

ADMIN

Network & Security

QUMBU backup and maintenance tool for Microsoft SQL Server

According to Plan

QUMBU lets database administrators – even those with less experience – perform straightforward SQL Server backups and maintenance checks. By Sandro Lucifora

Databases are the tech heart of the enterprise. If they stop beating, the entire work process usually grinds to a standstill, which is why it is so important constantly to maintain and secure your databases. I took a look at how this can be done with QUMBU [1] – new software by WSW Software for creating backups and maintaining the SQL Server database management system (**Table 1**) – and discovered that the developers have put a great deal of thought into ease of use and a planned approach. In the lab, I

checked whether the software can measure up to the claim that these processes can be organized in an effortless and secure way. For this example, the tool was installed on Windows Server 2019; it will run on Windows 7 or Windows Server 2012 with a .NET Framework. As the SQL server, the QUMBU developers assume version 2008 R2 or newer. The lab had SQL server 2019 in place. The well-structured instructions will also help administrators get through the installation and familiarization process,

even if you do not work with SQL servers on a daily basis.

The setup wizard gives you the option of installing a single QUMBU client, the QUMBU server, or both components together. Therefore, the server component and the client can be installed separately (e.g., on the server and an administrative desktop computer). For this example, the full install was chosen. The QUMBU server runs in the background as a Windows service; it requires a logged in user account to run, which you specify during the installation.

After the software is installed, the wizard expects you to enter a QUMBU account name, which is the user who has access to all SQL Server instances and can be a user who authenticates locally on Windows; in other words, you do not need a domain controller. This process is simple; you can view all the potential users and select the one you want from a list. After entering the password, you can then complete the installation. Note that you do need admin privileges for the

Table 1: QUMBU

Product	Software for Microsoft SQL Server backup and maintenance
Manufacturer	WSW Software GmbH
Price	12 months: EUR59 per month and per server 36 months: EUR39 per month and per server
Supported operating systems	Windows 7 and later with .NET Framework 3.5; Windows Server 2012, 2016, or 2019
System requirements	<ul style="list-style-type: none"> • Processor: 2GHz or faster • Memory: At least 4GB of RAM • Disk space: At least 1GB of free hard drive space • Graphics resolution: At least 1024x768 • Microsoft SQL Server from version 2008 R2 on Windows • Internet access for licensing the program

installation; otherwise, the required services will not start.

During the installation on Windows Server 2019, I initially had problems because the QUMBU service failed to launch. The manufacturer's support team then analyzed the problem and determined that – contrary to the stated system requirements – .NET Framework 3.5 was required. After installing it, the setup completed successfully.

No Connection to Azure Database

After the launch, an uncluttered window came up; this is the QUMBU cockpit. Next to the toolbar is an area for displaying the servers, the tasks, and the task details and notifications. To begin, add a Microsoft SQL server. To do so, right-click in the *Servers* area and select the *Connect to Server* entry. In the dialog that follows, enter the server name and your login details.

After connecting, you will see that the device has been added to the *Servers* area. Additional servers can also be entered at this point, but they must be accessible on the network. Connections with a virtual private network (VPN) are also possible, as are publicly accessible servers, which is already an advantage over Microsoft SQL Server

Management Studio. QUMBU also lets you include and manage multiple servers. However, QUMBU refused to connect to an SQL database I was running in Azure, saying that the server did not run on a Windows operating system. The requirement is that you run the SQL server on a dedicated Windows server – physical or virtual – for QUMBU to be able to handle it. The license model also reflects this requirement. The manufacturer only requires one license per physical or virtual server. The license is then valid for managing any number of database instances and users. The properties of the server instance and database can be seen in the context menu. For example, you can see whether the server version and the user used for the connection had full authorization. For the database, QUMBU shows the current status as well as the *created on* and *last backup* dates.

