_%Y%d%m%H%M%S.%EXT%" -CkDB -ReorgIDX -MAINTDEL -DelDeviceType DISK -
DELTYPE FileBkup -DELUSEAGE -DELUNIT "3" -DELUNITTYPE "DAYS" -DELSUBFOLDERS -
DELEXTENSION "bak" -DELFOLDER "c:\temp" -CleanHis 90DAYS MAINTPLN Activity
PurgeInfo ShrinkLog -RebldIdx 100 -RmUnusedSpace 50 10 -UpdOptiStats 50 -
UpdStatsTarget All -UpdStatsScan FullScan -WriteHistory -LogEngine 0
```

6. Perform an Amazon S3 cloud backup.

   Tip: When using variables in a command line script, use double %% or else they will not be parsed correctly.

   - Correct example: -Default "%%T-%%m-%%H-%%M-%%S"
   - Incorrect example: -Default "%T-%m-%H-%M-%S"

```
slssqlmaint.exe -D "master" -BkUpMedia CLOUD -BkUpDB -BkFileName -Logging 0 -
CompressionLevel 2 -Default "%%T-%%m-%%H-%%M-%%S" -OPTOLR -CSType "AmazonS3" -
CSAccessKey "******" -CSSecretKeyPlain "*********" -CSContainer "california" -
CSAccessPoint "us-west-1" -CSObject "test\" -Exclude Deleted Offline LogShippng
-T -S servername
```

7. Azure Cloud Storage:

```
Slssqlmaint.exe -D "prodb" -BkUpMedia CLOUD -BkUpDB -BkFileName -NOINIT -Logging
0 -CompressionLevel 7 -Default "%D_%T_%Y-%m-%d-%H%M%S.%EXT%" -OPTOLR -CSType
"AzureBlob" -CSAccessKeyPlain "***" -CSSecretKey "***" -CSContainer
"mycontainer" -CSObject "backups" -AutoStrip -UseSSL -CSBlobType "Block" -
Exclude Deleted Offline LogShippng
```

8. Google Cloud Storage:

```
slssqlmaint.exe -D "prodb" -BkUpMedia CLOUD -BkUpDB -BkFileName -NOINIT -Logging
0 -CompressionLevel 7 -Default "%D_%T_%Y-%m-%d-%H%M%S.%EXT%" -OPTOLR -CSType
"GoogleStorage" -CSAccessKey "***" -CSSecretKeyPlain "***" -CSContainer
"mybucket" -CSAccessPoint "australia-southeast1" -CSObject "mybackups\" -
CSProjectID "***" -CSStorageClass 2 -Exclude Deleted Offline LogShippng
```

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. For more information, see Examples on page 257.

Syntax

```
SLSRecast.exe ( -? | <source_options> <target_options> <other_options> )
```

Source Options:

```
-E|--SrcBackupFiles <path>
[-N|--SrcBackupIndex <file_number> ]
[-P|--SrcKey <password> ]
```

Target Options:

```
-F|--TgtBackupFiles <path>
[-K|--TgtKey <password> ]
[-I|--Overwrite ]
[(-C|--CompressionLevel 0...8 )|--AdaptiveCompression (Speed|Size)]
[-e|--EncryptionLevel 0...8 ]
[-y|--Expiration <time> ]
[-r|--RetainDays <number> ]
[-J|--DoubleClick]
[(-M|--OLRMap)
[--TempDirectory <path>] ]
```

Other Options:

```
[ -A|--Affinity <affinity_mask> ]
[ -p|--Priority -1...2 ]
[ -h|--Throttle 1...100 ]

[ -b|--BlockSize ]

[ -X|--IOFlags ]
```

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> ]
[ --TSMPointInTime]
[ --TSMDeviceTimeoutMinutes <number> ]
[ --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
(none)	--AdaptiveCompression	Automatically selects the optimal compression level based on CPU usage or Disk IO. For more information, see Compression Methods on page 121. You can tell Adaptive Compression to optimize backups either for size or for speed. This argument accepts one of the following values: • Size • Speed
-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 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

-Argument	--Argument	Description
		the threading as well. You may also want to consider using Adaptive Compression to maintain backup performance. For more information, see Adaptive Compression on page 122.
-O	--BaseSize	The smallest chunk of memory LiteSpeed attempts to write to disk at any given time.
-b	--BlockSize	Specifies the physical block size, in bytes. Supported values are: 512, 1024, 2048, 4096, 8192, 16384, 32768, and 65536 (Default).
-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 for disk backups and *7* is the default value for cloud backups. 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. For more information, see Test Optimal Backup Settings on page 83. **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.
-J	--DoubleClick	Creates a Double Click Restore executable. This argument accepts one of the following values: • 1–Creates one Double-Click Restore executable file. Note the following warning: The executable may be greater than 4GB for large databases. Windows Server is unable to run executable files larger than 4GB. However, the file will be convertible/restorable by LiteSpeed file. • 2–Creates a Double Click Restore loader in the same location. (Default) For more information, see Double Click Restore Executables on page 120.
-e	--EncryptionLevel	Specifies encryption level. Works in conjunction with the Key (K) parameter. This argument accepts one of the following values: • 0–40-bit RC2 • 1–56 bit RC2

-Argument	--Argument	Description
		• 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 • 9–MS_AES_128 • 10–MS_AES_192 • 11–MS_AES_256
-y	--Expiration	Specifies the date and time when the backup expires. LiteSpeed will not overwrite this file until expiration datetime is passed. This argument accepts one of the following formats: • yyyy-mm-dd • yyyy-mm-dd hh:mm:ss
-X	--IOFlags	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 and cloud operations. For more information, see Network Resilience on page 124.
-L	--LogLevel	Creates a log file. This argument accepts one of the following values: • 0–Logging off.

-Argument	--Argument	Description
		• 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 (or C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs)* (or *C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs*). To log to a different directory run this utility with the following argument: `--trace logpath = "path"`. For more information, see Configure Logging in LiteSpeed on page 576.
-M	--OLRMap	Generates a map file during a backup for Object Level Recovery. This argument accepts one of the following values: • 0–False (default) • 1–True
-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: • -1–Below Normal • 0–Normal (Default) • 1–AboveNormal • 2–High
-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.

-Argument	--Argument	Description
		For TSM backups and TSM archives, this argument accepts the following formats: • tsmbkp:<filespace>\<high>\<low> • tsmarc:<filespace>\<high>\<low> For more information, see Examples on page 257. **Tip:** Multiple -E parameters are used for stripe files (Example 2). Converting multiple files to a single file is accomplished by running the commands more than once (Example 3).
-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: • 0–Do not format (default) • 1–Write new header • 2–Long erase / write new header • 3–Low level controller format / write new header **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.
-w	--TapeRewind	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
-u	--TapeUnload	Applies to tape backups and restores. This argument accepts one of the following values: • 0–Keep tape loaded (default)

-Argument	--Argument	Description
		• 1–Unload and eject tape from the drive after operation
(none)	--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\SQL Server\LiteSpeedSettings.ini*.)
-F	--TgtBackupFiles	Location and name of the target backup/restore file device(s). You can supply multiple instances of this argument. Examples: UNC Path: \\servername\share\path\filename Local path: c:\filedirectory\filename **NOTE:** You cannot use the same location for the source and target files if you want to recast files with the same names.
-K	--TgtKey	Password/key used to encrypt new backup.
-h	--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. For more information, see Adaptive Compression on page 122.
(none)	--TSMAdminName	Specifies the TSM administrative user name that has client authority for the TSM node. Some operations may require an administrative user with client owner authority to be specified in order to open a TSM session. The correct username and password may be obtained from the TSM administrator.
(none)	--TSMOptions	Specifies the options that are used during the TSM session.

-Argument	--Argument	Description
(none)	--TSMAdminPwd	Specifies the plain text password of the administrative user which is used to log in to the TSM server and start the TSM session.
(none)	--TSMarchive	Specifies to store the backup as a TSM archive. This argument accepts one of the following values:
-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.
(none)	--TSMdsmi_dir	DSMI_DIR path if needed.
(none)	--TSMdsmi_log	DSMI_LOG path.
-i	--TSMFile	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 the actual name of the file. **NOTE:** You may only store one item at 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.
(none)	--TSMLogname	Log name.
-z	--TSMMgmtClass	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.

-Argument	--Argument	Description
		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. Convert a backup to a Double Click Restore executable:

```
Slsrecast.exe -E 1.bkp --DoubleClick 2 -F new
```

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

```
Slsrecast.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:

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

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

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

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

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

6. Encrypt a backup:

```
Slsrecast.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 -j c:\dsm.opt -E tsmbkp:test\test\test -c nodename -k password -F"D:\test.exe" -J2

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 <tsm_client_node>]

        [-i <tsm_filespace>]
        [-k <tsm_owner_password>]
        [-j <tsm_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
-?	--ShowHelp	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	--MTFFile	Specify the location and name of the Microsoft Tape Format (MSTF) base file. The extractor utility will create one backup device file for each thread used in a LiteSpeed backup. The extracted files containing the native SQL Server backup will

-Argument	--Argument	Description
		have the following format: `base_file_name`x. Where: • `base_file_name` is the specified Microsoft Tape Format base file. • 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. NOTES: • You can specify a network destination for this parameter. • You only need to specify this parameter once. The extraction utility will create all the necessary files automatically. • You cannot tell the extraction utility to extract a different number of native SQL Server files. However, you can specify different destinations 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. See example 4 for more information. • If a full path is not specified, the extracted files will be created in the current directory.
-F	--BackupFile	The name of the LiteSpeed backup device file to be extracted. This argument accepts network destinations. For TSM backups and TSM archives, this argument accepts the following formats: • tsmbkp:<filespace>\<high>\<low> • tsmarc:<filespace>\<high>\<low> You can supply multiple instances of this argument.
-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.
-i	--TSMFile	Defines the TSM filespace, high level and low level. This argument accepts the following format: `tsm_filespace\tsm_high_level\tsm_low_level` where:

-Argument	--Argument	Description
		• `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.
(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. **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.
-L	--LogLevel	Creates a log file. 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 (or C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs)* (or *C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs*). To log to a different directory run this utility with the following argument: `--trace logpath = "path"`. For more information, see Configure Logging in LiteSpeed on page 576.
-K	--Key	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. If you use the wrong encryption key, the restore will fail. **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'
-N	--BackupIndex	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

-Argument	--Argument	Description
		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. For more information, see Arguments on page 258. about the -E parameter.

4. Extract a LiteSpeed backup to different locations.

   a. See how many destination files the extractor utility is going to create for the LiteSpeed backup:

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

   b. Specify one filename for each destination file. The number of file names you specify with the -E parameter 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"
      "e:\Data\NW2.bak" \\my_server\my_share\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 ( -? | <list_backup_contents> | <view_or_restore_tables> | <restore_other_
objects> | <execute_select_script>)
```

Connection Options:

```
-U <username>
-P <password>
```

Backup Files:

```
-F <full_backup_filename>
    [-N <file_number>]
    [-K <encryption_key>] ]
[-D <diff_backup_filename>
    [-N <file_number>]
    [-K <encryption_key>] ]
[-g <log_backup_filename>
    [ ( [-N <file_number>]
      [-K <encryption_key>] )
    | (-h <striped_log_backup_filename>
        [-N <file_number>]
        [-K <encryption_key>] ) ] ]
[-L (Create|Keep|Delete)]
```

Cloud connection options:

```
[--CloudVendor <vendor name>]
[--CloudAccessKey <key name>]
[--CloudAccessKeyEnc <encrypted key name>]
[--CloudSecretKey <key name>]
[--CloudSecretKeyEnc <encrypted key name>]
[--CloudBucketName <bucket name>]
[--CloudRegionName <cloud region name>]
[--CloudGovRegion <government region number>]
[--CloudStorageClass <standard, standard-ia, standard-rrs>]
[--AWSUseServerSideEncryption <1, 0>]
[--AzureBlobType <block, page>]
[--CloudAutoStriping <1, 0>]
[--CloudAutoStripingThreshold <param size in GB>]
[--UseSSL <1, 0>]
```

Script Options:

```
[ -G <ON_filegroup_name> ]
[ -I <TEXTIMAGE_ON_filegroup_name> ]
[ -i <table_objects> ]
[ -p <prefix> ]
[ -s <suffix> ]
[ --UDT <0|1> ]
```

List Contents:

```
-V
[-Y <object_type>]
<backup_files>
```

View or Recover Tables:

```
-O <table_name>
-E <destination_server_name>
[ (-S <database_name>
[-T <table_name>]
[-W <temp_directory>]
[-X <ship_directory>] ) |
-J <output_filename> ]
[<script_options>]
<backup_files>
<connection_options>
```

Recover Objects Other than Tables:

```
( -C [ <object_name> ]
-Y ( object_type | 'Database' )
 | -Z <objects_filename> )
 [ -Q <script_filename> ]
[<script_options>]
<backup_files>
[-y ]
[-W <temp_directory>]
-E <destination_server_name>
<connection_options>
```

Execute Select Script:

```
-B
(-H <script_text> | -Q <script_file_name>)
[-y ]
-E <destination_server_name>
[ (-S <database_name>
[-T <table_name>]
[-W <temp_directory>]
 )
| -J <output_filename> ]
<backup_files>
<connection_options>
```

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.
- -B, -C, -O, -V and -Z parameters are mutually exclusive.

Arguments

Argument	Name	Description
-?	--Help	Displays the syntax summary of the LiteSpeed command-line utility.
-B	--ExecuteScript	Indicates Execute SELECT mode of operation.
-b	--BackEnd	Object Level Recovery can restore tables using two different internal techniques to handle the record inserts. The first and default method uses BCP files and a TSQL BULK INSERT statement. Object Level Recovery will write a BCP format file and an accompanying binary data file to the local file system. These files may become very large depending on the table size and will require permissions to write to a temporary directory. The default TEMP location can be set by using the @TempDirectory parameter or by setting a permanent temp location in the LiteSpeed configuration file. An alternate insertion method can be specified to use Sql Server's Sql Native Client capabilities. This method inserts row-data directly into the destination database bypassing any storage on the local file-system. To enable this method, use the parameter `@backend='SQLNativeClient'` (or `-b 1` from the command line). To make this the default method set the value "BackEnd=SQLNativeClient" in the Object Level Recovery section of the LiteSpeed configuration file. Regardless of the insertion method used, the batch size can be globally managed by setting the value "BulkImportBatchSize=<N>". This will set the number of row inserts for each batched transaction.
-C <object_name>	--CreateScript	Specifies the name of the object to recover.
-d	--RestoreAsOnDiskTable	This option allows you to restore an in-memory table as a regular table.
-D <diff_backup_filename>	--DifferentialBackupFileName	Name of backup file to restore. Used for differential backups instead of full backup files. You can supply multiple instances of this argument.
-E <destination_server_name>	--DestinationServer	Name of the destination server.
-F <full_backup_	--FullBackupFilename	Location and name of the backup file device containing the object to recover.

Argument	Name	Description
filename>		Examples: UNC Path: \\servername\share\path\filename Local path: c:\filedirectory\filename **NOTE:** There can be multiple files but they must be listed in the order in which they were backed up.
-G <filegroup_name>	--OnFileGroup	Destination ON filegroup name.
-g <log_filename>	--LogBackupFileName	Specifies location and name of the log backup file. You can supply multiple instances of this argument.
-H <script_text>	--ScriptText	SELECT Script literal text. For more information, see Supported SELECT Statements on page 172.
-h <striped_logfilename>	--LogBackupStripeFileName	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.
-i <table_objects>	--IncludeTableObjects	Instructs LiteSpeed to script or recover one or more of the following: • Constraints–But not foreign keys • ForeignKeys • Indexes • Statistics • Triggers • All–All of the above The value is a list of options, separated with commas.
-I <filegroup_name>	--TextImageOnFileGroup	Destination TEXTIMAGE_ON filegroup name. Used to restore a BLOB (binary large object).
-J <filename>	--ResultsFileName	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	--KeepComputedColumns	Instructs LiteSpeed to keep the computed columns with the object restore. This argument accepts one of the following values: • 0–False • 1–True

Argument	Name	Description
-K <encryption_key>	--Key	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. If you use the wrong encryption key, the restore will fail. **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'
-l	--LSMPath	Specify a custom path for finding or creating the LSM file.
-L <option>	--LSM	Specifies handling for OLR LSM mapfile(s). • Create–Reads backup and creates a new mapfile. It will ignore attached LSM. • Keep–Does not delete mapfile(s) when complete. • Delete–Always deletes mapfile(s) when complete.
-M	--FileStreamOnFileGroup	Specifies a file stream filegroup to include in the object restore.
-m	--PersistLogProcessing	Instructs LiteSpeed to persist log processing, so the same database backup does not have to be processed for each Object Level Recovery operation. This argument accepts one of the following values: • 0–False (Default). • 1–True. LiteSpeed will persist transaction log backups specified and the tail log for future use. This option can offer a huge performance gain for working with databases with large tail logs that could possibly take a long time to process.
-N <file_number>	--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.
-O <object_name>	--RestoreTable	Specifies the name of the object to recover. **NOTE:** Currently only tables. Table name must be preceded by database owner.
-p <prefix>	--PrefixTableObjects	Adds a prefix to the names of the table's objects you selected to script or recover.
-P <password>	--Password	Specifies the user password. Passwords are case-sensitive. Required if the connection type is not a trusted connection.
-Q <script_	--ScriptFileName	The file name that the script is output into. When used with

Argument	Name	Description
file_name>		execute-select, this file contains the select statement(s). For more information, see Supported SELECT Statements on page 172.
-R	--Trusted	This is Windows Authentication.
-s <suffix>	--SuffixTableObjects	Adds a suffix to the names of the table's objects you selected to script or recover.
-S <destination_database_name>	--DestinationDatabase	Specifies the destination database.
-t	--Trace	Used by LiteSpeed to activate trace logging.
-T <destination_table_name>	--TableName	Specifies the name of the destination table. LiteSpeed will not overwrite an existing table. If you select the same server instance and database as the original table, you must use a different table name. **NOTE:** For Execute-Select operations, LiteSpeed will attempt to insert (append) all selected records into existing table.
-U <username>	--UserName	Specifies user login ID. Required if the connection type is not a trusted connection. Login IDs are case-sensitive.
--UDT		Create table script: • 0–Off. Create table with native types, if possible; othervise (CLR UDT) create with UDT. (Default). • 1–On. Create table with UDT.
-V	--ViewContents	View contents.
-W <temp_directory>	--WriteDirectory	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\SQL Server\LiteSpeedSettings.ini*.)
-X <ship_directory>	--ShipDirectory	Ship directory for packaging files for subsequent restore.

Argument	Name	Description
-Y <object_type>	--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 [1, 3] • Database • Default • ExtendedProcedure • Function • IndexedView • MemoryOptimizedTable • PartitionFunction • PartitionScheme • Role [1] • Rule • StoredProcedure • SystemTable • Table • TableConstraintClustered [2] • TableConstraints [2] • TableForeignKeys [2] • TableIndexClustered [2] • TableIndexes [2] • TableStatistics[2] • TableTriggers [2] • Trigger • Type • User [1] • View • XmlSchemaCollection Notes: [1] These values cannot be used to create scripts. [2] 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 -V or xp_objectrecovery_viewcontents. When one of these values is used with -C or xp_objectrecovery_createscript, @ObjectName (-C) is not the name of the object, but the name of the owning table. [3] 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.
-y	--DisableLogProcessing	Instructs LiteSpeed to skip all transaction log backups and tail log processing. This may improve read and recovery times.
-Z <filename>	--ObjFileName	Identifies a file that contains a list of objects. The format of this file is "ObjectType,ObjectName" per line. **Tip:** You can use -V and -Y arguments to create the objects list.

Cloud-Specific Arguments

Cloud-specific arguments work in conjunction with the LiteSpeed arguments. See Syntax and Examples for more information.

-Argument	--Argument	Description
(none)	--AWSUseServerSideEncryption	The @AWSUseServerSideEncryption argument enables the encryption of data stored at rest in Amazon S3. This argument accepts one of the following values: • 0–Do not use Server Side Encryption • 1–Use Server Side Encryption
(none)	--AzureBlobType	The @AzureBlobType argument specifies the types of blobs that can be stored in the Microsoft Azure cloud storage. This argument accepts one of the following values: "Block", "Page". note: The LiteSpeed auto striping logic used in the @CloudAutoStriping and @CloudAutoStripingThreshold parameters can override the Azure blob limit for LiteSpeed backups.
(none)	--CloudAccessKey	The @CloudAccessKey argument specifies the name of the unique Cloud Web Service alphanumeric access key that identifies each user. The selections include Amazon Access Key, Azure Account Name, Google e-mail styled account.
(none)	--CloudAccessKeyEnc	The @CloudAccessKeyEnc argument specifies the name of the encrypted unique Cloud Web Service alphanumeric access key that identifies each user.
(none)	--CloudAutoStriping	This parameter enables automatic file striping for LiteSpeed cloud backups.
(none)	--CloudAutoStripingThreshold	This parameter contains the stripe size in GBs. LiteSpeed logic uses the database size to make a decision about the number of stripes needed for LiteSpeed cloud backups. For example, if you have a database with a size of 200GB and set @CloudAutoStripingThreshold = 50, then LiteSpeed uses 200/50 = 4 stripes.
(none)	--CloudBucketName	The @CloudBucketName argument specifies the name of the container for cloud objects. Bucket names must be at least 3 and no more than 63 characters

-Argument	--Argument	Description
		long. The selections are Amazon Bucket Name, Azure Container Name, Google Bucket Name. Google Bucket Name requirements are described at https://cloud.google.com/storage/docs/naming.
(none)	--CloudGovRegion	The @CloudGovRegion argument enables a special restricted region for the US Government use in Amazon S3 and Azure Clouds. This argument accepts one of the following values: • 0–Do not use government cloud (default) • 1–Use government cloud
(none)	--CloudRegionName	The @CloudRegionName argument specifies the name of the Cloud Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-east-2, us-west-1, us-west-2, ca-central-1, eu-central-1, eu-west-1, eu-west-2, ap-south-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, ap-northeast-2, sa-east-1, N'Germany' and N'China'.
(none)	--CloudSecretKey	The @CloudSecretKey argument specifies the name of the Cloud Web Service secret key that is assigned when you initially get a Cloud account.
(none)	--CloudSecretKeyEnc	The @CloudSecretKeyEnc argument specifies the name of the encrypted Cloud Web Service secret key that is assigned when you initially get a Cloud account.
(none)	--CloudStorageClass	The @CloudStorageClass argument specifies a range of storage classes established for different use cases including: For Amazon S3: • Standard: Standard storage - for general-purpose storage of frequently accessed data. • Standard-IA: Standard Infrequent Access - for long-lived, but less frequently accessed data. • RRS: Reduced Redundancy Storage - for non-critical data considering lower level of redundancy rather than Standard storage. Important: : In versions less than 8.5 you should use --AWSStorageClass. The @AWSStorageClass argument is no longer valid in subsequent LiteSpeed versions after 8.5. For Google Storage:

-Argument	--Argument	Description
		• Multi_regional - for frequently accessed data around the world as per serving website content, streaming videos, or gaming and mobile applications. • Regional - for frequently accessed data in the same region as your Google Cloud DataProc or the Google Compute Engine instances that use it, as per data analytics. • Nearline - for infrequently accessed data (data you expect to access no more than once per month). • Coldline - for infrequently accessed data (data you expect to access no more than once per year).
(none)	--CloudVendor	The @CloudVendor argument specifies the name of the cloud service provider. The argument accepts one of the following values: "AmazonS3", "AzureBlob" or "GoogleStorage".
(none)	--GSProject	DEPRECATED LiteSpeed 8.8: Was used to store for the Google Cloud Storage project ID; the project ID is now obtained from login. This parameter is retained for compatibility with old backup/restore scripts.
(none)	--UseSSL	The @UseSSL argument specifies that the connection uses SSL security. This argument accepts one of the following values: • 0–Do not use SSL • 1–Use SSL (default)

Examples

1. List all objects in the LiteSpeedLocal database in the LITESPEED_full.bak backup file:

```
olr.exe -V -F "C:\temp\LITESPEED_full.bak" -Y "All"
```

2. Preview a table, do not use tail log:

```
olr.exe -F "C:\temp\LITESPEED_full.bak" -N 1 -O dbo.LitespeedActivity -y -E
LITESPEED\SQL2005
```

3. Recover the contents of the LiteSpeedActivity table to the LITESPEED\SQL2005 server, TEST database, LiteSpeedActivity117 table:

```
olr.exe -F "C:\temp\LITESPEED_full.bak" -N2 -D "c:\temp\LITESPEED_diff.bak" -N3
-O dbo.LiteSpeedActivity -E LITESPEED\SQL2005 -S TEST -Tdbo.LiteSpeedActivity117
```

4. Recover the contents of the LiteSpeedActivity table to the LITESPEED\SQL2005 server, TEST database, LiteSpeedActivity table using custom temp directory:

```
olr.exe -F "C:\temp\LITESPEED_full.bak" -N2 -D "c:\temp\LITESPEED_diff.bak" -N3
-O dbo.LiteSpeedActivity -E LITESPEED\SQL2005 -S TEST -Tdbo.LiteSpeedActivity -
Wd:\products
```

5. Recover dbo.Employees from a striped backup:

```
olr.exe -F "C:\temp\FOX_full.bak" -K****** -g"C:\temp\FOX_tlog1.bak" -
h"C:\temp\FOX_tlog2.bak" -h"C:\temp\FOX_tlog3.bak" -K****** -Odbo.Employees -i
"constraints, foreignKeys" -E LITESPEED\SQL2005 -S HR -Tdbo.Employees
```

6. Query the backup:

```
olr.exe -F "C:\temp\LITESPEED_full.bak" -B -H "select top (100)* from
dbo.LiteSpeedActivity" -E LITESPEED\SQL2005
```

7. Execute select script and save results in a database:

```
olr.exe -F "C:\temp\LITESPEED_full.bak" -B -Q "C:\temp\New Folder\select_
script.sql" -E LITESPEED\SQL2005 -SQResults -T DBID6
```

8. Query the backup and save results in a .csv file:

```
olr.exe -F "C:\temp\LITESPEED_full.bak" -B -H "select * from
dbo.LiteSpeedActivity where PercentCompleted < 100" -E LITESPEED\SQL2005 -J
"C:\LS_Activity.csv"
```

9. Create scripts in the "c:\temp\scripts" folder to recover dbo.Employees later:

```
olr.exe -F "C:\temp\FOX_full.bak" -K****** -g"C:\temp\FOX_tlog1.bak" -
h"C:\temp\FOX_tlog2.bak" -h"C:\temp\FOX_tlog3.bak" -K****** -Odbo.Employees -i
"constraints, foreignKeys" -E LITESPEED\SQL2005 -S NHR -T dbo.Employees -X
c:\temp\scripts
```

10. Generate a script to restore the table schema only:

```
olr.exe -F "C:\temp\LITESPEED_full.bak" -N 1 -C dbo.LitespeedActivity -i
"constraints, foreignkeys" -s "_restored"
```

11. Generate a script for an object other than a table:

```
olr.exe -F "C:\temp\FOX_full.bak" -K****** -N3 -C dbo.FOX_view -Y View
```

12. Create a script file for objects listed in the objects.txt file:

```
olr.exe -F "C:\temp\FOX_full.bak" -K****** -N3 -Z "c:\temp\objects.txt" -Q
d:\temp\create_view.sql
```

13. Create a 'Create Database' script:

```
olr.exe -F "C:\temp\FOX_full.bak" -K****** -N3 -C -Y Database -Q d:\temp\create_
database_FOX.sql
```

14. Restore Objects

```
olr.exe -F "C:\temp\FOX_full.bak" -K****** -N3 -C -Y Database -Q d:\temp\create_
database_FOX.sql --UDT 0
```

15. Read files direct from cloud storage

```
olr.exe -F "s3:\\bucket.US\abyr-full-s3.bak" -V -Y All --CloudVendor "AmazonS3"
--CloudAccessKeyEnc "***" --CloudSecretKeyEnc "***"
```

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_key> [-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
```

3. Register the supplied license key

```
LicenseInfoCmd.exe -l C20TM3Q3K2HD74UDLBMHC6KYV6HZ3MQFNXZFB-123-45678-34 -s
```

4. Remove the currently installed license

```
LicenseInfoCmd.exe -r
```

NOTE: LicenseInfoCmd needs to be run from an elevated command prompt on Windows Vista/2008/7/8/10 to be able to store or remove the license key.

On upgrade from a LiteSpeed 8.5 or earlier installation the site message parameter is required as in:

```
LicenseInfoCmd.exe -l C20TM3Q3K2HD74UDLBMHC6KYV6HZ3MQFNXZFB-123-45678-34 -m
```

```
"Trial Version"
```

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

11

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 Toad for SQL Server.

NOTES:

- You can only run LiteSpeed commands on SQL Servers on which the LiteSpeed stored procedures have been registered.
- Extended stored procedures are executed against the master database.
- Review the Syntax sections to see which arguments are mandatory, which are optional, and which are mutually exclusive. Mutually exclusive arguments are separated by a vertical bar. Optional arguments are enclosed in square brackets. Round brackets are used to group arguments.
- Review the Arguments sections for more information about the arguments and accepted values.
- LiteSpeed arguments 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.

Create Backups

If you want to…	Use…
Back up a database (full, differential, file, or filegroup)	xp_backup_database
Back up a transaction log	xp_backup_log
Back up using Fast Compression	xp_slsFastCompression
Backup databases and perform other maintenance tasks	xp_slssqlmaint
Convert a backup to a Double Click Restore	xp_slsCreateDCR

Verify Backups

If you want to…	Use…
Verify the backup without restoring it	xp_restore_verifyonly
Validate that a file has not been corrupted	xp_restore_checksumonly

View Information about Backups

If you want to…	Use…
List header information for all LiteSpeed backups on a backup device	xp_restore_headeronly
View information about a stripe set	xp_restore_setinfo
List the logical file names in a backup	xp_restore_filelistonly
Check a password against a backup	xp_restore_checkpassword

Clean Up Old Backups

If you want to…	Use…
Delete old backups	xp_slsSmartCleanup

Restore Backups and Files

If you want to…	Use…
Restore a database	xp_restore_database
Restore transaction logs	xp_restore_log
Automate restore operations	xp_restore_automated
Restore only files attached to a backup	xp_restore_attachedfilesonly

Recover Objects from Backups

If you want to…	Use…
Restore tables from backups	xp_objectrecovery
List restorable objects in a backup	xp_objectrecovery_viewcontents
Create DDL scripts to recover objects	xp_objectrecovery_createscript
Execute a SELECT statement against the backup (can be used for row-level restores)	xp_objectrecovery_executeselect

Encrypt Passwords

If you want to…	Use…
Check a password against a backup	xp_restore_checkpassword
Encrypt a password for backup	xp_encrypt_backup_key
Encrypt a password for restore	xp_encrypt_restore_key

TSM-Specific Tasks

If you want to…	Use…
Delete an object from a specified TSM location	xp_delete_tsmfile
Retrieve TSM-specific information	xp_view_tsmcontents
See available TSM management classes	xp_view_tsmmc

Check Progress and Memory

If you want to…	Use…
Check the progress of an activity	xp_slsreadprogress
Check available memory	xp_memory_size

LiteSpeed Information

If you want to…	Use…
Manage licensing information	xp_sqllitespeed_licenseinfo
View LiteSpeed components version	xp_sqllitespeed_version

Backup examples - extended stored procedures

Backup examples are included for often used extended stored procedures. All backup type examples, disk backup examples, cloud backup examples, TSM backup examples, and tape examples are categorized. In each category locate the extended stored procedure in the **Use this procedure..** column and then select the example link in the **To run these backups…** column.

All backup type examples

Use this procedure…	To run these backups…
xp_backup_database	Back Up Database with Init
"	Create Differential Backup
"	Back Up Database with Encryption
"	Back Up Database with Multiple Threads
"	Multiple Backup Devices (Striped Backup)
"	Create Filegroup Backup
"	Create Differential Filegroup Backup
"	Create Partial Backup (includes the primary filegroup, all read/write secondary filegroups, and a specified read-only file)

Use this procedure…	To run these backups…
"	Backup all databases
"	Backup user databases
"	Backup selected databases
xp_slsFastCompression	Full backup change of 40%
"	Full backup to multiple locations
"	Force full backup
"	Backup showing FastCompressionExtension argument

Disk backup examples

Use this procedure…	To run these backups…
xp_backup_database	xp_backup_database (Disk)
xp_backup_log	Back Up Log (Disk)

Cloud backup examples

Use this procedure…	To run these backups…
xp_backup_database	Backup database to Amazon S3
"	Backup database to Amazon S3 using @CloudStorageClass
"	Backup database to Microsoft Azure
xp_backup_log	Back Up Log (Microsoft Azure)
xp_slsFastCompression	sls_FastCompression (Microsoft Azure)

TSM backup examples

Use this procedure…	To run TSM backups…
xp_backup_database	xp_backup_database (TSM)
"	Create TSM Archive

Use this procedure…	To run TSM backups…
xp_backup_log	Back Up Log (TSM)
xp_slsFastCompression	Backup to TSM change of 40%

Tape backup examples

Use this procedure…	To run these backups…
xp_backup_database	Back Up Database to Tape
"	xp_backup_database (Tape)
xp_backup_log	Back Up Log (Tape)

xp_backup_database

Performs full, differential, file, or partial filegroup backups.

Examples and Syntax

NOTE: You can replace argument values with variables. For more information, see LiteSpeed Variables on page 125.

All Backup 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='model'
, @filename='I:\no.BAK'
, @init=1
, @encryptionkey= N'Password'
, @nowrite = 0
, @returndetails = 1

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'
```

Backup all databases

```
exec  master.dbo.xp_backup_database
@MultiDatabaseType  =  N'all',
@backupname  =  N'%DATABASENAME%  -  Full  Database  Backup',
@desc  =  N'Full  Backup  of  %DATABASENAME%  on  5/31/2013  6:32:44  AM',
@compressionlevel  =  5,
@filename  =  N'D:\Backups\mdb-%D_%T_%z.bak',
@init  =  1,
@comment  =  'multi',
@with  =  N'STATS  =  10'
GO
```

Backup user databases

```
exec  master.dbo.xp_backup_database
@MultiDatabaseType  =  N'user',
@backupname  =  N'%DATABASENAME%  -  Full  Database  Backup',
@desc  =  N'Full  Backup  of  %DATABASENAME%  on  5/31/2013  6:32:44  AM',
@compressionlevel  =  5,
@filename  =  N'D:\Backups\mdb-%D_%T_%z.bak',
@init  =  1,
@comment  =  'multi',
@with  =  N'STATS  =  10'
GO
```

Backup selected databases

```
exec master.dbo.xp_backup_database
@MultiDatabaseType = N'selected',
@Database = N'db1',
@Database = N'db2',
@backupname = N'%DATABASENAME% - Full Database Backup',
@desc = N'Full Backup of %DATABASENAME% on 5/31/2013 6:32:44 AM',
@compressionlevel = 5,
```

```
@filename = N'D:\Backups\mdb-%D_%T_%z.bak',
@init = 1,
@comment = 'multi',
@with = N'STATS = 10'
GO
```

xp_backup_database (partial backup)

NOTE: The following example shows the syntax for performing partial backups using the “read_write_ filegroups” parameter. The database used in the example below, FGBackups_PROD, contains the following filegroups: Primary, FG1. FG2, and FG3).

