Click Studios

Passwordstate Backups

Local Windows Account using Local Folder

Table of Contents

1. Overview

Passwordstate is an application that communicates to, and stores all or your data in a Microsoft SQL Database. In the event of a disaster, you may need to restore your database and Passwordstate installation files, which will require you to have them backed up.

Passwordstate has a built-in automatic backup feature which can be configured to suit your requirements. For example, you may already have another solution for your SQL database backups, so you can set Passwordstate to not backup up your database, but maybe just the install files.

Not only is the SQL database critical to have a backup of, but there are also encryption keys which are located in your web.config file. These too are also critical to have a copy of in the event of a disaster, so setting up the Passwordstate automated Backup feature will ensure you have everything you need to restore your environment.

This document will help you configure Passwordstate to use a Local Windows account for the backups, and will also use a local folder to store the data. The reason you may want to use a local account and local folder is because you are hosting Passwordstate on a non domain joined machine. The idea would be to have Passwordstate back up to a local folder, and then use an external process to automate moving the data to some other location, like a secure network share or cloud-based storage solution.

Please Note: If using cloud-based database services like Azure SQL or Amazon RDS, you cannot perform database backups using our software, as those platforms do not support it.

2. Backup Permissions and Prerequisites

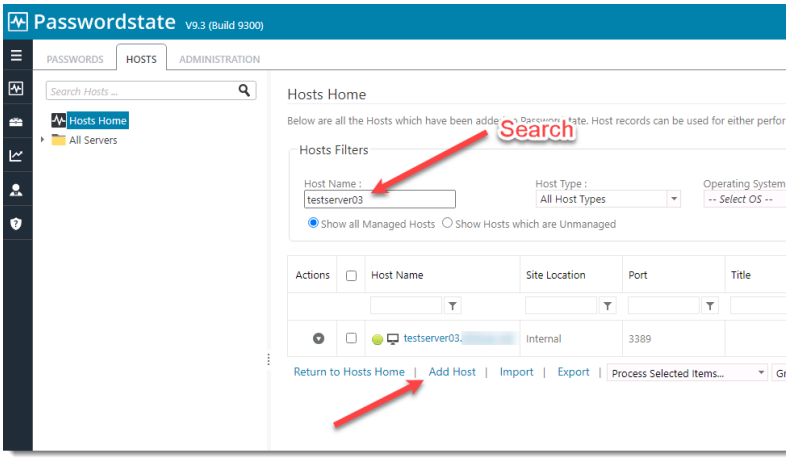
All examples below are using a local account that has previously been created on our Passwordstate webserver called "**Passwordstate**". This account is a Local Administrator.

The Passwordstate webserver, which also hosts the SQL database is called **testserver03.**

The backup folder that we've created on our Passwordstate webserver for this example is **C:\Data\Backups**.

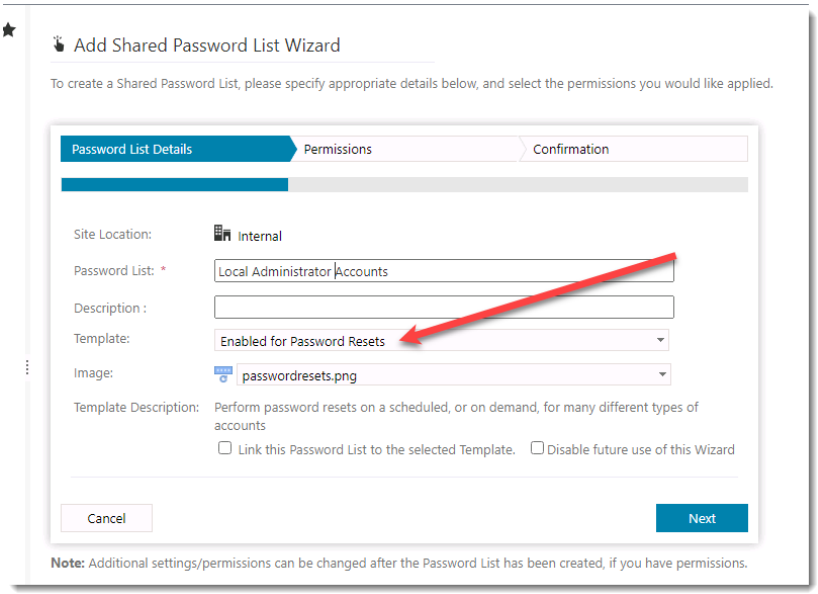
2.1 Add Passwordstate Server into Passwordstate