Cross-Server Tasks

Once the server is set up, you can turn your attention to the central task area. After selecting a server in the corresponding area, you can proceed to configure it and then select *All Servers* in the top level of the tree structure (Figure 1) to apply the task you defined to all servers. The *Job*

details area displays the history of executed tasks in a table, where you can filter for specific servers or tasks. Without this restriction, QUMBU shows all tasks for all SQL servers. The lower window shows the current system health notifications that relate to individual tasks or individual SQL server instances. I liked the fact that I could also see the urgency level (i.e., whether the message was just a notification or a warning that needed a response).

Tasks Everywhere

No matter what you want to do with QUMBU, the developers packaged all the functions into tasks. Both backing up and restoring a database are tasks, as are the various maintenance functions.

To begin, create a database backup task by selecting the *Backup* icon in the toolbar, which reveals options to *Backup All* databases, *Backup Selected* databases, or *Backup by Pattern* databases that match a search pattern. Clicking on the respective entry takes you to the corresponding task configuration. The operation is always the same: Right-click in the resulting dialog to open the context menu and select *Add*. The configuration dialog prompts you for parameters relating to the task. To back up all the databases,

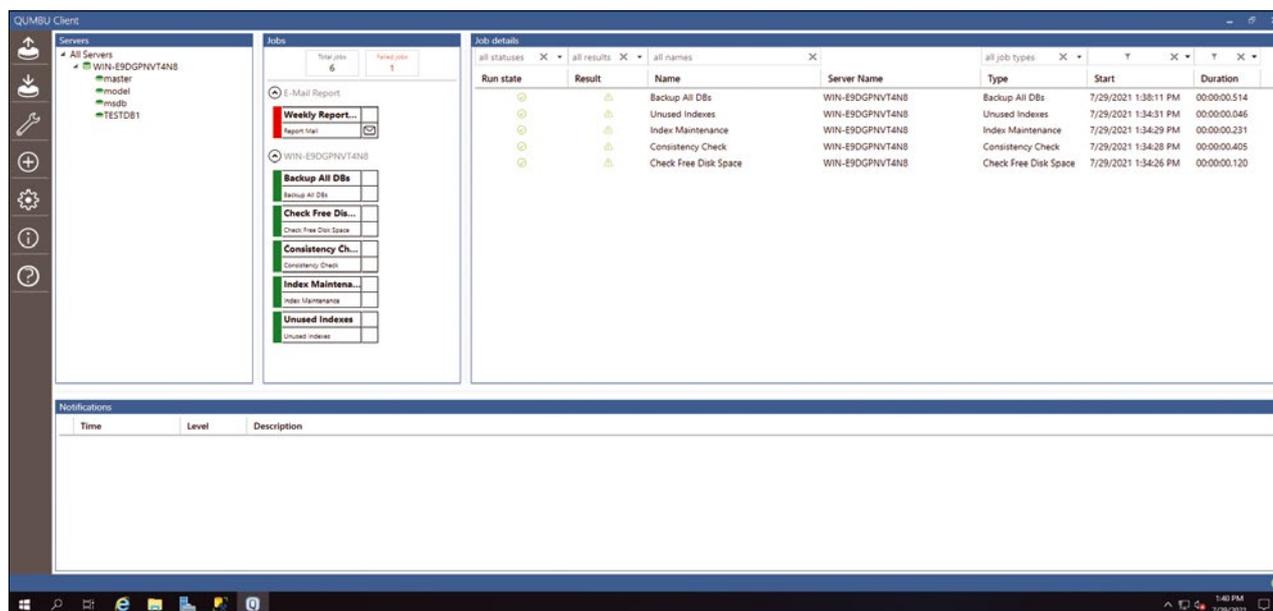


Figure 1: The QUMBU client has a state-of-the-art, clear-cut interface.