Tip: The following example takes a partial backup of the primary and secondary read write file groups (Primary, FG1, and FG2).

```
EXEC master.dbo.xp_backup_database

@database = N'FGBackups_PROD',

@backupname = N'FGBackups_PROD - Full Database Backup',

@desc = N'Full Backup of FGBackups_PROD on %Y-%m-%d %I:%M:%S %p',

@compressionlevel = 2,

@filename = N'I:\FGBackups_FULL.bkp',

@read_write_filegroups = 1,

@init = 1,

@with = N'STATS = 10'
```

Tape Backup Examples

Back Up Database to Tape

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

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 ]
```

Cloud Backup Examples

Backup database to Amazon S3

```
exec master.dbo.xp_backup_database
@Database = N'AA_3'
, @FileName = N'folder2\AA3Backup.bak'
, @CloudVendor = N'AmazonS3'
, @CloudBucketName = N'aabucket1'
, @CloudAccessKey = N'***' -- my key
, @CloudSecretKey = N'***' -- my key
, @UseSSL = 1
, @CloudRegionName = N'us-west-2' -- us-east-1, us-west-2, us-west-1, eu-west-1, ap-
southeast-1, ap-southeast-2, ap-northeast-1, sa-east-1
(This is an optional parameter. Regions for selected @CloudBucketName are used.)
, @ProxyHost = N'proxy.sitelocal'
, @ProxyPort = 8080
, @ProxyLogin = N'DOMAIN\tst-xyz-MYtester'
, @ProxyPassword = N'****'
, @AWSUseServerSideEncryption = 1
, @AWSStorageClass = 'RRS'
```

Backup database to Amazon S3 using @CloudStorageClass

```
exec xp_backup_database
@Database = N'a'
, @FileName = N'a1.bak'
, @CloudVendor = N'AmazonS3'
, @CloudAccessKey = N'***'
```

```
, @CloudSecretKey = N'***'
, @CloudBucketName = N'bucket'
, @CloudStorageClass = N'standard-ia' -- standard / rrs / standard-ia
```

Backup database to Microsoft Azure

```
EXECmaster.dbo.xp_backup_database
@database = N'model',
@backupname = N'model - Full Database Backup',
@desc = N'Full Backup of model on %Y-%m-%d %I:%M:%S %p',
@compressionlevel = 7,
@filename = N'test\test.bak',
@CloudVendor = N'AzureBlob',
@CloudAccessKey = N'*******',
@CloudSecretKey = N'******************',
@CloudBucketName = N'test',
@AzureBlobType = N'Page',
@UseSSL = 1,
@init = 0,@with = N'STATS = 10'
```

Backup database to Google Cloud Storage

```
exec master.dbo.xp_backup_database
@database = N'db'
, @backupname = N'%DATABASENAME% - Full Database Backup'
, @desc = N'Full Backup of %DATABASENAME% on %Y-%m-%d %I:%M:%S %p'
, @compressionlevel = 7
, @filename = N'%D.bak'
, @UseSSL = 1
, @init = 0
, @OLRMAP = 1
, @with = N'STATS = 10'
, @CloudVendor = N'GoogleStorage'
, @CloudBucketName = N'bucketname'
, @CloudAccessKey = N'***' -- my key
, @CloudSecretKey = N'***' -- my key
, @GSProject = N'***' -- my project ID
, @CloudStorageClass = N'nearline' -- as an example
, @CloudRegionName = N'us-central1' -- as an example
```

Disk Backup Examples

xp_backup_database (Disk)

```
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']
[, @verify = 0 | 1 ]
[, @returndetails = 0 |1 ]
```

TSM Backup Examples

xp_backup_database (TSM)

```
EXEC master.dbo.xp_backup_database
@database = 'database_name'
[, @nowrite = 0 | 1 ]
[, @desc = 'backup_description']
[, @backupname = 'backupset_name']
[, @threads = 1..32]
[, @init = 0 | 1 ]
[, @LSECompatible = 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]
[, @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']
[, @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 ]
```

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
```

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 or Disk IO. For more information, see Compression Methods on page 121.

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

Decimal Value	Binary Bit Mask	Allow LiteSpeed Threads on Processors
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. For more information, see Adaptive Compression on page 122.

@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.

@AWSAccessKey

The @AWSAccessKey argument specifies the name of the unique Amazon Web Service alphanumeric access key that identifies each user.

> Important: This @AWSAccessKey argument is replaced by @CloudAccessKey. The @AWSAccessKey argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSAccessKeyEnc

The @AWSAccessKeyEnc argument specifies the name of the encrypted unique Amazon Web Service alphanumeric access key that identifies each user.

> Important: This @AWSAccessKeyEnc argument is replaced by @CloudAccessKeyEnc. The @AWSAccessKeyEnc argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSBucketName

The @AWSBucketName argument specifies the name of the container for AWS objects. Bucket names must be at least 3 and no more than 63 characters long.

Important: This @AWSBucketName argument is replaced by @CloudBucketName. The @AWSBucketName argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSMaxParts

Is the number of parts that are simultaneously uploaded to Amazon S3. The LiteSpeed default is 3 parts. The number of parts can be up to 5 if there is enough memory available during the upload. If you override this parameter, you may impact memory usage.

Important: This @AWSMaxParts argument is replaced by @CloudParallelUpload. The @AWSMaxParts argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSPartSize

The size of each part that is uploaded to Amazon S3 (in MB). The LiteSpeed default for Part Size is calculated as a database size divided into 9,000. The default Part Size = 25MB.

Important: This @AWSPartSize argument is replaced by @CloudPartSize. The @AWSPartSize argument is no longer valid in subsequent LiteSpeed versions after 8.2.

Notes:

- Amazon S3 has a maximum allowable 10,000 parts per file. If you override this parameter, you may inadvertently go over the 10,000 limit.
- Minimum and maximum values for Part Size are defined by Amazon S3: 5MB and 5120MB (5GB) relatively.
- The maximum object size is 5TB.

TIP: Quest Software recommends using LiteSpeed defaults.

@AWSRegionName

The @AWSRegionName argument specifies the name of the Amazon Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-west-2, us-west-1, eu-west-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, and sa-east-1.

Important: This @AWSRegionName argument is replaced by @CloudRegionName. The @AWSRegionName argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSSecretKey

The @AWSSecretKey argument specifies the name of the Amazon Web Service secret key that is assigned when you initially get an AWS account.

Important: This @AWSSecretKey argument is replaced by @CloudSecretKey. The @AWSSecretKey argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSSecretKeyEnc

The @AWSSecretKeyEnc argument specifies the name of the encrypted Amazon Web Service secret key that is assigned when you initially get an AWS account.

Important: This @AWSSecretKeyEnc argument is replaced by @CloudSecretKeyEnc. The @AWSSecretKeyEnc is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSUseGovCloud

The @AWSUseGovCloud argument enables a special restricted region for the US Government use in Amazon S3. This argument accepts one of the following values:

- 0–Do not use government cloud
- 1–Use government cloud

Important: This @AWSUseGovCloud argument is replaced by @CloudGovRegion. The @AWSUseGovCloud argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSUseReducedRedundancy

The @AWSUseReducedRedundancy argument specifies the use of reduced redundancy storage in Amazon S3. This argument accepts one of the following values:

- 0–Do not use reduced redundancy storage
- 1–Use reduced redundancy storage

Note: This @AWSUseReducedRedundancy argument is replaced with the @CloudStorageClass = 'rrs' argument.

@AWSUseServerSideEncryption

The @AWSUseServerSideEncryption argument enables the encryption of data stored at rest in Amazon S3. This argument accepts one of the following values:

- 0–Do not use Server Side Encryption
- 1–Use Server Side Encryption

@AzureBlobType

The @AzureBlobType argument specifies the types of blobs that can be stored in the Microsoft Azure cloud storage. This argument accepts one of the following values: "Block", "Page".

note: The LiteSpeed auto striping logic used in the @CloudAutoStriping and @CloudAutoStripingThreshold parameters can override the Azure blob limit for LiteSpeed backups.

@backupname

Specifies the name of the backup set.

This argument accepts variables. For more information, see LiteSpeed Variables on page 125.

@buffercount

Specifies the number of SQL Server buffers available for a LiteSpeed operation. The default value is set by SQL Server.

@CloudAccessKey

The @CloudAccessKey argument specifies the name of the unique Cloud Web Service alphanumeric access key that identifies each user. The selections include Amazon Access Key, Azure Account Name, Google e-mail styled account.

@CloudAccessKeyEnc

The @CloudAccessKeyEnc argument specifies the name of the encrypted unique Cloud Web Service alphanumeric access key that identifies each user.

@CloudAutoStriping

This parameter enables automatic file striping for LiteSpeed cloud backups.

@CloudAutoStripingThreshold

This parameter contains the stripe size in GBs. LiteSpeed logic uses the database size to make a decision about the number of stripes needed for LiteSpeed cloud backups. For example, if you have a database with a size of 200GB and set @CloudAutoStripingThreshold = 50, then LiteSpeed uses 200/50 = 4 stripes.

@CloudBucketName

The @CloudBucketName argument specifies the name of the container for cloud objects. Bucket names must be at least 3 and no more than 63 characters long. The selections are Amazon Bucket Name, Azure Container Name, Google Bucket Name. Google Bucket Name requirements are described at https://cloud.google.com/storage/docs/naming.

@CloudGovRegion

The @CloudGovRegion argument enables a special restricted region for the US Government use in Amazon S3 and Azure Clouds. This argument accepts one of the following values:

- 0–Do not use government cloud (default)
- 1–Use government cloud

@CloudParallelUpload

The @CloudParallelUpload argument, parallel parts transfers, is used to create fast uploads to the Azure Cloud or Amazon S3. The default number of parallel uploads:

- Amazon S3 = 3
- Azure Blob = 20

@CloudPartSize

The @CloudPartSize argument determines the size of each part that is uploaded to the cloud. The default part size:

- Amazon S3 = 25MB
- Azure Blob = 4MB

notes:

- Minimum part size for Azure Blob = 4MB
- Minimum part size for Amazon S3 = 5MB

TIP: Quest Software recommends using LiteSpeed defaults.

@CloudRegionName

The @CloudRegionName argument specifies the name of the Cloud Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-east-2, us-west-1, us-west-2, ca-central-1, eu-central-1, eu-west-1, eu-west-2, ap-south-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, ap-northeast-2, sa-east-1, N'Germany' and N'China'.

@CloudSecretKey

The @CloudSecretKey argument specifies the name of the Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudSecretKeyEnc

The @CloudSecretKeyEnc argument specifies the name of the encrypted Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudStorageClass

The @CloudStorageClass argument specifies a range of storage classes established for different use cases including:

For Amazon S3:

- Standard: Standard storage - for general-purpose storage of frequently accessed data.
- Standard-IA: Standard Infrequent Access - for long-lived, but less frequently accessed data.
- RRS: Reduced Redundancy Storage - for non-critical data considering lower level of redundancy rather than Standard storage.

Important: : In versions less than 8.5 you should use --AWSStorageClass. The @AWSStorageClass argument is no longer valid in subsequent LiteSpeed versions after 8.5.

For Google Storage:

- Multi_regional - for frequently accessed data around the world as per serving website content, streaming videos, or gaming and mobile applications.
- Regional - for frequently accessed data in the same region as your Google Cloud DataProc or the Google Compute Engine instances that use it, as per data analytics.
- Nearline - for infrequently accessed data (data you expect to access no more than once per month).
- Coldline - for infrequently accessed data (data you expect to access no more than once per year).

@CloudVendor

The @CloudVendor argument specifies the name of the cloud service provider. The argument accepts one of the following values: "AmazonS3", "AzureBlob" or "GoogleStorage".

@comment

Appends a user comment to the backup.

This argument accepts variables. For more information, see LiteSpeed Variables on page 125.

@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 for disk backups and *7* is the default value for cloud backups.

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. For more information, see Test Optimal Backup Settings on page 83.

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. Test Optimal Backup Settings 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
- 9–MS_AES_128
- 10–MS_AES_192
- 11–MS_AES_256

@database

Name of database to be backed up or restored.

This parameter specifies a database:

- to be backed up (xp_backup_database and xp_slsFastCompression)
- 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. For more information, see LiteSpeed Variables on page 125.

@doubleclick

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

- 1–Creates one Double-Click Restore executable file. Note the following warning: The executable may be greater than 4GB for large databases. Windows Server is unable to run executable files larger than 4GB. However, the file will be convertible/restorable by LiteSpeed file.
- 2–Creates a Double Click Restore loader in the same location. (Default)

For more information, see Double Click Restore Executables on page 120.

@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. If you use the wrong encryption key, the restore will fail.

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'

@excludedatabase

Name of database(s) to exclude from this backup.

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

Tip: The @ExcludeDatabase argument can be applied together with @MultiDatabaseType to exclude several databases from the process.

@expiration

Specifies the date and time when the backup expires. LiteSpeed will not overwrite this file until expiration datetime is passed. This 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

Specifies a backup location (e.g. C:\backups\AdventureWorks.bak). This argument accepts network destinations. You can supply multiple instances of this argument to use stripe backups.

This argument accepts variables. For more information, see LiteSpeed Variables on page 125.

@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–Write new header
- 2–Long erase / write new header
- 3–Low level controller format / write new header

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.

@GSProject

DEPRECATED LiteSpeed 8.8: Was used to store for the Google Cloud Storage project ID; the project ID is now obtained from login. This parameter is retained for compatibility with old backup/restore scripts.

@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 and cloud 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'
```

Network Resilience

@jobp

Specifies an encrypted key. (Similar to @EncryptionKey).

You can use xp_encrypt_backup_key to convert the password (encryption_key) for use with @jobp. The original password (or encrypted key generated by xp_encrypt_restore_key) 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 (or C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs)* (or *C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs*). To log to a different directory add @Trace='logpath=path'.

See Configure Logging in LiteSpeed 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 (default)
- 1–True

@maxtransfersize

Specifies the largest unit of transfer in bytes to be used between SQL Server and LiteSpeed. The possible values are multiples of 65536 bytes (64 KB) ranging up to 4,194,304 bytes (4 MB). The default is 1048576 (1 MB).

@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. For more information, see LiteSpeed Variables on page 125.

@MultiDatabaseType

Produces a backup that includes several types of databases. Types can include: all, system, user, or selected databases.

This argument accepts one of the following values:

- All - Backup all system and user databases.
- System - Backup only system databases.
- User - Backup only user databases.
- Selected - Backup specifically selected databases.

@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 (default)
- 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 (default)
- 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

@ProxyHost

The @ProxyHost argument is optional and specifies the name of the proxy host name that is running the proxy server.

note: If the @ProxyHost argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyLogin

The @ProxyLogin argument is optional and specifies the proxy server login credential.

note: If not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPassword

The @ProxyPassword argument is optional and specifies the proxy server password credential.

note: If the @ProxyPassword argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPasswordEnc

The @ProxyPasswordEnc argument is optional and specifies the encrypted proxy server password credential.

note: If the @ProxyPasswordEnc argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPort

The @ProxyPort argument is optional and contains the port number of the proxy server. The TCP/IP port values can be 1-65535.

note: If the @ProxyPort argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@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.
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	Specifies the largest unit of transfer in bytes to be used between SQL Server and LiteSpeed. The possible values are multiples of 65536 bytes (64 KB) ranging up to 4,194,304 bytes (4 MB). The default is 1048576 (1 MB) .
BaseSize	int	The smallest chunk of memory LiteSpeed attempts to write to disk at any

Column Name	Data Type	Description
		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 and overwrites the backup that has not expired.

- 0–False (default)
- 1–True

@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. For more information, see Adaptive Compression on page 122.

@tsmarchive

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

- 0–False (default)
- 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.

You can use the %TSMDEFAULTPATH% variable to make LiteSpeed detect the default TSM configuration file path automatically (by getting from LiteSpeed defaults as a priority or the registry - HKEY_LOCAL_MACHINE\SOFTWARE\IBM\ADSM\CurrentVersion\BackupClient)

@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

@UseSSL

The @UseSSL argument specifies that the connection uses SSL security. This argument accepts one of the following values:

- 0–Do not use SSL
- 1–Use SSL (default)

@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 (default)
- 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. xp_restore_verifyonly

@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. **NOTE:** 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.

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)

```
EXEC master.dbo.xp_backup_log
@database = 'database_name'
(, @filename = 'backup_file') [,...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']
[, @verify = 0 | 1 ]
[, @returndetails = 0 | 1 ]
```

Back Up Log (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 ]
```

Back Up Log (Microsoft Azure)

```
EXEC master.dbo.xp_backup_log

@database = N'model',

@backupname = N'model - Transaction Log Backup',

@desc = N'Transaction Log Backup of model on %Y-%m-%d %I:%M:%S %p',

@compressionlevel = 7,

@filename = N'test\test.bak',

@CloudVendor = N'AzureBlob',

@CloudAccessKeyEnc = N'*************',

@CloudSecretKeyEnc = N'*******',

@CloudBucketName = N'test',

@AzureBlobType = N'Page',
```

```
@UseSSL = 1,

@init = 0,@with = N'STATS = 10'
```

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 or Disk IO. For more information, see Compression Methods on page 121.

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. For more information, see Adaptive Compression on page 122.

@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.

@AWSAccessKey

The @AWSAccessKey argument specifies the name of the unique Amazon Web Service alphanumeric access key that identifies each user.

Important: This @AWSAccessKey argument is replaced by @CloudAccessKey. The @AWSAccessKey argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSAccessKeyEnc

The @AWSAccessKeyEnc argument specifies the name of the encrypted unique Amazon Web Service alphanumeric access key that identifies each user.

Important: This @AWSAccessKeyEnc argument is replaced by @CloudAccessKeyEnc. The @AWSAccessKeyEnc argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSBucketName

The @AWSBucketName argument specifies the name of the container for AWS objects. Bucket names must be at least 3 and no more than 63 characters long.

Important: This @AWSBucketName argument is replaced by @CloudBucketName. The @AWSBucketName argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSMaxParts

Is the number of parts that are simultaneously uploaded to Amazon S3. The LiteSpeed default is 3 parts. The number of parts can be up to 5 if there is enough memory available during the upload. If you override this parameter, you may impact memory usage.

Important: This @AWSMaxParts argument is replaced by @CloudParallelUpload. The @AWSMaxParts argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSPartSize

The size of each part that is uploaded to Amazon S3 (in MB). The LiteSpeed default for Part Size is calculated as a database size divided into 9,000. The default Part Size = 25MB.

Important: This @AWSPartSize argument is replaced by @CloudPartSize. The @AWSPartSize argument is no longer valid in subsequent LiteSpeed versions after 8.2.

Notes:

- Amazon S3 has a maximum allowable 10,000 parts per file. If you override this parameter, you may inadvertently go over the 10,000 limit.
- Minimum and maximum values for Part Size are defined by Amazon S3: 5MB and 5120MB (5GB) relatively.
- The maximum object size is 5TB.

TIP: Quest Software recommends using LiteSpeed defaults.

@AWSRegionName

The @AWSRegionName argument specifies the name of the Amazon Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-west-2, us-west-1, eu-west-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, and sa-east-1.

Important: This @AWSRegionName argument is replaced by @CloudRegionName. The @AWSRegionName argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSSecretKey

The @AWSSecretKey argument specifies the name of the Amazon Web Service secret key that is assigned when you initially get an AWS account.

Important: This @AWSSecretKey argument is replaced by @CloudSecretKey. The @AWSSecretKey argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSSecretKeyEnc

The @AWSSecretKeyEnc argument specifies the name of the encrypted Amazon Web Service secret key that is assigned when you initially get an AWS account.

Important: This @AWSSecretKeyEnc argument is replaced by @CloudSecretKeyEnc. The @AWSSecretKeyEnc is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSUseGovCloud

The @AWSUseGovCloud argument enables a special restricted region for the US Government use in Amazon S3. This argument accepts one of the following values:

- 0–Do not use government cloud
- 1–Use government cloud

Important: This @AWSUseGovCloud argument is replaced by @CloudGovRegion. The @AWSUseGovCloud argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSUseReducedRedundancy

The @AWSUseReducedRedundancy argument specifies the use of reduced redundancy storage in Amazon S3. This argument accepts one of the following values:

- 0–Do not use reduced redundancy storage
- 1–Use reduced redundancy storage

Note: This @AWSUseReducedRedundancy argument is replaced with the @CloudStorageClass = 'rrs' argument.

@AWSUseServerSideEncryption

The @AWSUseServerSideEncryption argument enables the encryption of data stored at rest in Amazon S3. This argument accepts one of the following values:

- 0–Do not use Server Side Encryption
- 1–Use Server Side Encryption

@AzureBlobType

The @AzureBlobType argument specifies the types of blobs that can be stored in the Microsoft Azure cloud storage. This argument accepts one of the following values: "Block", "Page".

note: The LiteSpeed auto striping logic used in the @CloudAutoStriping and @CloudAutoStripingThreshold parameters can override the Azure blob limit for LiteSpeed backups.

@backupname

Specifies the name of the backup set.

This argument accepts variables. For more information, see LiteSpeed Variables on page 125.

@buffercount

Specifies the number of SQL Server buffers available for a LiteSpeed operation. The default value is set by SQL Server.

@CloudAccessKey

The @CloudAccessKey argument specifies the name of the unique Cloud Web Service alphanumeric access key that identifies each user. The selections include Amazon Access Key, Azure Account Name, Google e-mail styled account.

@CloudAccessKeyEnc

The @CloudAccessKeyEnc argument specifies the name of the encrypted unique Cloud Web Service alphanumeric access key that identifies each user.

@CloudAutoStriping

This parameter enables automatic file striping for LiteSpeed cloud backups.

@CloudAutoStripingThreshold

This parameter contains the stripe size in GBs. LiteSpeed logic uses the database size to make a decision about the number of stripes needed for LiteSpeed cloud backups. For example, if you have a database with a size of 200GB and set @CloudAutoStripingThreshold = 50, then LiteSpeed uses 200/50 = 4 stripes.

@CloudBucketName

The @CloudBucketName argument specifies the name of the container for cloud objects. Bucket names must be at least 3 and no more than 63 characters long. The selections are Amazon Bucket Name, Azure Container Name, Google Bucket Name. Google Bucket Name requirements are described at https://cloud.google.com/storage/docs/naming.

@CloudGovRegion

The @CloudGovRegion argument enables a special restricted region for the US Government use in Amazon S3 and Azure Clouds. This argument accepts one of the following values:

- 0–Do not use government cloud (default)
- 1–Use government cloud

@CloudParallelUpload

The @CloudParallelUpload argument, parallel parts transfers, is used to create fast uploads to the Azure Cloud or Amazon S3. The default number of parallel uploads:

- Amazon S3 = 3
- Azure Blob = 20

@CloudPartSize

The @CloudPartSize argument determines the size of each part that is uploaded to the cloud. The default part size:

- Amazon S3 = 25MB
- Azure Blob = 4MB

notes:

- Minimum part size for Azure Blob = 4MB
- Minimum part size for Amazon S3 = 5MB

TIP: Quest Software recommends using LiteSpeed defaults.

The @CloudBucketName argument specifies the name of the container for cloud objects. Bucket names must be at least 3 and no more than 63 characters long. The selections are Amazon Bucket Name, Azure Container Name, Google Bucket Name. Google Bucket Name requirements are described at https://cloud.google.com/storage/docs/naming.

@CloudRegionName

The @CloudRegionName argument specifies the name of the Cloud Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-east-2, us-west-1, us-west-2, ca-central-1, eu-central-1, eu-west-1, eu-west-2, ap-south-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, ap-northeast-2, sa-east-1, N'Germany' and N'China'.

@CloudSecretKey

The @CloudSecretKey argument specifies the name of the Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudSecretKeyEnc

The @CloudSecretKeyEnc argument specifies the name of the encrypted Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudStorageClass

The @CloudStorageClass argument specifies a range of storage classes established for different use cases including:

For Amazon S3:

- Standard: Standard storage - for general-purpose storage of frequently accessed data.
- Standard-IA: Standard Infrequent Access - for long-lived, but less frequently accessed data.
- RRS: Reduced Redundancy Storage - for non-critical data considering lower level of redundancy rather than Standard storage.

Important: : In versions less than 8.5 you should use --AWSStorageClass. The @AWSStorageClass argument is no longer valid in subsequent LiteSpeed versions after 8.5.

For Google Storage:

- Multi_regional - for frequently accessed data around the world as per serving website content, streaming videos, or gaming and mobile applications.
- Regional - for frequently accessed data in the same region as your Google Cloud DataProc or the Google Compute Engine instances that use it, as per data analytics.
- Nearline - for infrequently accessed data (data you expect to access no more than once per month).
- Coldline - for infrequently accessed data (data you expect to access no more than once per year).

@CloudVendor

The @CloudVendor argument specifies the name of the cloud service provider. The argument accepts one of the following values: "AmazonS3", "AzureBlob" or "GoogleStorage".

@comment

Appends a user comment to the backup.

This argument accepts variables. For more information, see LiteSpeed Variables on page 125.

@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 for disk backups and *7* is the default value for cloud backups.

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. For more information, see Test Optimal Backup Settings on page 83.

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. Test Optimal Backup Settings 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
- 9–MS_AES_128
- 10–MS_AES_192
- 11–MS_AES_256

@database

Name of database to be backed up or restored.

This parameter specifies a database:

- to be backed up (xp_backup_database and xp_slsFastCompression)
- 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. For more information, see LiteSpeed Variables on page 125.

@doubleclick

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

- 1–Creates one Double-Click Restore executable file. Note the following warning: The executable may be greater than 4GB for large databases. Windows Server is unable to run executable files larger than 4GB. However, the file will be convertible/restorable by LiteSpeed file.
- 2–Creates a Double Click Restore loader in the same location. (Default)

For more information, see Double Click Restore Executables on page 120.

@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. If you use the wrong encryption key, the restore will fail.

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'

@excludedatabase

Name of database(s) to exclude from this backup.

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

Tip: The @ExcludeDatabase argument can be applied together with @MultiDatabaseType to exclude several databases from the process.

@expiration

Specifies the date and time when the backup expires. LiteSpeed will not overwrite this file until expiration datetime is passed. This 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

Specifies a backup location (e.g. C:\backups\AdventureWorks.bak). This argument accepts network destinations. You can supply multiple instances of this argument to use stripe backups.

This argument accepts variables. For more information, see LiteSpeed Variables on page 125.

@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–Write new header
- 2–Long erase / write new header
- 3–Low level controller format / write new header

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.

@GSProject

DEPRECATED LiteSpeed 8.8: Was used to store for the Google Cloud Storage project ID; the project ID is now obtained from login. This parameter is retained for compatibility with old backup/restore scripts.

