

SQL virtual database helps you quickly and easily access data and objects from your backup files without performing costly restore operations or impacting your production servers.

You can get a quick glimpse into the newest features, fixed issues, and known issues in this release of SQL virtual database by accessing the [Release Notes](#).

- [Review issues fixed by this release](#)
- [Review previous features and fixed issues](#)
- [See known issues](#)
- [See list of recommended Idera Solutions](#)

Updated: 1/10/2012 1:24 PM

Use the following checklist to get started with SQL virtual database.

<input checked="" type="checkbox"/>	Get Started Checklist Steps
<input type="checkbox"/>	Locate the backup files for the database you want to attach.
<input type="checkbox"/>	Decide whether you want to attach a full backup file or attach multiple data files such as transaction log backups.
<input type="checkbox"/>	Create the virtual database according to your specific recovery needs.
<input type="checkbox"/>	Use the Query tool to verify your virtual database was attached correctly .
<input type="checkbox"/>	Use SQL Server Management Studio or another database management tool to view, compare, recover, and report on objects and data from your virtual databases.
<input type="checkbox"/>	Decide whether you want to update your existing SQL Server backup jobs to create file maps automatically after the regularly scheduled backup operation is completed. This improves the virtual database performance, allowing you to more quickly access future backups.
<input type="checkbox"/>	Monitor the amount of disk space used by your virtual databases . Typically, virtual databases require a fraction of the disk space consumed by a fully restored backup. <i>If you are virtualizing large databases</i> , you may need to allocate additional disk space over time.

You can easily and quickly upgrade to the latest version of SQL virtual database. Upgrading will not delete or alter any existing virtual databases hosted by the SQL Server instance on the installation computer. For more information about this release, see what's new.

To upgrade:

1. Use an administrator account to log onto the target computer.
2. Start SQL toolbox.
3. Click the **Updates** tab.
4. Select **SQL virtual database**, and then click **install**. SQL toolbox silently installs the latest version.
5. Start the SQLvdb Console. You can access your existing virtual databases or create new ones.

Idera SQL virtual database (SQLvdb) is a powerful one-of-a-kind solution that lets you attach SQL Server backup files and query them like real databases. With its revolutionary, patent-pending technology, you gain instant access to critical data in a backup file without spending the time and storage previously required for restore. In minutes, you can create a virtual database and then use any native SQL Server or third party tools to query and extract the data you need.

Features

Virtual recovery

Provides instant, feature-rich access to all data from within SQLsafe backup and native SQL Server backup files.

Point-in-time selection

Provides point-in-time selection and recovery, allowing granular control over the state of the data displayed in the virtual database.

Native SQL Server and third-party application access

Use existing SQL Server tools such as SQL Server Management Studio and third-party applications to interact with the new virtual database as though it were an actual physical database.

Intuitive Console

Allows virtual databases to be quickly and easily created, edited, or removed.

No impact to production servers

Installs to a single non-critical server and attaches all virtual databases to a single SQL Server instance.

Database administrators are often asked to recover a backup in order to recover objects and data, compare two systems, or run a report. Previously, the only way to do this is to restore the database, a consuming activity that requires time and storage.

SQL virtual database lets you create a 'virtual database' that appears to SQL Server and third-party applications as a physical database. This virtual database provides direct access to the data without a restore. This direct access greatly reduces the time required for data recovery, increases your productivity, and reduces resource requirements.

With SQL virtual database, you can:

- Eliminate long restore times
- Access a backup file and retrieve the objects or data you need *immediately*
- Save massive amounts of storage
- Execute any query
- Use backups for reporting, data extraction, data analysis, and more
- Take the load off your production servers and experience zero impact on production databases
- Access full, differential, and transaction log backup files made by [Idera SQLsafe](#) and SQL Server

SQL virtual database is driven by a unique, patent-pending recovery technology that makes backup files behave like regular physical databases. This technology allows you to create virtual databases by directly attaching backup files to SQL Server without requiring a restore operation.

Each virtual database is hosted by the SQL Server instance you specify. All objects and data in the virtual databases are accessible to SQL Server jobs and stored procedures, T-SQL scripts, and other third-party applications.

The SQLvdb Engine, which consists of a driver and a service, manages the virtual database and services the SQL Server I/O requests to the virtual data files. As the SQLvdb Engine attaches your selected backup files and creates the virtual database, it consolidates all the transactions and presents them to SQL Server for recovery. During this process, SQL virtual database "maps" the data from the backup files. *If the data has already been "mapped"*, such as through SQLsafe or the [Map Generation utility](#), then SQL virtual database proceeds to the virtual database creation. You can create a virtual database by attaching specific data sets from multiple full, differential, and transaction log backups.

You can then use the SQLvdb Console, or any other third-party tool, to perform queries against specific virtual databases or simply check the status of your virtual databases. For more information, see our instructions on [how to use your virtual database](#), as well as [tips and tricks](#).

The product features, such as the virtual database creation, are also available through the SQLvdb CLI. For more information, see the [product component](#) descriptions.

How does SQL virtual database work with SQLsafe?

SQL virtual database seamlessly converts SQLsafe backup files into virtual databases. You can use SQL virtual database with any backup file generated by SQLsafe 5.0 or later. For more information, see the [product requirements](#).

When you attach a backup file previously created by SQLsafe 5.0, the SQLvdb Engine performs the same steps as it does with the native SQL Server backup. To optimize performance for large databases, use the [Map Generation utility](#) to generate the required metadata before attaching the backup files.

With SQLsafe 6.0 or later, you can configure your backup operation to automatically generate the metadata. By default, this option is enabled when you perform a new manual backup or define a new backup policy. For more information, see the SQLsafe online Help.

What virtual data files are created?