You will need to add your Passwordstate web server into Passwordstate, specifically under the Hosts tab. If you do not see your server already listed, then use the Add Host button and fill out the appropriate details:



2.2 Create Password List with the Enabled for Resets Option Selected

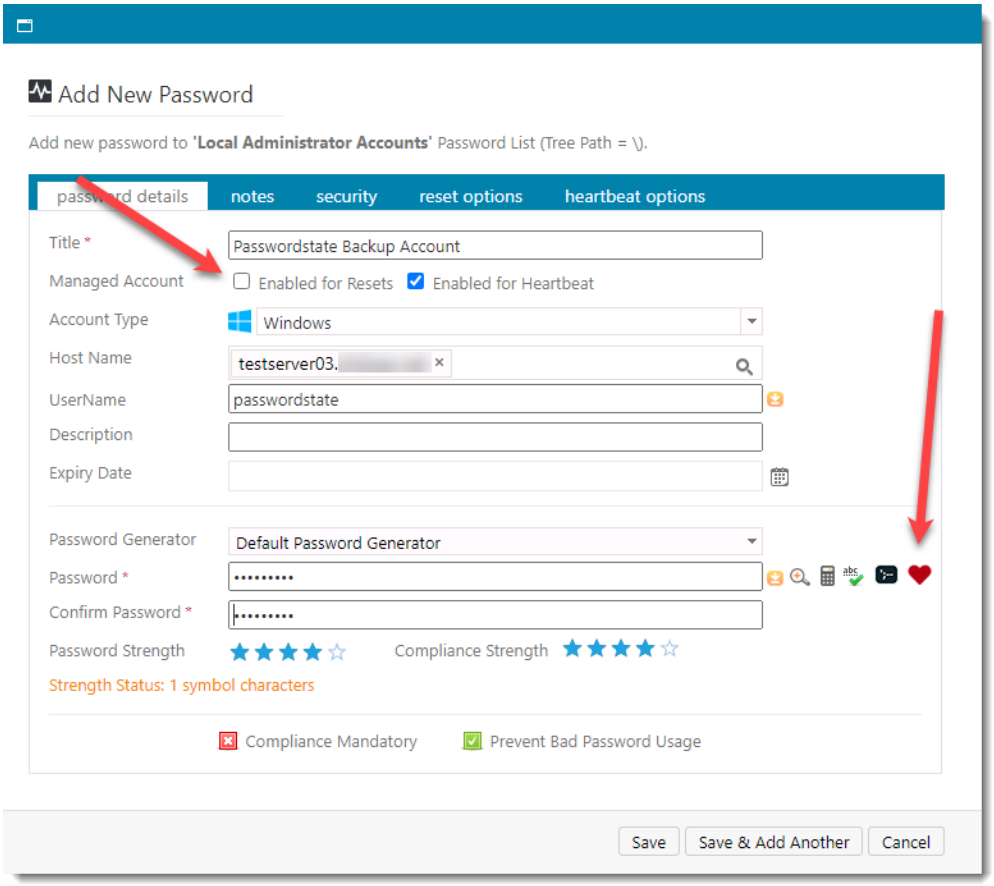
You may already have a Password List that is enabled for resets, but if you don't then you'll need to create one. Creating a List with this option will allow you to add in a Password Record that can automatically reset and validate the Local Windows account you will be using for backups:



2.3 Create Password Record

Once you have a Password List ready, you can now add in a new Password Record. When creating this record, deselect the option "**Enabled for Resets**". By deselecting this option, Passwordstate will not try to automatically reset the password for the account. More information about this in the FAQ at the end of this document.

You should then choose the "**Windows**" account type, set the **Server Name**, **Username** and finally add in the current **Password** for the account. You can test the password is valid by clicking the heartbeat icon:



2.4 PowerShell Requirements on Database Server

As a once off process, the SQL Server PowerShell module must be installed on your Passwordstate database server, which in this guide is the same server where Passwordstate website is installed. This module can be installed by opening an elevated PowerShell ISE session on your server, and execute the following lines of PowerShell code:

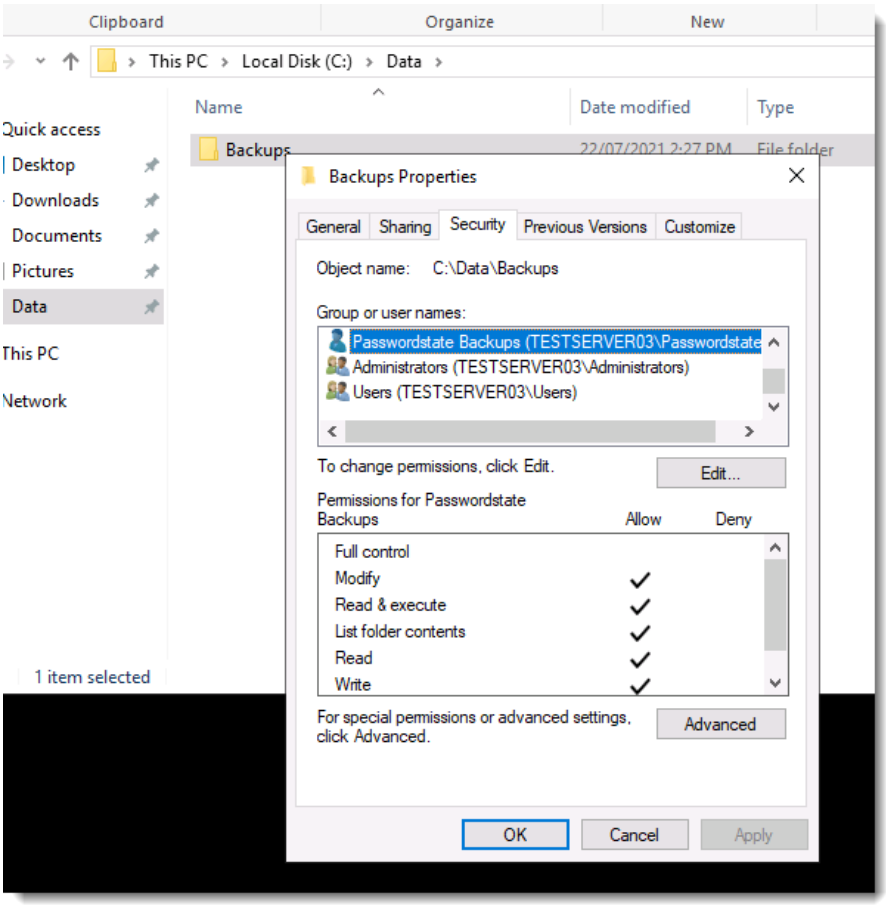
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls -bor [Net.SecurityProtocolType]::Tls11 -bor [Net.SecurityProtocolType]::Tls12

Install-Module sqlserver -Force -AllowClobber

Installing this module will reach out to some online Microsoft repositories to which you should agree to any prompts to ensure a successful install. More information about this can be found here: https://docs.microsoft.com/en-us/sql/powershell/download-sql-server-ps-module?view=sql-server-ver15

2.5 Backup Folder Permissions

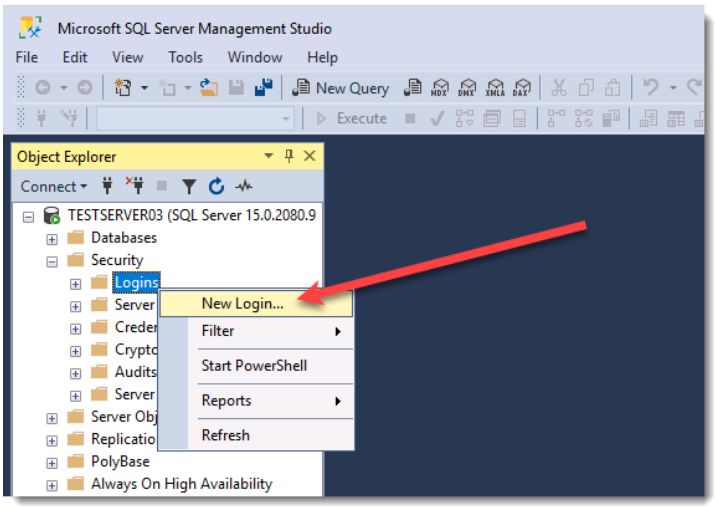
Even though the account you are using is a Local Administrator, you will still need to grant that account **Modify** permissions to the **Backup** folder. This is due to the way we impersonate the account when performing the backup operations:



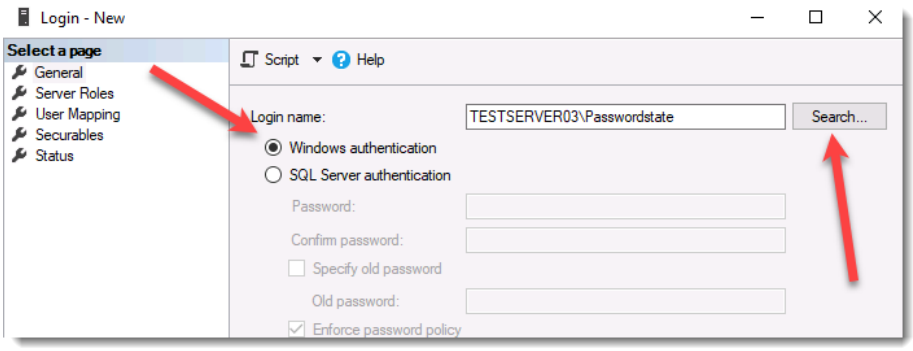
2.6 SQL Database and Server Permissions/Requirements

Using SQL Management Studio tools, connect to **testserver03** with any SQL admin account of your choice, and add in the **Passwordstate** account user under **Security** -> **Logins**. When adding this user, ensure you give it **db_backupoperator** permissions to the Passwordstate database:

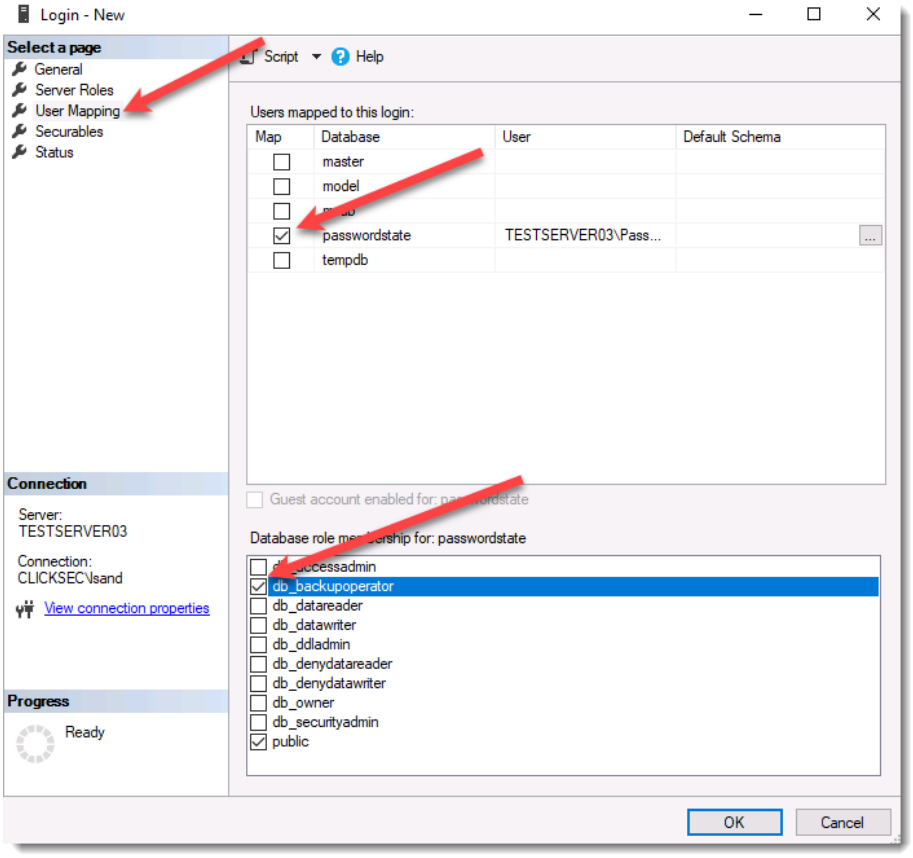
Screenshot #1:



Screenshot #2:

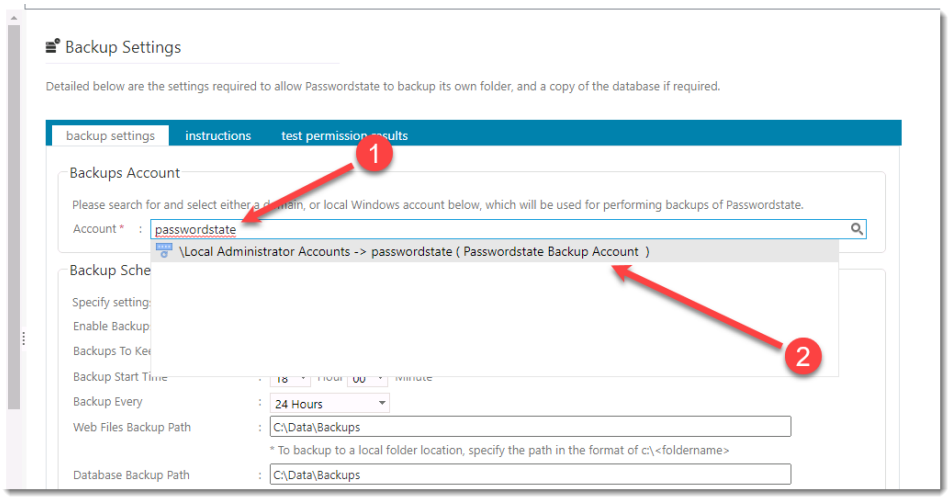


Screenshot #3:

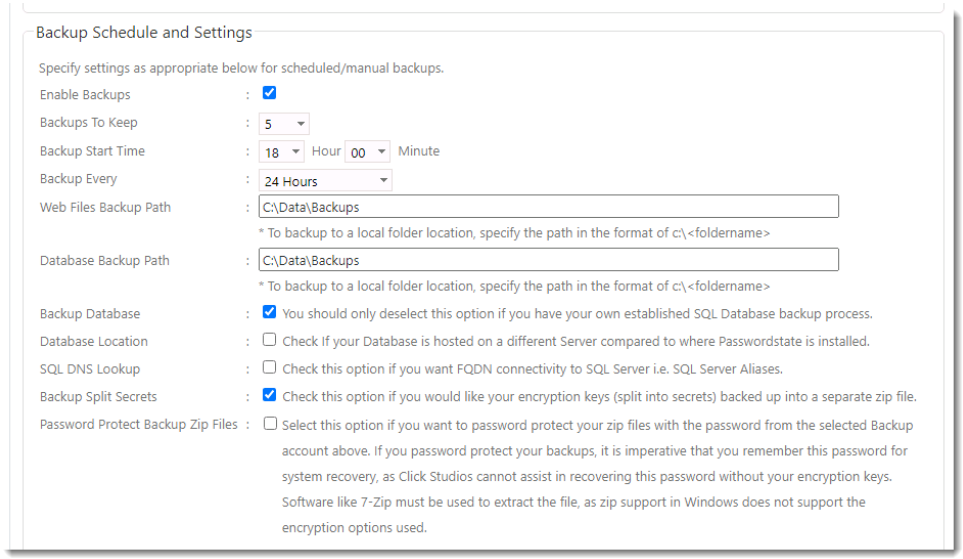


3. Configure Backups Screen

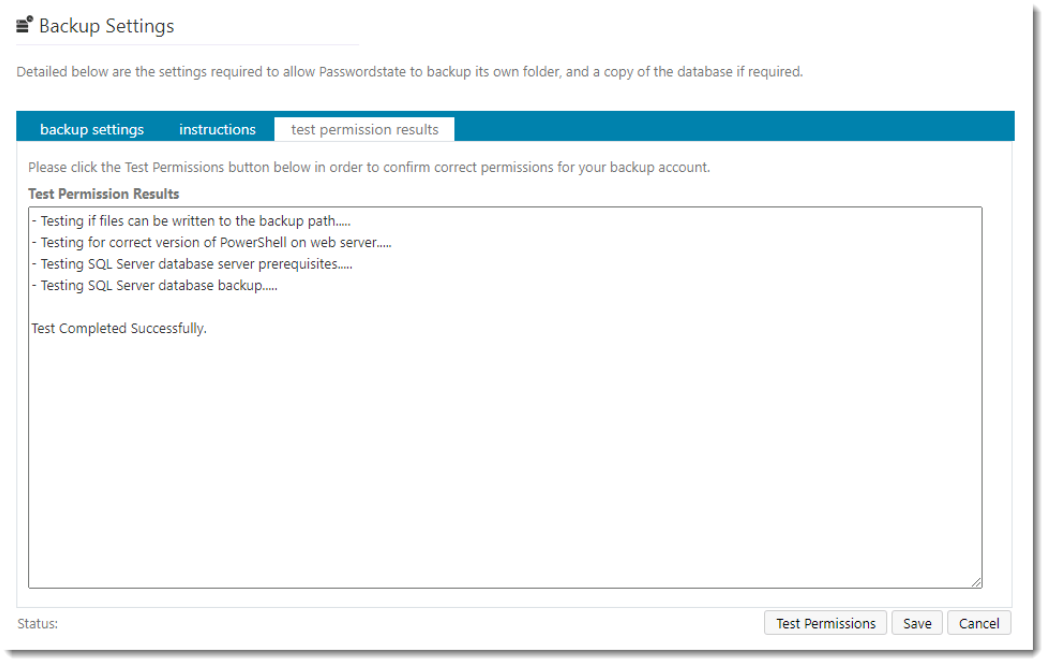
In Passwordstate, go to **Administration** -> **Backups and Upgrades** -> **Settings** and add in the backup account that you previously set up in **Step 2.3** of this document. Search for the account name and select the matching result to set the account:



Next, select the option to enable backups and choose a schedule of your choice. Add in the backup folder path for both web files, and database backup paths, and configure any other settings as appropriate on this screen:



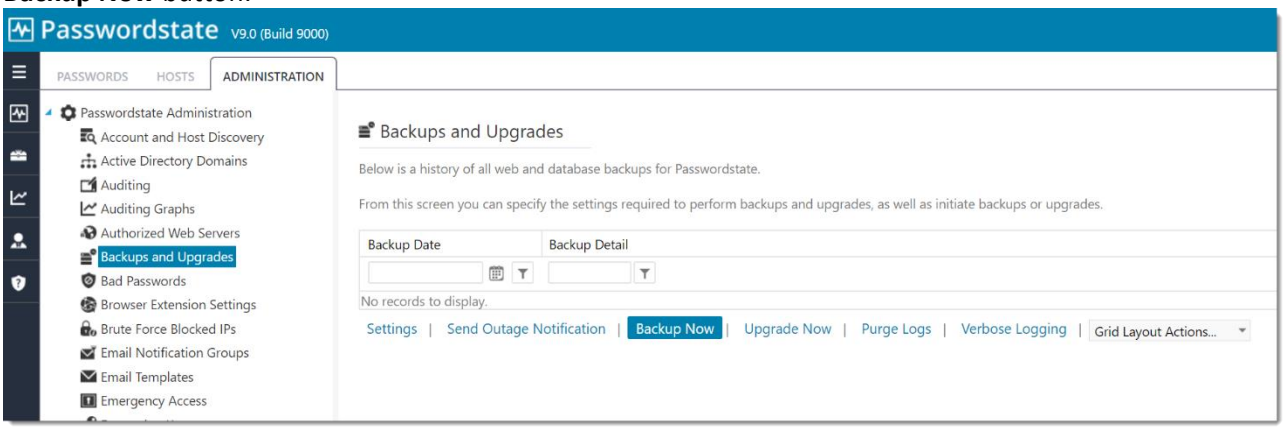
You should now be able to press the **Test Permissions** button on this page and if all configured correctly, you will get a successful result and your configuration of backups is complete.



4. FAQ

4.1 Manual Backups

You can run a manual backup at any time if you have configured your settings correctly by clicking the **Backup Now** button:



4.2 Session Recording Files

If using the Browser Based version of the Remote Session Launcher, Session Recordings will not be included in the standard Passwordstate backup functionality, due to the potential size of the files. If you have left the recording folder in the default path, then you need to organize your own backups of these files

4.3 Password Protecting the Backup Files

One of the options on the backups page is to Password Protect your zip files. If you have selected this option, then it is imperative you know the password. We'd recommend storing this in a safe place outside of Passwordstate. Without knowing this password, you will not be able to access your backup files.

4.4 Automatic Reset of Backup Account

If you select the option '**Enable for Resets**' for your account on your Password Record, and you have the option to password protect your zip files on the backups page, then it is imperative you know the password after it is reset. We'd recommend storing this in a safe place outside of Passwordstate. Without knowing this password, you will not be able to access your backup files.

4.5 Local Administration Requirements

There's no specific need to make your user a Local Administrator on your server, and backups will work without Local Admin rights. The only issue you would see if not making the account a Local Administrator, is Passwordstate will not be able to automatically reset the password for the account.

4.6 Backup Split Secrets

Backup Split Secrets is another option on the backup **Settings** page you can choose to ensure you have a copy of your encryption keys, half of which reside in your web.config file, and the other half are located in your database. These encryption keys are critical when restoring your environment. By enabling this option, these keys will get backed up to the **Web Files Backup Path**.