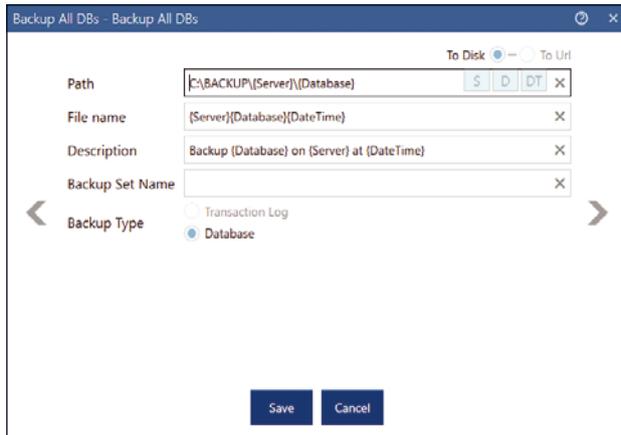


Figure 2: Variables can be used in the specifications for the path and file name of a backup.

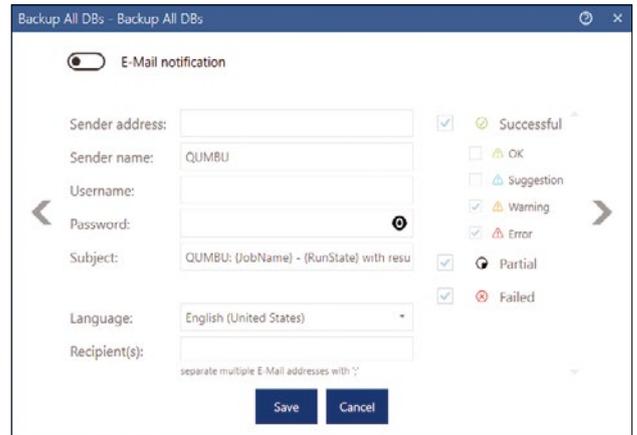


Figure 3: The email notification informs you of the success or failure of a task.

you just need to assign a name to the task and select the SQL server to be backed up.

On the next page of the dialog, you define the path, file name, and description (Figure 2). QUMBU also gives you variables for the server name, database, and backup date. The *Expert* tab has a variety of options regarding the backup itself. For example, you can specify the compression ratio and decide whether you want a differential backup or a full copy. The optional encryption is yet another level of protection for your data. Other options included the number of buffers, the maximum transfer rate, and the block size setting.

Notifications

QUMBU only sends notifications by email. To configure the email settings for backup status notification (Figure 3), you first have to specify an SMTP server, which QUMBU uses to dispatch email. After that, you specify parameters for backup notification, such as deciding on the events about which you want to be notified. What is missing at this point are alternative notification paths, such as a web-hook, which you could use to integrate Slack, for example.

In the last window, you define the schedule for the backup (Figure 4). In addition to choosing daily, weekly, or monthly backups, you have options for specifications such as the

day of the week, time, and the start and end dates. Configuring a backup for individual databases was identical, except for the option to select one or more databases at the beginning instead of just the SQL server.

One very helpful feature is backing up by pattern matching (i.e., performing an action against all databases that match a certain filter). This option can be helpful if, for example, all databases for production operations start with *Prod* and databases for development start with *Dev*. Perhaps you don't want to back up the Dev databases daily, but you do want to for the production databases. In this scenario, you would

filter by database name, starting with *Prod*. This option is especially useful where databases are added or deleted dynamically and a server has large numbers of databases.

You also need a function to restore a backed up database to the same or another Microsoft SQL Server. The *Restore* task in QUMBU does this for you. As options, the developers implemented recovery from the backup history, which is only possible on the same server, and recovery from a file, which lets you recover the database on another SQL server. For both options, the configuration dialog for the task is just as simple as the backup task: Right-click and select