For each virtual database you create, several files are also created to support the new virtual database. These virtual data files are:

File	Description	Location	Can you move it?	When is it deleted?
*.xbm *.xcm *.xdm *.xtm	Files that the SQLvdb Driver creates to map the data in the selected backup files.	In the same folder where you store your backup files.	No ... Moving these files will cause errors when you attempt to access the virtual database.	Never ... After you have retrieved the data you need from the virtual database, you can use the groom feature to delete unused data files, manually delete these files, or schedule a job to periodically groom old files.
*.lcf *.mcf *.ncf	Files that the SQLvdb Service creates when	In the VDB/Sparse	No but you can specify a new location ...	When the virtual database is removed (dropped) from the SQL Server

File	Description	Location	Can you move it?	When is it deleted?
	reading changed pages from the transaction log for the selected backup files.	folder under the Program Files directory on the client computer.	<p>Moving these files when the virtual database is attached will cause errors when you attempt to access its data.</p> <p>To choose a different location for these files, delete your virtual databases, and then specify a new directory path. When you create the next virtual database, the SQLvdb Engine will use the new path.</p>	instance.
*.lxf *.mxf *.nxf	Virtualized versions of the .lcf and .mcf files associated with the selected backup set.	None ... These files are placeholders for SQL Server.	No ... These files do not exist on your disk.	When the virtual database is removed (dropped) from the SQL Server instance.
*.vdb	An XML file that contains the configuration information for the virtual database.	In the Repository folder under the Program Files directory on the client computer.	No ... Moving these files will cause errors when you attempt to access the virtual database.	When the virtual database is removed (dropped) from the SQL Server instance.

How much disk space should I allocate for the virtual data files?

The amount of disk space consumed by the virtual data files depends on the size and content of your backup files. For more information, see the [recommended disk space allocations](#).

TIP The size for these files may display incorrectly in tools such as Management Studio. To see the correct size, use Windows Explorer to navigate to the file and then view the file properties.

This documentation set includes a comprehensive online Help system as well as additional resources that support you as you install and use the product. You can also search the Idera Solutions knowledge base, available at the [Idera Customer Service Portal](http://www.idera.com/support) (www.idera.com/support).

Please contact us with your questions and comments. We look forward to hearing from you. For support around the world, please contact us or your local partner. For a complete list of our partners, please see our Web site.

Who?	How?
Sales	713.523.4433 1.877.GO.IDERA (464.3372) (only in the United States and Canada) sales@idera.com
Licensing	licensing@idera.com
Support	http://www.idera.com/Support/Tools-and-Utilities/
Web site	www.idera.com

Idera documentation uses consistent conventions to help you identify items throughout the printed online library.

Convention	Specifies
Bold	Window items
<i>Italics</i>	Book and CD titles Variable names New terms
Fixed Font	File and directory names Commands and code examples Text typed by you
Straight brackets, as in [value]	Optional command parameters
Curly braces, as in {value}	Required command parameters
Logical OR, as in value 1 value 2	Exclusively command parameters where only one of the options can be specified

At Idera, we have engineered a new generation of tools for managing and administering the world's fastest growing database management system - Microsoft SQL Server. Headquartered in Houston, Texas, Idera boasts numerous systems and database management industry veterans devoted to bringing proven solutions to you, the database professional. We provide products that install in minutes, configure in hours and deploy worldwide in days to dramatically increase the performance, availability and efficiency of SQL Server operations, and the productivity of database administrators.

Our tools are engineered to scale - from managing a single server to enterprise deployments with thousands of servers. Idera products combine ease of use with a design that installs in minutes, configure in hours, and deploy worldwide in days. To learn more about Idera products, visit <http://www.idera.com/products>.

Idera – a division of BBS Technologies, Inc. IntelliCompress, DTx, Idera Newsfeed, Newsfeed, Point admin toolset, PowerShellPlus, SharePoint admin toolset, SharePoint backup, SharePoint performance monitor, SQL admin toolset, SQL change manager, SQLcheck, SQLcompliance, SQL compliance manager, SQLconfig, SQL defrag manager, SQL diagnostic manager, SQL diagnostic manager mobile, SQLdm, SQLdm Mobile, SQL doctor, SQL job manager, SQLpermissions, SQL mobile manager, SQLsafe, SQLsafe Freeware Edition, SQLscaler, SQLschedule, SQLsecure, SQLsmarts, SQLstats, SQLtool, Idera, BBS Technologies and the Idera logo are trademarks or registered trademarks of BBS Technologies, Inc., or its subsidiaries in the United States and other jurisdictions. All other company and product names may be trademarks or registered trademarks of their respective companies. © 2012 BBS Technologies, Inc., all rights reserved.

THIS DOCUMENT AND THE SOFTWARE DESCRIBED IN THIS DOCUMENT ARE FURNISHED UNDER AND ARE SUBJECT TO THE TERMS OF A LICENSE AGREEMENT. EXCEPT AS EXPRESSLY SET FORTH IN SUCH LICENSE AGREEMENT, BBS TECHNOLOGIES, INC., PROVIDES THIS DOCUMENT AND THE SOFTWARE DESCRIBED IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. SOME STATES DO NOT ALLOW DISCLAIMERS OF EXPRESS OR IMPLIED WARRANTIES IN CERTAIN TRANSACTIONS; THEREFORE, THIS STATEMENT MAY NOT APPLY TO YOU. YOU ARE ENCOURAGED TO READ THE LICENSE AGREEMENT BEFORE INSTALLING OR USING THIS DOCUMENTATION OR SOFTWARE.

Some companies, names, and data in this document are used for illustration purposes and may not represent real companies, individuals, or data.

This document could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein. These changes may be incorporated in new editions of this document. BBS Technologies, Inc., may make improvements in or changes to the software described in this document at any time.

© 2003-2012 BBS Technologies, Inc., all rights reserved.

U.S. Government Restricted Rights: The software and the documentation are commercial computer software and documentation developed at private expense. Use, duplication, or disclosure by the Government is subject to the terms of the BBS Technologies, Inc., standard commercial license for the software, and where applicable, the restrictions set forth in the Rights in Technical Data and Computer Software clauses and any successor rules or regulations.

Deploy an installation of SQL virtual database to each SQL Server computer that hosts the databases you want to virtualize. Install all of the [product components](#) on the same computer.

How do I install SQL virtual database?

1. Review the [product requirements](#).
2. Log onto the target computer using an account with administrator privileges.
3. Download and install the SQL toolbox.
4. From the **Tools available for install** list, select **SQL virtual database**.
5. Click **install**.
6. Start the SQLvdb Console. When you start the Console, the [licensing console](#) displays.
7. Choose the appropriate licensing option. You will need a production license for each computer on which you install SQL virtual database. For more information, see [product licensing](#).

