



App Server Install and Administration Guide

Table of Contents

1	APP SERVER OVERVIEW	3
2	PREREQUISITES.....	4
3	INSTALLING THE APP SERVER.....	5
4	SSL CERTIFICATE CONSIDERATIONS	9
5	INITIAL APP SEVER CONFIGURATION.....	10
5.1	AUTHORIZED WEB SERVER.....	10
5.2	WEB.CONFIG FILE CHANGES.....	10
5.3	URL AND APP SERVER PUBLIC KEY	12
6	MOBILE APP PERMISSIONS	13
6.1	FEATURE ACCESS.....	13
6.2	MOBILE ACCESS TO PASSWORD LISTS.....	14
7	MOBILE APP SYSTEM SETTINGS.....	15
7.1	BRUTE FORCE PROTECTION	15
7.2	MASK PASSWORDS OR MAKE THEM VISIBLE.....	15
7.3	PASSWORD STRENGTH POLICY	15
7.4	OFFLINE CACHE	16
8	ENCRYPTING THE DATABASE CONNECTION STRING IN THE WEB.CONFIG FILE.....	17
9	ENCRYPTING THE APPSETTINGS SECTION WITHIN THE WEB.CONFIG FILE	18
10	BROWSER EXTENSION AND SELF DESTRUCT WEB SITE USAGE	19
11	APPSERVER TROUBLESHOOTING GUIDE	21

1 App Server Overview

This document will detail instructions for installing the Passwordstate App Server, and configuring System wide settings related to this feature. This feature will allow you to use the native Android and iOS Mobile App, but can also be used as a portal for the Self Destruct Messages feature.

Typically, you would install this App Server in your DMZ, so you can allow access to it from outside your network. The App Server communicates securely back to your core Passwordstate website, on the port you use to access your Passwordstate site.

Alternatively, you can install the App Server on your Passwordstate webserver, and you may want to do this if you have limited servers in your environment.

For more information on how to use the Mobile App once this App Server has been installed, please refer to the **Mobile App User Manual** found under the **Help Menu** in Passwordstate.

For more information about using the App Server with the Self Destruct Portal, please see **Administration - > System Settings -> Self Destruct Messages**.

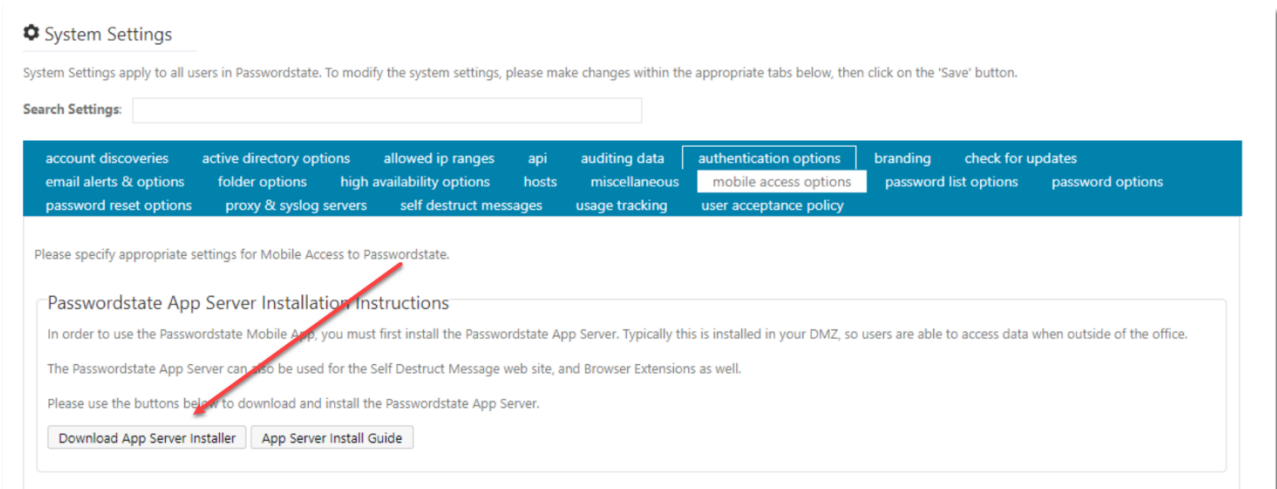
2 Prerequisites

In order for the App Server to function correctly, you will need to following requirements:

- An existing install of **Passwordstate 9** or above
- A Windows server running either **Windows 11**, or **Windows Server 2016** or above (*This can be your Passwordstate web server, or another server of your choice*)
- **Microsoft .NET Framework 4.7.2**
- A trusted SSL certificate (*More information on this in Section 4 of this guide*)
- From where your App Server is installed, you must have an open port for it to connect back to your SQL Server – this port is 1433 by default. If you are using a SQL Listener for database connectivity, you will need to allow port 1434 UDP – unless you have modified the default port.

3 Installing the App Server

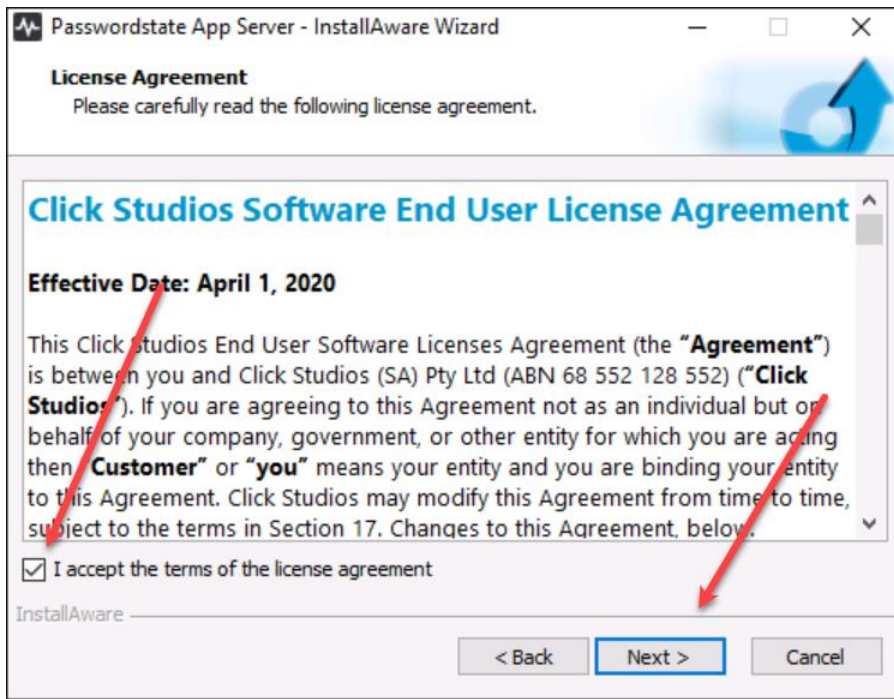
1. In Passwordstate, go to **Administration -> System Settings -> Mobile Access Options** and click the **Download App Server Installer** button:



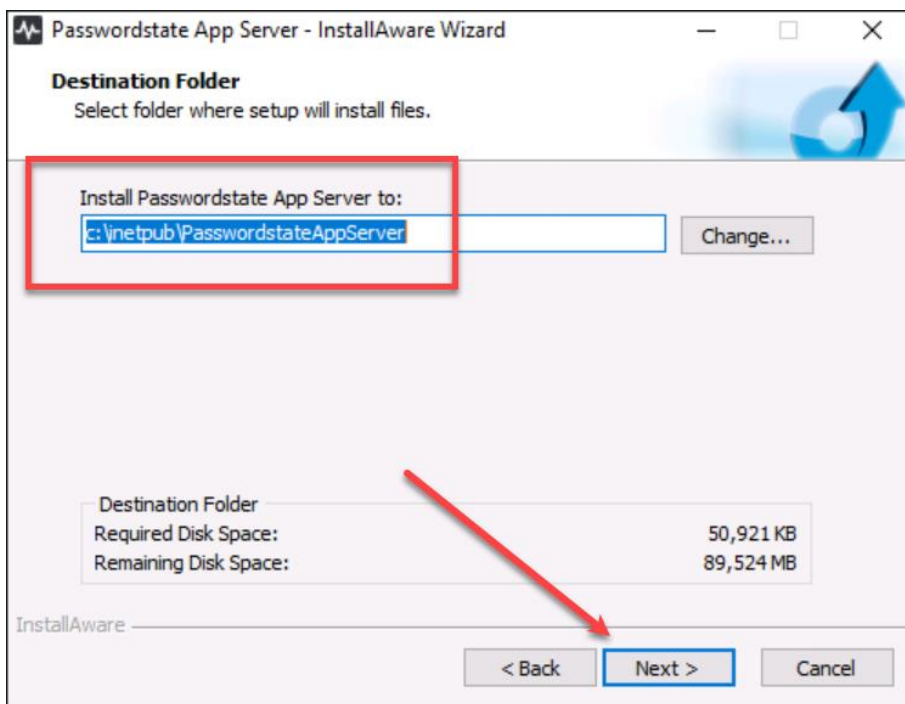
2. A file will download in your browser, and you'll need to transfer this to the server you are installing the App Server on
3. As an Administrator, run the **PasswordstateAppServer.exe** file
4. Click **Next**:



5. Accept the terms of the license agreement, and click **Next**:



6. Accept the default install path, and click **Next**:

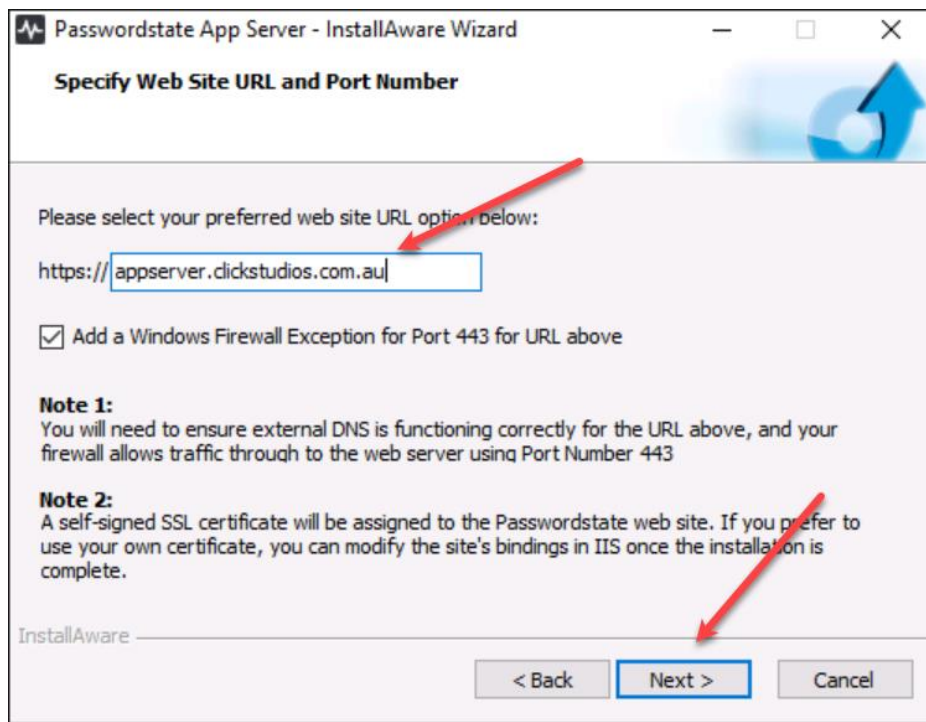


7. Set a URL of your choice, and click **Next**

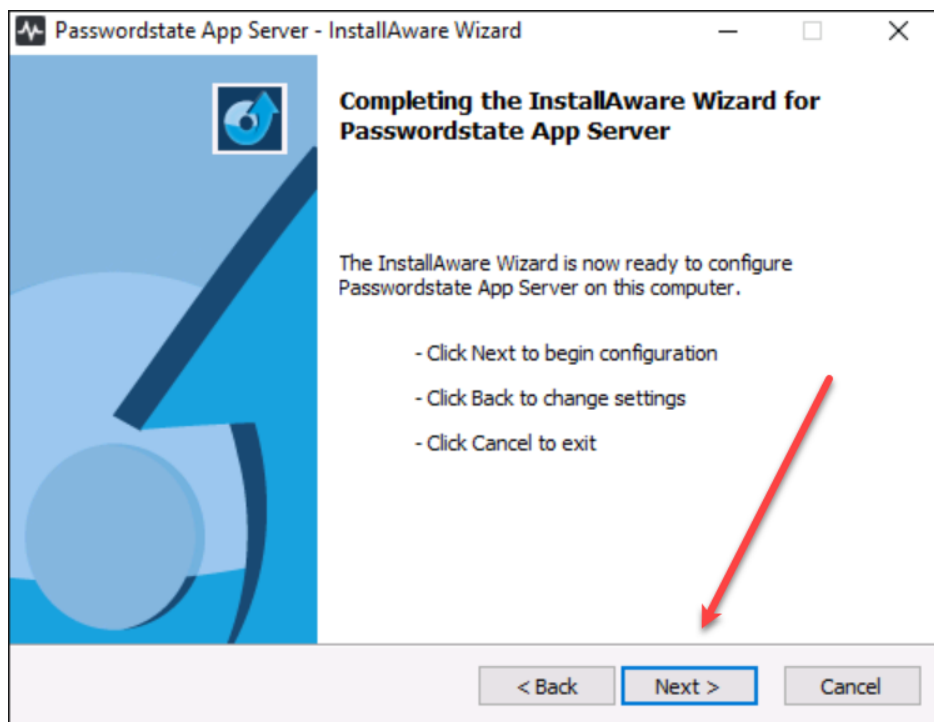
Note 1: This URL will require a functioning DNS entry, whether it be internal, or external from your network

Note 2: The default port set is 443, but this can be changed in IIS later if you require a custom port for any reason

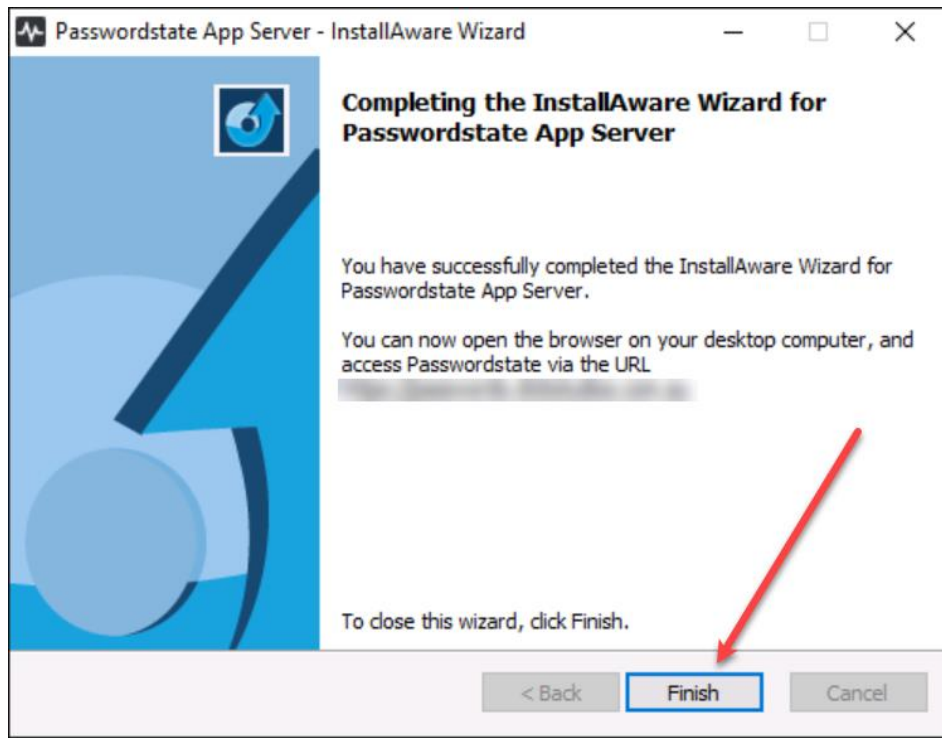
Note 3: This installer will set up a website in IIS with a Self Signed Certificate. You will need to change this certificate to a trusted one post install, to ensure for seamless access to your Mobile App.