@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 and cloud 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'
```

Network Resilience

@jobp

Specifies an encrypted key. (Similar to @EncryptionKey).

You can use xp_encrypt_backup_key to convert the password (encryption_key) for use with @jobp. The original password (or encrypted key generated by xp_encrypt_restore_key) 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 (or C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs)* (or *C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs*). To log to a different directory add @Trace='logpath=path'.

See Configure Logging in LiteSpeed 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 (default)
- 1–True

@maxtransfersize

Specifies the largest unit of transfer in bytes to be used between SQL Server and LiteSpeed. The possible values are multiples of 65536 bytes (64 KB) ranging up to 4,194,304 bytes (4 MB). The default is 1048576 (1 MB).

@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. For more information, see LiteSpeed Variables on page 125.

@MultiDatabaseType

Produces a backup that includes several types of databases. Types can include: all, system, user, or selected databases.

This argument accepts one of the following values:

- All - Backup all system and user databases.
- System - Backup only system databases.
- User - Backup only user databases.
- Selected - Backup specifically selected databases.

@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 (default)
- 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.

@ProxyHost

The @ProxyHost argument is optional and specifies the name of the proxy host name that is running the proxy server.

note: If the @ProxyHost argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyLogin

The @ProxyLogin argument is optional and specifies the proxy server login credential.

note: If not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPassword

The @ProxyPassword argument is optional and specifies the proxy server password credential.

note: If the @ProxyPassword argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPasswordEnc

The @ProxyPasswordEnc argument is optional and specifies the encrypted proxy server password credential.

note: If the @ProxyPasswordEnc argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPort

The @ProxyPort argument is optional and contains the port number of the proxy server. The TCP/IP port values can be 1-65535.

note: If the @ProxyPort argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@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.
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	Specifies the largest unit of transfer in bytes to be used between SQL Server and LiteSpeed. The possible values are multiples of 65536 bytes (64 KB) ranging up to 4,194,304 bytes (4 MB). The default is 1048576 (1 MB) .
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 and overwrites the backup that has not expired.

- 0–False (default)
- 1–True

@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. For more information, see Adaptive Compression on page 122.

@tsmarchive

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

- 0–False (default)
- 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.

You can use the %TSMDEFAULTPATH% variable to make LiteSpeed detect the default TSM configuration file path automatically (by getting from LiteSpeed defaults as a priority or the registry - HKEY_LOCAL_MACHINE\SOFTWARE\IBM\ADSM\CurrentVersion\BackupClient)

@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

@UseSSL

The @UseSSL argument specifies that the connection uses SSL security. This argument accepts one of the following values:

- 0–Do not use SSL
- 1–Use SSL (default)

@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 (default)
- 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. xp_restore_verifyonly

@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. **NOTE:** 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_backup_parameters

Enter parameters to connect to Cloud storage. Use in conjunction with xp_backup_database.

Syntax

Enter parameters to connect to Cloud storage:

```
declare @mirrorId1 as nvarchar(100)
exec master.dbo.xp_backup_parameters
@FileName = @mirrorId1 OUTPUT,
@CloudVendor = N'AmazonS3',
@CloudBucketName = N'bucketname',
@CloudAccessKeyEnc = N'***',
@CloudSecretKeyEnc = N'***',
@CloudRegionName = N'us-east-1',
@UseSSL = 1
```

Run a backup:

```
declare @mirrorPath1 nvarchar(1024)
set @mirrorPath1 = @mirrorId1 + N':' + N''
exec master.dbo.xp_backup_database
@database = N'TestDatabase',
@backupname = N'TestDatabase - Full Database Backup',
@desc = N'Full Backup of TestDatabase on %Y-%m-%d %I:%M:%S %p',
@compressionlevel = 2,
@filename = N'c:\backup\TestDatabase_Full_1503360019.bak',
@init = 1,
@mirror = @mirrorPath1,
@OLRMAP = 1 ,
@with = N'STATS = 10'
```

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>

@AzureBlobType

The @AzureBlobType argument specifies the types of blobs that can be stored in the Microsoft Azure cloud storage. This argument accepts one of the following values: "Block", "Page".

note: The LiteSpeed auto striping logic used in the @CloudAutoStriping and @CloudAutoStripingThreshold parameters can override the Azure blob limit for LiteSpeed backups.

@CloudAccessKey

The @CloudAccessKey argument specifies the name of the unique Cloud Web Service alphanumeric access key that identifies each user. The selections include Amazon Access Key, Azure Account Name, Google e-mail styled account.

@CloudAccessKeyEnc

The @CloudAccessKeyEnc argument specifies the name of the encrypted unique Cloud Web Service alphanumeric access key that identifies each user.

@CloudAutoStriping

This parameter enables automatic file striping for LiteSpeed cloud backups.

@CloudAutoStripingThreshold

This parameter contains the stripe size in GBs. LiteSpeed logic uses the database size to make a decision about the number of stripes needed for LiteSpeed cloud backups. For example, if you have a database with a size of 200GB and set @CloudAutoStripingThreshold = 50, then LiteSpeed uses 200/50 = 4 stripes.

@CloudBucketName

The @CloudBucketName argument specifies the name of the container for cloud objects. Bucket names must be at least 3 and no more than 63 characters long. The selections are Amazon Bucket Name, Azure Container Name, Google Bucket Name. Google Bucket Name requirements are described at https://cloud.google.com/storage/docs/naming.

@CloudGovRegion

The @CloudGovRegion argument enables a special restricted region for the US Government use in Amazon S3 and Azure Clouds. This argument accepts one of the following values:

- 0–Do not use government cloud (default)
- 1–Use government cloud

@CloudParallelUpload

The @CloudParallelUpload argument, parallel parts transfers, is used to create fast uploads to the Azure Cloud or Amazon S3. The default number of parallel uploads:

- Amazon S3 = 3
- Azure Blob = 20

@CloudPartSize

The @CloudPartSize argument determines the size of each part that is uploaded to the cloud. The default part size:

- Amazon S3 = 25MB
- Azure Blob = 4MB

notes:

- Minimum part size for Azure Blob = 4MB
- Minimum part size for Amazon S3 = 5MB

TIP: Quest Software recommends using LiteSpeed defaults.

@CloudRegionName

The @CloudRegionName argument specifies the name of the Cloud Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-east-2, us-west-1, us-west-2, ca-central-1, eu-central-1, eu-west-1, eu-west-2, ap-south-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, ap-northeast-2, sa-east-1, N'Germany' and N'China'.

@CloudSecretKey

The @CloudSecretKey argument specifies the name of the Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudSecretKeyEnc

The @CloudSecretKeyEnc argument specifies the name of the encrypted Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudStorageClass

The @CloudStorageClass argument specifies a range of storage classes established for different use cases including:

For Amazon S3:

- Standard: Standard storage - for general-purpose storage of frequently accessed data.
- Standard-IA: Standard Infrequent Access - for long-lived, but less frequently accessed data.
- RRS: Reduced Redundancy Storage - for non-critical data considering lower level of redundancy rather than Standard storage.

Important: : In versions less than 8.5 you should use --AWSStorageClass. The @AWSStorageClass argument is no longer valid in subsequent LiteSpeed versions after 8.5.

For Google Storage:

- Multi_regional - for frequently accessed data around the world as per serving website content, streaming videos, or gaming and mobile applications.
- Regional - for frequently accessed data in the same region as your Google Cloud DataProc or the Google Compute Engine instances that use it, as per data analytics.
- Nearline - for infrequently accessed data (data you expect to access no more than once per month).
- Coldline - for infrequently accessed data (data you expect to access no more than once per year).

@CloudVendor

The @CloudVendor argument specifies the name of the cloud service provider. The argument accepts one of the following values: "AmazonS3", "AzureBlob" or "GoogleStorage".

@filename

Specifies a backup location (e.g. C:\backups\AdventureWorks.bak). This argument accepts network destinations. You can supply multiple instances of this argument to use stripe backups.

This argument accepts variables. For more information, see LiteSpeed Variables on page 125.

@GSProject

DEPRECATED LiteSpeed 8.8: Was used to store for the Google Cloud Storage project ID; the project ID is now obtained from login. This parameter is retained for compatibility with old backup/restore scripts.

@ProxyHost

The @ProxyHost argument is optional and specifies the name of the proxy host name that is running the proxy server.

> note: If the @ProxyHost argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyLogin

The @ProxyLogin argument is optional and specifies the proxy server login credential.

> note: If not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPassword

The @ProxyPassword argument is optional and specifies the proxy server password credential.

> note: If the @ProxyPassword argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPasswordEnc

The @ProxyPasswordEnc argument is optional and specifies the encrypted proxy server password credential.

> note: If the @ProxyPasswordEnc argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPort

The @ProxyPort argument is optional and contains the port number of the proxy server. The TCP/IP port values can be 1-65535.

note: If the @ProxyPort argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@UseSSL

The @UseSSL argument specifies that the connection uses SSL security. This argument accepts one of the following values:

- 0–Do not use SSL
- 1–Use SSL (default)

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>

@tsmarchive

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

- 0–False (default)
- 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.

You can use the %TSMDEFAULTPATH% variable to make LiteSpeed detect the default TSM configuration file path automatically (by getting from LiteSpeed defaults as a priority or the registry - HKEY_LOCAL_MACHINE\SOFTWARE\IBM\ADSM\CurrentVersion\BackupClient)

@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_extractor

Converts LiteSpeed backups to native SQL Server backups.

Syntax

NOTE: You can replace argument values with variables. For more information, see LiteSpeed Variables on page 125.

xp_extractor

```
exec master..xp_extractor
@FileName = N'C:\Backups\backup.bkp',
@FileNumber = 1,
@Init = 1,
@MTFFile = N'C:\Backups\backup.bkp.bak'
```

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. For more information, see Adaptive Compression on page 122.

@backupfile

The name of the LiteSpeed backup device file to be extracted. This argument accepts network destinations.

For TSM backups and TSM archives, this argument accepts the following formats:

- tsmbkp:<filespace>\<high>\<low>
- tsmarc:<filespace>\<high>\<low>

You can supply multiple instances of this argument.

@backupindex

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.

@basesize

The smallest chunk of memory LiteSpeed attempts to write to disk at any given time.

@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. If you use the wrong encryption key, the restore will fail.

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'

@filename

Specifies a backup location (e.g. C:\backups\AdventureWorks.bak). This argument accepts network destinations. You can supply multiple instances of this argument to use stripe backups.

@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.

@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 and cloud operations.

@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 (or C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs)* (or *C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs*). To log to a different directory add @Trace='logpath=path'.

See Configure Logging in LiteSpeed for information about LiteSpeed logging.

@maxtransfersize

Specifies the largest unit of transfer in bytes to be used between SQL Server and LiteSpeed. The possible values are multiples of 65536 bytes (64 KB) ranging up to 4,194,304 bytes (4 MB). The default is 1048576 (1 MB).

@mtffile

Specify the location and name of the Microsoft Tape Format (MSTF) base file.

The extractor utility will create one backup device file for each thread used in a LiteSpeed backup.

The extracted files containing the native SQL Server backup will have the following format: base_file_namex.

Where:

- base_file_name is the specified Microsoft Tape Format base file.
- 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.

NoteS:

- You can specify a network destination for this parameter.
- You only need to specify this parameter once. The extraction utility will create all the necessary files automatically.
- You cannot tell the extraction utility to extract a different number of native SQL Server files. However, you can specify different destinations 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. See example 4 for more information.
- If a full path is not specified, the extracted files will be created in the current directory.

@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

@showhelp

Displays the syntax summary of the LiteSpeed command-line utility.

@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.

@trace

Used by LiteSpeed to activate trace logging.

@tsmarchive

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

- 0–False (default)
- 1–True

@tsmfile

Specifies the TSM 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.

@tsmconfigfile

Specifies the TSM configuration file.

@tsmdevicetimeoutminutes

Specifies how long to wait for a TSM device.

@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.

@tsmpassword

The TSM username password. Passwords are case-sensitive.

@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.

@tsmusername

The TSM username ID.

@vlf

Virtual log files are the unit of truncation for the transaction log.

@vlfmaxsize

The maximum size of virtual log files. The number of virtual log files can grow based on the auto growth settings for the log file and how often the active transactions are written to disk. Too many virtual log files can cause transaction log backups to slow down and can also slow down database recovery.

Examples

Encrypted backup conversion

exec master..xp_extractor
@FileName = N'C:\Backups\backup.bkp',
@FileNumber = 1,
@EncryptionKey = N'password',
@Init = 1,
@MTFFile = N'C:\Backups\backup.bkp.bak'

Striped backups

exec master..xp_extractor
@FileName = N'C:\Backups\backupStripe1.bkp',
@FileName = N'C:\Backups\backupStripe2.bkp',
@FileName = N'C:\Backups\backupStripe3.bkp',
@Init = 1,
@MTFFile = N'C:\Backups\backup.1.bak',
@MTFFile = N'C:\Backups\backup.2.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..<xp_extractor> <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..<xp_extractor> <arguments>, @resultmsg=@rmsg output, @resultcode=@rc
output
select @rc, @rmsg
```

Convert LiteSpeed Backups to SQL Server Backups

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 can restore objects directly from the Cloud. It is recommended to use this in cases where there is a fast connection between OLR and the Cloud.
- You cannot restore objects directly from TSM files or tape backups. For more information, see Object Level Restores from TSM Backups on page 170.
- 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, the **Optimize Object Level Recovery speed** option on the Backup wizard Options page is selected by default to create the index during the backup.
- Objects are recovered as they existed at the time they were backed up. You cannot recover data to a random point in time.

- **Direct mode** - In scenarios where you want the application to work with SQL Server directly using a TCP/IP connection without involving the SQL Server client, you can enable direct mode which significantly improves deployment and configuration of your applications. You can enable and disable the use of direct mode from the the Recover Table Wizard.
- **Tail log processing** - In scenarios when you do not require any transaction log backups and the tail log, you can select to bypass tail log processing. Object Level Recovery operations may work much faster in this case. You can enable and disable bypass tail log processing from the toolbar, and when running the Object Level Recovery Wizard and the Recover Table Wizard.

Syntax

Preview table data

```
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']
[, @LSM = 'option' ]
, @objectname = 'object_name'
[, @destinationserver = 'dest_server_name']
[, @tempdirectory = 'recovery_temp_dir']
[, @disablelogprocessing = 0 | 1 ]
```

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']
[, @LSM = 'option' ]
, @objectname = 'object_name'
, @destinationdatabase = 'database_name'
[, @destinationtable = 'dest_table_name']
[, @prefixtableobjects = N'prefix']
[, @suffixtableobjects = N'suffix']
[, @destinationserver = 'dest_server_name']
[, @tempdirectory = 'recovery_temp_dir']
[, @onfilegroup = 'table_filegroup_name']
[, @textimageonfilegroup = 'blob_filegroup_name']
[, @disablelogprocessing = 0 | 1 ]
[, @includetableobjects = 'options']
[, @OLRUDT = <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']
[, @LSM = 'option' ]
, @objectname = 'object_name'
, @shipdirectory = 'recovery_ship_dir'
[, @destinationtable = 'dest_table_name']
[, @prefixtableobjects = N'prefix']
[, @suffixtableobjects = N'suffix']
[, @onfilegroup = 'table_filegroup_name']
[, @textimageonfilegroup = 'blob_filegroup_name']
[, @disablelogprocessing = 0 | 1 ]
[, @includetableobjects = 'options']
[, @OLRUDT = <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']
[, @LSM = 'option' ]
, @objectname = 'object_name'
, @destinationfilename = 'csv_file_name'
[, @disablelogprocessing = 0 | 1 ]
[, @includetableobjects = 'options']
```

If the backup is stored in the cloud (Amazon S3) these parameters help us with access

```
[, @CloudBucketName = N'aabucket1']
[, @CloudAccessKey = N'***']
[, @CloudSecretKey = N'***']
[, @CloudRegionName = N'us-west-2']
[, @ProxyHost = N'proxy.sitelocal']
[, @ProxyPort = 8080]
[, @ProxyLogin = N'DOMAIN\***']
[, @ProxyPassword = N'***']
```

If the backup is stored in the cloud (Microsoft Azure) these parameters help us with access

```
@CloudVendor = N'AzureBlob',
@CloudBucketName = N'test',
@CloudAccessKeyEnc = N'*******',
@CloudSecretKeyEnc = N'******',
@UseSSL = 1,
@affinity = 0,
@logging = 0
```

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>

@AsOnDisk

Instructs LiteSpeed to restore an in-memory table as a regular table. This argument accepts one of the following values:

- 0–False (Default)
- 1–True

@backend

Object Level Recovery can restore tables using two different internal techniques to handle the record inserts.

The first and default method uses BCP files and a TSQL BULK INSERT statement. Object Level Recovery will write a BCP format file and an accompanying binary data file to the local file system. These files may become very large depending on the table size and will require permissions to write to a temporary directory. The default TEMP location can be set by using the @TempDirectory parameter or by setting a permanent temp location in the LiteSpeed configuration file.

An alternate insertion method can be specified to use Sql Server's Sql Native Client capabilities. This method inserts row-data directly into the destination database bypassing any storage on the local file-system. To enable this method, use the parameter `@backend='SQLNativeClient'` (or `-b 1` from the command line). To make this the default method set the value "BackEnd=SQLNativeClient" in the Object Level Recovery section of the LiteSpeed configuration file.

Regardless of the insertion method used, the batch size can be globally managed by setting the value "BulkImportBatchSize=<N>". This will set the number of row inserts for each batched transaction.

@CloudAccessKey

The @CloudAccessKey argument specifies the name of the unique Cloud Web Service alphanumeric access key that identifies each user. The selections include Amazon Access Key, Azure Account Name, Google e-mail styled account.

@CloudAccessKeyEnc

The @CloudAccessKeyEnc argument specifies the name of the encrypted unique Cloud Web Service alphanumeric access key that identifies each user.

@CloudBucketName

The @CloudBucketName argument specifies the name of the container for cloud objects. Bucket names must be at least 3 and no more than 63 characters long. The selections are Amazon Bucket Name, Azure Container Name, Google Bucket Name. Google Bucket Name requirements are described at https://cloud.google.com/storage/docs/naming.

@CloudGovRegion

The @CloudGovRegion argument enables a special restricted region for the US Government use in Amazon S3 and Azure Clouds. This argument accepts one of the following values:

- 0–Do not use government cloud (default)
- 1–Use government cloud

@CloudRegionName

The @CloudRegionName argument specifies the name of the Cloud Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-east-2, us-west-1, us-west-2, ca-central-1, eu-central-1, eu-west-1, eu-west-2, ap-south-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, ap-northeast-2, sa-east-1, N'Germany' and N'China'.

@CloudSecretKey

The @CloudSecretKey argument specifies the name of the Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudSecretKeyEnc

The @CloudSecretKeyEnc argument specifies the name of the encrypted Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudVendor

The @CloudVendor argument specifies the name of the cloud service provider. The argument accepts one of the following values: "AmazonS3", "AzureBlob" or "GoogleStorage".

@destinationdatabase

Specifies the destination database.

@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

Specifies the name of the destination table. LiteSpeed will not overwrite an existing table. If you select the same server instance and database as the original table, you must use a different table name.

NOTE: For Execute-Select operations, LiteSpeed will attempt to insert (append) all selected records into existing table.

@diffencryptionkey

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. If you use the wrong encryption key, the restore will fail.

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'

Equivalent to @encryptionkey, but used for differential backups instead of full backup files.

@difffilename

Name of backup file to restore. Used for differential backups instead of full backup files. You can supply multiple instances of this argument.

@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 may improve read and recovery times. 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.

@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. If you use the wrong encryption key, the restore will fail.

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'

@filename

Specifies a backup location (e.g. C:\backups\AdventureWorks.bak). This argument accepts network destinations. You can supply multiple instances of this argument to use stripe backups.

@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.

@FilestreamOnFileGroup

Specifies a file stream filegroup to include in the object restore.

@includetableobjects

Instructs LiteSpeed to script or recover one or more of the following:

- Constraints–But not foreign keys
- ForeignKeys
- Indexes
- Statistics
- Triggers
- All–All of the above

The value is a list of options, separated with commas.

@logencryptionkey

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. If you use the wrong encryption key, the restore will fail.

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'

Equivalent to @encryptionkey, but used for transaction log backups.

@logfilename

Specifies location and name of the log backup file. You can supply multiple instances of this argument.

Syntax

@logfilenumber

Identifies the log backup file within the backup set. Equivalent to @filenumber, but used for log backups.

@LSM

Specifies handling for OLR LSM mapfile(s).

- Create–Reads backup and creates a new mapfile. It will ignore attached LSM.
- Keep–Does not delete mapfile(s) when complete.
- Delete–Always deletes mapfile(s) when complete.

@objectname

Specifies the name of the object to recover.

@OLRUDT

Create table script:

- 0–Off. Create table with native types, if possible; othervise (CLR UDT) create with UDT. (Default).
- 1–On. Create table with UDT.

@onfilegroup

Filegroup with the object to restore.

@PersistLogProcessing

Instructs LiteSpeed to persist log processing, so the same database backup does not have to be processed for each Object Level Recovery operation. This argument accepts one of the following values:

- 0–False (Default).
- 1–True. LiteSpeed will persist transaction log backups specified and the tail log for future use. This option can offer a huge performance gain for working with databases with large tail logs that could possibly take a long time to process.

@prefixtableobjects

Adds a prefix to the names of the table's objects you selected to script or recover.

@ProxyHost

The @ProxyHost argument is optional and specifies the name of the proxy host name that is running the proxy server.

note: If the @ProxyHost argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyLogin

The @ProxyLogin argument is optional and specifies the proxy server login credential.

note: If not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPassword

The @ProxyPassword argument is optional and specifies the proxy server password credential.

note: If the @ProxyPassword argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPasswordEnc

The @ProxyPasswordEnc argument is optional and specifies the encrypted proxy server password credential.

note: If the @ProxyPasswordEnc argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPort

The @ProxyPort argument is optional and contains the port number of the proxy server. The TCP/IP port values can be 1-65535.

note: If the @ProxyPort argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@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.

@Status_FileName

Specifies the status of a backup location.

@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.

@suffixtableobjects

Adds a suffix to the names of the table's objects you selected to script or recover.

@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\SQL Server\LiteSpeedSettings.ini*.)

@textimageonfilegroup

Destination TEXTIMAGE_ON filegroup name. Used to restore a BLOB (binary large object).

@UseSSL

The @UseSSL argument specifies that the connection uses SSL security. This argument accepts one of the following values:

- 0–Do not use SSL
- 1–Use SSL (default)

@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

Parameter	Description
	replication with log shipping). You cannot specify this parameter with NORECOVERY.

MOVE	`MOVE = ''logical_file_name'' TO ''operating_system_file_name''`
	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.
	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.
	Example:
	`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'''`
	`, @with = 'MOVE ''MyDB_Log'' TO ''C:\MSSQL\Data\MyDB_log.LDF'''`
	NOTE: Use xp_restore_filelistonly to obtain a list of the logical files from the backup set. xp_restore_filelistonly

NORECOVERY	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). **NOTE:** When specifying the NORECOVERY option, the database is not usable in this intermediate, non-recovered state. 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: • a restore script is being run and the log is always being applied. • a sequence of file restores is used and the database is not intended to be usable between two of the restore operations.
PARTIAL	Specifies a partial restore operation. 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.
RECOVERY	Instructs the restore operation to roll back any uncommitted transactions. After the recovery

Parameter	Description
	process, the database is ready for use. If subsequent LiteSpeed restore operations (xp_restore_log or xp_restore_database from differential) are planned, NORECOVERY or STANDBY should be specified instead. If NORECOVERY, RECOVERY, or STANDBY is not specified, RECOVERY is the default. 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.
REPLACE	Instructs LiteSpeed to create the specified database and its related files even if another database already exists with the same name. The existing database is deleted. When the this option is not specified, LiteSpeed performs a check to ensure that the database is not restored to the current server if: • the database named in the xp_restore_database statement already exists on the current server, and • the database name is different from the database name recorded in the LiteSpeed backup set. LiteSpeed will overwrite an existing file which cannot be verified as belonging to the database being restored. Normally, LiteSpeed will refuse to overwrite pre-existing files.
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.
STANDBY	`STANDBY = ''undo_file_name''` 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. 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. If the specified undo file name does not exist, LiteSpeed creates it. If the file does exist, LiteSpeed overwrites it. The same undo file can be used for consecutive LiteSpeed restores of the same database. **NOTE:** If free disk space is exhausted on the drive containing the specified undo file name, the LiteSpeed restore operation stops.
STATS	Displays a message each time a percentage of the activity completes. The default is 10%.
CHECKSUM	Causes checksums to be verified when a LiteSpeed backup is created. **NOTE:** 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.

Examples

Preview a table from a full backup file

```
EXEC master.dbo.xp_objectrecovery
@filename='C:\MSSQL\Backup\MyDB_Backup.BAK',
@filenumber = 1,
@objectname='dbo.Customers'
```

Restore a table from a full backup file into a database

```
EXEC master.dbo.xp_objectrecovery
@filename = 'C:\MSSQL\Backup\MyDB_Backup.BAK'
, @filenumber = 1,
, @objectname = 'dbo.Customers'
, @destinationdatabase = 'tempdb'
```

Restore multiple tables and tables' constraints and indexes

```
exec xp_objectrecovery
@FileName = N'C:\temp\HiServ_full.bak',
@FileNumber = 1,
@ObjectName = N'dbo.Roles',
@DestinationTable = N'[dbo].[Roles]',
@DestinationDatabase = N'HiServ',
@DestinationServer = N'w2k3-22\LITESPEED',
@IncludeTableObjects = N'indexes, constraints, foreignKeys'
```

```
exec xp_objectrecovery
@FileName = N'C:\temp\HiServ_full.bak',
@FileNumber = 1,
@ObjectName = N'dbo.Employees',
@DestinationTable = N'[dbo].[Employees]',
@DestinationDatabase = N'HiServ',
@DestinationServer = N'w2k3-22\LITESPEED',
@IncludeTableObjects = N'indexes, constraints, foreignKeys'
```

```
exec xp_objectrecovery
@FileName = N'C:\temp\HiServ_full.bak',
@FileNumber = 1,
@ObjectName = N'dbo.Customers',
@DestinationTable = N'[dbo].[Customers]',
@DestinationDatabase = N'HiServ',
@DestinationServer = N'w2k3-22\LITESPEED',
@IncludeTableObjects = N'indexes, constraints, foreignKeys'
```

```
exec xp_objectrecovery
@FileName = N'C:\temp\HiServ_full.bak',
@FileNumber = 1,
```

```
@ObjectName = N'dbo.Projects',
@DestinationTable = N'[dbo].[Projects]',
@DestinationDatabase = N'HiServ',
@DestinationServer = N'w2k3-22\LITESPEED',
@IncludeTableObjects = N'indexes, constraints, foreignKeys'

exec xp_objectrecovery
@FileName = N'C:\temp\HiServ_full.bak',
@FileNumber = 1,
@ObjectName = N'dbo.Positions',
@DestinationTable = N'[dbo].[Positions]',
@DestinationDatabase = N'HiServ',
@DestinationServer = N'w2k3-22\LITESPEED',
@IncludeTableObjects = N'indexes, constraints, foreignKeys'
```

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).

NOTE:

When using scripts, the message output results are rendered in a multi row, single column result set so other products can programmatically acquire the script without having to parse the message results. The script then remains in the message output and result set locations.Object Level Restores from TSM Backups

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']
[, @LSM = 'option' ]
, @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 ]
[, @includetableobjects = 'options']
[, @prefixtableobjects = N'prefix']
[, @suffixtableobjects = N'suffix']
[, @OLRUDT = <0|1>]
```

Create the DDL script (Amazon S3)

```
EXEC master.dbo.xp_objectrecovery_createscript
@filename = 'backup_file_name'
, @CloudVendor = N'AmazonS3'
, @Database = N'AA_5_restored88'
[, @CloudBucketName = N'aabucket1']
[, @CloudAccessKey = N'***']
[, @CloudSecretKey = N'***']
[, @CloudRegionName = N'us-west-2']
[, @ProxyHost = N'proxy.sitelocal']
[, @ProxyPort = 8080]
[, @ProxyLogin = N'DOMAIN\temp-xyz-MYtester']
[, @ProxyPassword = N'***']
[, @OLRUDT = <0|1>]
```

Create the DDL script (Microsoft Azure)

```
EXEC master.dbo.xp_objectrecovery_createscript
@filename = 'backup_file_name'
@database = N'model' ,
@CloudVendor = N'AzureBlob',
@CloudBucketName = N'test',
@CloudAccessKeyEnc = N'*******',
@CloudSecretKeyEnc = N'******',
@UseSSL = 1,
@affinity = 0,
@logging = 0,
@OLRUDT = 0
```

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>

@AsOnDisk

Instructs LiteSpeed to restore an in-memory table as a regular table. This argument accepts one of the following values:

- 0–False (Default)
- 1–True

@CloudAccessKey

The @CloudAccessKey argument specifies the name of the unique Cloud Web Service alphanumeric access key that identifies each user. The selections include Amazon Access Key, Azure Account Name, Google e-mail styled account.

@CloudAccessKeyEnc

The @CloudAccessKeyEnc argument specifies the name of the encrypted unique Cloud Web Service alphanumeric access key that identifies each user.

@CloudBucketName

The @CloudBucketName argument specifies the name of the container for cloud objects. Bucket names must be at least 3 and no more than 63 characters long. The selections are Amazon Bucket Name, Azure Container Name, Google Bucket Name. Google Bucket Name requirements are described at https://cloud.google.com/storage/docs/naming.

@CloudGovRegion

The @CloudGovRegion argument enables a special restricted region for the US Government use in Amazon S3 and Azure Clouds. This argument accepts one of the following values:

- 0–Do not use government cloud (default)
- 1–Use government cloud

@CloudRegionName

The @CloudRegionName argument specifies the name of the Cloud Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-east-2, us-west-1, us-west-2, ca-central-1, eu-central-1, eu-west-1, eu-west-2, ap-south-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, ap-northeast-2, sa-east-1, N'Germany' and N'China'.

@CloudSecretKey

The @CloudSecretKey argument specifies the name of the Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudSecretKeyEnc

The @CloudSecretKeyEnc argument specifies the name of the encrypted Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudVendor

The @CloudVendor argument specifies the name of the cloud service provider. The argument accepts one of the following values: "AmazonS3", "AzureBlob" or "GoogleStorage".

@diffencryptionkey

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. If you use the wrong encryption key, the restore will fail.

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'

Equivalent to @encryptionkey, but used for differential backups instead of full backup files.

@difffilename

Name of backup file to restore. Used for differential backups instead of full backup files. You can supply multiple instances of this argument.

@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 may improve read and recovery times. 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.

@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. If you use the wrong encryption key, the restore will fail.

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'

@filename

Specifies a backup location (e.g. C:\backups\AdventureWorks.bak). This argument accepts network destinations. You can supply multiple instances of this argument to use stripe backups.

@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.

@FilestreamOnFileGroup

Specifies a file stream filegroup to include in the object restore.

@includetableobjects

Instructs LiteSpeed to script or recover one or more of the following:

- Constraints–But not foreign keys
- ForeignKeys
- Indexes
- Statistics
- Triggers
- All–All of the above

The value is a list of options, separated with commas.

@logencryptionkey

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. If you use the wrong encryption key, the restore will fail.

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'

Equivalent to @encryptionkey, but used for transaction log backups.

@logfilename

Specifies location and name of the log backup file. You can supply multiple instances of this argument.

Syntax

@logfilenumber

Identifies the log backup file within the backup set. Equivalent to @filenumber, but used for log backups.

@LSM

Specifies handling for OLR LSM mapfile(s).

- Create–Reads backup and creates a new mapfile. It will ignore attached LSM.
- Keep–Does not delete mapfile(s) when complete.
- Delete–Always deletes mapfile(s) when complete.

@objectfilename

Identifies a file that contains a list of objects. The format of this file is "ObjectType,ObjectName" per line.

You can create the list using xp_objectrecovery_viewcontents. xp_objectrecovery_viewcontents

@objectname

Specifies the name of the object to recover.

@OLRUDT

Create table script:

- 0–Off. Create table with native types, if possible; othervise (CLR UDT) create with UDT. (Default).
- 1–On. Create table with UDT.

@onfilegroup

Filegroup with the object to restore.

@PersistLogProcessing

Instructs LiteSpeed to persist log processing, so the same database backup does not have to be processed for each Object Level Recovery operation. This argument accepts one of the following values:

- 0–False (Default).
- 1–True. LiteSpeed will persist transaction log backups specified and the tail log for future use. This option can offer a huge performance gain for working with databases with large tail logs that could possibly take a long time to process.

@prefixtableobjects

Adds a prefix to the names of the table's objects you selected to script or recover.

@ProxyHost

The @ProxyHost argument is optional and specifies the name of the proxy host name that is running the proxy server.

note: If the @ProxyHost argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyLogin

The @ProxyLogin argument is optional and specifies the proxy server login credential.

note: If not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPassword

The @ProxyPassword argument is optional and specifies the proxy server password credential.

note: If the @ProxyPassword argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPasswordEnc

The @ProxyPasswordEnc argument is optional and specifies the encrypted proxy server password credential.

note: If the @ProxyPasswordEnc argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPort

The @ProxyPort argument is optional and contains the port number of the proxy server. The TCP/IP port values can be 1-65535.

note: If the @ProxyPort argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@scriptfilename

Name of the script file to save the generated SQL scripts.

@Status_FileName

Specifies the status of a backup location.

@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.

@suffixtableobjects

Adds a suffix to the names of the table's objects you selected to script or recover.

@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 [1, 3]
- Database
- Default
- ExtendedProcedure
- Function
- IndexedView
- MemoryOptimizedTable
- PartitionFunction
- PartitionScheme
- Role [1]
- Rule
- StoredProcedure
- SystemTable
- Table
- TableConstraintClustered [2]
- TableConstraints [2]
- TableForeignKeys [2]
- TableIndexClustered [2]
- TableIndexes [2]
- TableStatistics[2]
- TableTriggers [2]
- Trigger
- Type
- User [1]
- View
- XmlSchemaCollection

Notes:
[1] These values cannot be used to create scripts.
[2] 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 -V or xp_objectrecovery_viewcontents. When one of these values is used with -C or xp_objectrecovery_createscript, @ObjectName (-C) is not the name of the object, but the name of the owning table.
[3] 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.

@UseSSL

The @UseSSL argument specifies that the connection uses SSL security. This argument accepts one of the following values:

- 0–Do not use SSL
- 1–Use SSL (default)

@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	`MOVE = ''logical_file_name'' TO ''operating_system_file_name''`
	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.
	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.
	Example:
	`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'''`
	`, @with = 'MOVE ''MyDB_Log'' TO ''C:\MSSQL\Data\MyDB_log.LDF'''`
	NOTE: Use xp_restore_filelistonly to obtain a list of the logical files from the backup set. xp_restore_filelistonly

NORECOVERY	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). **NOTE:** When specifying the NORECOVERY option, the database is not usable in this intermediate, non-recovered state. 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:

Parameter	Description
	• a restore script is being run and the log is always being applied. • a sequence of file restores is used and the database is not intended to be usable between two of the restore operations.
PARTIAL	Specifies a partial restore operation. 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.
RECOVERY	Instructs the restore operation to roll back any uncommitted transactions. After the recovery process, the database is ready for use. If subsequent LiteSpeed restore operations (xp_restore_log or xp_restore_database from differential) are planned, NORECOVERY or STANDBY should be specified instead. If NORECOVERY, RECOVERY, or STANDBY is not specified, RECOVERY is the default. 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.
REPLACE	Instructs LiteSpeed to create the specified database and its related files even if another database already exists with the same name. The existing database is deleted. When the this option is not specified, LiteSpeed performs a check to ensure that the database is not restored to the current server if: • the database named in the xp_restore_database statement already exists on the current server, and • the database name is different from the database name recorded in the LiteSpeed backup set. LiteSpeed will overwrite an existing file which cannot be verified as belonging to the database being restored. Normally, LiteSpeed will refuse to overwrite pre-existing files.
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.
STANDBY	`STANDBY = ''undo_file_name''` 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. 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. If the specified undo file name does not exist, LiteSpeed creates it. If the file does exist, LiteSpeed overwrites it.

Parameter	Description
	The same undo file can be used for consecutive LiteSpeed restores of the same database. **NOTE:** If free disk space is exhausted on the drive containing the specified undo file name, the LiteSpeed restore operation stops.
STATS	Displays a message each time a percentage of the activity completes. The default is 10%.
CHECKSUM	Causes checksums to be verified when a LiteSpeed backup is created. **NOTE:** 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.

Examples

Generate SQL script to create a database

```
EXEC master.dbo.xp_objectrecovery_createscript
@filename = 'C:\MSSQL\Backup\MyDB_Backup.BAK'
, @type = 'Database'
, @scriptfilename = 'C:\sql\CREATE_DATABASE.sql'
```

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 restore a table, including table's indexes, constraints and foreign keys

```
EXEC master.dbo.xp_objectrecovery_createscript
@filename='C:\MSSQL\Backup\MyDB_Backup.BAK'
, @type='table'
, @objectname='tbl'
, @prefixtableobjects='test_'
, @includetableobjects='indexes, constraints, foreignkeys'
```

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_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.