Can I deploy SQL virtual database to a clustered SQL Server environment?

Yes. SQL virtual database supports [Windows 2003](#) and [Windows 2008](#) clusters.

Before installing SQL virtual database, check the following product component definitions and architecture, and review [how the product works](#).

Product components

CLI

Is the command line interface. The SQLvdb CLI calls the Management Service to manage the creation, modification, and removal of your virtual databases. You can call the CLI from new or existing T-SQL scripts, such as your SQL Server backup jobs. To access the SQLvdb CLI, start the Windows Command Prompt, and then enter `sqlvdbcli` to see which options you can use.

Console

Is the user interface that allows you to manage your virtual databases and query any data or object from the virtual databases. You can also use the SQLvdb Console to change your product configuration and access online Help.

Driver

Is a component of the SQLvdb Engine that is attached to a specific drive on the target SQL Server computer. By default, the SQLvdb Driver is attached to the `C:\` drive. The driver handles the I/O requests required to create the virtual database. You can [change the drive](#) to which the driver is attached.

Engine

Consists of the driver and the service. The engine components communicate with SQL Server, and create and maintain your virtual database.

Service

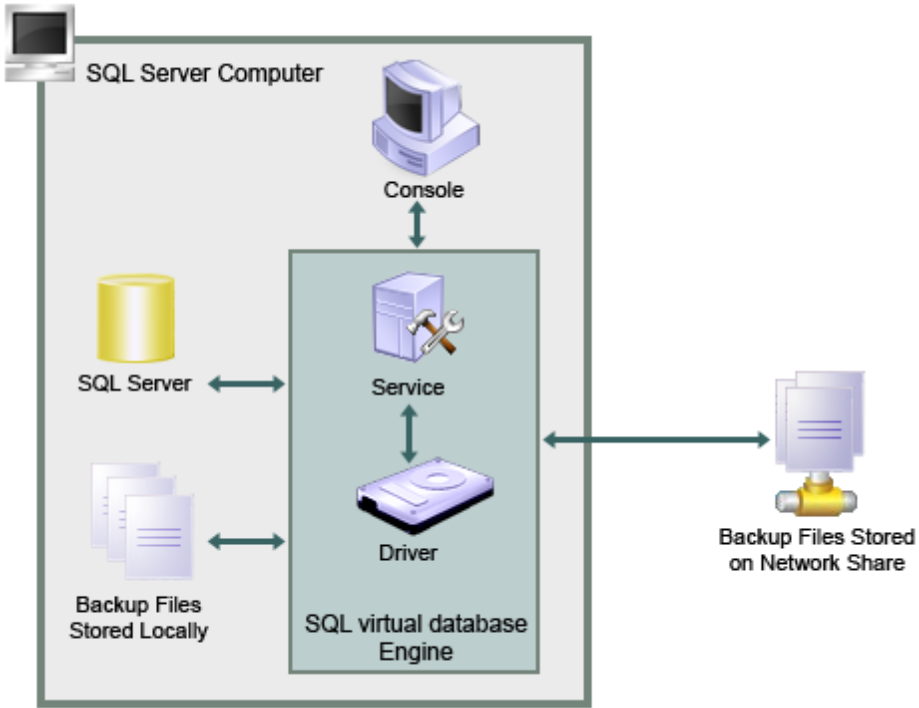
Is a component of the SQLvdb Engine that is installed on the target SQL Server computer. The SQLvdb Service uses the configured Windows account or SQL Server login credentials to communicate with SQL Server. This service also coordinates with the driver to retrieve data from the specified backup files and then create data files for your virtual database. You can [specify a different folder](#) in which to store the virtual data files.

Virtual Data Files

Is a set of files that includes information about your product configuration, virtual database settings, and other data. For more information about these files, review [how SQLvdb works](#).

Product architecture

The following graphic illustrates the relationship between the product components. All SQLvdb components are installed on the same physical SQL Server computer.



You can easily and quickly install SQL virtual database on any computer that meets or exceeds the following hardware, software, and permission requirements. Also review the [product components and architecture](#) as well as [how the product works](#).

Console requirements

Ensure the computer on which you want to install the SQLvdb Console meets or exceeds the following requirements.

Hardware/Software	Requirement
CPU	2.0 GHz or higher
Memory	1 GB
32-bit Operating System	Either: <ul style="list-style-type: none"> ○ Windows Server 2003 SP2 or later ○ Windows XP SP2 or later ○ Windows Vista Business and Enterprise ○ Windows 7 ○ Windows Server 2008 ○ Windows Server 2008 R2 Plus: <ul style="list-style-type: none"> ○ .NET Framework 2.0 or later ○ MDAC 2.8 or later
64-bit Operating System	Either: <ul style="list-style-type: none"> ○ Windows Server 2003 SP1 (x64) ○ Windows XP Professional (x64) ○ Windows 7 ○ Windows Server 2008 ○ Windows Server 2008 R2 Plus: <ul style="list-style-type: none"> ○ .NET Framework 2.0 or later ○ MDAC 2.8 for later
SQL Server versions	<ul style="list-style-type: none"> ○ SQL Server 2000 SP4 or later ○ SQL Server 2005 SP1 or later ○ SQL Server 2008 ○ SQL Server 2008 R2
Browser	Internet Explorer 7.0 or later (for the online Help)

Backup file requirements

Backup File Attributes	Requirements
SQLsafe versions	SQLsafe 5.0 or later
SQL Server versions	<ul style="list-style-type: none"> ○ SQL Server 2000 SP4

Backup File Attributes	Requirements
	<ul style="list-style-type: none"> ○ or later ○ SQL Server 2005 SP1 or later ○ SQL Server 2008 ○ SQL Server 2008 R2
Backup Types	<ul style="list-style-type: none"> ○ Full ○ Differential ○ Transaction log
File Types	<ul style="list-style-type: none"> ○ Single file ○ Multiple files ○ Striped files

Permission requirements

The SQLvdb Service account requires the following permissions. By default, setup program will assign the Local System account to the SQLvdb Service. During install, you can input credentials for a Windows user account or SQL Server login.