8. Click **Next**



9. Click **Finish**



4 SSL Certificate Considerations

Typically, this Passwordstate App Server will be installed in your DMZ, and you will then permit access to this website from outside your network, by opening your firewall on Port 443 by default, and having an external functioning DNS entry. There are two reasons for this:

1. Users on mobile phones or tablets that aren't joined to your domain can connect to the AppServer, and use the Mobile App
2. Users can access the Self Destruct Messages portal from any machine outside your network, from any internet connection, and read Self Destruct Messages.

When installing the App Server, it will create a new website in IIS called **passwordstateappserver**. If IIS has not been installed already, the installer will automatically set up and configure IIS for you.

The installer will create a Self-Signed Certificate using the name of the URL that you have entered during the install process.

It is highly recommended you change this certificate and use a purchased SSL certificate and assign it to your **passwordstateappserver** website, to ensure functional use of the Mobile App. Also, if you intend on accessing your Self Destruct Portal using the Passwordstate App Server, instead of the embedded Self Destruct Portal, then you should also assign a purchased trusted certificate from an online authority.

The Self Signed certificate that comes with the installer or even a certificate issued by your internal Certificate Authority will present issues for your end users, as the devices from outside the network will not trust these certificates.

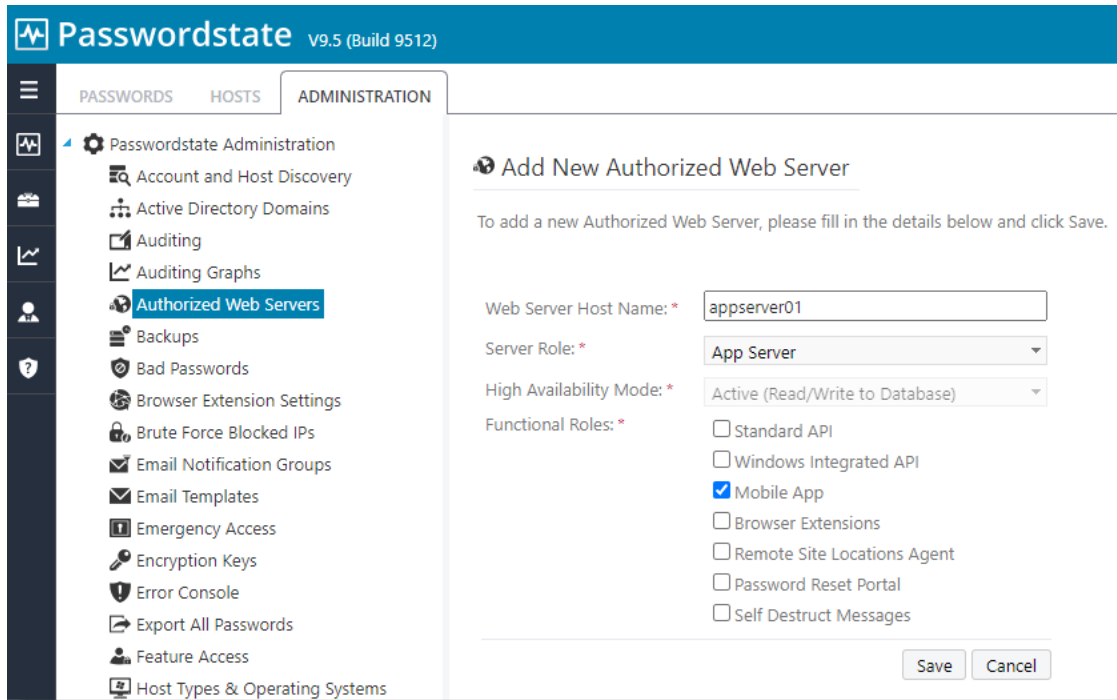
If your URL for the App Server is <https://appserver.contoso.com>, then you should use a certificate with a matching name of appserver.contoso.com, or even a wildcard certificate of *.contoso.com.