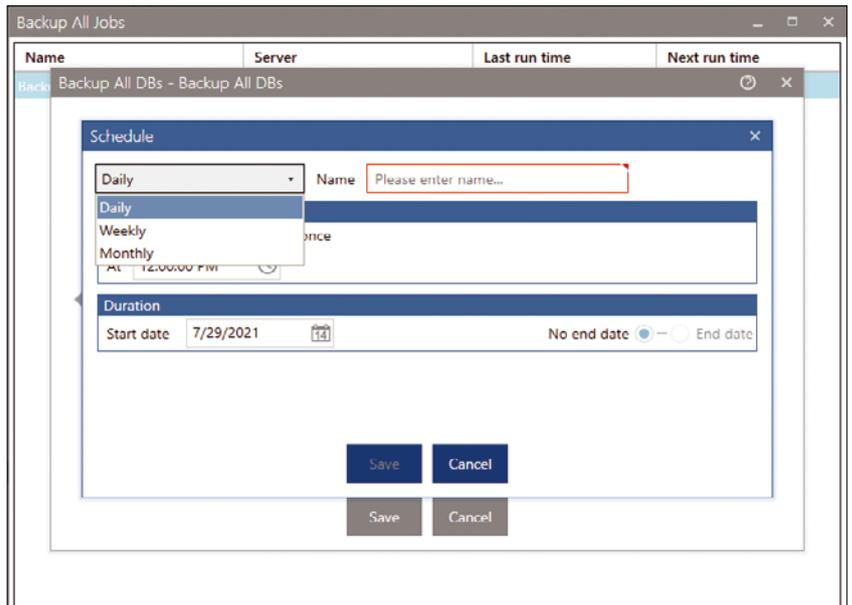


Figure 4: The schedule configuration is identical for each task and covers all reasonable possibilities.

Add to open the dialog; then, select the server and the database if you are restoring from the history. In the next window, the program shows you the history, from which you then select the desired restore start time. After that, check the expert and email settings for further tweaks. The schedule configuration dialog lets you specify when you want the restore to take place. Daily, weekly, and monthly intervals are also possible. When restoring from a file, the task definition differs only in terms of the first dialog, in which you select the backup file and specify the SQL server and target database for the restore. The remainder of the configuration process is exactly the same as restoring from the history. To clone a database from one server to another, the vendor uses sequential backup and restore tasks. One use case for this example is regularly replicating a database to another server.

Simple Recovery

To check the consistency of a backup that you created, you have the option of simulating a restore. In this case, QUMBU reads and processes the data exactly as if the real restore were taking place, the only difference being that the software does not write any data at the end. In this way, you can check whether the backup from another database server is compatible with the target server if the two SQL servers are running different versions. The tasks for the simulated restore are identical to the processes of the actual restore described earlier. In addition to the main tasks of backing up and restoring databases, the developers also offer the ability to perform some maintenance tasks, including consistency checks, index maintenance, finding unused indexes, and checking for free hard disk capacity.

Here, too, I liked the identical process of creating tasks as was used in previously described procedures. I added a new task from the context menu and made the task-specific settings. For the consistency check, I selected the server and the database. In the expert settings, I determined the database console command (DBCC) I wanted to run for the check. In addition to the familiar email notification, I also specified the schedule for the check. For regular index maintenance, I again created a task for which I selected the server and the database. In the expert settings this time, I was able to set the thresholds for the reorganization and rebuild as percentages. The developers also include an option for the minimum number of index pages. According to the manufacturer, however, index maintenance only makes sense after seven days of operation, because no reliable state-

GOT CLUSTER?

Tune in to the HPC Update newsletter for news, views, and real-world technical articles on high-performance computing.

<https://bit.ly/HPC-ADMIN-Update>

ADMIN HPC

HPC Up Close

- OpenACC - Parallelizing Loops
- DES Completes Dark Matter Survey
- Open Compute Project Call for Posters
- Dolphin Announces New Switch for Composable Architectures
- Usenet

Highlights

OpenACC - Parallelizing Loops
OpenACC is a great tool for parallelizing applications for a variety of processors. In this article, I look at one of the most powerful...

DES Completes Dark Matter Survey
Scientists will continue to analyze the results until 2021

Open Compute Project Call for Posters
Submissions for the OCP Future Technologies Symposium are due on January 31

Dolphin Announces New Switch for Composable Architectures
The MX2000 will support software-defined configuration and device sharing at low latency

Usenet
Usenet is a gigantic Internet forum with thousands of subforums. The Usenet system is designed as a federated network, which means you just need...

Most Read

GUI or Text-based Interface?
Sys admins are like smokejumpers who parachute into fires, fighting them until they are out, or at least under control. When you jump into the...