Permissions	Actions Taken
Member of the Built-in Administrators Windows group on the SQL Server computer where the SQLvdb Engine is installed	Create or edit virtual database
Read and write access rights on the folder where backup files are stored	Read backup files and write temporary files

Disk space recommendations

Use the following table to help you set aside the appropriate amount of disk space for your virtual databases. Virtual databases typically require a fraction of the disk space consumed by a fully restored backup. Before creating virtual databases, ensure you allocate the recommended additional disk space for each virtual database you plan to create.

Size of Backup	Additional Disk Space
1 TB	105 MB
500 GB	51 MB
100 GB	10 MB
1 GB	105 kB
500 MB	51 kB

For more information about virtual data files, see [how SQL virtual database works](#).

Port Requirements

By default, the SQLvdb Service uses port 5303. Contact your network administrator to verify whether this port is open and available. To comply with existing port assignments or firewall settings, you can [change which port SQLvdb Service uses](#).

Use the following instructions to deploy SQL virtual database in a clustered SQL Server environment. When deployment is complete, you can attach virtual databases to a virtual SQL Server instance located on an active Windows 2003 cluster node. To ensure virtual databases can be used after a failover occurs, deploy SQL virtual database on each node in the Windows cluster.

TIP Ensure the backup files you want to virtualize are stored on either a network share or the drive assigned to the SQL Server resource group.

Install and configure SQL virtual database

1. On each node (computer) in the target cluster, verify that a resource group exists for your target SQL Server cluster. *If the target cluster does not have a SQL Server resource group*, create one.
2. On each node, [install SQL virtual database](#).
3. Log onto the currently active cluster node (primary node) using an administrator account.
4. Identify which drive is being used by the target instance on this node. The target instance is the SQL Server instance to which you intend to attach virtual databases.
 - a. Start the Microsoft Cluster Administrator tool.
 - b. In the left navigation pane, expand the **Groups** node.
 - c. Select the cluster resource group to which the target instance belongs, and then note the drive letter.
5. On each node in the target cluster, configure SQL virtual database to use the SQL Server drive.
 - a. Start SQL virtual database.
 - b. [Change the virtual data files directory path](#) to use a folder on the SQL Server drive.
 - c. [Attach the SQLvdb Driver](#) to the SQL Server drive.
6. On each node in the target cluster, change the SQLvdb Service properties to require a manual start.
 - a. Start the Windows Services tool.
 - b. Stop the SQLvdb Service.
 - c. On the General tab of the Service Properties window, set the **Startup type** to `Manual`.

Configure the Windows cluster to recognize SQL virtual database

1. Log onto the currently active cluster node (primary node) using an administrator account, and then start the Microsoft Cluster Administrator tool.
2. Create an IP Address resource for the SQLvdb Service. *If a Network Name resource already exists*, skip this step.
 - a. Select the target SQL Server cluster resource group.
 - b. Create a Network Name resource that will map to the TCP/IP address.
 - c. Add the TCP/IP address as a dependency.
3. Create a Generic Service resource for the SQLvdb Service (`SQLvdbService.exe`).
 - a. Select the target SQL Server cluster resource group.
 - b. Add a Generic Service resource and specify the following settings:
 - On the New Resource window, verify that the resource name is `SQLvdb Service`.
 - On the Dependencies window, add the SQL Server service and the name of the disk used by the SQL Server resource cluster group.
 - On the Registry Replication window, add the following root registry key:`SOFTWARE\Idera\SQLvdb\Service`
4. Bring the new Generic Service online.

Use the following instructions to deploy SQL virtual database in a clustered SQL Server environment. When deployment is complete, you can attach virtual databases to a virtual SQL Server instance located on an active Windows 2008 cluster node. To ensure virtual databases can be used after a failover occurs, deploy SQL virtual database on each node in the Windows cluster.

TIP Ensure the backup files you want to virtualize are stored on either a network share or the drive assigned to the SQL Server resource group.

Install and configure SQL virtual database

1. On each node (computer) in the target cluster, verify that a resource group exists for your target SQL Server cluster. *If the target cluster does not have a SQL Server resource group*, create one.
2. On each node, [install SQL virtual database](#).
3. Log onto the currently active cluster node (primary node) using an administrator account.
4. Identify which drive is being used by the target instance on this node. The target instance is the SQL Server instance to which you intend to attach virtual databases.
 - a. Start the Microsoft Failover Cluster Administrator tool.
 - b. In the left navigation pane, expand the appropriate domain node, and then expand **Services and Applications**.
 - c. Select the cluster resource group to which the target instance belongs, expand **Disk Drives** in the right pane, and then note the volume letter.
5. On each node in the target cluster, configure SQL virtual database to use the SQL Server drive.
 - a. Start SQL virtual database.
 - b. [Change the virtual data files directory path](#) to use a folder on the SQL Server drive.
 - c. [Attach the SQLvdb Driver](#) to the SQL Server drive.
6. On each node in the target cluster, change the SQLvdb Service properties to require a manual start.
 - a. Start the Windows Services tool.
 - b. Stop the SQLvdb Service.
 - c. On the General tab of the Service Properties window, set the **Startup type** to `Manual`.

Configure the Windows cluster to recognize SQL virtual database

1. Log onto the currently active cluster node (primary node) using an administrator account, and then start the Microsoft Failover Cluster Administrator tool.
2. Create an IP Address resource for the SQLvdb Service. *If a Network Name resource already exists*, skip this step.
 - a. Select the target SQL Server cluster resource group.
 - b. Create a Network Name resource that will map to the TCP/IP address.
 - c. Add the TCP/IP address as a dependency.

3. Create a Generic Service resource for the SQLvdb Service (SQLvdbService.exe).
 - a. Select the target SQL Server cluster resource group.
 - b. Add a Generic Service resource using the default settings.
 - c. On the Service Properties window, specify the following settings:
 - o On the New Resource window, verify that the resource name is SQLvdb Service.
 - o On the Dependencies tab, add the SQL Server service and the name of the disk used by the SQL Server resource cluster group.
 - o On the Registry Replication window, add the following root registry key:SOFTWARE\Idera\SQLvdb\Service
4. Bring the new Generic Service online.