5 Initial App Server Configuration

Please follow these instructions for the initial configuration of the App Server, before it can be used.

5.1 Authorized Web Server

In Passwordstate, under **Administration** -> **Authorized Web Servers**, click the **Add** button to add in the App Server. Enter in the netbios name of the server where you installed the App Server, and also assign the App Server role. Once this is done, click Save.



5.2 Web.config File Changes

As a once off process, you will need to copy the following 3 lines from the web.config file from your main Passwordstate web site folder, to the web.config file in this App Server folder.

- Core Passwordstate vault web.config file default location: c:\inetpub\Passwordstate\web.config
- App Server web.config file default location: C:\inetpub\PasswordstateAppServer\web.config

Note: Only copy **PasswordstateConnectionString** and **Secret1** and **Secret2** lines

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <configuration>
3   <connectionStrings>
4     <add name="PasswordstateConnectionString" connectionString="Data Source=localhost;Initial Catalog=passwordstate;U
5   </connectionStrings>
6   <appSettings>
7     <add key="webpages:Version" value="3.0.0.0" />
8     <add key="webpages:Enabled" value="false" />
9     <add key="ClientValidationEnabled" value="true" />
10    <add key="UnobtrusiveJavaScriptEnabled" value="true" />
11    <add key="Secret1" value="671-1-6a54fc36c25a68de6a6cb960c2fa00cc1c891109df84afd63f1d80173c81daf65db1dda024a0f6999
12    <add key="Secret2" value="d51-1-61dfeb48c345203509d67b9b358e3576ac3c69a29ef9670eb6ad2d5c034dc503e78b01a6edd68efb
13  </appSettings>
14  <system.web>
15    <authentication mode="Forms">
16      <forms loginUrl="~/Default" name="Passwordstate" />
```

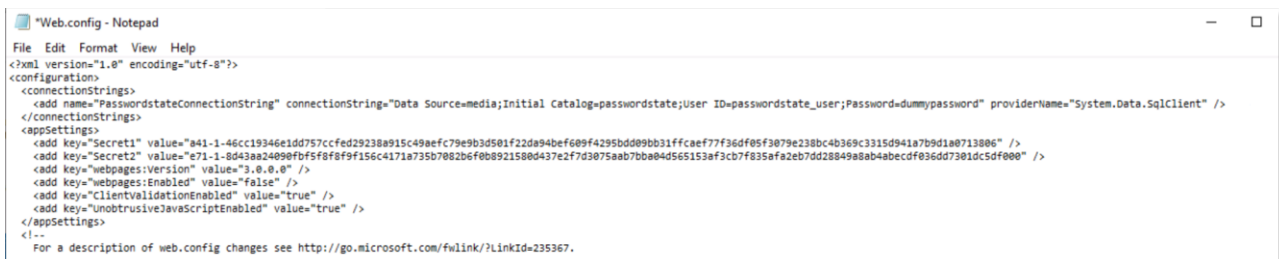
When looking at the web.config file for your main Passwordstate installation, if you do not see your database connection string, or appSetting section, then most likely you have encrypted these sections of the file, and you will need to decrypt them in order to copy across the contents. To do this, follow the instructions below – and only do this when users are not using Passwordstate:

- Open command prompt as Administrator, and run the following commands
 - CD C:\Windows\Microsoft.NET\Framework64\v4.0.30319
 - aspnet_regiis.exe -pdf "connectionStrings" "c:\inetpub\passwordstate"
 - aspnet_regiis.exe -pdf "appSettings" "c:\inetpub\passwordstate"

To re-encrypt them again afterwards, use the commands:

- aspnet_regiis.exe -pef "connectionStrings" "c:\inetpub\passwordstate"
- aspnet_regiis.exe -pef "appSettings" "c:\inetpub\passwordstate"

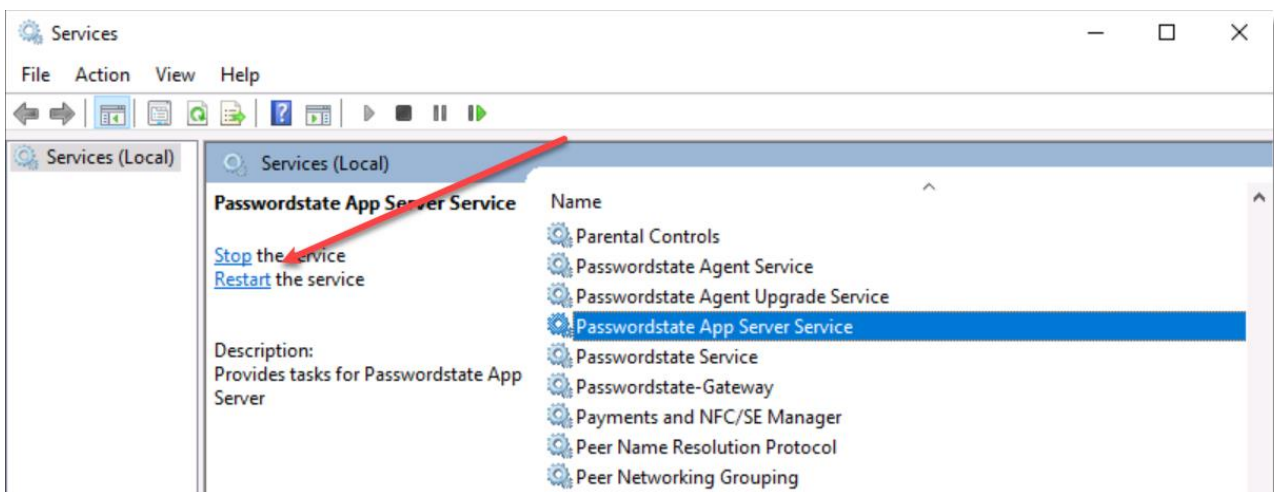
Example of App Server web.config file after inserting connection string and secret keys