Syntax

View the SELECT query results

```
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']
[, @LSM = 'option' ]
{, @scripttext = 'script_text' |
, @scriptfilename = 'script_file_name'}
[, @destinationserver = 'dest_server_name']
[, @tempdirectory = 'recovery_temp_dir']
[, @disablelogprocessing = 0 | 1 ]
```

Restore the SELECT query 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']
[, @LSM = 'option' ]
{, @scripttext = 'script_text' |
, @scriptfilename = 'script_file_name'}
, @destinationtable = 'dest_table_name'
[, @prefixtableobjects = N'prefix']
[, @suffixtableobjects = N'suffix']
, @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 query results into a 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']
[, @LSM = 'option' ]
{, @scripttext = 'script_text' |
, @scriptfilename = 'script_file_name'}
, @destinationtable = 'dest_table_name'
[, @prefixtableobjects = N'prefix']
[, @suffixtableobjects = N'suffix']
, @shipdirectory = 'recovery_ship_dir'
[, @onfilegroup = 'table_filegroup_name']
[, @textimageonfilegroup = 'blob_filegroup_name']
[, @disablelogprocessing = 0 | 1 ]
```

Recover the SELECT query 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']
[, @LSM = 'option' ]
{, @scripttext = 'script_text' |
, @scriptfilename = 'script_file_name'}
, @destinationfilename = 'csv_file_name'
[, @disablelogprocessing = 0 | 1 ]
```

If the backup is stored in the cloud (Amazon S3) these parameters help us with access

```
[, @CloudBucketName = N'aabucket1']
[, @CloudAccessKey = N'***']
[, @CloudSecretKey = N'***']
[, @CloudRegionName = N'us-west-2']
[, @ProxyHost = N'proxy.sitelocal']
[, @ProxyPort = 8080]
[, @ProxyLogin = N'DOMAIN\temp-xyz-MYtester']
[, @ProxyPassword = N'***']
```

If the backup is stored in the cloud (Microsoft Azure) these parameters help us with access

```
@CloudVendor = N'AzureBlob',
@CloudBucketName = N'test',
@CloudAccessKeyEnc = N'*******',
@CloudSecretKeyEnc = N'******',
@UseSSL = 1,
@affinity = 0,
@logging = 0
```

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

Object Level Recovery can restore tables using two different internal techniques to handle the record inserts.

The first and default method uses BCP files and a TSQL BULK INSERT statement. Object Level Recovery will write a BCP format file and an accompanying binary data file to the local file system. These files may become very large depending on the table size and will require permissions to write to a temporary directory. The default TEMP location can be set by using the @TempDirectory parameter or by setting a permanent temp location in the LiteSpeed configuration file.

An alternate insertion method can be specified to use Sql Server's Sql Native Client capabilities. This method inserts row-data directly into the destination database bypassing any storage on the local file-system. To enable this method, use the parameter `@backend='SQLNativeClient'` (or `-b 1` from the command line). To make this the default method set the value "BackEnd=SQLNativeClient" in the Object Level Recovery section of the LiteSpeed configuration file.

Regardless of the insertion method used, the batch size can be globally managed by setting the value "BulkImportBatchSize=<N>". This will set the number of row inserts for each batched transaction.

@CloudAccessKey

The @CloudAccessKey argument specifies the name of the unique Cloud Web Service alphanumeric access key that identifies each user. The selections include Amazon Access Key, Azure Account Name, Google e-mail styled account.

@CloudAccessKeyEnc

The @CloudAccessKeyEnc argument specifies the name of the encrypted unique Cloud Web Service alphanumeric access key that identifies each user.

@CloudBucketName

The @CloudBucketName argument specifies the name of the container for cloud objects. Bucket names must be at least 3 and no more than 63 characters long. The selections are Amazon Bucket Name, Azure Container Name, Google Bucket Name. Google Bucket Name requirements are described at https://cloud.google.com/storage/docs/naming.

@CloudGovRegion

The @CloudGovRegion argument enables a special restricted region for the US Government use in Amazon S3 and Azure Clouds. This argument accepts one of the following values:

- 0–Do not use government cloud (default)
- 1–Use government cloud

@CloudRegionName

The @CloudRegionName argument specifies the name of the Cloud Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-east-2, us-west-1, us-west-2, ca-central-1, eu-central-1, eu-west-1, eu-west-2, ap-south-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, ap-northeast-2, sa-east-1, N'Germany' and N'China'.

@CloudSecretKey

The @CloudSecretKey argument specifies the name of the Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudSecretKeyEnc

The @CloudSecretKeyEnc argument specifies the name of the encrypted Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudVendor

The @CloudVendor argument specifies the name of the cloud service provider. The argument accepts one of the following values: "AmazonS3", "AzureBlob" or "GoogleStorage".

@destinationdatabase

Specifies the destination database.

@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

Specifies the name of the destination table. LiteSpeed will not overwrite an existing table. If you select the same server instance and database as the original table, you must use a different table name.

NOTE: For Execute-Select operations, LiteSpeed will attempt to insert (append) all selected records into existing table.

@diffencryptionkey

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. If you use the wrong encryption key, the restore will fail.

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'

Equivalent to @encryptionkey, but used for differential backups instead of full backup files.

@difffilename

Name of backup file to restore. Used for differential backups instead of full backup files. You can supply multiple instances of this argument.

@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 may improve read and recovery times. 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.

@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. If you use the wrong encryption key, the restore will fail.

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'

@filename

Specifies a backup location (e.g. C:\backups\AdventureWorks.bak). This argument accepts network destinations. You can supply multiple instances of this argument to use stripe backups.

@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.

@KeepComputedColumns

Instructs LiteSpeed to keep the computed columns with the object restore. This argument accepts one of the following values:

- 0–False
- 1–True

@logencryptionkey

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. If you use the wrong encryption key, the restore will fail.

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'

Equivalent to @encryptionkey, but used for transaction log backups.

@logfilename

Specifies location and name of the log backup file. You can supply multiple instances of this argument.

Syntax

@logfilenumber

Identifies the log backup file within the backup set. Equivalent to @filenumber, but used for log backups.

@LSM

Specifies handling for OLR LSM mapfile(s).

- Create–Reads backup and creates a new mapfile. It will ignore attached LSM.
- Keep–Does not delete mapfile(s) when complete.
- Delete–Always deletes mapfile(s) when complete.

@OLRUDT

Create table script:

- 0–Off. Create table with native types, if possible; othervise (CLR UDT) create with UDT.
- 1–On. Create table with UDT (Default).

@onfilegroup

Filegroup with the object to restore.

@PersistLogProcessing

Instructs LiteSpeed to persist log processing, so the same database backup does not have to be processed for each Object Level Recovery operation. This argument accepts one of the following values:

- 0–False (Default).
- 1–True. LiteSpeed will persist transaction log backups specified and the tail log for future use. This option can offer a huge performance gain for working with databases with large tail logs that could possibly take a long time to process.

@ProxyHost

The @ProxyHost argument is optional and specifies the name of the proxy host name that is running the proxy server.

note: If the @ProxyHost argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyLogin

The @ProxyLogin argument is optional and specifies the proxy server login credential.

note: If not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPassword

The @ProxyPassword argument is optional and specifies the proxy server password credential.

note: If the @ProxyPassword argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPasswordEnc

The @ProxyPasswordEnc argument is optional and specifies the encrypted proxy server password credential.

note: If the @ProxyPasswordEnc argument is not defined, then the LiteSpeed core engine checks the local

.ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPort

The @ProxyPort argument is optional and contains the port number of the proxy server. The TCP/IP port values can be 1-65535.

note: If the @ProxyPort argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@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\SQL Server\LiteSpeedSettings.ini*.)

@textimageonfilegroup

Destination TEXTIMAGE_ON filegroup name. Used to restore a BLOB (binary large object).

@UseSSL

The @UseSSL argument specifies that the connection uses SSL security. This argument accepts one of the following values:

- 0–Do not use SSL
- 1–Use SSL (default)

@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_	Instructs the restore operation to keep the replication settings when restoring a published

Parameter	Description
REPLICATION	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	`MOVE = ''logical_file_name'' TO ''operating_system_file_name''`
	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.
	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.
	Example:
	`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'''`
	`, @with = 'MOVE ''MyDB_Log'' TO ''C:\MSSQL\Data\MyDB_log.LDF'''`
	NOTE: Use xp_restore_filelistonly to obtain a list of the logical files from the backup set. xp_restore_filelistonly

NORECOVERY	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). **NOTE:** When specifying the NORECOVERY option, the database is not usable in this intermediate, non-recovered state. 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: • a restore script is being run and the log is always being applied. • a sequence of file restores is used and the database is not intended to be usable between two of the restore operations.
PARTIAL	Specifies a partial restore operation. 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.

Parameter	Description
RECOVERY	Instructs the restore operation to roll back any uncommitted transactions. After the recovery process, the database is ready for use. If subsequent LiteSpeed restore operations (xp_restore_log or xp_restore_database from differential) are planned, NORECOVERY or STANDBY should be specified instead. If NORECOVERY, RECOVERY, or STANDBY is not specified, RECOVERY is the default. 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.
REPLACE	Instructs LiteSpeed to create the specified database and its related files even if another database already exists with the same name. The existing database is deleted. When the this option is not specified, LiteSpeed performs a check to ensure that the database is not restored to the current server if: • the database named in the xp_restore_database statement already exists on the current server, and • the database name is different from the database name recorded in the LiteSpeed backup set. LiteSpeed will overwrite an existing file which cannot be verified as belonging to the database being restored. Normally, LiteSpeed will refuse to overwrite pre-existing files.
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.
STANDBY	`STANDBY = ''undo_file_name''` 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. 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. If the specified undo file name does not exist, LiteSpeed creates it. If the file does exist, LiteSpeed overwrites it. The same undo file can be used for consecutive LiteSpeed restores of the same database. **NOTE:** If free disk space is exhausted on the drive containing the specified undo file name, the LiteSpeed restore operation stops.
STATS	Displays a message each time a percentage of the activity completes. The default is 10%.
CHECKSUM	Causes checksums to be verified when a LiteSpeed backup is created. **NOTE:** 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.

Examples

View the SELECT query results

```
xp_objectrecovery_executeselect
@filename = 'D:\temp\LiteSpeedLocal.bak'
, @scripttext = 'select top (4)* from dbo.LiteSpeedActivity'
```

Restore the SELECT query results 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 results 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 results 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 results 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_objectrecovery_viewcontents

Lists the objects within the backup file.

Syntax

Parameters to describe backup file(s)

```
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']
```

If the backup is stored in the cloud (Amazon S3) these parameters help us with access

```
[, @CloudBucketName = N'aabucket1']
[, @CloudAccessKey = N'***']
[, @CloudSecretKey = N'***']
[, @CloudRegionName = N'us-west-2']
[, @ProxyHost = N'proxy.sitelocal']
[, @ProxyPort = 8080]
[, @ProxyLogin = N'DOMAIN\temp-xyz-MYtester']
[, @ProxyPassword = N'***']
```

If the backup is stored in the cloud (Microsoft Azure) these parameters help us with access

```
@CloudVendor = N'AzureBlob',
@CloudBucketName = N'test',
@CloudAccessKeyEnc = N'*******',
@CloudSecretKeyEnc = N'******',
@UseSSL = 1,
@affinity = 0,
@logging = 0
```

Parameters to describe OLR

```
, @Database = N'AA_5_restored88'
[, @type = 'object_type']
[, @disablelogprocessing = 0 | 1 ]
[, @LSM = '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>

@CloudAccessKey

The @CloudAccessKey argument specifies the name of the unique Cloud Web Service alphanumeric access key that identifies each user. The selections include Amazon Access Key, Azure Account Name, Google e-mail styled account.

@CloudAccessKeyEnc

The @CloudAccessKeyEnc argument specifies the name of the encrypted unique Cloud Web Service alphanumeric access key that identifies each user.

@CloudBucketName

The @CloudBucketName argument specifies the name of the container for cloud objects. Bucket names must be at least 3 and no more than 63 characters long. The selections are Amazon Bucket Name, Azure Container Name, Google Bucket Name. Google Bucket Name requirements are described at https://cloud.google.com/storage/docs/naming.

@CloudGovRegion

The @CloudGovRegion argument enables a special restricted region for the US Government use in Amazon S3 and Azure Clouds. This argument accepts one of the following values:

- 0–Do not use government cloud (default)
- 1–Use government cloud

@CloudRegionName

The @CloudRegionName argument specifies the name of the Cloud Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-east-2, us-west-1, us-west-2, ca-central-1, eu-central-1, eu-west-1, eu-west-2, ap-south-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, ap-northeast-2, sa-east-1, N'Germany' and N'China'.

@CloudSecretKey

The @CloudSecretKey argument specifies the name of the Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudSecretKeyEnc

The @CloudSecretKeyEnc argument specifies the name of the encrypted Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudVendor

The @CloudVendor argument specifies the name of the cloud service provider. The argument accepts one of the following values: "AmazonS3", "AzureBlob" or "GoogleStorage".

@diffencryptionkey

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. If you use the wrong encryption key, the restore will fail.

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'

Equivalent to @encryptionkey, but used for differential backups instead of full backup files.

@difffilename

Name of backup file to restore. Used for differential backups instead of full backup files. You can supply multiple instances of this argument.

@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 may improve read and recovery times. 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.

@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. If you use the wrong encryption key, the restore will fail.

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'

@filename

Specifies a backup location (e.g. C:\backups\AdventureWorks.bak). This argument accepts network destinations. You can supply multiple instances of this argument to use stripe backups.

@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

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. If you use the wrong encryption key, the restore will fail.

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'

Equivalent to @encryptionkey, but used for transaction log backups.

@logfilename

Specifies location and name of the log backup file. You can supply multiple instances of this argument.

Syntax

@logfilenumber

Identifies the log backup file within the backup set. Equivalent to @filenumber, but used for log backups.

@LSM

Specifies handling for OLR LSM mapfile(s).

- Create–Reads backup and creates a new mapfile. It will ignore attached LSM.
- Keep–Does not delete mapfile(s) when complete.
- Delete–Always deletes mapfile(s) when complete.

@PersistLogProcessing

Instructs LiteSpeed to persist log processing, so the same database backup does not have to be processed for each Object Level Recovery operation. This argument accepts one of the following values:

- 0–False (Default).
- 1–True. LiteSpeed will persist transaction log backups specified and the tail log for future use. This option can offer a huge performance gain for working with databases with large tail logs that could possibly take a long time to process.

@ProxyHost

The @ProxyHost argument is optional and specifies the name of the proxy host name that is running the proxy server.

note: If the @ProxyHost argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyLogin

The @ProxyLogin argument is optional and specifies the proxy server login credential.

note: If not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPassword

The @ProxyPassword argument is optional and specifies the proxy server password credential.

note: If the @ProxyPassword argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPasswordEnc

The @ProxyPasswordEnc argument is optional and specifies the encrypted proxy server password credential.

note: If the @ProxyPasswordEnc argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPort

The @ProxyPort argument is optional and contains the port number of the proxy server. The TCP/IP port values can be 1-65535.

note: If the @ProxyPort argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@Status_FileName

Specifies the status of a backup location.

@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 [1, 3]
- Database
- Default
- ExtendedProcedure
- Function
- IndexedView
- MemoryOptimizedTable
- PartitionFunction
- PartitionScheme
- Role [1]
- Rule
- StoredProcedure
- SystemTable
- Table
- TableConstraintClustered [2]
- TableConstraints [2]
- TableForeignKeys [2]
- TableIndexClustered [2]
- TableIndexes [2]
- TableStatistics[2]
- TableTriggers [2]
- Trigger
- Type
- User [1]
- View
- XmlSchemaCollection

Notes:
[1] These values cannot be used to create scripts.
[2] 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 -V or xp_objectrecovery_viewcontents. When one of these values is used with -C or xp_objectrecovery_createscript, @ObjectName (-C) is not the name of the object, but the name of the owning table.
[3] 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.

@UseSSL

The @UseSSL argument specifies that the connection uses SSL security. This argument accepts one of the following values:

- 0–Do not use SSL
- 1–Use SSL (default)

@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	`MOVE = ''logical_file_name'' TO ''operating_system_file_name''` 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. 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. Example:

Parameter	Description

	`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'''`
	`, @with = 'MOVE ''MyDB_Log'' TO ''C:\MSSQL\Data\MyDB_log.LDF'''`
	NOTE: Use xp_restore_filelistonly to obtain a list of the logical files from the backup set. xp_restore_filelistonly

NORECOVERY	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). **NOTE:** When specifying the NORECOVERY option, the database is not usable in this intermediate, non-recovered state. 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: • a restore script is being run and the log is always being applied. • a sequence of file restores is used and the database is not intended to be usable between two of the restore operations.
PARTIAL	Specifies a partial restore operation. 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.
RECOVERY	Instructs the restore operation to roll back any uncommitted transactions. After the recovery process, the database is ready for use. If subsequent LiteSpeed restore operations (xp_restore_log or xp_restore_database from differential) are planned, NORECOVERY or STANDBY should be specified instead. If NORECOVERY, RECOVERY, or STANDBY is not specified, RECOVERY is the default. 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.
REPLACE	Instructs LiteSpeed to create the specified database and its related files even if another database already exists with the same name. The existing database is deleted. When the this option is not specified, LiteSpeed performs a check to ensure that the

Parameter	Description
	database is not restored to the current server if: • the database named in the xp_restore_database statement already exists on the current server, and • the database name is different from the database name recorded in the LiteSpeed backup set. LiteSpeed will overwrite an existing file which cannot be verified as belonging to the database being restored. Normally, LiteSpeed will refuse to overwrite pre-existing files.
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.
STANDBY	`STANDBY = ''undo_file_name''` 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. 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. If the specified undo file name does not exist, LiteSpeed creates it. If the file does exist, LiteSpeed overwrites it. The same undo file can be used for consecutive LiteSpeed restores of the same database. **NOTE:** If free disk space is exhausted on the drive containing the specified undo file name, the LiteSpeed restore operation stops.
STATS	Displays a message each time a percentage of the activity completes. The default is 10%.
CHECKSUM	Causes checksums to be verified when a LiteSpeed backup is created. **NOTE:** 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.

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 = N'C:\temp\8_20101007183923.bak'
, @filenumber = 1
, @type = 'All'
, @logfilename = N'C:\temp\8_20101007183923_20101007184136.bak'
, @logfilenumber = 1
, @logfilename = N'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_remove_file

Deletes a backup file from a specified location (disk, TSM, or cloud).

Syntax

Syntax (disk)

```
EXEC master.dbo.xp_remove_file
@filename = N'I:\test\test.bkp'
```

Syntax (TSM)

```
EXEC master.dbo.xp_remove_file
@filename = N'tsmbkp:test\test\backup'
,@TSMUserName = 'nodename'
, @TSMPassword= 'password'
, @TSMconfigfile= 'C:\dsm.opt'
```

Syntax (cloud)

```
EXEC master.dbo.xp_remove_file

@filename = N'test\model.bak',

@CloudVendor = N'AmazonS3',

@CloudBucketName = N'test',

@CloudAccessKeyEnc = N'***************,

@CloudSecretKeyEnc = N'**********',

@CloudRegionName = N'us-west-1',

@UseSSL = 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>
- The command does not support TSM file remove operations.
- Use the slsmedia command to delete a TSM object. See the example below.
  SLSMedia.exe -r tsm:file space\high level\low level --TSMConfigFile C:\ProgramFiles\Tivoli\tsm\api\dsm.opt --tsmclientnode nodename --TSMClientOwnerPwd password

@AWSAccessKey

The @AWSAccessKey argument specifies the name of the unique Amazon Web Service alphanumeric access key that identifies each user.

Important: This @AWSAccessKey argument is replaced by @CloudAccessKey. The @AWSAccessKey argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSAccessKeyEnc

The @AWSAccessKeyEnc argument specifies the name of the encrypted unique Amazon Web Service alphanumeric access key that identifies each user.

Important: This @AWSAccessKeyEnc argument is replaced by @CloudAccessKeyEnc. The @AWSAccessKeyEnc argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSBucketName

The @AWSBucketName argument specifies the name of the container for AWS objects. Bucket names must be at least 3 and no more than 63 characters long.

Important: This @AWSBucketName argument is replaced by @CloudBucketName. The @AWSBucketName argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSRegionName

The @AWSRegionName argument specifies the name of the Amazon Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-west-2, us-west-1, eu-west-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, and sa-east-1.

Important: This @AWSRegionName argument is replaced by @CloudRegionName. The @AWSRegionName argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSSecretKey

The @AWSSecretKey argument specifies the name of the Amazon Web Service secret key that is assigned when you initially get an AWS account.

Important: This @AWSSecretKey argument is replaced by @CloudSecretKey. The @AWSSecretKey argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSSecretKeyEnc

The @AWSSecretKeyEnc argument specifies the name of the encrypted Amazon Web Service secret key that is assigned when you initially get an AWS account.

Important: This @AWSSecretKeyEnc argument is replaced by @CloudSecretKeyEnc. The

@AWSSecretKeyEnc is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSUseGovCloud

The @AWSUseGovCloud argument enables a special restricted region for the US Government use in Amazon S3. This argument accepts one of the following values:

- 0–Do not use government cloud
- 1–Use government cloud

Important: This @AWSUseGovCloud argument is replaced by @CloudGovRegion. The @AWSUseGovCloud argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSUseReducedRedundancy

The @AWSUseReducedRedundancy argument specifies the use of reduced redundancy storage in Amazon S3. This argument accepts one of the following values:

- 0–Do not use reduced redundancy storage
- 1–Use reduced redundancy storage

Note: This @AWSUseReducedRedundancy argument is replaced with the @CloudStorageClass = 'rrs' argument.

@AWSUseServerSideEncryption

The @AWSUseServerSideEncryption argument enables the encryption of data stored at rest in Amazon S3. This argument accepts one of the following values:

- 0–Do not use Server Side Encryption
- 1–Use Server Side Encryption

@CloudAccessKey

The @CloudAccessKey argument specifies the name of the unique Cloud Web Service alphanumeric access key that identifies each user. The selections include Amazon Access Key, Azure Account Name, Google e-mail styled account.

@CloudAccessKeyEnc

The @CloudAccessKeyEnc argument specifies the name of the encrypted unique Cloud Web Service alphanumeric access key that identifies each user.

@CloudBucketName

The @CloudBucketName argument specifies the name of the container for cloud objects. Bucket names must be at least 3 and no more than 63 characters long. The selections are Amazon Bucket Name, Azure Container Name, Google Bucket Name. Google Bucket Name requirements are described at https://cloud.google.com/storage/docs/naming.

@CloudGovRegion

The @CloudGovRegion argument enables a special restricted region for the US Government use in Amazon S3 and Azure Clouds. This argument accepts one of the following values:

- 0–Do not use government cloud (default)
- 1–Use government cloud

@CloudRegionName

The @CloudRegionName argument specifies the name of the Cloud Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-east-2, us-west-1, us-west-2, ca-central-1, eu-central-1, eu-west-1, eu-west-2, ap-south-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, ap-northeast-2, sa-east-1, N'Germany' and N'China'.

@CloudSecretKey

The @CloudSecretKey argument specifies the name of the Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudSecretKeyEnc

The @CloudSecretKeyEnc argument specifies the name of the encrypted Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudVendor

The @CloudVendor argument specifies the name of the cloud service provider. The argument accepts one of the following values: "AmazonS3", "AzureBlob" or "GoogleStorage".

@FileName

Specifies a backup location (e.g. C:\backups\AdventureWorks.bak). This argument accepts network destinations. You can supply multiple instances of this argument to use stripe backups.

@ProxyHost

The @ProxyHost argument is optional and specifies the name of the proxy host name that is running the proxy server.

note: If the @ProxyHost argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyLogin

The @ProxyLogin argument is optional and specifies the proxy server login credential.

note: If not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPassword

The @ProxyPassword argument is optional and specifies the proxy server password credential.

note: If the @ProxyPassword argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPasswordEnc

The @ProxyPasswordEnc argument is optional and specifies the encrypted proxy server password credential.

note: If the @ProxyPasswordEnc argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPort

The @ProxyPort argument is optional and contains the port number of the proxy server. The TCP/IP port values can be 1-65535.

note: If the @ProxyPort argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@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.

@TSMConfigFile

Specifies the TSM configuration file.

@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.

@UseSSL

The @UseSSL argument specifies that the connection uses SSL security. This argument accepts one of the following values:

- 0–Do not use SSL
- 1–Use SSL (default)

Examples

Remove file from Microsoft Azure

```
EXEC master.dbo.xp_remove_file

@filename = N'tst\test.bak',

@CloudVendor = N'AzureBlob',

@CloudBucketName = N'test',

@CloudAccessKeyEnc = N'******',

@CloudSecretKeyEnc = N'********'
```

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_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)

```
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]
```

xp_restore_attachedfilesonly (TSM)

```
EXEC master.dbo.xp_restore_attachedfilesonly

( (@attachedfile = 'pathname [ to new_pathname ]']) [,..n]

, @tsmobject = 'TSM_object' [,..n]

, @tsmconfigfile = 'TSM_configuration_file'

[, @encryptionkey = 'encryption_key']

[, @filenumber = n]

[, @logging = 0 | 1 | 2 ]

[, @affinity = 0..2147483648]

[, @throttle = 1..100]

[, @tsmclientnode = 'TSM_client_node']
```

```
[, @tsmclientownerpwd = 'TSM_client_owner_password']

[, @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]
```

xp_restore_attachedfilesonly (Microsoft Azure)

```
EXEC master.dbo.xp_restore_attachedfilesonly

@filename = N'test\test,bak',

@filenumber = 1,

@CloudVendor = N'AzureBlob',

@CloudBucketName = N'test',

@CloudAccessKeyEnc = N'********',

@CloudSecretKeyEnc = N'********',

@UseSSL = 1,

@affinity = 0,

@logging = 0,

@attachedfile = N'''C:\test.txt'' to ''C:\test_1.txt'''
```

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. For more information, see Adaptive Compression on page 122.

@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.

@AWSAccessKey

The @AWSAccessKey argument specifies the name of the unique Amazon Web Service alphanumeric access key that identifies each user.

> Important: This @AWSAccessKey argument is replaced by @CloudAccessKey. The @AWSAccessKey argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSAccessKeyEnc

The @AWSAccessKeyEnc argument specifies the name of the encrypted unique Amazon Web Service alphanumeric access key that identifies each user.

Important: This @AWSAccessKeyEnc argument is replaced by @CloudAccessKeyEnc. The @AWSAccessKeyEnc argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSBucketName

The @AWSBucketName argument specifies the name of the container for AWS objects. Bucket names must be at least 3 and no more than 63 characters long.

Important: This @AWSBucketName argument is replaced by @CloudBucketName. The @AWSBucketName argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSRegionName

The @AWSRegionName argument specifies the name of the Amazon Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-west-2, us-west-1, eu-west-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, and sa-east-1.

Important: This @AWSRegionName argument is replaced by @CloudRegionName. The @AWSRegionName argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSSecretKey

The @AWSSecretKey argument specifies the name of the Amazon Web Service secret key that is assigned when you initially get an AWS account.

Important: This @AWSSecretKey argument is replaced by @CloudSecretKey. The @AWSSecretKey argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSSecretKeyEnc

The @AWSSecretKeyEnc argument specifies the name of the encrypted Amazon Web Service secret key that is assigned when you initially get an AWS account.

Important: This @AWSSecretKeyEnc argument is replaced by @CloudSecretKeyEnc. The @AWSSecretKeyEnc is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSUseGovCloud

The @AWSUseGovCloud argument enables a special restricted region for the US Government use in Amazon S3. This argument accepts one of the following values:

- 0–Do not use government cloud
- 1–Use government cloud

Important: This @AWSUseGovCloud argument is replaced by @CloudGovRegion. The @AWSUseGovCloud argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSUseReducedRedundancy

The @AWSUseReducedRedundancy argument specifies the use of reduced redundancy storage in Amazon S3. This argument accepts one of the following values:

- 0–Do not use reduced redundancy storage
- 1–Use reduced redundancy storage

Note: This @AWSUseReducedRedundancy argument is replaced with the @CloudStorageClass = 'rrs' argument.

@AWSUseServerSideEncryption

The @AWSUseServerSideEncryption argument enables the encryption of data stored at rest in Amazon S3. This argument accepts one of the following values:

- 0–Do not use Server Side Encryption
- 1–Use Server Side Encryption

@CloudAccessKey

The @CloudAccessKey argument specifies the name of the unique Cloud Web Service alphanumeric access key that identifies each user. The selections include Amazon Access Key, Azure Account Name, Google e-mail styled account.

@CloudAccessKeyEnc

The @CloudAccessKeyEnc argument specifies the name of the encrypted unique Cloud Web Service alphanumeric access key that identifies each user.

@CloudBucketName

The @CloudBucketName argument specifies the name of the container for cloud objects. Bucket names must be at least 3 and no more than 63 characters long. The selections are Amazon Bucket Name, Azure Container Name, Google Bucket Name. Google Bucket Name requirements are described at https://cloud.google.com/storage/docs/naming.

@CloudGovRegion

The @CloudGovRegion argument enables a special restricted region for the US Government use in Amazon S3 and Azure Clouds. This argument accepts one of the following values:

- 0–Do not use government cloud (default)
- 1–Use government cloud

@CloudRegionName

The @CloudRegionName argument specifies the name of the Cloud Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-east-2, us-west-1, us-west-2, ca-central-1, eu-central-1, eu-west-1, eu-west-2, ap-south-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, ap-northeast-2, sa-east-1, N'Germany' and N'China'.

@CloudSecretKey

The @CloudSecretKey argument specifies the name of the Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudSecretKeyEnc

The @CloudSecretKeyEnc argument specifies the name of the encrypted Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudVendor

The @CloudVendor argument specifies the name of the cloud service provider. The argument accepts one of the following values: "AmazonS3", "AzureBlob" or "GoogleStorage".

@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. If you use the wrong encryption key, the restore will fail.

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'

@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

Specifies a backup location (e.g. C:\backups\AdventureWorks.bak). This argument accepts network destinations. You can supply multiple instances of this argument to use stripe backups.

@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 and cloud 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'
```

Network Resilience

@JobP

Specifies an encrypted key. (Similar to @EncryptionKey).

You can use xp_encrypt_backup_key to convert the password (encryption_key) for use with @jobp. The original password (or encrypted key generated by xp_encrypt_restore_key) 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 (or C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs)* (or *C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs*). To log to a different directory add @Trace='logpath=path'.

See Configure Logging in LiteSpeed for information about LiteSpeed logging.

@MaxTransferSize

Specifies the largest unit of transfer in bytes to be used between SQL Server and LiteSpeed. The possible values are multiples of 65536 bytes (64 KB) ranging up to 4,194,304 bytes (4 MB). The default is 1048576 (1 MB).

@ProxyHost

The @ProxyHost argument is optional and specifies the name of the proxy host name that is running the proxy server.

note: If the @ProxyHost argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyLogin

The @ProxyLogin argument is optional and specifies the proxy server login credential.

note: If not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPassword

The @ProxyPassword argument is optional and specifies the proxy server password credential.

note: If the @ProxyPassword argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPasswordEnc

The @ProxyPasswordEnc argument is optional and specifies the encrypted proxy server password credential.

note: If the @ProxyPasswordEnc argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPort

The @ProxyPort argument is optional and contains the port number of the proxy server. The TCP/IP port values can be 1-65535.

note: If the @ProxyPort argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@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. For more information, see Adaptive Compression on page 122.

@TSMarchive

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

- 0–False (default)
- 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

@UseSSL

The @UseSSL argument specifies that the connection uses SSL security. This argument accepts one of the following values:

- 0–Do not use SSL
- 1–Use SSL (default)

Examples

1. Restore a complete directory:

```
EXEC master.dbo.xp_restore_attachedfilesonly

@filename= 'C:\MSSQL\Backup\MyDB_Backup.BAK'

@attachedfile = N'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 = N'''C:\DATA\Images'' to ''c:\DATA\Old_Images'''

@attachedfile = N'''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_automated

Restores the most recent full disk and cloud backup created with LiteSpeed and optionally differential and transaction log backups. You can use this extended stored procedure to automate restore operations even if backup files have unique names.

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

```
EXEC master.dbo.xp_restore_automated
[@database = 'database_name'
[, @datafilepath = 'path']
[, @logfilepath = 'path'] ]
, ( @filename = 'backup_filename' | ( @backuppath = 'path'
, @backupextension = 'extensions'
, @checksubfolders = 0 | 1 ) ) [,...n]
, @sourceserver = 'server_name'
, @sourcedatabase = 'database_name'
, @backuptype = N'option',
[, ( @encryptionkey = 'encryption_key' | @jobp = 'encrypted_key' ) ]
[, @with = 'additional_with_parameters'] [,...n]
[, @withreplace = 0 | 1 ]
[, @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']
[, @restoreasreadonly = 0 | 1]
[, @restoreascompressed = 0 | 1]
[, @dryrun = 0 | 1]
[, @dropdatabaseonfailure = 0 | 1 ]
[, @dropdatabaseonsuccess = 0 | 1 ]
```

xp_restore_automated (Amazon S3)

```
EXEC master.dbo.xp_restore_automated

@database = N'filegroups' ,

@filename = N'test\test.bak',

@sourceserver = N'test\test',
```

```
@sourcedatabase = N'filegroups',

@backuptype = N'diff',

@CloudVendor = N'AmazonS3',

@CloudBucketName = N'test',

@CloudAccessKeyEnc = N'******',

@CloudSecretKeyEnc = N'*******',

@UseSSL = 1,

@CloudGovRegion = 1,

@proxyhost = N'10.1.1.1',

@proxyport = 80,

@proxylogin = N'test',

@ProxyPasswordEnc = N'******',

@affinity = 0,

@logging = 0,

@DontUseReplication = 1,

@withreplace = 1,

@checkdb = 1,

@checkdbphysicalonly = 1,

@checkdbnoindex = 1,

@checkdbnoinfomessages = 1,

@read_write_filegroups = 1,

@returndetails = 1,

@with = N'STATS = 10'
```

xp_restore_automated (Google Cloud Storage)

```
exec master.dbo.xp_restore_automated
@database = N'db-ar'
, @backuppath = N''
, @backupextension = N''
, @checksubfolders = 0
, @sourceserver = N'SOURCE\SERVER'
, @sourcedatabase = N'source-db'
, @backuptype = N'diff'
, @affinity = 0
, @logging = 0
, @DontUseReplication = 1
, @checkdb = 1
, @checkdbphysicalonly = 1
, @checkdbnoindex = 1
, @checkdbnoinfomessages = 1
, @with = N'RECOVERY'
, @with = N'STATS = 10'
, @CloudVendor = N'GoogleStorage'
, @CloudBucketName = N'bucketname'
, @CloudAccessKey = N'***' -- my key'
, @CloudSecretKey = N'***' -- my key'
```

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. For more information, see Adaptive Compression on page 122.