After initially installing and setting up SQLvdb, there are several tasks you might want to do in order to further customize and streamline your deployment.

- [Change the port that SQLvdb Service uses](#)
- [Change where the SQLvdb Engine stores cached files during the virtual database creation](#)
- [Collect SQLvdb logs for troubleshooting](#)
- [Groom unused virtual data files](#)
- [Increase performance by preparing backup files for future data retrieval](#)
- [Manage your product license](#)
- [Select a different drive on which the SQLvdb Driver should run](#)

Use the Advanced tab of the Configure window to attach the SQLvdb Driver to a different drive (volume). By default, the SQLvdb Driver is attached to a fixed drive on the local computer, such as the C : \ volume. Consider re-attaching the SQLvdb Driver to a volume that is used infrequently by other applications.

For more information about other configuration tasks, see [Configure Your Deployment](#).

Use the Advanced tab of the Configure window to choose a different port. The SQLvdb Service account uses the specified port to connect to remote SQL Server instances and network shares.

By default, the SQLvdb Service uses port 5303. Due to network firewall settings and application security policies, this port may not be unavailable or you may experience a connection issue. ***If you encounter connection issues***, contact your network administrator to verify whether the specified port is open.

For more information about other configuration tasks, see [Configure Your Deployment](#).

Use the System Diagnostics window to collect verbose logs that can be used for troubleshooting by Idera Support. By default, high-level logging is always enabled for SQL virtual database activities and events.

TIP *If you are encountering connection issues*, contact your network administrator to verify whether the [port configured for the SQLvdb Service](#) is open and available.

How do I access system diagnostics?

1. On the SQL virtual database Welcome window, click **Help & More**.
2. Click **System Diagnostics**.

How do I collect the logs?

1. Click **Collect**. SQLvdb creates a zipped folder of the specified log files and places this folder on the Windows desktop. By default, the zip folder is named `IderaSQLvdbCollectionLogInstanceNameTimeStamp.zip`.
2. Email the zipped log folder to [Idera Support](#).

How do I collect logs for issues encountered when installing SQL virtual database?

1. Start the Windows Command Prompt.
2. Navigate to the installer directory (by default, `C:\Documents and Settings\User Name\My Documents`).
3. Type one of the following commands:

Server Type	Installation Command
x86 (32-bit)	<code>msiexec /i SQLvdb.msi /!*v C:\MyFolder\SQLvdb_install.log</code>
x64 (64-bit)	<code>msiexec /i SQLvdb-X64.msi /!*v C:\MyFolder\SQLvdb_install.log</code>

4. Step through the setup program until you recreate the issue. SQL virtual database generates the resultant log file in the location you specified.
5. Email the log file to [Idera Support](#).

Use the Basic tab of the Configure window to groom (delete) any unused [virtual data files](#). The **Clean Up Files** chart shows how much disk space is currently being used by the virtual data files.

To groom your data files, click **Clean Up**. SQL virtual database automatically detects and deletes any virtual data files that are not being used or are not needed to support your existing virtual databases.

TIP This feature is also available through the `-cleanup` option in the command line interface (CLI). To get more information, enter `sqlvdbcli` in the Windows Command Prompt.

For more information about other configuration tasks, see [Configure Your Deployment](#).

Use the **Manage SQLvdb License** option to view details about your trial or production license and perform actions such as activating a license.

From the licensing console, you can:

- Verify current status of your license
- Extend your trial period
- Register and activate your license
- Deactivate your license

For more information about other configuration tasks, see [Configure Your Deployment](#). For more information about the licensing console, see the [licensing online Help](#).

How do I access the licensing console?

1. On the main SQLvdb Console window, click **Help & More**.
2. On the Help window, click **Manage SQLvdb License**. This option opens the licensing console.

What is the trial license?

By default, SQL virtual database installs with a limited-time trial license key that is valid for the computer on which you have installed the tool.

When you decide to move from a trial implementation of SQL virtual database to your production environment, contact and [obtain a license key](#) from Idera.

How does SQLvdb licensing work?

SQL virtual database checks for a valid license each time you start the tool. A valid license is a trial or production key that has been registered for the computer on which you have installed the SQL virtual database tool.

If your license expires, such as when the trial period ends, the Console functionality will be limited to removing previously created virtual databases and managing your product license. You can continue to access existing virtual databases using tools such as SQL Server Management Studio.

If your license is deactivated, you will no longer be able to use the tool. You can re-install the tool and reuse a previously deactivated production key.

Use the Map Generation utility to prepare your backups for future data retrieval.

You can use this utility to automatically generate virtual data files for specific backups on a routine basis. For example, you can schedule a SQL Server job that searches for new backup files and then generates the corresponding virtual data.

By building the virtual data before attempting to attach the backup file, you can decrease the time required to create the virtual database and get faster access to your data and objects. The larger the database is, the more time you can save by preparing your backups.

Consider running this utility on backups for mission-critical and sensitive databases, as well as databases that often require object recovery and data retrieval.

How do I deploy the utility?

You can deploy this utility to any computer. The utility does not require access to the SQLvdb Service or any other SQLvdb components.

The utility setup program is automatically installed with SQL virtual database (by default, `C:\Program Files\Idera\SQL virtual database\Install`). You can share this folder across your network for remote deployments or download and run the utility setup program directly on the target computer.

To deploy this utility locally:

1. Verify that the target SQL Server computer meets the [SQL virtual database product requirements](#).
2. Log onto the target SQL Server computer using an account that has local administrator privileges.
3. Copy the utility setup program (`SQLvdbMapGen.msi`) to a file system folder on the target SQL Server computer.
4. Run the utility setup program.
5. Verify that the utility has been correctly installed on this computer. By default, the utility is installed in the SQL virtual database Program Files directory.
6. To start using the utility, open the Windows Command Prompt, and then enter `sqlvdbmapgen`.

When should I use this utility?

You should use the utility whenever you need to decrease the time required to create a virtual database. Consider using this utility for the following types of backups:

- Large SQL Server backups
- SQLsafe 5.0 backups
- SQLsafe 6.0 backups that were not run with the **Generate metadata** option enabled

How do I run the utility in a job?