4.7 Access Denied Errors when Performing Backup Tests

If you have configured everything as per this documentation, but are still getting errors performing the Permission Tests, please check the following:

Troubleshooting Step #1:

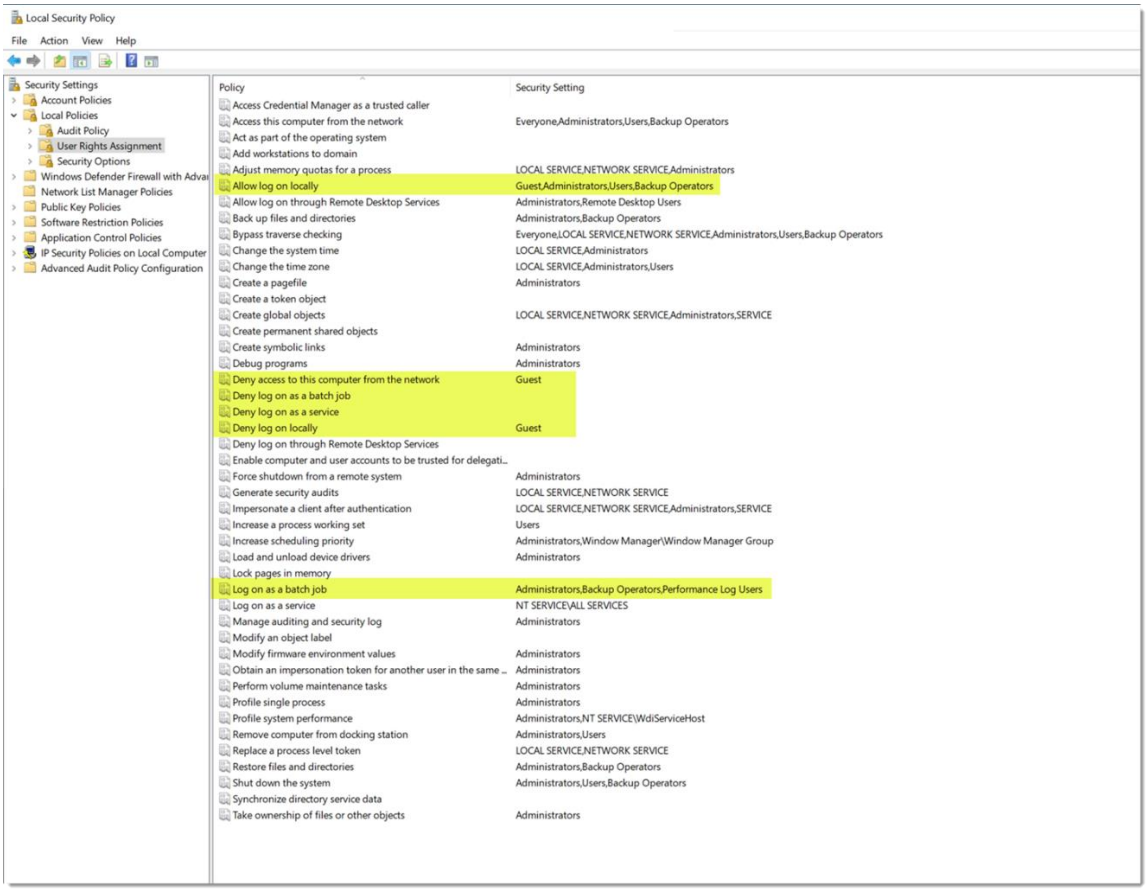
Confirm that the account you set up for backups, has a correct password set within Passwordstate. You can confirm this by opening the Password Record you created, and click on the Heartbeat icon.

Troubleshooting Step #2:

Ensure the account is not disabled on your server, or the password has expired.

Troubleshooting Step #3:

Investigate if you have a policy applying to your server that is denying the Backup Account from logging on. To check this, run the command "**secpol.msc**" on your Passwordstate server, and investigate these options:



Troubleshooting Step #4:

Another reason this error may happen is due to the way we impersonate the backup account. As the Passwordstate website runs under the identity of Network Service by default, which is a built in IIS account, we need to impersonate the backup account when performing tests, so it's the backup account you have configured that effectively performs the backup.

There are three Microsoft values for this impersonation, and the default value is 2. You can change this impersonation type to either 4, or 9 using SQL Management Studio Tools, when connected to your Passwordstate database server.