@ARPeriod

Specifies a point in time to restore from where the time is measured in days, hours, minutes and seconds from the restore time.

Set 0's for periods not used.

@ARPeriod = N'DD.HH:MM:SS'

@ARPointInTime

Specifies a point in time to restore from: year, month, day, hours, minutes, seconds.

@ARPointInTime = N'YYYY-MM-DD HH:MM:SS'

@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.

@AWSAccessKey

The @AWSAccessKey argument specifies the name of the unique Amazon Web Service alphanumeric access key that identifies each user.

Important: This @AWSAccessKey argument is replaced by @CloudAccessKey. The @AWSAccessKey argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSAccessKeyEnc

The @AWSAccessKeyEnc argument specifies the name of the encrypted unique Amazon Web Service alphanumeric access key that identifies each user.

Important: This @AWSAccessKeyEnc argument is replaced by @CloudAccessKeyEnc. The @AWSAccessKeyEnc argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSBucketName

The @AWSBucketName argument specifies the name of the container for AWS objects. Bucket names must be at least 3 and no more than 63 characters long.

Important: This @AWSBucketName argument is replaced by @CloudBucketName. The @AWSBucketName argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSRegionName

The @AWSRegionName argument specifies the name of the Amazon Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-west-2, us-west-1, eu-west-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, and sa-east-1.

Important: This @AWSRegionName argument is replaced by @CloudRegionName. The @AWSRegionName argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSSecretKey

The @AWSSecretKey argument specifies the name of the Amazon Web Service secret key that is assigned when you initially get an AWS account.

Important: This @AWSSecretKey argument is replaced by @CloudSecretKey. The @AWSSecretKey argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSSecretKeyEnc

The @AWSSecretKeyEnc argument specifies the name of the encrypted Amazon Web Service secret key that is assigned when you initially get an AWS account.

Important: This @AWSSecretKeyEnc argument is replaced by @CloudSecretKeyEnc. The

@AWSSecretKeyEnc is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSUseGovCloud

The @AWSUseGovCloud argument enables a special restricted region for the US Government use in Amazon S3. This argument accepts one of the following values:

- 0–Do not use government cloud
- 1–Use government cloud

Important: This @AWSUseGovCloud argument is replaced by @CloudGovRegion. The @AWSUseGovCloud argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@backupextension

When looking for database backups, LiteSpeed will only consider backup files that have the extensions you specify. The value of this parameter is a list of extensions, separated with commas. No value or asterisk (*) specifies any file extension.

@backuppath

Specifies the directory where to search for the backup files.

@backuptype

Specifies backup types to use for the restore. This argument accepts one of the following values:

- full–LiteSpeed will only restore the most recent full database backup.
- diff–LiteSpeed will restore the most recent full database backup and any existing differential backups based on this full.
- tlog–LiteSpeed will restore the most recent full database backup and any existing differential and/or transaction log backups created after the most recent full backup.

@buffercount

Specifies the number of SQL Server buffers available for a LiteSpeed operation. The default value is set by SQL Server.

@checkdb

Checks database integrity after restore. This argument accepts one of the following values:

- 0–False. LiteSpeed will not confirm integrity of the database after restore.
- 1–True. LiteSpeed will confirm integrity of the database after restore.

@checkdbdatapurity

Checks data purity validations on every column value in all rows of the table or tables in the database. This argument accepts one of the following values:

- 0–False. LiteSpeed will not confirm integrity of the database column values after restore.
- 1–True. LiteSpeed will confirm integrity of the database column values after restore.

@checkdbextendedlogical

Checks logical consistency on an indexed view, XML indexes, and spatial indexes after restore. This argument accepts one of the following values:

- 0–False. LiteSpeed will not confirm logical consistency after restore.
- 1–True. LiteSpeed will confirm logical consistency after restore.

@checkdbnoindex

Checks the database but does not check the index. This argument accepts one of the following values:

- 0–False. LiteSpeed will check the database and check the index.
- 1–True. LiteSpeed will check the database but not check the index.

@checkdbnoinfomessages

Causes check database to suppress all informational messages. after restore. This argument accepts one of the following values:

- 0–False. LiteSpeed will include informational messages in notification report after restore.
- 1–True. LiteSpeed will not include informational messages in notification report after restore.

@checkdbphysicalonly

Checks physical structure of the database only. This argument accepts one of the following values:

- 0–False. LiteSpeed will not confirm physical structure of the database.
- 1–True. LiteSpeed will confirm physical structure of the database.

@checkdbtablelocks

Causes check database to obtain locks instead of using an internal database snapshot. This includes a short-term exclusive (X) lock on the database. This argument accepts one of the following values:

- 0–False. LiteSpeed will not use locks instead of using an internal database snapshot after restore.
- 1–True. LiteSpeed will use locks instead of using an internal database snapshot after restore.

@checksubfolders

Specifies whether to use subfolders to look for database backups. This argument accepts one of the following values:

- 0–False. LiteSpeed will only use backups located in the specified folder.
- 1–True. LiteSpeed will look for backups in the specified folder and in its subfolders.

@CloudAccessKey

The @CloudAccessKey argument specifies the name of the unique Cloud Web Service alphanumeric access key that identifies each user. The selections include Amazon Access Key, Azure Account Name, Google e-mail styled account.

@CloudAccessKeyEnc

The @CloudAccessKeyEnc argument specifies the name of the encrypted unique Cloud Web Service alphanumeric access key that identifies each user.

@CloudBucketName

The @CloudBucketName argument specifies the name of the container for cloud objects. Bucket names must be at least 3 and no more than 63 characters long. The selections are Amazon Bucket Name, Azure Container Name, Google Bucket Name. Google Bucket Name requirements are described at https://cloud.google.com/storage/docs/naming.

@CloudGovRegion

The @CloudGovRegion argument enables a special restricted region for the US Government use in Amazon S3 and Azure Clouds. This argument accepts one of the following values:

- 0–Do not use government cloud (default)
- 1–Use government cloud

@CloudRegionName

The @CloudRegionName argument specifies the name of the Cloud Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-east-2, us-west-1, us-west-2, ca-central-1, eu-central-1, eu-west-1, eu-west-2, ap-south-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, ap-northeast-2, sa-east-1, N'Germany' and N'China'.

@CloudSecretKey

The @CloudSecretKey argument specifies the name of the Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudSecretKeyEnc

The @CloudSecretKeyEnc argument specifies the name of the encrypted Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudVendor

The @CloudVendor argument specifies the name of the cloud service provider. The argument accepts one of the following values: "AmazonS3", "AzureBlob" or "GoogleStorage".

@database

Name of database to be backed up or restored.

This parameter specifies a database:

- to be backed up (xp_backup_database and xp_slsFastCompression)
- 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.

@datafilepath

Specifies a location for data files.

@dontusecopyonly

Specifies whether copy only backups are included during automated restores. This argument accepts one of the following values:

- 0–False. Include copy only backups.
- 1–True. Do not include copy only backups.

@DontUseReplication

Specifies whether replication is used during automated restores. This argument accepts one of the following values:

- 0–False. Use replication.
- 1–True. Do not use replication.

@dropdatabaseonfailure

Drops the restored database only if the restore fails. Use this option if you no longer need the restored database. For example, if you are only restoring the latest backup for testing purposes. This option contains two additional options to select. One or both options can be selected.On success restore and check database integrity operations - The database is dropped after a successful restore and database integrity check.On failure any of restore or check databases integrity operations - The database is dropped after failing the restore or database integrity check. This argument accepts one of the following values:

- 0–False (default)
- 1–True

@dropdatabaseonsuccess

Drops database on success only. Use this option if you no longer need the restored database. For example, if you are only restoring the latest backup for testing purposes. This option contains two additional options to select. One or both options can be selected.On success restore and check database integrity operations - The database is dropped after a successful restore and database integrity check.On failure any of restore or check databases integrity operations - The database is dropped after failing the restore or database integrity check. This argument accepts one of the following values:

- 0–False (default)
- 1–True

@dryrun

Shows backups that are candidates for restore at this time, but does not restore them. This argument accepts one of the following values:

- 0–False (default)
- 1–True

@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. If you use the wrong encryption key, the restore will fail.

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'

NOTE: Automated Restore requires that you use the same password for all encrypted backups.

@FileName

Specifies a backup location (e.g. C:\backups\AdventureWorks.bak). This argument accepts network destinations. You can supply multiple instances of this argument to use stripe backups.

@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 and cloud 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'
```

Network Resilience

@jobp

Specifies an encrypted key. (Similar to @EncryptionKey).

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

NOTE: Automated Restore requires that you use the same password for all encrypted backups.

@logfilepath

Specifies a location for log files.

@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 (or C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs)* (or *C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs*). To log to a different directory add @Trace='logpath=path'.

See Configure Logging in LiteSpeed for information about LiteSpeed logging.

@maxtransfersize

Specifies the largest unit of transfer in bytes to be used between SQL Server and LiteSpeed. The possible values are multiples of 65536 bytes (64 KB) ranging up to 4,194,304 bytes (4 MB). The default is 1048576 (1 MB).

@ProxyHost

The @ProxyHost argument is optional and specifies the name of the proxy host name that is running the proxy server.

note: If the @ProxyHost argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyLogin

The @ProxyLogin argument is optional and specifies the proxy server login credential.

note: If not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPassword

The @ProxyPassword argument is optional and specifies the proxy server password credential.

note: If the @ProxyPassword argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPasswordEnc

The @ProxyPasswordEnc argument is optional and specifies the encrypted proxy server password credential.

note: If the @ProxyPasswordEnc argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPort

The @ProxyPort argument is optional and contains the port number of the proxy server. The TCP/IP port values can be 1-65535.

note: If the @ProxyPort argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@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.

@RestoreAsCompressed

Works in conjunction with @restoreasreadonly, creates a folder if it does not exist, and then compresses it. This argument accepts one of the following values:

- 0–False (default)
- 1–True

@RestoreAsReadOnly

Instructs the restore operation to leave the database in read-only mode. This argument accepts one of the following values:

- 0–False (default)
- 1–True

Using this option, you can restore a user database into an NTFS compressed folder or restore a tlog to a read-only database in a compressed folder.

NOTES:

- When using an NTFS-compressed folder for a database, it can only be restored as read-only.
- You can only use this feature on Windows NTFS file systems.

@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.

Column Name	Data Type	Description
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	Specifies the largest unit of transfer in bytes to be used between SQL Server and LiteSpeed. The possible values are multiples of 65536 bytes (64 KB) ranging up to 4,194,304 bytes (4 MB). The default is 1048576 (1 MB) .
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.

@sourcedatabase

Backups of this database are the source for restore.

@sourceserver

Backups created on this instance of SQL Server are the source for restore.

@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. For more information, see Adaptive Compression on page 122.

@UseSSL

The @UseSSL argument specifies that the connection uses SSL security. This argument accepts one of the following values:

- 0–Do not use SSL
- 1–Use SSL (default)

@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	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). **NOTE:** When specifying the NORECOVERY option, the database is not usable in this intermediate, non-recovered state. 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: • a restore script is being run and the log is always being applied. • a sequence of file restores is used and the database is not intended to be usable between two of the restore operations.
RECOVERY	Instructs the restore operation to roll back any uncommitted transactions. After the recovery process, the database is ready for use.

Parameter	Description
	If subsequent LiteSpeed restore operations (xp_restore_log or xp_restore_database from differential) are planned, NORECOVERY or STANDBY should be specified instead. If NORECOVERY, RECOVERY, or STANDBY is not specified, RECOVERY is the default. 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.
STATS	Displays a message each time a percentage of the activity completes. The default is 10%.
PASSWORD	Specifies the password for the backup set.

@withreplace

Instructs LiteSpeed to create the specified database and its related files even if another database already exists with the same name. The existing database is deleted. This argument accepts one of the following values:

- 0–False (default)
- 1–True

Examples

Restore the Most Recent Full Database Backup to a New Database

```
EXEC master.dbo.xp_restore_automated
@database='NEWDB'
, @datafilepath = 'D:\DATA'
, @logfilepath = 'D:\DATA'
, @backuppath = N'D:\temp'
, @backupextension = 'bak,bkp'
, @checksubfolders = 1
, @sourceserver = N'LITESPEED\SQL2005'
, @sourcedatabase = N'FOX'
, @backuptype = N'full'
```

Restore the Most Recent Full and Drop Database

```
EXEC master.dbo.xp_restore_automated
@database='TESTDB'
, @datafilepath = 'D:\DATA'
, @logfilepath = 'D:\DATA'
, @backuppath = N'D:\temp'
, @backupextension = ''
, @checksubfolders = 1
, @sourceserver = N'LITESPEED\SQL2005'
```

```
, @sourcedatabase = N'FOX'
, @backuptype = N'full'
, @dropdatabaseonfailure = 1
, @dropdatabaseonsuccess = 1
, @withreplace = 1
```

Restore the Most Recent Fast Compression Backups

```
exec master.dbo.xp_restore_automated
@database = N'LiteSpeedLocal_AutomatedRestore'
, @backuppath = N'D:\temp\FC\'
, @backupextension = ''
, @checksubfolders = 0
, @sourceserver = N'LITESPEED\SQL2005'
, @sourcedatabase = N'LiteSpeedLocal'
, @backuptype = N'diff'
, @jobp = N'5jzOEztgLxQ='
, @withreplace = 1
```

Restore the Most Recent Striped Backup

```
EXEC master.dbo.xp_restore_automated
@database='NEWDB'
, @datafilepath = 'D:\DATA'
, @logfilepath = 'D:\DATA'
, @backuppath = N'D:\temp'
, @backupextension = 'stripe1'
, @checksubfolders = 0
, @backuppath = N'E:\temp'
, @backupextension = 'stripe2'
, @checksubfolders = 0
, @sourceserver = N'LITESPEED\SQL2005'
, @sourcedatabase = N'FOX'
, @backuptype = N'full'
```

Restore with Database Integrity Enabled

```
exec master.dbo.xp_restore_automated
@database = N'CM_1_Testing_1' ,
@backuppath = N'C:\backup',
@backupextension = N'',
@checksubfolders = 0,
@sourceserver = N'SPB8595',
@sourcedatabase = N'CM_1_Testing',
@backuptype = N'diff',
@affinity = 0,
@logging = 0,
@withreplace = 1,
@checkdb = 1,
@checkdbphysicalonly = 1,
@checkdbnoindex = 1,
@with = N'STATS = 10'
```

View Candidates for Automated Restore

```
EXEC master.dbo.xp_restore_automated
@backuppath = N'D:\temp'
, @backupextension = ''
, @checksubfolders = 1
, @encryptionkey = N'******'
, @sourceserver = N'LITESPEED\SQL2005'
, @sourcedatabase = N'FOX'
, @backuptype = N'tlog'
, @dryrun = 1
```

Restore from Amazon S3

```
exec master.dbo.xp_restore_automated @database = N'model' ,

@backuppath = N'test',

@backupextension = N'',

@checksubfolders = 0,

@sourceserver = N 'servername',

@sourcedatabase = N'model',

@backuptype = N'tlog',

@CloudVendor = N'AmazonS3',

@CloudBucketName = N'california',

@CloudAccessKey = N'********',
```

```
@CloudSecretKey = N'*********',

@UseSSL = 1,

@affinity = 0,

@logging = 0,

@DontUseReplication = 1,

@checkdb = 1,

@checkdbphysicalonly = 1,

@checkdbnoindex = 1,

@checkdbnoinfomessages = 1,

@progressname = N'f94ee3d8-f6ac-47e0-80cd-a6326a532dd9',

@with = N'STATS = 10'
```

Restore from Microsoft Azure

```
exec master.dbo.xp_restore_automated @database = N'newtest' ,

@backuppath = N'test',

@backupextension = N'bak',

@checksubfolders = 1,

@sourceserver = N'Server\SQL_instance',

@sourcedatabase = N'test',

@backuptype = N'diff',

@CloudVendor = N'AzureBlob',

@CloudBucketName = N'test',

@CloudAccessKeyEnc = N'******',

@CloudSecretKeyEnc = N'******',

@UseSSL = 1,

@affinity = 0,

@logging = 0,

@DontUseReplication = 1,

@withreplace = 1,

@checkdb = 1,

@checkdbphysicalonly = 1,
```

```
@checkdbnoindex = 1,

@checkdbnoinfomessages = 1,

@with = N'STATS = 10'
```

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_automated_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

```
EXEC master.dbo.xp_restore_automated_verifyonly
[@database = 'database_name'
[, @datafilepath = 'path']
[, @logfilepath = 'path'] ]
, ( @filename = 'backup_filename' | ( @backuppath = 'path'
, @backupextension = 'extensions'
, @checksubfolders = 0 | 1 ) ) [,...n]
, @sourceserver = 'server_name'
, @sourcedatabase = 'database_name'
, @backuptype = N'option',
[, ( @encryptionkey = 'encryption_key' | @jobp = 'encrypted_key' ) ]
[, @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']
[, @maxtransfersize = 'maximum_transfer_size']
[, @dryrun = 0 | 1]
```

xp_restore_automated (Cloud)

```
EXEC master.dbo.xp_restore_automated_verifyonly

@database = N'filegroups' ,

@filename = N'test\test.bak',

@sourceserver = N'test\test',

@sourcedatabase = N'filegroups',

@backuptype = N'diff',

@CloudVendor = N'AmazonS3',

@CloudBucketName = N'test',

@CloudAccessKeyEnc = N'******',

@CloudSecretKeyEnc = N'*******',

@UseSSL = 1,

@CloudGovRegion = 1,

@proxyhost = N'10.1.1.1',

@proxyport = 80,

@proxylogin = N'test',

@ProxyPasswordEnc = N'******',

@affinity = 0,

@logging = 0,

@DontUseReplication = 1,

@with = N'STATS = 10'
```

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. For more information, see Adaptive Compression on page 122.

@ARPeriod

Specifies a point in time to restore from where the time is measured in days, hours, minutes and seconds from the restore time.

Set 0's for periods not used.

@ARPeriod = N'DD.HH:MM:SS'

@ARPointInTime

Specifies a point in time to restore from: year, month, day, hours, minutes, seconds.

@ARPointInTime = N'YYYY-MM-DD HH:MM:SS'

@AWSAccessKey

The @AWSAccessKey argument specifies the name of the unique Amazon Web Service alphanumeric access key that identifies each user.

Important: This @AWSAccessKey argument is replaced by @CloudAccessKey. The @AWSAccessKey argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSAccessKeyEnc

The @AWSAccessKeyEnc argument specifies the name of the encrypted unique Amazon Web Service alphanumeric access key that identifies each user.

Important: This @AWSAccessKeyEnc argument is replaced by @CloudAccessKeyEnc. The @AWSAccessKeyEnc argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSBucketName

The @AWSBucketName argument specifies the name of the container for AWS objects. Bucket names must be at least 3 and no more than 63 characters long.

Important: This @AWSBucketName argument is replaced by @CloudBucketName. The @AWSBucketName argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSRegionName

The @AWSRegionName argument specifies the name of the Amazon Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-west-2, us-west-1, eu-west-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, and sa-east-1.

Important: This @AWSRegionName argument is replaced by @CloudRegionName. The @AWSRegionName argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSSecretKey

The @AWSSecretKey argument specifies the name of the Amazon Web Service secret key that is assigned when you initially get an AWS account.

Important: This @AWSSecretKey argument is replaced by @CloudSecretKey. The @AWSSecretKey argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSSecretKeyEnc

The @AWSSecretKeyEnc argument specifies the name of the encrypted Amazon Web Service secret key that is assigned when you initially get an AWS account.

Important: This @AWSSecretKeyEnc argument is replaced by @CloudSecretKeyEnc. The

@AWSSecretKeyEnc is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSUseGovCloud

The @AWSUseGovCloud argument enables a special restricted region for the US Government use in Amazon S3. This argument accepts one of the following values:

- 0–Do not use government cloud
- 1–Use government cloud

Important: This @AWSUseGovCloud argument is replaced by @CloudGovRegion. The @AWSUseGovCloud argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@backupextension

When looking for database backups, LiteSpeed will only consider backup files that have the extensions you specify. The value of this parameter is a list of extensions, separated with commas. No value or asterisk (*) specifies any file extension.

@backuppath

Specifies the directory where to search for the backup files.

@backuptype

Specifies backup types to use for restore and verify. This argument accepts one of the following values:

- full–LiteSpeed will only restore the most recent full database backup.
- diff–LiteSpeed will restore the most recent full database backup and any existing differential backups based on this full.
- tlog–LiteSpeed will restore the most recent full database backup and any existing differential and/or transaction log backups created after the most recent full backup.
- verifyFullDiff–LiteSpeed will verify the most recent full database backup and any existing differential backups based on that full database backup.

@buffercount

Specifies the number of SQL Server buffers available for a LiteSpeed operation. The default value is set by SQL Server.

@CloudAccessKey

The @CloudAccessKey argument specifies the name of the unique Cloud Web Service alphanumeric access key that identifies each user. The selections include Amazon Access Key, Azure Account Name, Google e-mail styled account.

@CloudAccessKeyEnc

The @CloudAccessKeyEnc argument specifies the name of the encrypted unique Cloud Web Service alphanumeric access key that identifies each user.

@CloudBucketName

The @CloudBucketName argument specifies the name of the container for cloud objects. Bucket names must be at least 3 and no more than 63 characters long. The selections are Amazon Bucket Name, Azure Container Name, Google Bucket Name. Google Bucket Name requirements are described at https://cloud.google.com/storage/docs/naming.

@CloudGovRegion

The @CloudGovRegion argument enables a special restricted region for the US Government use in Amazon S3 and Azure Clouds. This argument accepts one of the following values:

- 0–Do not use government cloud (default)
- 1–Use government cloud

@CloudRegionName

The @CloudRegionName argument specifies the name of the Cloud Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-east-2, us-west-1, us-west-2, ca-central-1, eu-central-1, eu-west-1, eu-west-2, ap-south-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, ap-northeast-2, sa-east-1, N'Germany' and N'China'.

@CloudSecretKey

The @CloudSecretKey argument specifies the name of the Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudSecretKeyEnc

The @CloudSecretKeyEnc argument specifies the name of the encrypted Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudVendor

The @CloudVendor argument specifies the name of the cloud service provider. The argument accepts one of the following values: "AmazonS3", "AzureBlob" or "GoogleStorage".

@database

Name of database to be backed up or restored.

This parameter specifies a database:

- to be backed up (xp_backup_database and xp_slsFastCompression)
- 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.

@dontusecopyonly

Specifies whether copy only backups are included during automated restores. This argument accepts one of the following values:

- 0–False. Include copy only backups.
- 1–True. Do not include copy only backups.

@DontUseReplication

Specifies whether replication is used during automated restores. This argument accepts one of the following values:

- 0–False. Use replication.
- 1–True. Do not use replication.

@dryrun

Shows backups that are candidates for restore at this time, but does not restore them. This argument accepts one of the following values:

- 0–False (default)
- 1–True

@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. If you use the wrong encryption key, the restore will fail.

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'

NOTE: Automated Restore requires that you use the same password for all encrypted backups.

@FileName

Specifies a backup location (e.g. C:\backups\AdventureWorks.bak). This argument accepts network destinations. You can supply multiple instances of this argument to use stripe backups.

@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 and cloud 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'
```

Network Resilience

@jobp

Specifies an encrypted key. (Similar to @EncryptionKey).

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

NOTE: Automated Restore requires that you use the same password for all encrypted backups.

@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 (or C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs)* (or *C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs*). To log to a different directory add @Trace='logpath=path'.

See Configure Logging in LiteSpeed for information about LiteSpeed logging.

@maxtransfersize

Specifies the largest unit of transfer in bytes to be used between SQL Server and LiteSpeed. The possible values are multiples of 65536 bytes (64 KB) ranging up to 4,194,304 bytes (4 MB). The default is 1048576 (1 MB).

@ProxyHost

The @ProxyHost argument is optional and specifies the name of the proxy host name that is running the proxy server.

note: If the @ProxyHost argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyLogin

The @ProxyLogin argument is optional and specifies the proxy server login credential.

note: If not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPassword

The @ProxyPassword argument is optional and specifies the proxy server password credential.

note: If the @ProxyPassword argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPasswordEnc

The @ProxyPasswordEnc argument is optional and specifies the encrypted proxy server password credential.

note: If the @ProxyPasswordEnc argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPort

The @ProxyPort argument is optional and contains the port number of the proxy server. The TCP/IP port values can be 1-65535.

note: If the @ProxyPort argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@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.

@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	Specifies the largest unit of transfer in bytes to be used between SQL Server and LiteSpeed. The possible values are multiples of 65536 bytes (64 KB) ranging up to 4,194,304 bytes (4 MB). The default is 1048576 (1 MB) .
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.

@sourcedatabase

Backups of this database are the source for restore.

@sourceserver

Backups created on this instance of SQL Server are the source for restore.

@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. For more information, see Adaptive Compression on page 122.

@UseSSL

The @UseSSL argument specifies that the connection uses SSL security. This argument accepts one of the following values:

- 0–Do not use SSL
- 1–Use SSL (default)

@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	Instructs the restore operation to not roll back any uncommitted transactions. Either the

Parameter	Description
	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). **NOTE:** When specifying the NORECOVERY option, the database is not usable in this intermediate, non-recovered state. 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: • a restore script is being run and the log is always being applied. • a sequence of file restores is used and the database is not intended to be usable between two of the restore operations.
RECOVERY	Instructs the restore operation to roll back any uncommitted transactions. After the recovery process, the database is ready for use. If subsequent LiteSpeed restore operations (xp_restore_log or xp_restore_database from differential) are planned, NORECOVERY or STANDBY should be specified instead. If NORECOVERY, RECOVERY, or STANDBY is not specified, RECOVERY is the default. 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.
STATS	Displays a message each time a percentage of the activity completes. The default is 10%.
PASSWORD	Specifies the password for the backup set.

Examples

Verify the Most Recent Full Database Backup

```
EXEC master.dbo.xp_restore_automated_verifyonly

@database='NEWDB'

, @backuppath = N'D:\temp'

, @backupextension = 'bak,bkp'

, @checksubfolders = 1

, @sourceserver = N'LITESPEED\SQL2005'

, @sourcedatabase = N'FOX'
```

```
, @backuptype = N'full'
```

Verify the Most Recent Fast Compression Backups

```
exec master.dbo.xp_restore_automated_verifyonly

@database = N'LiteSpeedLocal_AutomatedRestore'

, @backuppath = N'D:\temp\FC\'

, @backupextension = ''

, @checksubfolders = 0

, @sourceserver = N'LITESPEED\SQL2005'

, @sourcedatabase = N'LiteSpeedLocal'

, @backuptype = N'diff'

, @jobp = N'5jzOEztgLxQ='
```

Verify the Most Recent Striped Backup

```
EXEC master.dbo.xp_restore_automated_verifyonly

@database='NEWDB'

, @backuppath = N'D:\temp'

, @backupextension = 'stripe1'

, @checksubfolders = 0

, @backuppath = N'E:\temp'

, @backupextension = 'stripe2'

, @checksubfolders = 0

, @sourceserver = N'LITESPEED\SQL2005'

, @sourcedatabase = N'FOX'

, @backuptype = N'full'
```

View Candidates for Automated Verify

```
EXEC master.dbo.xp_restore_automated_verifyonly

@backuppath = N'D:\temp'

, @backupextension = ''

, @checksubfolders = 1
```

```
, @encryptionkey = N'******'
, @sourceserver = N'LITESPEED\SQL2005'
, @sourcedatabase = N'FOX'
, @backuptype = N'tlog'
, @dryrun = 1
```

Verify from Amazon S3

```
exec master.dbo.xp_restore_automated_verifyonly
@database = N'model' ,
@backuppath = N'test',
@backupextension = N'',
@checksubfolders = 0,
@sourceserver = N 'servername',
@sourcedatabase = N'model',
@backuptype = N'tlog',
@CloudVendor = N'AmazonS3',
@CloudBucketName = N'california',
@CloudAccessKey = N'********',
@CloudSecretKey = N'*********',
@UseSSL = 1,
@affinity = 0,
@logging = 0,
@DontUseReplication = 1,
@with = N'STATS = 10'
```

Verify from Microsoft Azure

```
exec master.dbo.xp_restore_automated_verifyonly
@database = N'newtest' ,
@backuppath = N'test',
@backupextension = N'bak',
@checksubfolders = 1,
```

```
@sourceserver = N'Server\SQL_instance',

@sourcedatabase = N'test',

@backuptype = N'diff',

@CloudVendor = N'AzureBlob',

@CloudBucketName = N'test',

@CloudAccessKeyEnc = N'******',

@CloudSecretKeyEnc = N'******',

@UseSSL = 1,

@affinity = 0,

@logging = 0,

@DontUseReplication = 1,

@with = N'STATS = 10'
```

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

@tsmconfigfile = N'I:\dsm.opt',

@tsmobject = N'test\test\test',

@tsmpointintime = '2016-03-09 13:19:34',

@tsmclientnode = N'nodename',

@tsmclientownerpwd = N'password',

@tsmarchive = 1,

@encryptionkey = N'key',

@logging = 0
```

Note: The parameter @encryptionkey is required for this extended procedure.

Arguments

The procedure accepts the following arguments:

@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. If you use the wrong encryption key, the restore will fail.

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'

@filename

Specifies a backup location (e.g. C:\backups\AdventureWorks.bak). This argument accepts network destinations. You can supply multiple instances of this argument to use stripe backups.

@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.

@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 (or C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs)* (or *C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs*). To log to a different directory add @Trace='logpath=path'.

See Configure Logging in LiteSpeed for information about LiteSpeed logging.

@TSMArchive

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

- 0–False (default)
- 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.

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 full and partial backups created with LiteSpeed. 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)

```
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]
[, @returndetails = 0 | 1]
[, @restoreasreadonly = 0 | 1]
[, @restoreascompressed = 0 | 1]
```

xp_restore_database (TSM)

```
EXEC master.dbo.xp_restore_database
@database = 'database_name'
, @tsmobject = 'TSM_object' [,...n]
, @tsmconfigfile = 'TSM_configuration_file'
[, ( @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']

[, @tsmpointintime = 'date_time']

[, @tsmarchive = 0 | 1 ]

[, @returndetails = 0 | 1]

[, @restoreasreadonly = 0 | 1]

[, @restoreascompressed = 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]
```

xp_restore_database (Amazon S3)

```
EXEC master.dbo.xp_restore_database
@CloudVendor = N'AmazonS3'
, @Database = N'AA_5_restored88'
[, @FileName = N'AA_5_1.bak']
[, @FileNumber = 1]
[, @CloudBucketName = N'aabucket1']
[, @CloudAccessKey = N'***']
[, @CloudSecretKey = N'***']
[, @CloudRegionName = N'us-west-2']
[, @With = N'MOVE ''AA_5'' TO ''F:\Databases.SQL2005\QQ_3_restored88.mdf'', MOVE ''AA_
3_log'' TO ''F:\Databases.SQL2005\AA_5_restored88_log.ldf'', REPLACE']
[, @ProxyHost = N'proxy.sitelocal']
[, @ProxyPort = 8080]
[, @ProxyLogin = N'DOMAIN\temp-xyz-MYtester']
[, @ProxyPassword = N'***']
```

xp_restore_database (Microsoft Azure)

```
EXEC master.dbo.xp_restore_database

@database = N'model' ,

@filename = N'test\test.bak',

@filenumber = 1,

@CloudVendor = N'AzureBlob',

@CloudBucketName = N'test',

@CloudAccessKeyEnc = N'*******',

@CloudSecretKeyEnc = N'******',

@UseSSL = 1,

@with = N'REPLACE',

@with = N'STATS = 10',

@affinity = 0,

@logging = 0
```

xp_restore_database (Google Cloud Storage)

```
exec xp_restore_database

@Database = N'db-1'

, @FileName = N'db.bak'
```

```
, @FileNumber = 1

, @With = N'MOVE ''db'' TO ''path\to\db_1.mdf'', MOVE ''db_log'' TO ''path\to\db_
1_log.ldf'''

, @CloudVendor = N'GoogleStorage'

, @CloudBucketName = N'bucketname'

, @CloudAccessKey = N'***' -- my key'

, @CloudSecretKey = N'***' -- my key
```

xp_restore_database (restore partial backup - filegroup offline)

NOTE: The following example shows the syntax for performing partial restores using the “read_write_ filegroups” parameter. The database used in the example below, FGBackups_PROD, contains the following filegroups: Primary, FG1. FG2, and FG3).

Tip: The read only filegroup FG3 is offline after the restore is complete as indicated in the following example.