1. Install the utility on the SQL Server computer that hosts the databases you routinely back up.
2. Use Microsoft Management Studio to connect to the associated SQL Server instance.
3. Create a new job for this instance.
4. At the beginning of the job, insert a new step.

5. Input the following code for this step:

```
sqlvdbmapgen -BackupFile MyFolder\MyBackupFiles\Name.bak -WindowsUsername tar-  
getdomain\account -WindowsPassword password
```

6. Schedule the job to run on a routine basis, such as once a week during off-hours.

7. Save the job.

8. Test your update by manually running the job.

Can I run this utility manually?

Yes. To run the utility manually, open the Windows Command Prompt, and then type the appropriate command when the prompt displays.

What options are available?

You can specify any of the following options:

- Individual backup files
- Specific data sets contained in the backup files
- Connection credentials
- Password used to decrypt an encrypted SQLsafe backup file

To see a detailed description of each option, enter `sqlvdbmapgen` at the command prompt.

Use the Basic tab on the Configure window to specify which folder SQLvdb uses to store data files when creating a virtual database. For more information about virtual data files, review [how SQLvdb works](#). For more information about other configuration tasks, see [Configure Your Deployment](#).

You can type the directory path for the folder you want to use or click the browse button to navigate to the folder. Once you have specified the appropriate folder, click **OK**.

TIP Ensure the SQLvdb Service account has read and write access rights on the specified folder. For more information, see the [product requirements](#).

Idera provides several tips and tricks so you can get the most value from SQL virtual database. You can watch a [video demonstration](#) or read [customer testimonials](#) on the Idera Web site (<http://www.idera.com/Products/SQL-toolbox/SQL-virtual-database/>).

The following tasks are a small sample of the many ways you can use your virtual database:

- Create DDL scripts for entire databases, tables, or rows
- Identify rows that were added, modified, or deleted since your last backup
- Recover stored procedures
- Recover tables using bulk copy
- Report on data changes using T-SQL queries
- Run historical reports using Microsoft Reporting Services
- View user, role, or schema changes

All of these tasks can now be done without restoring the database!

Create virtual databases to access, compare, recover, and report on data from previous backups of databases in your environment.

For example, you can create multiple virtual databases, at different points in time, from backups of the same live database. This strategy allows you to compare column values and schema objects from one version of the database to another version several days later without restoring the database.

You can create a virtual database by either:

- [Attaching a single full backup file](#)
- [Attaching multiple backup files](#), including transaction log backups

Virtual databases can be created through the SQLvdb Console or CLI.

What is a virtual database?

A virtual database (vdb) is a patent-pending software implementation of a database that allows for data storage and retrieval like a physical database.

Because a virtual database acts like a physical database, you can access any data or object in the virtual database using T-SQL scripts, third-party applications, reporting tools, and database management tools such as Microsoft SQL Server Management Studio. Database management tools interface with and display the virtual database as a physical database.

Each SQL Server instance can host multiple virtual databases.

What can I do with a virtual database?

Any operation that you can perform on a physical database can be performed on a virtual database. Likewise, applications that rely on getting information from this database can continue using the virtual version.

You can use the built-in query tool to execute commands on objects in this virtual database. You can also access this virtual database using Microsoft SQL Server tools, such as Management Studio, and other third-party applications. For more information, see our [tip and tricks](#).

TIP You can modify the data and objects in the virtual database. However, because the virtual database is based on archived data, your changes will not persist when you recreate the same virtual database later. To preserve your changes, back up the modified virtual database and then create a new virtual database using those backup files.

How do I access the CLI?

To access the SQLvdb CLI, start the Windows Command Prompt, and then enter `sqlvdbcli` to see which commands and options you can use. For more information about the SQL virtual database components, see the [product components](#).

Use the Attach Full Backup window to easily create a virtual database from a single full backup file. For more information about how you can use the virtual database, see our [tips and tricks](#).

TIP When you select a backup file, SQL virtual database opens the file in shared read mode. When Microsoft SQL Server or Idera SQLsafe attempts to append (write data to) or delete this file, the backup operation will fail because the file is locked for read. ***If your backup policies or jobs are configured to append new data sets to this file***, change your backup settings so that a new and unique file is created for each backup.

Which backup files can I use?

You can select any full backup files created by Microsoft SQL Server or Idera SQLsafe. For more information, check the [supported backup file formats](#).

Fields

Backup File

Type or browse to the directory path and filename of the backup file you want to use. You can specify a single file for any full backup performed on the target database.

Backup files can be located on the local computer or a network share. To specify a network path, use UNC syntax (\\MyServer\MyFolder\MyBackup.bak).

Host Instance

Type the name of the SQL Server instance that should host the virtual database. You can browse for instances running on the local computer.

Virtual Database Name

Type a unique name you want to use for your new virtual database. This name will be displayed in the Console and in other database management tools, such as Microsoft SQL Server Management Studio.

The Attach Multiple Backups wizard lets you build a live virtual database based on data from specific backup files. With this wizard, you will:

- [Select the backup files you want to include](#)
- [Select the date and time up to which you want data](#)
- [Specify how you want the virtual database to display in the Console and other tools](#)

For more information about how you can use the virtual database, see our [tips and tricks](#).

Use the Backup Files tab to select which backup files you want to use to create your virtual database.

TIP When you select a backup file, SQL virtual database opens the file in shared read mode. When Microsoft SQL Server or Idera SQLsafe attempts to append (write data to) or delete this file, the backup operation will fail because the file is locked for read. *If your backup policies or jobs are configured to append new data sets to this file*, change your backup settings so that a new and unique file is created for each backup.

Which backup files can I use?

You can select any full, differential, and transaction log backup files created by Microsoft SQL Server or Idera SQLsafe. For more information, check the [supported backup file formats](#).

Why are one or more of the selected backup files displayed in orange text?

After you add your backup files, SQL virtual database checks the LSN (log sequence number) range of each file to ensure it is in sequence. When the LSN range is out of sequence, the affected files do not contain all of the logged transactions required to create an accurate virtual database.

If an LSN range is out of sequence, such as a gap between one file and the next, the affected files are displayed in orange text.

To resolve this issue, add more backup files to fill the LSN gap. Also consider verifying the backup files to ensure a complete backup had been performed. For more information, see Microsoft SQL Server Books Online.