To change the Impersonation value to 4, run the following SQL Script: (change values to 9 if 4 does not work)

```
Use Passwordstate
Update BackupSettings Set BackupDatabaseImpersonation = 4
Update BackupSettings Set BackupEncryptionKeysImpersonation = 4
Update BackupSettings Set BackupFilesImpersonation = 4
Update BackupSettings Set BackupTest1Impersonation = 4
Update BackupSettings Set BackupTest2Impersonation = 4
Update BackupSettings Set BackupTest3Impersonation = 4
Update BackupSettings Set BackupTest4Impersonation = 4
Update BackupSettings Set UIImpersonation = 4
Update BackupSettings Set ServiceImpersonation = 4
```

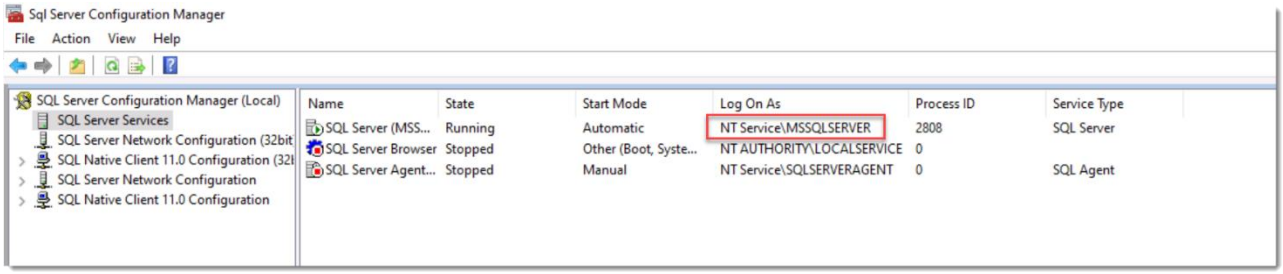
If neither of these impersonation values work, please revert the values back to the default values of 2.

If you still have issues running the tests, please contact Click Studio Support and send a screenshot of the test results for further analysis.

Troubleshooting Step #5:

If you find that the Passwordstate SQL database is not being backed up, either on a schedule or you notice an error when doing a test, please check the following settings:

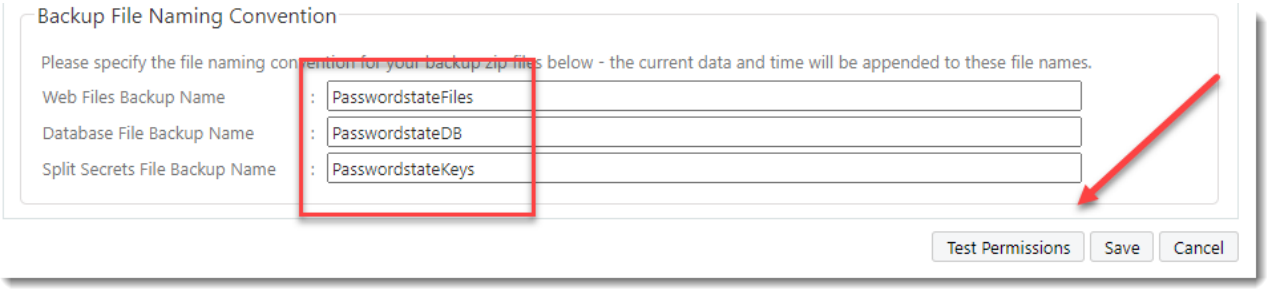
- If your database resides on a separate server to where you have Passwordstate installed, ensure you have configured the "**Database Location**" setting on the backups page appropriately
- If your SQL Services are configured to run under the identity of a custom account, then that account too will need access to your database, and network share. Unless you have a specific reason to run your services under a custom account, it may be easier to change the **Log On As** value back to the defaults, which can be seen in the screenshot below:



4.8 Backup File Naming Convention

The last section of the settings page gives you control of what your zip files should be called. These values can be changed to anything you like. The current date and time of the backup operation will be appended to these file names below.

The **Test Permissions** button at the bottom of the page can be used to check all your settings are valid.



4.9 Azure and Amazon SQL Databases

If you are using either an Azure or Amazon SQL database service to host your Passwordstate data, our backup process will not work with those services as they do not support the Powershell backup commands we use.

We'd recommend configuring those cloud services to back up your database, and then deselect the database backup option in Passwordstate.