```
EXEC master.dbo.xp_restore_database

@database = 'Prod_test' ,

@read_write_filegroups = 1,

@filename = N'I:\FGBackups_FULL.bkp',

@with = N'REPLACE'
```

xp_restore_database (restore partial backup - filegroup online)

Tip: To restore the read only filegroup FG3 and bring it online after the partial restore above has completed, a full backup containing the filegroup must be used for the restore as indicated in the following example.

```
EXEC master.dbo.xp_restore_database

@database = 'Prod_test'

, @filegroup = 'FG3'

, @filename = N'I:\FGBackups_FULL_keep.bak',

@with = N'REPLACE'
```

xp_restore_database (restore fast compression partial backup - offline)

NOTE: The read only file group, FG3, is offline after the restore is complete.

```
EXEC master.dbo.xp_restore_database

@database = 'Prod_test',
```

```
@read_write_filegroups = 1,

@filename = 'I:\SQLbackups\FGBackups_PROD.litespeed.f0.bkp' ,

@with = N'REPLACE'
```

xp_restore_database (restore fast compression partial backup - online)

Tip: To restore the read only file group, FG3, and bring it online after the partial restore above has completed, a full backup containing the filegroup must be used for the restore.

```
EXEC master.dbo.xp_restore_database

@database = 'Prod_test'

, @filegroup = 'FG3'

, @filename = N'I:\FGBackups_FULL.bak',

@with = N'REPLACE'
```

xp_restore_database (restore partial differential backup)

NOTE: The read only file group FG3 is offline after the restore is complete.

It is a two-step process to restore a LiteSpeed partial full backup followed by a restore of a LiteSpeed partial differential backup. Both steps are needed in order to restore a partial differential backup and bring the database online.

Step 1 - restore the LiteSpeed partial full backup

```
EXEC.master.dbo.xp_restore_database

@database = 'FGBackups_PROD' ,

@read_write_filegroups = 1,

@WITH = 'PARTIAL' ,

@filename = N'I:\FGBackups_FULL.bkp',

@with = N'REPLACE',

@with = N'NORECOVERY'
```

Step 2 -restore the LiteSpeed partial differential backup

```
EXEC master.dbo.xp_restore_database

@database = 'FGBackups_PROD' ,

@read_write_filegroups = 1,

@filename = N'I:\FGBackups_FULL_diff.bkp',
```

Tip: To restore the read only file group FG3 and bring it online after the partial restore above is completed, a full backup containing the file group must be used for the restore.

```
EXEC.master.dbo.xp_restore_database

@database = 'FGBackups_PROD'

, @filegroup = 'FG3'

, @filename = N'I:\FGBackups_FULL_keep.bak',

@with = N'REPLACE'
```

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. For more information, see Adaptive Compression on page 122.

@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.

@AWSAccessKey

The @AWSAccessKey argument specifies the name of the unique Amazon Web Service alphanumeric access key that identifies each user.

Important: This @AWSAccessKey argument is replaced by @CloudAccessKey. The @AWSAccessKey argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSAccessKeyEnc

The @AWSAccessKeyEnc argument specifies the name of the encrypted unique Amazon Web Service alphanumeric access key that identifies each user.

Important: This @AWSAccessKeyEnc argument is replaced by @CloudAccessKeyEnc. The @AWSAccessKeyEnc argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSBucketName

The @AWSBucketName argument specifies the name of the container for AWS objects. Bucket names must be at least 3 and no more than 63 characters long.

Important: This @AWSBucketName argument is replaced by @CloudBucketName. The @AWSBucketName argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSRegionName

The @AWSRegionName argument specifies the name of the Amazon Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-west-2, us-west-1, eu-west-1, ap-southeast-1, ap-southeast-

2, ap-northeast-1, and sa-east-1.

Important: This @AWSRegionName argument is replaced by @CloudRegionName. The @AWSRegionName argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSSecretKey

The @AWSSecretKey argument specifies the name of the Amazon Web Service secret key that is assigned when you initially get an AWS account.

Important: This @AWSSecretKey argument is replaced by @CloudSecretKey. The @AWSSecretKey argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSSecretKeyEnc

The @AWSSecretKeyEnc argument specifies the name of the encrypted Amazon Web Service secret key that is assigned when you initially get an AWS account.

Important: This @AWSSecretKeyEnc argument is replaced by @CloudSecretKeyEnc. The @AWSSecretKeyEnc is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSUseGovCloud

The @AWSUseGovCloud argument enables a special restricted region for the US Government use in Amazon S3. This argument accepts one of the following values:

- 0–Do not use government cloud
- 1–Use government cloud

Important: This @AWSUseGovCloud argument is replaced by @CloudGovRegion. The @AWSUseGovCloud argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSUseReducedRedundancy

The @AWSUseReducedRedundancy argument specifies the use of reduced redundancy storage in Amazon S3. This argument accepts one of the following values:

- 0–Do not use reduced redundancy storage
- 1–Use reduced redundancy storage

Note: This @AWSUseReducedRedundancy argument is replaced with the @CloudStorageClass = 'rrs' argument.

@AWSUseServerSideEncryption

The @AWSUseServerSideEncryption argument enables the encryption of data stored at rest in Amazon S3. This argument accepts one of the following values:

- 0–Do not use Server Side Encryption
- 1–Use Server Side Encryption

@buffercount

Specifies the number of SQL Server buffers available for a LiteSpeed operation. The default value is set by SQL Server.

@buffercount

Specifies the number of SQL Server buffers available for a LiteSpeed operation. The default value is set by SQL Server.

@CloudAccessKey

The @CloudAccessKey argument specifies the name of the unique Cloud Web Service alphanumeric access key that identifies each user. The selections include Amazon Access Key, Azure Account Name, Google e-mail styled account.

@CloudAccessKeyEnc

The @CloudAccessKeyEnc argument specifies the name of the encrypted unique Cloud Web Service alphanumeric access key that identifies each user.

@CloudBucketName

The @CloudBucketName argument specifies the name of the container for cloud objects. Bucket names must be at least 3 and no more than 63 characters long. The selections are Amazon Bucket Name, Azure Container Name, Google Bucket Name. Google Bucket Name requirements are described at https://cloud.google.com/storage/docs/naming.

@CloudGovRegion

The @CloudGovRegion argument enables a special restricted region for the US Government use in Amazon S3 and Azure Clouds. This argument accepts one of the following values:

- 0–Do not use government cloud (default)
- 1–Use government cloud

@CloudRegionName

The @CloudRegionName argument specifies the name of the Cloud Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-east-2, us-west-1, us-west-2, ca-central-1, eu-central-1, eu-west-1, eu-west-2, ap-south-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, ap-northeast-2, sa-east-1, N'Germany' and N'China'.

@CloudSecretKey

The @CloudSecretKey argument specifies the name of the Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudSecretKeyEnc

The @CloudSecretKeyEnc argument specifies the name of the encrypted Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudVendor

The @CloudVendor argument specifies the name of the cloud service provider. The argument accepts one of the following values: "AmazonS3", "AzureBlob" or "GoogleStorage".

@database

Name of database to be backed up or restored.

This parameter specifies a database:

- to be backed up (xp_backup_database and xp_slsFastCompression)
- 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.

@DisconnectUsers

Disconnect users on executing restore (in standby mode only). This argument accepts one of the following values:

- 0–Do not disconnect users (default).
- 1–Disconnect users.

@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. If you use the wrong encryption key, the restore will fail.

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'

@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

Specifies a backup location (e.g. C:\backups\AdventureWorks.bak). This argument accepts network destinations. You can supply multiple instances of this argument to use stripe backups.

@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 and cloud 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'
```

Network Resilience

@jobp

Specifies an encrypted key. (Similar to @EncryptionKey).

You can use xp_encrypt_backup_key to convert the password (encryption_key) for use with @jobp. The original password (or encrypted key generated by xp_encrypt_restore_key) 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 (or C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs)* (or *C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs*). To log to a different directory add @Trace='logpath=path'.

See Configure Logging in LiteSpeed for information about LiteSpeed logging.

@maxtransfersize

Specifies the largest unit of transfer in bytes to be used between SQL Server and LiteSpeed. The possible values are multiples of 65536 bytes (64 KB) ranging up to 4,194,304 bytes (4 MB). The default is 1048576 (1 MB).

@Page

The @Page argument allows you to restore one or more damaged pages without the need to restore the whole database. Restoring and recovering a few individual pages might be faster than a file restore by reducing the amount of data that is offline during a restore operation. However, if you have to restore more than a few pages in a file, it may be more efficient to restore the whole file. For example, if lots of pages on a device indicate a pending device failure, consider restoring the file, possibly to another location, and repairing the device. Please refer to the Microsoft SQL Server documentation on Restore Pages for additional information and guidelines on page restores.

@ProxyHost

The @ProxyHost argument is optional and specifies the name of the proxy host name that is running the proxy server.

note: If the @ProxyHost argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyLogin

The @ProxyLogin argument is optional and specifies the proxy server login credential.

note: If not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPassword

The @ProxyPassword argument is optional and specifies the proxy server password credential.

note: If the @ProxyPassword argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPasswordEnc

The @ProxyPasswordEnc argument is optional and specifies the encrypted proxy server password credential.

note: If the @ProxyPasswordEnc argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPort

The @ProxyPort argument is optional and contains the port number of the proxy server. The TCP/IP port values can be 1-65535.

note: If the @ProxyPort argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@read_write_filegroup

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.

@restoreascompressed

Works in conjunction with @restoreasreadonly, creates a folder if it does not exist, and then compresses it. This argument accepts one of the following values:

- 0–False (default)
- 1–True

@restoreasreadonly

Instructs the restore operation to leave the database in read-only mode. This argument accepts one of the following values:

- 0–False (default)
- 1–True

Using this option, you can restore a user database into an NTFS compressed folder or restore a tlog to a read-only database in a compressed folder.

NOTES:

- When using an NTFS-compressed folder for a database, it can only be restored as read-only.
- You can only use this feature on Windows NTFS file systems.

@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	Specifies the largest unit of transfer in bytes to be used between SQL Server and LiteSpeed. The possible values are multiples of 65536 bytes (64 KB) ranging up to 4,194,304 bytes (4 MB). The default is 1048576 (1 MB) .
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.

Column Name	Data Type	Description
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. For more information, see Adaptive Compression on page 122.

@tsmarchive

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

- 0–False (default)
- 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

@UseSSL

The @UseSSL argument specifies that the connection uses SSL security. This argument accepts one of the following values:

- 0–Do not use SSL
- 1–Use SSL (default)

@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	`MOVE = ''logical_file_name'' TO ''operating_system_file_name''`
	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.
	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.
	Example:
	`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'''`
	`, @with = 'MOVE ''MyDB_Log'' TO ''C:\MSSQL\Data\MyDB_log.LDF'''`
	NOTE: Use xp_restore_filelistonly to obtain a list of the logical files from the backup set. xp_restore_filelistonly

NORECOVERY	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

Parameter	Description
	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). **NOTE:** When specifying the NORECOVERY option, the database is not usable in this intermediate, non-recovered state. 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: • a restore script is being run and the log is always being applied. • a sequence of file restores is used and the database is not intended to be usable between two of the restore operations.
PARTIAL	Specifies a partial restore operation. 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.
RECOVERY	Instructs the restore operation to roll back any uncommitted transactions. After the recovery process, the database is ready for use. If subsequent LiteSpeed restore operations (xp_restore_log or xp_restore_database from differential) are planned, NORECOVERY or STANDBY should be specified instead. If NORECOVERY, RECOVERY, or STANDBY is not specified, RECOVERY is the default. 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.
REPLACE	Instructs LiteSpeed to create the specified database and its related files even if another database already exists with the same name. The existing database is deleted. When the this option is not specified, LiteSpeed performs a check to ensure that the database is not restored to the current server if: • the database named in the xp_restore_database statement already exists on the current server, and • the database name is different from the database name recorded in the LiteSpeed backup set. LiteSpeed will overwrite an existing file which cannot be verified as belonging to the database being restored. Normally, LiteSpeed will refuse to overwrite pre-existing files.
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.

Parameter	Description
STANDBY	`STANDBY = ''undo_file_name''` 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. 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. If the specified undo file name does not exist, LiteSpeed creates it. If the file does exist, LiteSpeed overwrites it. The same undo file can be used for consecutive LiteSpeed restores of the same database. **NOTE:** If free disk space is exhausted on the drive containing the specified undo file name, the LiteSpeed restore operation stops.
STATS	Displays a message each time a percentage of the activity completes. The default is 10%.
CHECKSUM	Causes checksums to be verified when a LiteSpeed backup is created. **NOTE:** 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.

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 an Encrypted Backup

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

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
```

Restore Database from Amazon S3

```
EXEC master.dbo.xp_restore_database
@CloudVendor = N'AmazonS3'
, @Database = N'AA_3_restored33'
, @FileName = N'AA_3_1.bak'
, @FileNumber = 1
, @CloudBucketName = N'aabucket1'
, @CloudAccessKey = N'***' -- my key
, @CloudSecretKey = N'***' -- my key
, @CloudRegionName = N'us-west-2' -- us-east-1, us-west-2, us-west-1, eu-west-1, ap-
southeast-1, ap-southeast-2, ap-northeast-1, sa-east-1
, @With = N'MOVE ''AA_3'' TO ''E:\Databases.SQL2005\AA_3_restored11.mdf'', MOVE ''AA_
3_log'' TO ''E:\Databases.SQL2005\AA_3_restored11_log.ldf'', REPLACE'
, @ProxyHost = N'proxy.sitelocal'
, @ProxyPort = 8080
, @ProxyLogin = N'DOMAIN\temp-xyz-MYtester'
, @ProxyPassword = N'***'
```

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)

```
EXEC master.dbo.xp_restore_filelistonly

( @filename = 'backup_file_name') [,...n]

[, @filenumber = n]
```

xp_restore_filelistonly (TSM)

```
EXEC master.dbo.xp_restore_filelistonly

( @tsmobject = 'TSM_object') [,...n]

, @tsmconfigfile = 'TSM_configuration_file'

[, @filenumber = n]

[, @tsmclientnode = 'TSM_client_node']

[, @tsmclientownerpwd = 'TSM_client_owner_password']

[, @tsmpointintime = 'date_time' ]
```

xp_restore_filelistonly (Tape)

```
EXEC master.dbo.xp_restore_filelistonly
@filename = 'tape_device_name'
[, @filenumber = n]
```

xp_restore_filelistonly (Amazon S3)

```
exec master.dbo.xp_restore_filelistonly

@filename = N'test/test/test.bkp',

@filenumber = 1,

@CloudVendor = N'AmazonS3',

@CloudBucketName = N'california',

@CloudAccessKey = N'***********',

@CloudSecretKey = N'****************',

@UseSSL = 1
```

xp_restore_filelistonly (Microsoft Azure)

```
exec master.dbo.xp_restore_filelistonly

@filename = N'test\test.bak',

@CloudVendor = N'AzureBlob',

@CloudAccessKeyEnc = N'*****',

@CloudSecretKeyEnc = N'******',

@CloudBucketName = N'test',@UseSSL = 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>

@AWSAccessKey

The @AWSAccessKey argument specifies the name of the unique Amazon Web Service alphanumeric access key that identifies each user.

Important: This @AWSAccessKey argument is replaced by @CloudAccessKey. The @AWSAccessKey argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSAccessKeyEnc

The @AWSAccessKeyEnc argument specifies the name of the encrypted unique Amazon Web Service alphanumeric access key that identifies each user.

Important: This @AWSAccessKeyEnc argument is replaced by @CloudAccessKeyEnc. The @AWSAccessKeyEnc argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSBucketName

The @AWSBucketName argument specifies the name of the container for AWS objects. Bucket names must be at least 3 and no more than 63 characters long.

Important: This @AWSBucketName argument is replaced by @CloudBucketName. The @AWSBucketName argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSRegionName

The @AWSRegionName argument specifies the name of the Amazon Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-west-2, us-west-1, eu-west-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, and sa-east-1.

Important: This @AWSRegionName argument is replaced by @CloudRegionName. The @AWSRegionName argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSSecretKey

The @AWSSecretKey argument specifies the name of the Amazon Web Service secret key that is assigned when you initially get an AWS account.

Important: This @AWSSecretKey argument is replaced by @CloudSecretKey. The @AWSSecretKey argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSSecretKeyEnc

The @AWSSecretKeyEnc argument specifies the name of the encrypted Amazon Web Service secret key that is assigned when you initially get an AWS account.

Important: This @AWSSecretKeyEnc argument is replaced by @CloudSecretKeyEnc. The @AWSSecretKeyEnc is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSUseGovCloud

The @AWSUseGovCloud argument enables a special restricted region for the US Government use in Amazon S3. This argument accepts one of the following values:

- 0–Do not use government cloud
- 1–Use government cloud

Important: This @AWSUseGovCloud argument is replaced by @CloudGovRegion. The @AWSUseGovCloud argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@CloudAccessKey

The @CloudAccessKey argument specifies the name of the unique Cloud Web Service alphanumeric access key that identifies each user. The selections include Amazon Access Key, Azure Account Name, Google e-mail styled account.

@CloudAccessKeyEnc

The @CloudAccessKeyEnc argument specifies the name of the encrypted unique Cloud Web Service alphanumeric access key that identifies each user.

@CloudBucketName

The @CloudBucketName argument specifies the name of the container for cloud objects. Bucket names must be at least 3 and no more than 63 characters long. The selections are Amazon Bucket Name, Azure Container Name, Google Bucket Name. Google Bucket Name requirements are described at https://cloud.google.com/storage/docs/naming.

@CloudGovRegion

The @CloudGovRegion argument enables a special restricted region for the US Government use in Amazon S3 and Azure Clouds. This argument accepts one of the following values:

- 0–Do not use government cloud (default)
- 1–Use government cloud

@CloudRegionName

The @CloudRegionName argument specifies the name of the Cloud Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-east-2, us-west-1, us-west-2, ca-central-1, eu-central-1, eu-west-1, eu-west-2, ap-south-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, ap-northeast-2, sa-east-1, N'Germany' and N'China'.

@CloudSecretKey

The @CloudSecretKey argument specifies the name of the Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudSecretKeyEnc

The @CloudSecretKeyEnc argument specifies the name of the encrypted Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudVendor

The @CloudVendor argument specifies the name of the cloud service provider. The argument accepts one of the following values: "AmazonS3", "AzureBlob" or "GoogleStorage".

@filename

Specifies a backup location (e.g. C:\backups\AdventureWorks.bak). This argument accepts network destinations. You can supply multiple instances of this argument to use stripe backups.

@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.

@JobP

Specifies an encrypted key. (Similar to @EncryptionKey).

You can use xp_encrypt_backup_key to convert the password (encryption_key) for use with @jobp. The original password (or encrypted key generated by xp_encrypt_restore_key) 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 (or C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs)* (or *C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs*). To log to a different directory add @Trace='logpath=path'.

See Configure Logging in LiteSpeed for information about LiteSpeed logging.

@ProxyHost

The @ProxyHost argument is optional and specifies the name of the proxy host name that is running the proxy server.

note: If the @ProxyHost argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyLogin

The @ProxyLogin argument is optional and specifies the proxy server login credential.

note: If not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPassword

The @ProxyPassword argument is optional and specifies the proxy server password credential.

note: If the @ProxyPassword argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPasswordEnc

The @ProxyPasswordEnc argument is optional and specifies the encrypted proxy server password credential.

note: If the @ProxyPasswordEnc argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPort

The @ProxyPort argument is optional and contains the port number of the proxy server. The TCP/IP port values can be 1-65535.

note: If the @ProxyPort argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@TSMArchive

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

- 0–False (default)
- 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

@UseSSL

The @UseSSL argument specifies that the connection uses SSL security. This argument accepts one of the following values:

- 0–Do not use SSL
- 1–Use SSL (default)

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: • D–Data file

Column Name	Data Type	Description for backup sets
		• L–Log file • S–Filestream file
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']
```

xp_restore_headeronly (Amazon S3)

```
EXEC  master.dbo.xp_restore_headeronly
@CloudVendor  =  N'AmazonS3'
,  @FileName  =  N'AA_3_1.bak'
[,  @CloudBucketName  =  N'aabucket7']
[,  @CloudAccessKey  =  N'***']
[,  @CloudSecretKey  =  ***']
[,  @CloudRegionName  =  N'us-west-2']
[,  @ProxyHost  =  N'proxy.sitelocal']
[,  @ProxyPort  =  8080]
[,  @ProxyLogin  =  N'DOMAIN\xyz-tst-MYTester']
[,  @ProxyPassword  =  N'***']
```

xp_restore_headeronly (Microsoft Azure)

```
EXEC master.dbo.xp_restore_headeronly

@filename = N'test\test.bak',

@CloudVendor = N'AzureBlob',

@CloudAccessKeyEnc = N'*******',

@CloudSecretKeyEnc = N'******',

@CloudBucketName = N'test',

@UseSSL = 1,

@logging = 0
```

xp_restore_headeronly (Google Cloud Storage)

```
exec xp_restore_headeronly
@FileName = N'path\to\backup'
, @FileNumber = 1
, @CloudVendor = N'GoogleStorage'
, @CloudBucketName = N'bucketname'
, @CloudAccessKey = N'***' -- my key'
, @CloudSecretKey = N'***' -- my key'
```

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

@AWSAccessKey

The @AWSAccessKey argument specifies the name of the unique Amazon Web Service alphanumeric access key that identifies each user.

Important: This @AWSAccessKey argument is replaced by @CloudAccessKey. The @AWSAccessKey argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSAccessKeyEnc

The @AWSAccessKeyEnc argument specifies the name of the encrypted unique Amazon Web Service alphanumeric access key that identifies each user.

Important: This @AWSAccessKeyEnc argument is replaced by @CloudAccessKeyEnc. The @AWSAccessKeyEnc argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSBucketName

The @AWSBucketName argument specifies the name of the container for AWS objects. Bucket names must be at least 3 and no more than 63 characters long.

Important: This @AWSBucketName argument is replaced by @CloudBucketName. The @AWSBucketName argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSRegionName

The @AWSRegionName argument specifies the name of the Amazon Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-west-2, us-west-1, eu-west-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, and sa-east-1.

Important: This @AWSRegionName argument is replaced by @CloudRegionName. The @AWSRegionName argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSSecretKey

The @AWSSecretKey argument specifies the name of the Amazon Web Service secret key that is assigned when you initially get an AWS account.

Important: This @AWSSecretKey argument is replaced by @CloudSecretKey. The @AWSSecretKey argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSSecretKeyEnc

The @AWSSecretKeyEnc argument specifies the name of the encrypted Amazon Web Service secret key that is assigned when you initially get an AWS account.

Important: This @AWSSecretKeyEnc argument is replaced by @CloudSecretKeyEnc. The

@AWSSecretKeyEnc is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSUseGovCloud

The @AWSUseGovCloud argument enables a special restricted region for the US Government use in Amazon S3. This argument accepts one of the following values:

- 0–Do not use government cloud
- 1–Use government cloud

Important: This @AWSUseGovCloud argument is replaced by @CloudGovRegion. The @AWSUseGovCloud argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSUseReducedRedundancy

The @AWSUseReducedRedundancy argument specifies the use of reduced redundancy storage in Amazon S3. This argument accepts one of the following values:

- 0–Do not use reduced redundancy storage
- 1–Use reduced redundancy storage

Note: This @AWSUseReducedRedundancy argument is replaced with the @CloudStorageClass = 'rrs' argument.

@AWSUseServerSideEncryption

The @AWSUseServerSideEncryption argument enables the encryption of data stored at rest in Amazon S3. This argument accepts one of the following values:

- 0–Do not use Server Side Encryption
- 1–Use Server Side Encryption

@CloudAccessKey

The @CloudAccessKey argument specifies the name of the unique Cloud Web Service alphanumeric access key that identifies each user. The selections include Amazon Access Key, Azure Account Name, Google e-mail styled account.

@CloudAccessKeyEnc

The @CloudAccessKeyEnc argument specifies the name of the encrypted unique Cloud Web Service alphanumeric access key that identifies each user.

@CloudBucketName

The @CloudBucketName argument specifies the name of the container for cloud objects. Bucket names must be at least 3 and no more than 63 characters long. The selections are Amazon Bucket Name, Azure Container Name, Google Bucket Name. Google Bucket Name requirements are described at https://cloud.google.com/storage/docs/naming.

@CloudGovRegion

The @CloudGovRegion argument enables a special restricted region for the US Government use in Amazon S3 and Azure Clouds. This argument accepts one of the following values:

- 0–Do not use government cloud (default)
- 1–Use government cloud

@CloudRegionName

The @CloudRegionName argument specifies the name of the Cloud Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-east-2, us-west-1, us-west-2, ca-central-1, eu-central-1, eu-west-1, eu-west-2, ap-south-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, ap-northeast-2, sa-east-1, N'Germany' and N'China'.

@CloudSecretKey

The @CloudSecretKey argument specifies the name of the Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudSecretKeyEnc

The @CloudSecretKeyEnc argument specifies the name of the encrypted Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudVendor

The @CloudVendor argument specifies the name of the cloud service provider. The argument accepts one of the following values: "AmazonS3", "AzureBlob" or "GoogleStorage".

@filename

Specifies a backup location (e.g. C:\backups\AdventureWorks.bak). This argument accepts network destinations. You can supply multiple instances of this argument to use stripe backups.

@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.

@JobP

Specifies an encrypted key. (Similar to @EncryptionKey).

You can use xp_encrypt_backup_key to convert the password (encryption_key) for use with @jobp. The original password (or encrypted key generated by xp_encrypt_restore_key) 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 (or C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs)* (or *C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs*). To log to a different directory add @Trace='logpath=path'.

See Configure Logging in LiteSpeed for information about LiteSpeed logging.

@ProxyHost

The @ProxyHost argument is optional and specifies the name of the proxy host name that is running the proxy server.

note: If the @ProxyHost argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyLogin

The @ProxyLogin argument is optional and specifies the proxy server login credential.

note: If not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPassword

The @ProxyPassword argument is optional and specifies the proxy server password credential.

note: If the @ProxyPassword argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPasswordEnc

The @ProxyPasswordEnc argument is optional and specifies the encrypted proxy server password credential.

note: If the @ProxyPasswordEnc argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPort

The @ProxyPort argument is optional and contains the port number of the proxy server. The TCP/IP port values can be 1-65535.

note: If the @ProxyPort argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@tsmarchive

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

- 0–False (default)
- 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.

@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

@UseSSL

The @UseSSL argument specifies that the connection uses SSL security. This argument accepts one of the following values:

- 0–Do not use SSL
- 1–Use SSL (default)

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_full2.bak'
, @attachedfiles = 2
```

2. List files attached to a tsm backup:

```
exec master.dbo.xp_restore_headeronly
@tsmclientnode = N'10.1.26.177',
@tsmclientownerpwd = N'Quest2013',
@tsmconfigfile = N'D:\dsm.opt'
, @tsmobject = N'test\test\model'
, @attachedfiles = 1
```

3. Restore headers from an AmazonS3 Cloud backup:

```
exec master.dbo.xp_restore_headeronly
@CloudVendor = N'AmazonS3'
, @FileName = N'AA_3_1.bak'
, @CloudBucketName = N'aabucket7'
, @CloudAccessKey = N'***' -- my key
, @CloudSecretKey = ***' -- my key
, @CloudRegionName = N'us-west-2' -- us-east-1, us-west-2, us-west-1, eu-west-1,
ap-southeast-1, ap-southeast-2, ap-northeast-1, sa-east-1
(this is an optional parameter. Region for selected @CloudBucketName will be
used.)
, @ProxyHost = N'proxy.sitelocal'
, @ProxyPort = 8080
, @ProxyLogin = N'DOMAIN\xyz-tst-MYTester'
, @ProxyPassword = N'***'
```

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.

Column Name	Data Type	Description
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: • 1–Database • 2–Transaction Log • 4–File • 5–Differential Database • 6–Differential File • 7–Partial • 8–Partial Differential
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 was backed up.
DatabaseVersion	int	Version of the database from which the backup was created.
DatabaseCreationDate	datetime	Date and time the database was created.

Column Name	Data Type	Description
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 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.

Column Name	Data Type	Description
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 • 0–not encrypted • 1–encrypted
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_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)

```
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']

[, @maxtransfersize = 'maximum_transfer_size']

[, @attachedfile = 'pathname'] [,..n]

[, @returndetails = 0 | 1 ]

[, @restoreasreadonly = 0 | 1]

[, @restoreascompressed = 0 | 1]
```

xp_restore_log (TSM)

```
EXEC master.dbo.xp_restore_log

@database = 'database_name'

, @tsmobject = 'TSM_object' [,...n]

, @tsmconfigfile = 'TSM_configuration_file'

[, ( @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']

[, @maxtransfersize = 'maximum_transfer_size']

[, @attachedfile = 'pathname'] [,..n]

[, @tsmclientnode = 'TSM_client_node']

[, @tsmclientownerpwd = 'TSM_client_owner_password']

[, @tsmpointintime = 'date_time' ]

[, @returndetails = 0 | 1 ]

[, @restoreasreadonly = 0 | 1]

[, @restoreascompressed = 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 ]
```

xp_restore_log (Amazon S3)

```
exec master.dbo.xp_restore_log @database = N'model' ,

@filename = N'test\tlogl.bak',

@filenumber = 1,

@CloudVendor = N'AmazonS3',

@CloudBucketName = N'california',
```

```
@CloudAccessKey = N'*******',

@CloudSecretKey = N'**********',

@UseSSL = 1,

@with = N'STATS = 10',

@with = N'RECOVERY',

@affinity = 0,

@logging = 0
```

xp_restore_log (Microsoft Azure)

```
exec master.dbo.xp_restore_log @database = N'newtest' ,

@filename = N'test\testlog.bak',

@filenumber = 1,

@CloudVendor = N'AzureBlob',

@CloudBucketName = N'test',

@CloudAccessKeyEnc = N'*****',

@CloudSecretKeyEnc = N'******',

@UseSSL = 1,

@with = N'STATS = 10',

@affinity = 0,@logging = 0
```

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. For more information, see Adaptive Compression on page 122.

@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.

@AWSAccessKey

The @AWSAccessKey argument specifies the name of the unique Amazon Web Service alphanumeric access key that identifies each user.

> Important: This @AWSAccessKey argument is replaced by @CloudAccessKey. The @AWSAccessKey argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSAccessKeyEnc

The @AWSAccessKeyEnc argument specifies the name of the encrypted unique Amazon Web Service alphanumeric access key that identifies each user.

Important: This @AWSAccessKeyEnc argument is replaced by @CloudAccessKeyEnc. The @AWSAccessKeyEnc argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSBucketName

The @AWSBucketName argument specifies the name of the container for AWS objects. Bucket names must be at least 3 and no more than 63 characters long.

Important: This @AWSBucketName argument is replaced by @CloudBucketName. The @AWSBucketName argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSRegionName

The @AWSRegionName argument specifies the name of the Amazon Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-west-2, us-west-1, eu-west-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, and sa-east-1.

Important: This @AWSRegionName argument is replaced by @CloudRegionName. The @AWSRegionName argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSSecretKey

The @AWSSecretKey argument specifies the name of the Amazon Web Service secret key that is assigned when you initially get an AWS account.

Important: This @AWSSecretKey argument is replaced by @CloudSecretKey. The @AWSSecretKey argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSSecretKeyEnc

The @AWSSecretKeyEnc argument specifies the name of the encrypted Amazon Web Service secret key that is assigned when you initially get an AWS account.

Important: This @AWSSecretKeyEnc argument is replaced by @CloudSecretKeyEnc. The @AWSSecretKeyEnc is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSUseGovCloud

The @AWSUseGovCloud argument enables a special restricted region for the US Government use in Amazon S3. This argument accepts one of the following values:

- 0–Do not use government cloud
- 1–Use government cloud

Important: This @AWSUseGovCloud argument is replaced by @CloudGovRegion. The @AWSUseGovCloud argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@buffercount

Specifies the number of SQL Server buffers available for a LiteSpeed operation. The default value is set by SQL Server.

@CloudAccessKey

The @CloudAccessKey argument specifies the name of the unique Cloud Web Service alphanumeric access key that identifies each user. The selections include Amazon Access Key, Azure Account Name, Google e-mail styled account.

@CloudAccessKeyEnc

The @CloudAccessKeyEnc argument specifies the name of the encrypted unique Cloud Web Service alphanumeric access key that identifies each user.

@CloudBucketName

The @CloudBucketName argument specifies the name of the container for cloud objects. Bucket names must be at least 3 and no more than 63 characters long. The selections are Amazon Bucket Name, Azure Container Name, Google Bucket Name. Google Bucket Name requirements are described at https://cloud.google.com/storage/docs/naming.

@CloudGovRegion

The @CloudGovRegion argument enables a special restricted region for the US Government use in Amazon S3 and Azure Clouds. This argument accepts one of the following values:

- 0–Do not use government cloud (default)
- 1–Use government cloud

@CloudRegionName

The @CloudRegionName argument specifies the name of the Cloud Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-east-2, us-west-1, us-west-2, ca-central-1, eu-central-1, eu-west-1, eu-west-2, ap-south-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, ap-northeast-2, sa-east-1, N'Germany' and N'China'.