Actions and Tasks

Add a backup file to the list

Click **Add** to select a backup file to use when creating this virtual database. To add a file, first type the path and filename in the provided field, or browse for the files. You can also use wildcards to find folders or files that match a naming convention.

If the selected backup file contains data for more than one database, you will be prompted to choose which data set you want.

If the selected backup file contains encrypted data, you will be prompted to provide the password required to decrypt the file.

If the backup was written (or striped) to multiple files, you will need to add each file in the stripe.

Browse for backup files

Click the browse button to find the files you want to use.

Configure encryption settings

Select the listed backup files that are encrypted and then click **Encryption Settings** to specify the password required to decrypt the data.

Remove a backup file from the list

Click **Remove** to delete a previously selected backup file from the list of files to use when creating this virtual database.

Columns and Fields

Created

Provides the date and time when the backup files were created (that is, when the corresponding backup operations were performed).

Database

Provides the name of the database whose data was archived in this backup file.

Filename

Provides the name of the backup files you selected.

First LSN

Provides the LSN at which each backup file starts. For more information about LSN ranges, see [Microsoft SQL Server Books Online](#).

Instance

Provides the name of the SQL Server instance hosting the archived database.

Last LSN

Provides the LSN at which each backup file ends. For more information about LSN ranges, see [Microsoft SQL Server Books Online](#).

Type

Indicates which type of backup was performed when the selected files were created.

Use the Point in Time tab to choose the exact date and time at which you want to create the database.

You can pick any date and time included in the data sets of the selected backup files. *If you do not see the point in time you want*, you may need to [choose different backup files](#). For example, to pick an hourly interval, select transaction log backup files.

Actions and Tasks

Pick a date and time

You can use either the slider or the calendar to pick the exact date and time you want to recover. After you select a date and time, SQL virtual database lists which data sets will be used to create the virtual database up to the selected point in time.

Pick an LSN

SQL Server 2005 and later allows you to use a LSN (log sequence number) to identify the point at which you want to recover data. To recover up to a specific LSN range, select **LSN** from the drop-down menu, and then choose the appropriate backup file.

Verify data sets

Use the grid to verify which data sets will be used to create the virtual database up to the selected point in time. Depending on the point in time you selected, SQL virtual database may not need to use all the data sets available in the backup files.

What is a point in time?

A point in time provides a historical perspective of a physical database at the specified date and time. You can create multiple virtual databases, representing different points in time, from the same set of backup files. For example, you can create three virtual databases that represent the physical database on Monday, Tuesday, and Wednesday. You can then access data and objects from the database as it existed on Monday, Tuesday, or Wednesday.

A point in time can help you:

- Analyze different versions of the same database
- Copy different versions of the same object
- Compare changes in data and schema objects from one time period to another
- Report on data that existed at a given date and time

How do I use the slider?

Use the slider to pick a point in time during any hour of the most recent day in the selected backup files. As you hover over the slider, the available time intervals display. To pick a time, click the corresponding interval.

How do I use the Calendar?

Use the calendar interface to choose a date and time in any available month by selecting the appropriate area on the displayed day. As you hover over a day, the available hour intervals display. To pick a time, click the corresponding interval.

The calendar interface indicates whether data is available for a given date and time, based on the backup files you selected. The calendar shows when different types of backups were performed to help you choose the correct files so that you have the data you need.

Use the Create As tab to specify how SQLvdb should create your new virtual database.

TIP The specified name and description will display in the Console and in other database management tools, such as Microsoft SQL Server Management Studio.

Actions and Tasks

Specify the host instance

Type the name of the SQL Server instance that should host the virtual database.

Specify the virtual database name

Type a unique name for your new virtual database.

Specify a description

Type a description for this virtual database. Use a description that includes details about this data set to help you troubleshoot issues, administer your databases, or recover specific objects.

Review the provided summary, and then click **Finish**. When you finish this wizard, SQL virtual recovery will create a virtual database according to your specifications.

After you complete this wizard, the Console will list your new virtual database in the Welcome window. You can use the built-in query tool to execute commands on objects in this virtual database. You can also access this virtual database using Microsoft SQL Server tools, such as Management Studio, and other third-party applications.

If you want to change a setting now, click **Back** to return to the appropriate window. You can also change these settings later using the **Edit** button on the SQLvdb Console Welcome window.

Use the Edit Credentials window to specify different connection credentials. SQL virtual database will use these credentials to connect to the target SQL Server instance, read the specified backup files, and create the virtual database.

By default, the SQLvdb Service credentials are used. However, the service account credentials may not have the appropriate privileges on the target SQL Server instance or the directory where the backup files are stored. This issue is more likely to occur when backups have been striped across multiple files located in different network shares.

Ensure you specify a Windows account that has the appropriate permissions, and then click **OK**. For more information, check the [permissions requirements](#).

Use the Missing Striped Backups window to resolve issues caused by missing data files. This window displays when a backup file you selected is one of several files created by a striped backup and you did not select all of the files required to create the virtual database.

To specify more backup files, type the path and filename of each file (or use the browse button to navigate to the target file), and then click **Add**.

For more information about how striped backups work, see [Microsoft SQL Server Books Online](#).

You can access your virtual database using any third-party tool, such as:

- SQL admin toolset Multi Query tool
- SQL comparison toolset
- SQL Server Management Studio

For more usage ideas, see our [tips and tricks](#).

How to use the Multi Query tool in the Idera SQL admin toolset

1. Click **Multi Query** from the SQL admin toolset Launchpad.
2. Add a new query or open an existing query. For more information, see the SQL admin toolset online Help.
3. Execute your query against the selected targets. In this case, the selected targets should include the virtual databases you want to query.

How to use SQL Server Management Studio

1. Connect to the SQL Server instance that is hosting your virtual database.
2. In the **Object Explorer**, expand the **Databases** node.
3. Select your virtual database from the list. Each virtual database will display as a physical database.
4. Perform an action or run a query. For usage ideas, see [Tips and Tricks](#).

You can test your new virtual database using Microsoft SQL Server Management Studio (Management Studio) or the built-in Query Database tool. With either application, you can test your virtual database by performing simple T-SQL queries, such as testing whether you can select data from a specific table.