Exploring the /proc Filesystem with Python
The proc filesystem (procfs for short), is a Linux pseudo-filesystem that provides an interface to the operating system's kernel data structures.

Quantum Computing Milestone Achieved
Researchers at QIPNL perform independent operations on two qubits encoded on photons of different frequencies.

ISC 2019 Call for Submissions

The ISC High Performance 2019 conference (June 19-20) is currently open to various opportunities for engineers, researchers, and scientists working in high performance computing.
[Learn more here.](#)

Further Reading

- Shared Storage with NFS and S3FS
- Human Brain Supercomputer Wakes Up
- Resource Management with Slurm
- Intel Unleashes New Processor Line
- Discovering ROCm

7 YEARS OF ADMIN On One DVD

Order Now!
Shop.linuxnewmedia.com

2018 Archives Available Now!

Clear off your bookshelf and get every issue from 2018 at 50% off the digiSub price!

The 2018 ADMIN Network & Security magazine digital archive bundle is available now (Issue #13 through #48). Order now, and get all 2018 issues added to your digital account for access at any time from any device.

[Order the archives today!](#)

ments about outdated or unused indexes can be made before that. However, if necessary, you can adjust this time in the configuration.

To make sure the database server has enough disk space, you need to set up the appropriate check. In the configuration dialog, select the SQL server and chose the drives you want to monitor from the list of available drives, setting the threshold values in gigabytes for warnings and notifications in case of errors individually for each drive.

The subsequent configuration for the email notification and the process of creating the schedule are familiar. For this task, however, you do not select one-off execution per day, weekday, or month but specify a daily check in a given time frame and a certain interval. In the test, this was the weekday test, without weekends, during production hours from 8am to 10pm with an interval of 15 minutes. This was a way of ensuring no downtime because of a lack of hard disk space, especially during peak hours when the database is in use.

User Management

It is worth mentioning that QUMBU has no user management with regard to the operation of the software. The target group of users is limited to the administrator, who is allowed to do everything, or users who just have read-only rights. These settings also showed up in the list of users under the *Settings* function. Here, QUMBU outputs all the users that have been saved for access to the databases. An additional read-only checkbox restricts their authorizations.

To round off the feature set, the manufacturer integrated – again, as a task – a reporting function, which made it possible to receive a weekly report by email (e.g., that clearly shows the results of the defined tasks). In addition to selecting the server and the possible tasks, you can also set the reporting period.

Conclusions

The software did a great job of handling the core tasks of backing up and restoring SQL databases (Table 2). The

additional maintenance tasks even allow less experienced database administrators to perform preemptive checks. Grouping all the functions as a single task and the consistent, easy-to-understand user interface simplify integration into an existing infrastructure.

In contrast to Microsoft SQL Server Management Studio, QUMBU can also be used to manage multiple Microsoft SQL Servers and to configure the tasks identically across servers. In terms of functionality, the developers have taken care in the configuration of the tasks of ensuring that the dialogs only differ where they expect function-specific input. The dialogs are not overloaded, and the configuration options are very well thought out.

For time-controlled cloning of databases, the developers rely on a backup with a subsequent restore. The additional maintenance functions for consistency checks, index maintenance, finding unused indexes, and the hard disk check set QUMBU apart from most other tools in this category. The maintenance functions might only be “little things,” but they do help to optimize a database continuously.

Not to be overlooked were teething problems, especially during the install, but these were solved quickly and competently by the support team. For me, QUMBU is a very useful tool with a well-thought-out and modern user interface, simple operation, and, above all, useful and reliable functions. ■

Info

[1] QUMBU: [https://www.qumbu.de/en/]

Table 2: Review

Rating (out of 10)	Backup databases: 7
	Database recovery: 8
	Maintenance functions: 8
	Configuration options: 7
	Notifications: 6
This product is suitable	<ul style="list-style-type: none"> • Great for backing up and maintaining multiple Microsoft SQL Servers; the tool is also designed for less experienced administrators • With restrictions as a plain vanilla maintenance tool for SQL Server • Not as a backup solution for databases that do not run on a Windows operating system, such as Azure or AWS