@CloudSecretKey

The @CloudSecretKey argument specifies the name of the Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudSecretKeyEnc

The @CloudSecretKeyEnc argument specifies the name of the encrypted Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudVendor

The @CloudVendor argument specifies the name of the cloud service provider. The argument accepts one of the following values: "AmazonS3", "AzureBlob" or "GoogleStorage".

@database

Name of database to be backed up or restored.

This parameter specifies a database:

- to be backed up (xp_backup_database and xp_slsFastCompression)
- 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.

@DisconnectUsers

Disconnect users on executing restore (in standby mode only). This argument accepts one of the following values:

- 0–Do not disconnect users (default).
- 1–Disconnect users.

@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. If you use the wrong encryption key, the restore will fail.

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'

@filename

Specifies a backup location (e.g. C:\backups\AdventureWorks.bak). This argument accepts network destinations. You can supply multiple instances of this argument to use stripe backups.

@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 and cloud 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'
```

Network Resilience

@jobp

Specifies an encrypted key. (Similar to @EncryptionKey).

You can use xp_encrypt_backup_key to convert the password (encryption_key) for use with @jobp. The original password (or encrypted key generated by xp_encrypt_restore_key) 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 (or C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs)* (or *C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs*). To log to a different directory add @Trace='logpath=path'.

See Configure Logging in LiteSpeed for information about LiteSpeed logging.

@maxtransfersize

Specifies the largest unit of transfer in bytes to be used between SQL Server and LiteSpeed. The possible values are multiples of 65536 bytes (64 KB) ranging up to 4,194,304 bytes (4 MB). The default is 1048576 (1 MB).

@Page

The @Page argument allows you to restore one or more damaged pages without the need to restore the whole database. Restoring and recovering a few individual pages might be faster than a file restore by reducing the amount of data that is offline during a restore operation. However, if you have to restore more than a few pages in a file, it may be more efficient to restore the whole file. For example, if lots of pages on a device indicate a pending device failure, consider restoring the file, possibly to another location, and repairing the device. Please refer to the Microsoft SQL Server documentation on Restore Pages for additional information and guidelines on page restores.

@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

@ProxyHost

The @ProxyHost argument is optional and specifies the name of the proxy host name that is running the proxy server.

note: If the @ProxyHost argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyLogin

The @ProxyLogin argument is optional and specifies the proxy server login credential.

note: If not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPassword

The @ProxyPassword argument is optional and specifies the proxy server password credential.

note: If the @ProxyPassword argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPasswordEnc

The @ProxyPasswordEnc argument is optional and specifies the encrypted proxy server password credential.

note: If the @ProxyPasswordEnc argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPort

The @ProxyPort argument is optional and contains the port number of the proxy server. The TCP/IP port values can be 1-65535.

note: If the @ProxyPort argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@restoreascompressed

Works in conjunction with @restoreasreadonly, creates a folder if it does not exist, and then compresses it. This argument accepts one of the following values:

- 0–False (default)
- 1–True

@restoreasreadonly

Instructs the restore operation to leave the database in read-only mode. This argument accepts one of the following values:

- 0–False (default)
- 1–True

Using this option, you can restore a user database into an NTFS compressed folder or restore a tlog to a read-only database in a compressed folder.

NOTES:

- When using an NTFS-compressed folder for a database, it can only be restored as read-only.
- You can only use this feature on Windows NTFS file systems.

@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	Specifies the largest unit of transfer in bytes to be used between SQL Server and LiteSpeed. The possible values are multiples of 65536 bytes (64 KB) ranging up to 4,194,304 bytes (4 MB). The default is 1048576 (1 MB) .
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.

Column Name	Data Type	Description
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. For more information, see Adaptive Compression on page 122.

@TSMArchive

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

- 0–False (default)
- 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.

@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

@UseSSL

The @UseSSL argument specifies that the connection uses SSL security. This argument accepts one of the following values:

- 0–Do not use SSL
- 1–Use SSL (default)

@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	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). NOTE: When specifying the NORECOVERY option, the database is not usable in this intermediate, non-recovered state. 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: • a restore script is being run and the log is always being applied. • a sequence of file restores is used and the database is not intended to be usable between two of the restore operations.
RECOVERY	Instructs the restore operation to roll back any uncommitted transactions. After the recovery process, the database is ready for use. If subsequent LiteSpeed restore operations (xp_restore_log or xp_restore_database from differential) are planned, NORECOVERY or STANDBY should be specified instead. If NORECOVERY, RECOVERY, or STANDBY is not specified, RECOVERY is the default. 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.
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.
STANDBY	`STANDBY = ''undo_file_name''`

Parameter	Description
	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. 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. If the specified undo file name does not exist, LiteSpeed creates it. If the file does exist, LiteSpeed overwrites it. The same undo file can be used for consecutive LiteSpeed restores of the same database. **NOTE:** If free disk space is exhausted on the drive containing the specified undo file name, the LiteSpeed restore operation stops.
STOPAT	`STOPAT = date_time | @date_time_var` 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. Only transaction log records written before the specified date and time are applied to the database. **NOTE:** 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.
STOPATMARK	`STOPATMARK = ''mark_name'' [ AFTER Datetime ]` Specifies recovery to the specified mark, including the transaction that contains the mark. 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.
STOPBEFOREMARK	`STOPBEFOREMARK = ''mark_name'' [ AFTER Datetime ]` Specifies recovery to the specified mark but does not include the transaction that contains the mark. 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.
CHECKSUM	Causes checksums to be verified when a LiteSpeed backup is created. **NOTE:** 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.

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_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']
```

xp_restore_setinfo (Amazon S3)

```
EXEC master.dbo.xp_restore_setinfo

@filename = 'file name'

@CloudVendor = 'AmazonS3',

@CloudBucketName = 'bucket name',

@CloudAccessKeyEnc = 'accesskeyenc',

@CloudSercretKeyEnc = 'secretkeyenc'
```

xp_restore_setinfo (Microsoft Azure)

```
EXEC master.dbo.xp_restore_setinfo

@filename = N'tst\test.bak',

@CloudVendor = N'AzureBlob',

@CloudBucketName = N'test',

@CloudAccessKeyEnc = N'******',

@CloudSecretKeyEnc = N'********'
```

Arguments

@AWSAccessKey

The @AWSAccessKey argument specifies the name of the unique Amazon Web Service alphanumeric access key that identifies each user.

Important: This @AWSAccessKey argument is replaced by @CloudAccessKey. The @AWSAccessKey argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSAccessKeyEnc

The @AWSAccessKeyEnc argument specifies the name of the encrypted unique Amazon Web Service alphanumeric access key that identifies each user.

Important: This @AWSAccessKeyEnc argument is replaced by @CloudAccessKeyEnc. The @AWSAccessKeyEnc argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSBucketName

The @AWSBucketName argument specifies the name of the container for AWS objects. Bucket names must be at least 3 and no more than 63 characters long.

Important: This @AWSBucketName argument is replaced by @CloudBucketName. The @AWSBucketName argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSRegionName

The @AWSRegionName argument specifies the name of the Amazon Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-west-2, us-west-1, eu-west-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, and sa-east-1.

Important: This @AWSRegionName argument is replaced by @CloudRegionName. The @AWSRegionName argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSSecretKey

The @AWSSecretKey argument specifies the name of the Amazon Web Service secret key that is assigned when you initially get an AWS account.

Important: This @AWSSecretKey argument is replaced by @CloudSecretKey. The @AWSSecretKey argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSSecretKeyEnc

The @AWSSecretKeyEnc argument specifies the name of the encrypted Amazon Web Service secret key that is assigned when you initially get an AWS account.

Important: This @AWSSecretKeyEnc argument is replaced by @CloudSecretKeyEnc. The @AWSSecretKeyEnc is no longer valid in subsequent LiteSpeed versions after 8.2.

@CloudAccessKey

The @CloudAccessKey argument specifies the name of the unique Cloud Web Service alphanumeric access key that identifies each user. The selections include Amazon Access Key, Azure Account Name, Google e-mail styled account.

@CloudAccessKeyEnc

The @CloudAccessKeyEnc argument specifies the name of the encrypted unique Cloud Web Service alphanumeric access key that identifies each user.

@CloudBucketName

The @CloudBucketName argument specifies the name of the container for cloud objects. Bucket names must be at least 3 and no more than 63 characters long. The selections are Amazon Bucket Name, Azure Container Name, Google Bucket Name. Google Bucket Name requirements are described at https://cloud.google.com/storage/docs/naming.

@CloudRegionName

The @CloudRegionName argument specifies the name of the Cloud Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-east-2, us-west-1, us-west-2, ca-central-1, eu-central-1, eu-west-1, eu-west-2, ap-south-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, ap-northeast-2, sa-east-1, N'Germany' and N'China'.

@CloudSecretKey

The @CloudSecretKey argument specifies the name of the Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudSecretKeyEnc

The @CloudSecretKeyEnc argument specifies the name of the encrypted Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudVendor

The @CloudVendor argument specifies the name of the cloud service provider. The argument accepts one of the following values: "AmazonS3", "AzureBlob" or "GoogleStorage".

@filename

Specifies a backup location (e.g. C:\backups\AdventureWorks.bak). This argument accepts network destinations. You can supply multiple instances of this argument to use stripe backups.

@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.

@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 (or C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs)* (or *C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs*). To log to a different directory add @Trace='logpath=path'.

See Configure Logging in LiteSpeed for information about LiteSpeed logging.

@ProxyHost

The @ProxyHost argument is optional and specifies the name of the proxy host name that is running the proxy server.

note: If the @ProxyHost argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyLogin

The @ProxyLogin argument is optional and specifies the proxy server login credential.

note: If not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPassword

The @ProxyPassword argument is optional and specifies the proxy server password credential.

note: If the @ProxyPassword argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPasswordEnc

The @ProxyPasswordEnc argument is optional and specifies the encrypted proxy server password credential.

note: If the @ProxyPasswordEnc argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPort

The @ProxyPort argument is optional and contains the port number of the proxy server. The TCP/IP port values can be 1-65535.

note: If the @ProxyPort argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@tsmarchive

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

- 0–False (default)
- 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.

@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

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.

Column Name	Data Type	Description
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_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]
```

xp_restore_verifyonly (Amazon S3)

```
exec master.dbo.xp_restore_verifyonly

@filename = N'test\tlogl.bak',

@filenumber = 1,

@CloudVendor = N'AmazonS3',

@CloudBucketName = N'california',

@CloudAccessKey = N'************',

@CloudSecretKey = N'**********',

@UseSSL = 1,

@with = N'STATS = 10',

@with = N'RECOVERY',

@affinity = 0,

@logging = 0
```

xp_restore_verifyonly (Microsoft Azure)

```
exec master.dbo.xp_restore_verifyonly

@filename = N'test\test.bak',

@CloudVendor = N'AzureBlob',

@CloudAccessKeyEnc = N'*******',

@CloudSecretKeyEnc = N'******',

@CloudBucketName = N'test',
```

```
@UseSSL = 1,

@with = N'STATS = 10',

@with = N'RECOVERY',

@affinity = 0,

@logging = 0
```

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. For more information, see Adaptive Compression on page 122.

@AWSAccessKey

The @AWSAccessKey argument specifies the name of the unique Amazon Web Service alphanumeric access key that identifies each user.

Important: This @AWSAccessKey argument is replaced by @CloudAccessKey. The @AWSAccessKey argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSAccessKeyEnc

The @AWSAccessKeyEnc argument specifies the name of the encrypted unique Amazon Web Service alphanumeric access key that identifies each user.

Important: This @AWSAccessKeyEnc argument is replaced by @CloudAccessKeyEnc. The @AWSAccessKeyEnc argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSBucketName

The @AWSBucketName argument specifies the name of the container for AWS objects. Bucket names must be at least 3 and no more than 63 characters long.

Important: This @AWSBucketName argument is replaced by @CloudBucketName. The @AWSBucketName argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSRegionName

The @AWSRegionName argument specifies the name of the Amazon Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-west-2, us-west-1, eu-west-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, and sa-east-1.

Important: This @AWSRegionName argument is replaced by @CloudRegionName. The @AWSRegionName argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSSecretKey

The @AWSSecretKey argument specifies the name of the Amazon Web Service secret key that is assigned when you initially get an AWS account.

Important: This @AWSSecretKey argument is replaced by @CloudSecretKey. The @AWSSecretKey argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSSecretKeyEnc

The @AWSSecretKeyEnc argument specifies the name of the encrypted Amazon Web Service secret key that is assigned when you initially get an AWS account.

Important: This @AWSSecretKeyEnc argument is replaced by @CloudSecretKeyEnc. The

@AWSSecretKeyEnc is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSUseGovCloud

The @AWSSecretKeyEnc argument specifies the name of the encrypted Amazon Web Service secret key that is assigned when you initially get an AWS account.

Important: This @AWSSecretKeyEnc argument is replaced by @CloudSecretKeyEnc. The @AWSSecretKeyEnc is no longer valid in subsequent LiteSpeed versions after 8.2.

@buffercount

Specifies the number of SQL Server buffers available for a LiteSpeed operation. The default value is set by SQL Server.

@CloudAccessKey

The @CloudAccessKey argument specifies the name of the unique Cloud Web Service alphanumeric access key that identifies each user. The selections include Amazon Access Key, Azure Account Name, Google e-mail styled account.

@CloudAccessKeyEnc

The @CloudAccessKeyEnc argument specifies the name of the encrypted unique Cloud Web Service alphanumeric access key that identifies each user.

@CloudBucketName

The @CloudBucketName argument specifies the name of the container for cloud objects. Bucket names must be at least 3 and no more than 63 characters long. The selections are Amazon Bucket Name, Azure Container Name, Google Bucket Name. Google Bucket Name requirements are described at https://cloud.google.com/storage/docs/naming.

@CloudGovRegion

The @CloudGovRegion argument enables a special restricted region for the US Government use in Amazon S3 and Azure Clouds. This argument accepts one of the following values:

- 0–Do not use government cloud (default)
- 1–Use government cloud

@CloudRegionName

The @CloudRegionName argument specifies the name of the Cloud Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-east-2, us-west-1, us-west-2, ca-central-1, eu-central-1, eu-

west-1, eu-west-2, ap-south-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, ap-northeast-2, sa-east-1, N'Germany' and N'China'.

@CloudSecretKey

The @CloudSecretKey argument specifies the name of the Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudSecretKeyEnc

The @CloudSecretKeyEnc argument specifies the name of the encrypted Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudVendor

The @CloudVendor argument specifies the name of the cloud service provider. The argument accepts one of the following values: "AmazonS3", "AzureBlob" or "GoogleStorage".

@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. If you use the wrong encryption key, the restore will fail.

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'

@filename

Specifies a backup location (e.g. C:\backups\AdventureWorks.bak). This argument accepts network destinations. You can supply multiple instances of this argument to use stripe backups.

@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 and cloud 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'
```

Network Resilience

@jobp

Specifies an encrypted key. (Similar to @EncryptionKey).

You can use xp_encrypt_backup_key to convert the password (encryption_key) for use with @jobp. The original password (or encrypted key generated by xp_encrypt_restore_key) 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 (or C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs)* (or *C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs*). To log to a different directory add @Trace='logpath=path'.

See Configure Logging in LiteSpeed for information about LiteSpeed logging.

@maxtransfersize

Specifies the largest unit of transfer in bytes to be used between SQL Server and LiteSpeed. The possible values are multiples of 65536 bytes (64 KB) ranging up to 4,194,304 bytes (4 MB). The default is 1048576 (1 MB).

@ProxyHost

The @ProxyHost argument is optional and specifies the name of the proxy host name that is running the proxy server.

note: If the @ProxyHost argument is not defined, then the LiteSpeed core engine checks the local .ini files

for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyLogin

The @ProxyLogin argument is optional and specifies the proxy server login credential.

note: If not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPassword

The @ProxyPassword argument is optional and specifies the proxy server password credential.

note: If the @ProxyPassword argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPasswordEnc

The @ProxyPasswordEnc argument is optional and specifies the encrypted proxy server password credential.

note: If the @ProxyPasswordEnc argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPort

The @ProxyPort argument is optional and contains the port number of the proxy server. The TCP/IP port values can be 1-65535.

note: If the @ProxyPort argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@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	Specifies the largest unit of transfer in bytes to be used between SQL Server and LiteSpeed. The possible values are multiples of 65536 bytes (64 KB) ranging up to 4,194,304 bytes (4 MB). The default is 1048576 (1 MB) .
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.

@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. For more information, see Adaptive Compression on page 122.

@tsmarchive

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

- 0–False (default)
- 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

@UseSSL

The @UseSSL argument specifies that the connection uses SSL security. This argument accepts one of the following values:

- 0–Do not use SSL
- 1–Use SSL (default)

@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. **NOTE:** 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_sls_cloud_browse

The xp_sls_cloud_browse command returns a list of objects located on the specified cloud storage path.

Syntax

```
exec master.dbo.xp_sls_cloud_browse
[, @CloudAccessKey = N'***']
[, @CloudSecretKey = N'***]
[, @CloudBucketName = N'aabucket6']
[, @CloudFolderName = N'f1']
[, @CloudMaxItems = 2]
```

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>

@CloudAccessKey

The @CloudAccessKey argument specifies the name of the unique Cloud Web Service alphanumeric access key that identifies each user. The selections include Amazon Access Key, Azure Account Name, Google e-mail styled account.

@CloudAccessKeyEnc

The @CloudAccessKeyEnc argument specifies the name of the encrypted unique Cloud Web Service alphanumeric access key that identifies each user.

@CloudBucketName

The @CloudBucketName argument specifies the name of the container for cloud objects. Bucket names must be at least 3 and no more than 63 characters long. The selections are Amazon Bucket Name, Azure Container Name, Google Bucket Name. Google Bucket Name requirements are described at https://cloud.google.com/storage/docs/naming.

@CloudFolderName

The @CloudFolderName argument specifies the name of the folder for searching in cloud browse. It is a filter for output files.

This argument accepts variables. For more information, see LiteSpeed Variables on page 125.

@CloudGovRegion

The @CloudGovRegion argument enables a special restricted region for the US Government use in Amazon S3 and Azure Clouds. This argument accepts one of the following values:

- 0–Do not use government cloud (default)
- 1–Use government cloud

@CloudMaxItems

The @CloudMaxItems argument specifies the number of rows to return from the cloud.

Note: The @CloudMaxItems argument can be very useful on large tables with thousands of records. Returning a large number of records can impact cloud performance.

@CloudRegionName

The @CloudRegionName argument specifies the name of the Cloud Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-east-2, us-west-1, us-west-2, ca-central-1, eu-central-1, eu-west-1, eu-west-2, ap-south-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, ap-northeast-2, sa-east-1, N'Germany' and N'China'.

@CloudSecretKey

The @CloudSecretKey argument specifies the name of the Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudSecretKeyEnc

The @CloudSecretKeyEnc argument specifies the name of the encrypted Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudVendor

The @CloudVendor argument specifies the name of the cloud service provider. The argument accepts one of the following values: "AmazonS3", "AzureBlob" or "GoogleStorage".

@ProxyHost

The @ProxyHost argument is optional and specifies the name of the proxy host name that is running the proxy server.

note: If the @ProxyHost argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyLogin

The @ProxyLogin argument is optional and specifies the proxy server login credential.

note: If not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPassword

The @ProxyPassword argument is optional and specifies the proxy server password credential.

note: If the @ProxyPassword argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPasswordEnc

The @ProxyPasswordEnc argument is optional and specifies the encrypted proxy server password credential.

note: If the @ProxyPasswordEnc argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPort

The @ProxyPort argument is optional and contains the port number of the proxy server. The TCP/IP port values can be 1-65535.

note: If the @ProxyPort argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@UseSSL

The @UseSSL argument specifies that the connection uses SSL security. This argument accepts one of the following values:

- 0–Do not use SSL
- 1–Use SSL (default)

Examples

```
exec master.dbo.xp_sls_cloud_browse
 @CloudAccessKey = N'***' -- my key
, @CloudSecretKey = N'***' -- my key
, @CloudBucketName = N'aabucket1'
, @CloudFolderName = N'f1' -- f1/f2 - for subfolders
, @CloudMaxItems = 2-- number of returning rows. Works like TOP command in SQL.
```

Returns

A list of objects or error on 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_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. Double Click Restore Executables

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–Creates one Double-Click Restore executable file. Note the following warning: The executable may be greater than 4GB for large databases. Windows Server is unable to run executable files larger than 4GB. However, the file will be convertible/restorable by LiteSpeed file.
- 2–Creates a Double Click Restore loader in the same location. (Default)

For more information, see Double Click Restore Executables on page 120.

Example

```
exec xp_slscreatedcr
@FileName = N'I:\test\test.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_slsFastCompression

xp_slsFastCompression (previously known as xp_slsSmartDiff) performs a full, partial, or differential database backup using Fast Compression technology. Fast Compression

Syntax

```
EXEC master.dbo.xp_slsFastCompression
 @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']
[, @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']
```

xp_slsFastCompression (restore partial backup with fast compression)

NOTE: The following example shows the syntax for performing partial restores using the “read_write_ filegroups” parameter. The database used in the example below, FGBackups_PROD, contains the following filegroups: Primary, FG1, FG2, and FG3).

Tip: This example takes a partial backup of the primary and secondary read write filegroups (Primary, FG1, and FG2).

```
EXEC master.dbo.xp_slsFastCompression

@database = N'FGBackups_PROD'

, @BackupDirectory = 'I:\SQLbackups'

, @ExtentsChgRatioRequireFull = '.4',

@ForceFull = 1,

@read_write_filegroups = 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 or Disk IO. For more information, see Compression Methods on page 121.

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. For more information, see Adaptive Compression on page 122.

@AlterDir

Specifies the directory where to search for the backup file.

Note: @AlterDIr replaced @SearchAlternateBackupDirectory in LiteSpeed 8.6. Support for the old @SearchAlternateBackupDirectory parameter will be gradually phased out.

@AppendDifferential

Appends data to an existing full backup file.This argument accepts one of the following values:

- 0–False (default)
- 1–True

@attachedfile

- 0–False (default)
- 1–True

@AWSAccessKey

The @AWSAccessKey argument specifies the name of the unique Amazon Web Service alphanumeric access key that identifies each user.

Important: This @AWSAccessKey argument is replaced by @CloudAccessKey. The @AWSAccessKey argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSAccessKeyEnc

The @AWSAccessKeyEnc argument specifies the name of the encrypted unique Amazon Web Service alphanumeric access key that identifies each user.

Important: This @AWSAccessKeyEnc argument is replaced by @CloudAccessKeyEnc. The @AWSAccessKeyEnc argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSBucketName

The @AWSBucketName argument specifies the name of the container for AWS objects. Bucket names must be at least 3 and no more than 63 characters long.

Important: This @AWSBucketName argument is replaced by @CloudBucketName. The @AWSBucketName argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSMaxParts

Is the number of parts that are simultaneously uploaded to Amazon S3. The LiteSpeed default is 3 parts. The number of parts can be up to 5 if there is enough memory available during the upload. If you override this parameter, you may impact memory usage.

Important: This @AWSMaxParts argument is replaced by @CloudParallelUpload. The @AWSMaxParts argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSPartSize

The size of each part that is uploaded to Amazon S3 (in MB). The LiteSpeed default for Part Size is calculated as a database size divided into 9,000. The default Part Size = 25MB.

Important: This @AWSPartSize argument is replaced by @CloudPartSize. The @AWSPartSize argument is no longer valid in subsequent LiteSpeed versions after 8.2.

Notes:

- Amazon S3 has a maximum allowable 10,000 parts per file. If you override this parameter, you may inadvertently go over the 10,000 limit.
- Minimum and maximum values for Part Size are defined by Amazon S3: 5MB and 5120MB (5GB) relatively.
- The maximum object size is 5TB.

TIP: Quest Software recommends using LiteSpeed defaults.

@AWSRegionName

The @AWSRegionName argument specifies the name of the Amazon Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-west-2, us-west-1, eu-west-1, ap-southeast-1, ap-southeast-

2, ap-northeast-1, and sa-east-1.

Important: This @AWSRegionName argument is replaced by @CloudRegionName. The @AWSRegionName argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSSecretKey

The @AWSSecretKey argument specifies the name of the Amazon Web Service secret key that is assigned when you initially get an AWS account.

Important: This @AWSSecretKey argument is replaced by @CloudSecretKey. The @AWSSecretKey argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSSecretKeyEnc

The @AWSSecretKeyEnc argument specifies the name of the encrypted Amazon Web Service secret key that is assigned when you initially get an AWS account.

Important: This @AWSSecretKeyEnc argument is replaced by @CloudSecretKeyEnc. The @AWSSecretKeyEnc is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSUseGovCloud

The @AWSUseGovCloud argument enables a special restricted region for the US Government use in Amazon S3. This argument accepts one of the following values:

- 0–Do not use government cloud
- 1–Use government cloud

Important: This @AWSUseGovCloud argument is replaced by @CloudGovRegion. The @AWSUseGovCloud argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSUseReducedRedundancy

The @AWSUseReducedRedundancy argument specifies the use of reduced redundancy storage in Amazon S3. This argument accepts one of the following values:

- 0–Do not use reduced redundancy storage
- 1–Use reduced redundancy storage

Note: This @AWSUseReducedRedundancy argument is replaced with the @CloudStorageClass = 'rrs' argument.

@AWSUseServerSideEncryption

The @AWSUseServerSideEncryption argument enables the encryption of data stored at rest in Amazon S3. This argument accepts one of the following values:

- 0–Do not use Server Side Encryption
- 1–Use Server Side Encryption

@AzureBlobType

The @AzureBlobType argument specifies the types of blobs that can be stored in the Microsoft Azure cloud storage. This argument accepts one of the following values: "Block", "Page".

note: The LiteSpeed auto striping logic used in the @CloudAutoStriping and @CloudAutoStripingThreshold parameters can override the Azure blob limit for LiteSpeed backups.

@BackupDirectory

Specifies a directory for the backup file.

@backupname

Specifies the name of the backup set.

This argument accepts variables. For more information, see LiteSpeed Variables on page 125.

@buffercount

Specifies the number of SQL Server buffers available for a LiteSpeed operation. The default value is set by SQL Server.

@CheckForFullBackup

Checks if the expected full backup exists when backing up to separate files and returns a failure message if it is not found. If the full backup file does not exist, it performs a full backup. If the full backup file does exist, the decision to perform a new full or a differential backup will depend on other conditions specified. Accepts the following values:

- 0–False (default)
- 1–True

@CloudAccessKey

The @CloudAccessKey argument specifies the name of the unique Cloud Web Service alphanumeric access key that identifies each user. The selections include Amazon Access Key, Azure Account Name, Google e-mail styled account.

@CloudAccessKeyEnc

The @CloudAccessKeyEnc argument specifies the name of the encrypted unique Cloud Web Service alphanumeric access key that identifies each user.

@CloudAutoStriping

This parameter enables automatic file striping for LiteSpeed cloud backups.

@CloudAutoStripingThreshold

This parameter contains the stripe size in GBs. LiteSpeed logic uses the database size to make a decision about the number of stripes needed for LiteSpeed cloud backups. For example, if you have a database with a size of 200GB and set @CloudAutoStripingThreshold = 50, then LiteSpeed uses 200/50 = 4 stripes.

@CloudBucketName

The @CloudBucketName argument specifies the name of the container for cloud objects. Bucket names must be at least 3 and no more than 63 characters long. The selections are Amazon Bucket Name, Azure Container Name, Google Bucket Name. Google Bucket Name requirements are described at https://cloud.google.com/storage/docs/naming.

@CloudGovRegion

The @CloudGovRegion argument enables a special restricted region for the US Government use in Amazon S3 and Azure Clouds. This argument accepts one of the following values:

- 0–Do not use government cloud (default)
- 1–Use government cloud

@CloudParallelUpload

The @CloudParallelUpload argument, parallel parts transfers, is used to create fast uploads to the Azure Cloud or Amazon S3. The default number of parallel uploads:

- Amazon S3 = 3
- Azure Blob = 20

@CloudPartSize

The @CloudPartSize argument determines the size of each part that is uploaded to the cloud. The default part size:

- Amazon S3 = 25MB
- Azure Blob = 4MB

notes:

- Minimum part size for Azure Blob = 4MB
- Minimum part size for Amazon S3 = 5MB

TIP: Quest Software recommends using LiteSpeed defaults.

@CloudRegionName

The @CloudRegionName argument specifies the name of the Cloud Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-east-2, us-west-1, us-west-2, ca-central-1, eu-central-1, eu-west-1, eu-west-2, ap-south-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, ap-northeast-2, sa-east-1, N'Germany' and N'China'.

@CloudSecretKey

The @CloudSecretKey argument specifies the name of the Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudSecretKeyEnc

The @CloudSecretKeyEnc argument specifies the name of the encrypted Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudStorageClass

The @CloudStorageClass argument specifies a range of storage classes established for different use cases including:

For Amazon S3:

- Standard: Standard storage - for general-purpose storage of frequently accessed data.
- Standard-IA: Standard Infrequent Access - for long-lived, but less frequently accessed data.
- RRS: Reduced Redundancy Storage - for non-critical data considering lower level of redundancy rather than Standard storage.

Important: : In versions less than 8.5 you should use --AWSStorageClass. The @AWSStorageClass argument is no longer valid in subsequent LiteSpeed versions after 8.5.

For Google Storage:

- Multi_regional - for frequently accessed data around the world as per serving website content, streaming videos, or gaming and mobile applications.
- Regional - for frequently accessed data in the same region as your Google Cloud DataProc or the Google Compute Engine instances that use it, as per data analytics.
- Nearline - for infrequently accessed data (data you expect to access no more than once per month).
- Coldline - for infrequently accessed data (data you expect to access no more than once per year).

@CloudVendor

The @CloudVendor argument specifies the name of the cloud service provider. The argument accepts one of the following values: "AmazonS3", "AzureBlob" or "GoogleStorage".

@comment

Appends a user comment to the backup.

This argument accepts variables. For more information, see LiteSpeed Variables on page 125.

@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 for disk backups and *7* is the default value for cloud backups.

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. For more information, see Test Optimal Backup Settings on page 83.

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. Test Optimal Backup Settings 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
- 9–MS_AES_128
- 10–MS_AES_192
- 11–MS_AES_256

@database

Name of database to be backed up or restored.

This parameter specifies a database:

- to be backed up (xp_backup_database and xp_slsFastCompression)
- 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. For more information, see LiteSpeed Variables on page 125.

@DiffToFullRatioRequireFull

Specifies the last differential backup size to last full backup size ratio. When exceeding the specified ratio LiteSpeed performs a full backup.

This argument accepts one of the following formats:

- '.4'
- '40%'

@DryRun

Shows backups that are candidates for restore at this time, but does not restore them. This argument accepts one of the following values:

- 0–False (default)
- 1–True

@ElapsedDaysRequireFull

Specifies the minimum number of days since last full backup required to perform full backup. The default value is *14*.

@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. If you use the wrong encryption key, the restore will fail.

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'

@excludedatabase

Name of database(s) to exclude from this backup.

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

Tip: The @ExcludeDatabase argument can be applied together with @MultiDatabaseType to exclude several databases from the process.

@ExtentsChgRatioRequireFull

Specifies the minimum amount of database changes required for the full backup.

This 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. This argument accepts one of the following formats:

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

@FastCompressionExtension

Specifies the fast compression file extension. This argument accepts one of the following formats:

- bak - the default for new items.
- bkp - for an existing item that does not have an extension defined

@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.

@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.

@ForceDifferential

Forces differential backup. It accepts the following values:

- 0–False (default)
- 1–True

@ForceFull

Forces full backup. This argument accepts one of the following values:

- 0–False (default)
- 1–True

@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–Write new header
- 2–Long erase / write new header
- 3–Low level controller format / write new header

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.

@FullBackupEscalation

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.

This argument accepts one of the following values:

- 0–False (default)
- 1–True

@GSProject

DEPRECATED LiteSpeed 8.8: Was used to store for the Google Cloud Storage project ID; the project ID is now obtained from login. This parameter is retained for compatibility with old backup/restore scripts.