When you click **Run Queries** on the Welcome window of the SQLvdb Console, SQL virtual database starts the Management Studio tool. *If Management Studio is not installed on the local computer*, SQL virtual database starts the built-in Query Database tool instead.

How do I use the Query Database tool?

You can immediately begin using the Query Database tool by typing or copying a script in to the editor pane, picking the virtual database on which you want to run your script, and then executing the query you entered. The following actions and tasks are available from this tool.

Choose the target database

Use the **Target Database(s)** field to select which virtual databases should be tested with this query.

Run the query

Click the execute button to run your script.

Control how your query is run

Use the **Query** tab on the **Options** window to set run properties, such as the maximum number of returned rows and the execution timeout value.

Change how the editor pane displays your code

Use the **Editor** tab on the **Options** window to set the editor pane properties, such as whether to show line numbers or highlight syntax errors.

The Edit Virtual Database Options wizard lets you change the configuration of a virtual database. With this wizard, you can:

- [Change the data set by specifying different backup files](#)
- [Choose a different date and time](#)
- [Rename the virtual database or edit the description](#)

When should I edit the virtual database settings?

You should edit your virtual database configuration when:

- You need to query or recover object data from a different time period
- You want to change how the virtual database is displayed in the SQLvdb Console and other third-party applications

However, you may want to consider creating a new virtual database to use along side the existing virtual database. This strategy can be useful when:

- You want to query data from the same database at two different time periods
- You need to track changes over time, such as daily values of sensitive data
- You want to copy or move objects, such as a table, from one database to another

For more information about tasks you can perform on a virtual database, see [tips and tricks](#).

Use the Backup Files tab to change the backup files being used for your virtual database. Choosing different backup files allows you to access a different data set, such as a different time period.

TIP When you select a backup file, SQL virtual database opens the file in shared read mode. When Microsoft SQL Server or Idera SQLsafe attempts to append (write) data to this file, the backup operation will fail because the file is locked for read. *If your backup policies or jobs are configured to append new data sets to this file*, change your backup settings so that a new and unique file is created for each backup.

Which backup files can I use?

You can select any full, differential, and transaction log backup files created by Microsoft SQL Server or Idera SQLsafe. For more information, check the [supported backup file formats](#).

Actions and Tasks

Add a backup file to the list

Click **Add** to select a backup file to use when creating this virtual database. To add a file, first type the path and filename in the provided field, or browse for the files. You can also use regular expressions to find folders or files that match a naming convention.

If the selected backup file contains data for more than one database, you will be prompted to choose which data set you want.

If the selected backup file contains encrypted data, you will be prompted to provide the password required to decrypt the file.

If the backup was written (or striped) to multiple files, you will need to add each file in the stripe.

Browse for backup files

Click the browse button to find the files you want to use.

Remove a backup file from the list

Click **Remove** to delete a previously selected backup file from the list of files to use when creating this virtual database.

Specify connection credentials

Click **Credentials** to specify which Windows account credentials should be used to access the selected files. By default, the SQLvdb Service credentials are used.

Ensure you specify a Windows account that has the appropriate permissions. For more information, check the [permissions requirements](#).

Columns

Created

Provides the date and time when the backup files were created (that is, when the corresponding backup operations were performed).

Database

Provides the name of the database whose data was archived in this backup file.

Data Set

Indicates which data set this backup type is. The selected backup file may contain more than one data set, depending on whether this file was appended with data from multiple backup operations.

Filename

Provides the name of the backup files you selected.

First LSN

Provides the LSN at which each backup starts. For more information, see Microsoft SQL Server Books Online.

Instance

Provides the name of the SQL Server instance hosting the archived database.

Last LSN

Provides the LSN at which each backup ends. For more information, see Microsoft SQL Server Books Online.

Stripe Seq No

Indicates which stripe sequence number the selected file has. *If the backup did not use striped files*, the sequence number for all selected files will be 1.

Type

Indicates which type of backup was performed when the selected files were created.

Use the Point in Time tab to change the date and time of the data included in the virtual database.

You can pick any date and time included in the data sets of the selected backup files. *If you do not see the point in time you want*, you may need to [choose different backup files](#).

Actions and Tasks

Pick a date and time

You can use either the slider or the calendar to pick the exact date and time you want to recover. After you select a date and time, SQL virtual database lists which data sets will be used to create the virtual database up to the selected point in time.

Pick an LSN

SQL Server 2005 and later allows you to use a LSN (log sequence number) to identify the point at which you want to recover data. To recover up to a specific LSN range, select **LSN** from the drop-down menu, and then choose the appropriate backup file.

Verify data sets

Use the grid to verify which data sets will be used to create the virtual database up to the selected point in time. Depending on the point in time you selected, SQL virtual database may not need to use all the data sets available in the backup files.

What is a point in time?

A point in time provides a historical perspective of a physical database at the specified date and time. You can create multiple virtual databases, representing different points in time, from the same set of backup files. For example, you can create three virtual databases that represent the physical database on Monday, Tuesday, and Wednesday. You can then access data and objects from the database as it existed on Monday, Tuesday, or Wednesday.

A point in time can help you:

- Analyze different versions of the same database
- Copy different versions of the same object
- Compare changes in data and schema objects from one time period to another
- Report on data that existed at a given date and time

How do I use the slider?

Use the slider to pick a point in time during any hour of the most recent day in the selected backup files. As you hover over the slider, the available time intervals display. To pick a time, click the corresponding interval.

How do I use the calendar?

Use the calendar interface to choose a date and time in any available month by selecting the appropriate area on the displayed day. As you hover over a day, the available hour intervals display. To pick a time, click the corresponding interval.

The calendar interface indicates whether data is available for a given date and time, based on the backup files you selected. The calendar shows when different types of backups were performed to help you choose the correct files so that you have the data you need.

Use the General tab to specify a different description for the virtual database.

TIP Each virtual database requires a unique name. To rename an existing virtual database, recreate the virtual database and assign a different name.

Actions and Tasks

Specify a description

Type a description for this virtual database. Use a description that includes details about this data set to help you troubleshoot issues, administer your databases, or recover specific objects.