@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 and cloud 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'
```

Network Resilience

@JobP

Specifies an encrypted key. (Similar to @EncryptionKey).

You can use xp_encrypt_backup_key to convert the password (encryption_key) for use with @jobp. The original password (or encrypted key generated by xp_encrypt_restore_key) 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 (or C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs)* (or

C:\ProgramData\Quest Software\LiteSpeed\SQL Server\Logs). To log to a different directory add @Trace='logpath=path'.

See Configure Logging in LiteSpeed for information about LiteSpeed logging.

@maxtransfersize

Specifies the largest unit of transfer in bytes to be used between SQL Server and LiteSpeed. The possible values are multiples of 65536 bytes (64 KB) ranging up to 4,194,304 bytes (4 MB). The default is 1048576 (1 MB).

@MirrorDirectory

Specifies a directory for a mirror backup. You can supply multiple instances of this argument.

@MultiDatabaseType

Produces a backup that includes several types of databases. Types can include: all, system, user, or selected databases.

This argument accepts one of the following values:

- All - Backup all system and user databases.
- System - Backup only system databases.
- User - Backup only user databases.
- Selected - Backup specifically selected databases.

@olrmap

Generates a map file during a backup for Object Level Recovery. This argument accepts one of the following values:

- 0–False (default)
- 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

@ProxyHost

The @ProxyHost argument is optional and specifies the name of the proxy host name that is running the proxy server.

note: If the @ProxyHost argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyLogin

The @ProxyLogin argument is optional and specifies the proxy server login credential.

note: If not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPassword

The @ProxyPassword argument is optional and specifies the proxy server password credential.

note: If the @ProxyPassword argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPasswordEnc

The @ProxyPasswordEnc argument is optional and specifies the encrypted proxy server password credential.

note: If the @ProxyPasswordEnc argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPort

The @ProxyPort argument is optional and contains the port number of the proxy server. The TCP/IP port values can be 1-65535.

note: If the @ProxyPort argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@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.

@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. For more information, see Adaptive Compression on page 122.

@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.

You can use the %TSMDEFAULTPATH% variable to make LiteSpeed detect the default TSM configuration file path automatically (by getting from LiteSpeed defaults as a priority or the registry - HKEY_LOCAL_MACHINE\SOFTWARE\IBM\ADSM\CurrentVersion\BackupClient)

@tsmdevicetimeoutminutes

Specifies how long to wait for a TSM device.

@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.

@UseSSL

The @UseSSL argument specifies that the connection uses SSL security. This argument accepts one of the following values:

- 0–Do not use SSL
- 1–Use SSL (default)

@Verify

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

- 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. NOTE: 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

Full backup change of 40%

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'
```

Full backup to multiple locations

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
```

Force full backup

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
```

Backup to TSM change of 40%

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
```

Backup showing FastCompressionExtension argument

Backup showing FastCompressionExtension argument using existing item bkp file extension.

```
exec master.dbo.xp_slsFastCompression

@database = N'A1',

@backupname = N'A1 - Fast Compression Backup',

@desc = N'Fast Compression Backup of A1',

@compressionlevel = 2,

@comment = N'',

@AppendDifferential = 0,

@CheckForFullBackup = 1,

@ExtentsChgRatioRequireFull = N'35%',

@BackupDirectory = N'D:\Backups',

@FullBackupEscalation = 1,

@ElapsedDaysRequireFull = 14,

@FastCompressionExtension = 'bkp'
```

sls_FastCompression (Microsoft Azure)

```
exec master.dbo.xp_slsFastCompression

@database = N'newtest',

@backupname = N'newtest - Fast Compression Backup',

@desc = N'Fast Compression Backup of test',

@compressionlevel = 7,

@comment = N'',

@ExtentsChgRatioRequireFull = N'35%',

@BackupDirectory = N'test',

@CloudVendor = N'AzureBlob',

@CloudAccessKeyEnc = N'**************',

@CloudSecretKeyEnc = N'***************',

@CloudBucketName = N'test',
```

```
@AzureBlobType = N'Page',

@UseSSL = 1,

@FullBackupEscalation = 1,

@ElapsedDaysRequireFull = 14
```

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 (either the file age or the date).

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
- Files that have the filesystem archive bit set (if that option is selected)

Syntax

```
exec master.dbo.xp_slsSmartCleanup
@database = 'database_name'
, (@BackupRetainDays = 1...365 | @BackupExpiration = 'date_time')
, (@LogRetainDays = 1...365 | @LogExpiration = 'date_time')
[, @KeepArchiveFiles = 0 | 1 ]
[, @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']
```

xp_slsSmartCleanup (Amazon S3)

```
exec master.dbo.xp_slsSmartCleanup
@database = N'test',
@CloudVendor = N'AmazonS3',
@CloudBucketName = N'test',
@CloudAccessKeyEnc = N'********',
@CloudSecretKeyEnc = N'*******',
@CloudRegionName = N'us-east-1',
@UseSSL = 1,
@BackupRetainDays = 28,
@LogRetainDays = 7
```

xp_slsSmartCleanup (Azure)

```
exec master.dbo.xp_slsSmartCleanup
@database = N'test',
@CloudVendor = N'AzureBlob',
@CloudAccessKeyEnc = N'******',
@CloudSecretKeyEnc = N'*****',
@CloudBucketName = N'test',
@UseSSL = 1,
@BackupRetainDays = 28,
@LogRetainDays = 7
```

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>

@AWSAccessKey

The @AWSAccessKey argument specifies the name of the unique Amazon Web Service alphanumeric access key that identifies each user.

> Important: This @AWSAccessKey argument is replaced by @CloudAccessKey. The @AWSAccessKey argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSAccessKeyEnc

The @AWSAccessKeyEnc argument specifies the name of the encrypted unique Amazon Web Service alphanumeric access key that identifies each user.

Important: This @AWSAccessKeyEnc argument is replaced by @CloudAccessKeyEnc. The @AWSAccessKeyEnc argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSBucketName

The @AWSBucketName argument specifies the name of the container for AWS objects. Bucket names must be at least 3 and no more than 63 characters long.

Important: This @AWSBucketName argument is replaced by @CloudBucketName. The @AWSBucketName argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSRegionName

The @AWSRegionName argument specifies the name of the Amazon Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-west-2, us-west-1, eu-west-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, and sa-east-1.

Important: This @AWSRegionName argument is replaced by @CloudRegionName. The @AWSRegionName argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSSecretKey

The @AWSSecretKey argument specifies the name of the Amazon Web Service secret key that is assigned when you initially get an AWS account.

Important: This @AWSSecretKey argument is replaced by @CloudSecretKey. The @AWSSecretKey argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSSecretKeyEnc

The @AWSSecretKeyEnc argument specifies the name of the encrypted Amazon Web Service secret key that is assigned when you initially get an AWS account.

Important: This @AWSSecretKeyEnc argument is replaced by @CloudSecretKeyEnc. The @AWSSecretKeyEnc is no longer valid in subsequent LiteSpeed versions after 8.2.

@AWSUseGovCloud

The @AWSUseGovCloud argument enables a special restricted region for the US Government use in Amazon S3. This argument accepts one of the following values:

- 0–Do not use government cloud
- 1–Use government cloud

Important: This @AWSUseGovCloud argument is replaced by @CloudGovRegion. The @AWSUseGovCloud argument is no longer valid in subsequent LiteSpeed versions after 8.2.

@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.

@CloudAccessKey

The @CloudAccessKey argument specifies the name of the unique Cloud Web Service alphanumeric access key that identifies each user. The selections include Amazon Access Key, Azure Account Name, Google e-mail styled account.

@CloudAccessKeyEnc

The @CloudAccessKeyEnc argument specifies the name of the encrypted unique Cloud Web Service alphanumeric access key that identifies each user.

@CloudBucketName

The @CloudBucketName argument specifies the name of the container for cloud objects. Bucket names must be at least 3 and no more than 63 characters long. The selections are Amazon Bucket Name, Azure Container Name, Google Bucket Name. Google Bucket Name requirements are described at https://cloud.google.com/storage/docs/naming.

@CloudGovRegion

The @CloudGovRegion argument enables a special restricted region for the US Government use in Amazon S3 and Azure Clouds. This argument accepts one of the following values:

- 0–Do not use government cloud (default)
- 1–Use government cloud

@CloudRegionName

The @CloudRegionName argument specifies the name of the Cloud Web Service region to use for a bucket. Example values are but not limited to: us-east-1, us-east-2, us-west-1, us-west-2, ca-central-1, eu-central-1, eu-west-1, eu-west-2, ap-south-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, ap-northeast-2, sa-east-1, N'Germany' and N'China'.

@CloudSecretKey

The @CloudSecretKey argument specifies the name of the Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudSecretKeyEnc

The @CloudSecretKeyEnc argument specifies the name of the encrypted Cloud Web Service secret key that is assigned when you initially get a Cloud account.

@CloudVendor

The @CloudVendor argument specifies the name of the cloud service provider. The argument accepts one of the following values: "AmazonS3", "AzureBlob" or "GoogleStorage".

@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.

@database

Name of database to be backed up or restored.
This parameter specifies a database:

- to be backed up (xp_backup_database and xp_slsFastCompression)
- 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.

@Destination

Specifies the destination where to delete files from. Possible values: All, Disk, TSM, Tape, Cloud.

@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.

@GSProject

DEPRECATED LiteSpeed 8.8: Was used to store for the Google Cloud Storage project ID; the project ID is now obtained from login. This parameter is retained for compatibility with old backup/restore scripts.

@KeepArchiveFiles

Turns on monitoring and refuses to delete files that have the archive filesystem bit set. When enabled dependent files are not deleted.

@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.

@MultiDatabaseType

Produces a cleanup that includes several types of databases. Types can include: all, system, user, or selected databases.
This argument accepts one of the following values:

- All - Clean up backups for all system and user databases.
- System - Clean up backups for only system databases.
- User - Clean up backups for only user databases.
- Selected - Clean up backups for specifically selected databases.

@ProxyHost

The @ProxyHost argument is optional and specifies the name of the proxy host name that is running the proxy server.

note: If the @ProxyHost argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyLogin

The @ProxyLogin argument is optional and specifies the proxy server login credential.

note: If not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPassword

The @ProxyPassword argument is optional and specifies the proxy server password credential.

note: If the @ProxyPassword argument is not defined, then the LiteSpeed core engine checks the local .ini

files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPasswordEnc

The @ProxyPasswordEnc argument is optional and specifies the encrypted proxy server password credential.

note: If the @ProxyPasswordEnc argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@ProxyPort

The @ProxyPort argument is optional and contains the port number of the proxy server. The TCP/IP port values can be 1-65535.

note: If the @ProxyPort argument is not defined, then the LiteSpeed core engine checks the local .ini files for the proxy parameters. If the proxy parameters are not detected, then the LiteSpeed core engine connects directly to the proxy servers.

@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.

You can use the %TSMDEFAULTPATH% variable to make LiteSpeed detect the default TSM configuration file path automatically (by getting from LiteSpeed defaults as a priority or the registry - HKEY_LOCAL_MACHINE\SOFTWARE\IBM\ADSM\CurrentVersion\BackupClient)

@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.

@UseSSL

The @UseSSL argument specifies that the connection uses SSL security. This argument accepts one of the following values:

- 0–Do not use SSL
- 1–Use SSL (default)

Examples

1. Delete disk full and differential backups older than 28 days, delete log backups older than 2 days, allow deletions of the copy-only backups:

```
exec master.dbo.xp_slsSmartCleanup
@database = 'test2'
, @BackupRetainDays = 28
, @LogRetainDays = 2
, @CopyOnlyBackups = 'AllowDeletes'
```

2. Delete disk full and differential backups created before 11/15/2012:

```
exec master.dbo.xp_slsSmartCleanup
@database = 'test2'
, @BackupExpiration = '2012-11-15'
```

3. Delete tsm log backups older than 2 days:

```
exec master.dbo.xp_slsSmartCleanup
@database = N'test_tsm'
, @tsmconfigfile = N'C:\Program Files\Tivoli\TSM\baclient\dsm.opt'
, @tsmclientnode = N'w2k3_TSM2'
, @tsmclientownerpwd = N'*****'
, @LogRetainDays = 2
```

4. Delete full, differential and transaction log TSM backups created before 06/15/2012, using the PASSWORDAccess generate option to connect to the TSM Server:

```
exec master.dbo.xp_slsSmartCleanup
@database = N'test_tsm'
, @tsmconfigfile = N'C:\Program Files\Tivoli\TSM\baclient\dsm_gp.opt'
, @BackupExpiration = '2012-06-15'
, @LogExpiration = '2012-06-15'
```

5. Keep archived files older than 2 days:

```
exec master.dbo.xp_slsSmartCleanup
@database = @dbname
, @BackupRetainDays = 2
, @LogRetainDays = 1
, @KeepArchiveFiles = 1
```

6. Microsoft Azure cloud cleanup

```
exec master.dbo.xp_slsSmartCleanup

@database = N'test',

@CloudVendor = N'AzureBlob',

@CloudAccessKeyEnc = N'*****',

@CloudSecretKeyEnc = N'*******',

@CloudBucketName = N'test',

@UseSSL = 1,

@BackupRetainDays = 28,

@LogRetainDays = 7
```

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_slssqlmaint

This extended stored procedure executes the slssqlmaint.exe utility. It accepts a string that contains the command-line arguments to be passed directly to slssqlmaint.exe.

NOTE: You can generate scripts by opening tasks in the LiteSpeed UI Console and clicking **View T-SQL**. About Creating Maintenance Plans

Syntax

```
EXEC master.dbo.xp_slssqlmaint '<task_options> '
```

Script Maintenance Plans Tasks about the task options and the syntax for scripting maintenance plan tasks.

Examples

Back up database

```
exec master.dbo.xp_slssqlmaint N'-D Regex:"LiteSpeed" Regex:"DB1" -BkUpMedia DISK -
DelBkUps 3DAYS -BkUpDB "C:\temp" -CrBkSubDir -BkExt "bak" -Logging 1 -Reliability 1 -
CompressionLevel 1 -Default "%D_%T_%z.%EXT%" -Exclude Offline LogShippng ReadOnly '
```

Mirror to Disk

```
execute master.dbo.xp_slssqlmaint N'-D "at1" -BkUpMedia DISK -BkUpDB "c:\backup" -
BkFileName -Logging 0 -CompressionLevel 2 -Mirror "c:\mirror\" -OPTOLR -SmartDiff
14DAYS -DataDelta 35 -SingleFile 0 -BackupEsc -Exclude Offline LogShippng
IgnoreReplica Secondary ReadOnly '
```

Mirror to Cloud

```
execute master.dbo.xp_slssqlmaint N'-D "TestDatabase" -BkUpMedia DISK -BkUpDB
"c:\backup\%SERVER%_%D_%T_%z.bak" -BkFileName -Logging 0 -CompressionLevel 2 -Mirror
"c:\mirror\" "
{""cloud"":""AmazonS3"",""accessKey"":""lkjflkjsldjiofsjrdfftgrux5j+OwkI"",""secretKey
"":""’plkljhlkwjnuildiIujUhjkHkldkflkdfe"",""container"":""irelandaatest"",""authType"
":""AccessAndSecretKeys"",""region"":""us-east-
1"",""storageClass"":""0"",""useSSE"":""False"",""useSSL"":""True"",""isGovCloud"":""F
alse"",""useAcceleration"":""False"",""useAutoStriping"":""True"",""autoStripSize"":""
0"",""paths"":[""test/""]}" -OPTOLR -Exclude Offline LogShippng IgnoreReplica
Secondary ReadOnly '
```

Clean up maintenance plans

```
exec master.dbo.xp_slssqlmaint '-MAINTDEL -DELTYPE FileBkup -DELSUBFOLDERS -
DelEmptyFolder -DELFOLDER "C:\temp\" -DELEXTENSION "bak" -DELUNIT "3" -DELUNITTYPE
"WEEKS" -DELUSEAGE -NO_OUTPUT '
```

xp_sqllitespeed_licenseinfo

Using xp_sqllitespeed_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_sqllitespeed_licenseinfo
[ @licensekey = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX-111-22222-YY'
[, @sitemessage = 'Site_Message' ]
[, @ShowLicenseNumber = 1 ]
[, @store = 1]
| @remove = 1]
```

Arguments

@licensekey	Passes license key value.
@sitemessage	Passes license site message. Applicable to upgrades from a LiteSpeed 8.5 or earlier installation.
@ShowLicenseNumber = 1	Show the Customer License Number. The Customer License Number is part of the license key and is required to access support.
@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_sqllitespeed_licenseinfo
   @licensekey = 'C20TM3Q3K2HD74UDLBMHC6KYV6HZ3MQFNXZFB-123-45678-34'
   ```

2. Register a license key:

```
EXEC master.dbo.xp_sqllitespeed_licenseinfo
@licensekey = 'C20TM3Q3K2HD74UDLBMHC6KYV6HZ3MQFNXZFB-123-45678-34'
, @store=1
```

3. Remove license key:

```
EXEC master.dbo.xp_sqllitespeed_licenseinfo
@remove= 1
```

4. View information about the supplied license key (upgrades from a LiteSpeed 8.5 or earlier installation):

```
EXEC master.dbo.xp_sqllitespeed_licenseinfo
@licensekey = 'C20TM3Q3K2HD74UDLBMHC6KYV6HZ3MQFNXZFB-123-45678-34'
, @sitemessage = 'Trial Version'
```

5. Register a license key (upgrades from a LiteSpeed 8.5 or earlier installation) :

```
EXEC master.dbo.xp_sqllitespeed_licenseinfo
@licensekey = 'C20TM3Q3K2HD74UDLBMHC6KYV6HZ3MQFNXZFB-123-45678-34'
, @sitemessage = 'Trial Version'
, @store=1
```

6. Show the Customer License Number. The Customer License Number is part of the license key and is required to access support.

```
EXEC master.dbo.xp_sqllitespeed_licenseinfo @ShowLicenseNumber = 1
```

Result Set

Column Name	Data Type	Description
Product Name	nvarchar (128)	LiteSpeed for SQL Server.
License Key	nvarchar (128)	Key license value.
License Site Message	nvarchar (128)	Site message. Applicable to upgrades from a LiteSpeed 8.5 or earlier installation.
License Type	nvarchar (128)	One of the following: • Trial–a dated trial license key that expires on a specific date • Permanent–a permanent license key • Term–similar to trial keys, except it comes with support
Edition	nvarchar (128)	Enterprise or Standard.
Site License	bit	Whether the key is a Site license key that is used company-wide.

Column Name	Data Type	Description
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. **NOTE:** 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).

About Using Extended Stored Procedures

xp_sqllitespeed_version

This extended stored procedure returns the name and version of the installed LiteSpeed components.

```
exec master.dbo.xp_sqllitespeed_version
```

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 = 1 | 0 ]
[, @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.

NOTE: @desc is only supported if @TSMBriefList=0.

@tsmarchive

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

- 0–False (default)
- 1–True

@tsmbrieflist

Returns a brief list. This argument accepts one of the following values:

- 0–False
- 1–True (default)

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.

You can use the %TSMDEFAULTPATH% variable to make LiteSpeed detect the default TSM configuration file path automatically (by getting from LiteSpeed defaults as a priority or the registry - HKEY_LOCAL_MACHINE\SOFTWARE\IBM\ADSM\CurrentVersion\BackupClient)

@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 (default)
- 1–True

@tsmsortbypit

Sorts the results by the point-in-time date. It accepts one of the following:

- 0–False (default)
- 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.

Column Name	Data Type	Description
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: • 1–Database • 2–Transaction Log • 4–File • 5–Differential Database • 6–Differential File • 7–Partial • 8–Partial Differential
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.
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.

Column Name	Data Type	Description
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.
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: • 0–not encrypted • 1–encrypted
IsCopyOnly	Bit	Indicates if the backup is copy-only:

Column Name	Data Type	Description
		• 1–A copy-only backup Note: A copy-only backup does not impact the overall backup and restore procedures for the database.

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.

You can use the %TSMDEFAULTPATH% variable to make LiteSpeed detect the default TSM configuration file path automatically (by getting from LiteSpeed defaults as a priority or the registry - HKEY_LOCAL_MACHINE\SOFTWARE\IBM\ADSM\CurrentVersion\BackupClient)

@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	Nvarchar	Specifies how many days all but the most recent backup version is retained. The

Column Name	Data Type	Description
extra versions	(128)	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.
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
```

12

Troubleshoot LiteSpeed

Review Known Issues

Refer to the Release Notes for the current list of issues known to exist in this version of LiteSpeed.

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.

Dell storage appliance not working with LiteSpeed

LiteSpeed has been tested and is certified to work with Dell DR4000, DR4100, and DR6000 storage appliances. In rare cases the DR storage appliance may not work with LiteSpeed.

To check the DR storage appliance

1. Confirm the access rights of the SQLserver\agent startup account to the DR resource.
2. Confirm network connectivity between the SQL server and the DR storage appliance.

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 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.

Analyze log information

Enable advanced logging and review log files for errors or work with Quest Software support to identify and resolve plan problems. Reporting and Logging in Maintenance Plans

NOTE: For all scenarios, the best is to upgrade all of your servers to the latest LiteSpeed version.

Upgrade LiteSpeed Maintenance Plans

LiteSpeed for SQL Server Upgrade

If upgrading LiteSpeed 5.x or later, the Maintenance Plans should remain intact and no additional steps are required. See the LiteSpeed *Installation Guide* for more information about the upgrade scenario.

If upgrading LiteSpeed 4.x, the SSMS plug-in maintenance plans need to be manually upgraded. LiteSpeed allows you to read SSMS plug-in maintenance plans and upgrade them to SSIS LiteSpeed maintenance plans.

To upgrade a maintenance plan

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

NOTE: The **Remove files older than** option in the Backup Database task does not remove LiteSpeed 4.x SSIS maintenance plans backups. To remove old backup files for these plans, you can use a separate **Clean up maintenance plans** task.

SQL Server In-Place Upgrade

SSIS LiteSpeed plans must be manually upgraded to use the appropriate library, otherwise they will fail after SQL Server upgrade. If you have at least one LiteSpeed task in a SSIS plan, then you need to upgrade the plan after upgrading SQL Server to a higher version.

To upgrade plans in LiteSpeed 6.1 or later

Right-click an instance in the Maintenance Plan pane and select **Upgrade LiteSpeed Maintenance plans**.

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. For more information, see xp_memory_size on page 341.

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 https://www.quest.com/products/litespeed-for-sql-server/ 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.

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. For more information, see Log File Naming and Location on page 578.

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.

To enable backup/restore logging, do one of the following:

- In wizards, access options or advanced options and set the logging level.
- Supply the @logging parameter if using the extended stored procedures.

- Supply the -L(--LogLevel) parameter from the command line. For more information, see LiteSpeed Command-Line Arguments on page 179..

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. For more information, see Reporting and Logging in Maintenance Plans on page 579. 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.
   - SmartCleanup–To log SmartCleanup activity.
   - SLSMedia–To log SLSMedia 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. For more information, see Log File Naming and Location on page 578.
Flush output to log file after every write (slower)	LiteSpeed will perform a disk flush after each record. Select this parameter if you experience program exceptions resulting in a process abort.
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 (or C:\ProgramData\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\SQL Server\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 'SLSInstall'-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 **Application button | 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
Write a text file report in directory	Select a directory in which to write a text file report. Alternately select to browse and locate a directory • Delete text files older than - Select to delete files based on their age in number of minutes, hours, days, weeks, months, and years.
Write an html file report in directory	Select a directory in which to write an html file report. Alternately select to browse and locate a directory. • Delete html files older than - Select to delete files based on their age in number of minutes, hours, days, weeks, months, and years.
Send report to an email recipient	Select to send an email report to a designated recipient. • Agent operator - Use the drop-down to select an agent operator to receive reports by email.
Log extended information	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. 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 history.
Log to remote Server	Writes the plan history to the remote server. Select the connection drop-down to choose the server. Alternately assign a new server.
Debug Logging	Logs LiteSpeed maintenance plan utility activity. Quest Software support may ask you to enable debug logging in maintenance plans to help

Option	Description
	troubleshoot product issues. Make sure LiteSpeed is installed on the server where you have a problem with maintenance plans. When a subplan is executed, the Maintenance Plan engine saves the archived <plan_name>_<subplan_name>[<GUID>].zip file in the default output directory.Configure Logging in LiteSpeed about the default output directory. Each archive has: • one or more files for all LiteSpeed tasks in a subplan, including the Reporting task • one file containing `-Gather` command-line command to collect statistics • one file containing `-CollectStats` command-line command for the process that starts collecting statistics to repositories (only created for SSIS plans) LiteSpeed's debug logging is not supported for the native SQL Server tasks. If the **Use LiteSpeed** option is cleared for a task, then that task is handled by the native SQL Server components.

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 **Application button | Help | Support Bundle**.

To create a server support bundle

1. Right-click a server in the tree and select **Support | LiteSpeed Core Details**.
2. Send the support bundle to Quest Software customer support or click **Clipboard** to save the generated content to file.

13

Review Additional Resources

LiteSpeed Community

Get the latest product information, find helpful resources, and join a discussion with the LiteSpeed team and other community members. Join the LiteSpeed community at: https://www.quest.com/community/products/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

Useful Web Resources

Share and find information about Quest Software products and SQL Server related technologies at:

ToadWorld	http://www.toadworld.com/platforms/sql-server/default.aspx
Microsoft SQL Server World Wide Users Group	http://www.sswug.org
Microsoft Product Support	http://support.microsoft.com

Microsoft Developer Network (MSDN)	http://msdn.microsoft.com
Professional Association for SQL Server	http://www.sqlpass.org
Microsoft SQL Server Developer Center	http://msdn.microsoft.com/en-us/sqlserver
TechNet Site	http://technet.microsoft.com
Microsoft SQL Server Web Site	http://www.microsoft.com/sql
IBM Spectrum Protect (Tivoli Storage Manager - TSM) Information Center	http://publib.boulder.ibm.com/infocenter/tsminfo/v6r3/index.jsp

About us

We are more than just a name

We are on a quest to make your information technology work harder for you. That is why we build community-driven software solutions that help you spend less time on IT administration and more time on business innovation. We help you modernize your data center, get you to the cloud quicker and provide the expertise, security and accessibility you need to grow your data-driven business. Combined with Quest's invitation to the global community to be a part of its innovation, and our firm commitment to ensuring customer satisfaction, we continue to deliver solutions that have a real impact on our customers today and leave a legacy we are proud of. We are challenging the status quo by transforming into a new software company. And as your partner, we work tirelessly to make sure your information technology is designed for you and by you. This is our mission, and we are in this together. Welcome to a new Quest. You are invited to Join the Innovation™.

Our brand, our vision. Together.

Our logo reflects our story: innovation, community and support. An important part of this story begins with the letter Q. It is a perfect circle, representing our commitment to technological precision and strength. The space in the Q itself symbolizes our need to add the missing piece – you – to the community, to the new Quest.

Contacting Quest

For sales or other inquiries, visit www.quest.com/contact.

Technical support resources

Technical support is available to Quest customers with a valid maintenance contract and customers who have trial versions. You can access the Quest Support Portal at https://support.quest.com.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request
- View Knowledge Base articles
- Sign up for product notifications
- Download software and technical documentation
- View how-to-videos
- Engage in community discussions
- Chat with support engineers online
- View services to assist you with your product

Third-party contributions

This product contains the following third-party components. For third-party license information, go to https://www.quest.com/legal/license-agreements.aspx. Source code for components marked with an asterisk (*) is available at https://opensource.quest.com.

Table 1: List of Third-Party Contributions

Component	License or acknowledgement
AWS SDK for .NET 3.3.24	AWS SDK for .NET Copyright 2009-2017 Amazon.com, Inc. or its affiliates. All Rights Reserved. ********** THIRD PARTY COMPONENTS ********** This software includes third party software subject to the following copyrights: -Json processing from LitJson - All the source code and related files distributed with this software have been dedicated to the public domain by the authors. Anyone is free to copy, modify, publish, use, compile, sell, or distribute the software, either in source code form or as a compiled binary, for any purpose, commercial or non-commercial, and by any means. - Parsing PEM files from Bouncy Castle - Copyright (c) 2000 - 2011 The Legion Of The Bouncy Castle (http://www.bouncycastle.org) Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING

Component	License or acknowledgement
	FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE. - Performing CRC32 checks from vbAccelerator.com vbAccelerator Software License Version 1.0 Copyright (c) 2002 vbAccelerator.com Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by vbAccelerator (http://vbaccelerator.com/)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear. The names "vbAccelerator" and "vbAccelerator.com" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact vbAccelerator through steve@vbaccelerator.com. Products derived from this software may not be called "vbAccelerator", nor may "vbAccelerator" appear in their name, without prior written permission of vbAccelerator. THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL VBACCELERATOR OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. This software consists of voluntary contributions made by many individuals on behalf of the vbAccelerator. For more information, please see http://vbaccelerator.com/ . The vbAccelerator licence is based on the Apache Software Foundation Software Licence, Copyright (c) 2000 The Apache Software Foundation. All rights reserved. - MD5 checks in the Windows Phone 8 variant of the SDK from Microsoft

Component	License or acknowledgement
	Corporation Microsoft Public License (MS-PL) This license governs use of the accompanying software. If you use the software, you accept this license. If you do not accept the license, do not use the software. 1. Definitions The terms "reproduce," "reproduction," "derivative works," and "distribution" have the same meaning here as under U.S. copyright law. A "contribution" is the original software, or any additions or changes to the software. A "contributor" is any person that distributes its contribution under this license. "Licensed patents" are a contributor's patent claims that read directly on its contribution. 2. Grant of Rights (A) Copyright Grant- Subject to the terms of this license, including the license conditions and limitations in section 3, each contributor grants you a non-exclusive, worldwide, royalty-free copyright license to reproduce its contribution, prepare derivative works of its contribution, and distribute its contribution or any derivative works that you create. (B) Patent Grant- Subject to the terms of this license, including the license conditions and limitations in section 3, each contributor grants you a non-exclusive, worldwide, royalty-free license under its licensed patents to make, have made, use, sell, offer for sale, import, and/or otherwise dispose of its contribution in the software or derivative works of the contribution in the software. 3. Conditions and Limitations (A) No Trademark License- This license does not grant you rights to use any contributors' name, logo, or trademarks. (B) If you bring a patent claim against any contributor over patents that you claim are infringed by the software, your patent license from such contributor to the software ends automatically. (C) If you distribute any portion of the software, you must retain all copyright, patent, trademark, and attribution notices that are present in the software. (D) If you distribute any portion of the software in source code form, you may do so only under this license by including a complete copy of this license with your distribution. If you distribute any portion of the software in compiled or object code form, you may only do so under a license that complies with this license. (E) The software is licensed "as-is." You bear the risk of using it. The contributors give no express warranties, guarantees or conditions. You may have additional consumer rights under your local laws which this license cannot change. To the extent permitted under your local laws, the contributors exclude the implied warranties of merchantability, fitness for a particular purpose and non-infringement. ** iOS4Unity - https://github.com/Hitcents/iOS4Unity/ ** SQLitePCLRaw - https://github.com/ericsink/SQLitePCL.raw Apache License

Component	License or acknowledgement
	Version 2.0, January 2004 http://www.apache.org/licenses/ TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION 1. Definitions. "License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document. "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License. "Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity. "You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License. "Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files. "Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types. "Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below). "Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof. "Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution." "Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

Component	License or acknowledgement
	2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form. 3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed. 4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions: (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and (b) You must cause any modified files to carry prominent notices stating that You changed the files; and (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License. 5. Submission of Contributions. Unless You explicitly state otherwise, any

Component	License or acknowledgement
	Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions. 6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file. 7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License. 8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages. 9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability. END OF TERMS AND CONDITIONS - Endian Conversion for EventStream Event decoding [Amazon.Util.EventStreams.EndianConversionUtility] from IPAddress.cs https://github.com/Microsoft/referencesource/ The MIT License (MIT) Copyright (c) Microsoft Corporation Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

Component	License or acknowledgement
	The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
BouncyCastle.Crypto.dll 1.8.1	Copyright (c) 2000 - 2017 The Legion of the Bouncy Castle Inc. (http://www.bouncycastle.org) Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Google APIs Client Library for .NET 1.30.0	Apache 2.0
Google.Api.Gax (Google API) 2.0	Copyright (c) Jerome Touffe-Blin ("Author") All rights reserved. The BSD License Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT

Component	License or acknowledgement
	SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Google.Api.Gax.Rest (Google API) 2.0	Copyright (c) Jerome Touffe-Blin ("Author") All rights reserved. The BSD License Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Google.Cloud.Storage.V1 (Google API) 2.0	Copyright 2017 Google Inc. Apache 2.0
JEDI JCL/JVCL 1.0	Mozilla Public License (MPL) 1.1
Json.NET 10.0.3	Copyright © 2017 Newtonsoft Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

Component	License or acknowledgement
	THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Log4Net 2.0.7	Apache 2.0 license Copyright 2004-2017 The Apache Software Foundation This product includes software developed at The Apache Software Foundation (http://www.apache.org/).
Microsoft Windows Azure Storage 8.0	Apache License Version 2.0, January 2004
Newtonsoft.Json 10.0.3	Copyright (c) 2007 James Newton-King Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
pugixml 1.9	Copyright (c) 2006-2015 Arseny Kapoulkine Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND

Component	License or acknowledgement
	NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
SharpZipLib 0.81.0.1407	The library is released under the GPL with the following exception: Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License cover the whole combination. As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version. Note The exception is changed to reflect the latest GNU Classpath exception. Older versions of #ziplib did have another exception, but the new one is clearer and it doesn't break compatibility with the old one. Bottom line In plain English this means you can use this library in commercial closed-source applications.
Spring4D 1.2.1	Copyright (c) 2009 - 2018 Spring4D Team. Apache License Version 2.0
Task Scheduler Managed Wrapper 2.7.3	Copyright (c) 2003-2010 David Hall Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
TimeSpan Helper Library 2.1.6	BSD Copyright (c) 2009, dahall All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Component	License or acknowledgement
	Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of CodePlex Community nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Virtual Treeview (JAM Software) 6.5	Mozilla Public License (MPL) 1.1
WinForms Group Controls 1.5.3	This component is governed by the Apache License 2.0 (Apache)
Wizard .NET Library 2.1.10	Copyright (c) 2013 David Hall Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
zlib 1.2.8	Portions copyright 1995-2012 Jean-loup Gailly and Mark Adler
zlib portable 1.11.0	Portions copyright 1995-2012 Jean-loup Gailly and Mark Adler

Index

%

.

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

X