```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <connectionStrings>
    <add name="PasswordstateConnectionString" connectionString="Data Source=media;Initial Catalog=passwordstate;User ID=passwordstate_user;Password=dummypassword" providerName="System.Data.SqlClient" />
  </connectionStrings>
  <appSettings>
    <add key="Secret1" value="a41-1-46cc19346e1dd757ccfed29238a915c49aefc79e9b3d501f22da94bef609f4295bd089bb31ffcaef77f36df05f3079e238bc4b369c3315d941a7b9d1a0713806" />
    <add key="Secret2" value="e71-1-8d43aa24090fbf5f8f9f15c4171a735b7082b6f0809215800437e2f7d3075aab7bba04d565153af3cb7f835fa2eb7dd288498ab4abecdf036dd7381dc5df000" />
    <add key="Webpages:Version" value="3.0.0.0" />
    <add key="Webpages:Enabled" value="false" />
    <add key="ClientValidationEnabled" value="true" />
    <add key="UnobtrusiveJavaScriptEnabled" value="true" />
  </appSettings>
</--
For a description of web.config changes see http://go.microsoft.com/fwlink/?LinkId=235367.
```

Note: From where your App Server is installed, you must have an open port for it to connect back to your SQL Server – this port is 1433 by default. If you are using a SQL Listener for database connectivity, you will need to allow port 1434 UDP – unless you have modified the default port.

Now restart the Passwordstate App Server Windows Service:



5.3 URL and APP Server Public Key

In Passwordstate, under **Administration** -> **System Settings** -> **Mobile Access Options**, set your URL for your App Server, including port if it is different to 443. Also, for added security, you should generate a SSL Public Key by clicking the **Query** button.

Once these fields are populated, click the **Save** button:

Mobile App URL and Security

Please note that changing any of the details below will require your users to rescan the Mobile App Server QR Code on their Preferences screen.

Specify the URL for your Passwordstate App Server installation (any time you change this URL, or update the SSL Certificate, you need to re-query the SSL Public Key below):

Reset App Pairing Secret for Passwordstate App Server:

App Pairing Secret is currently set

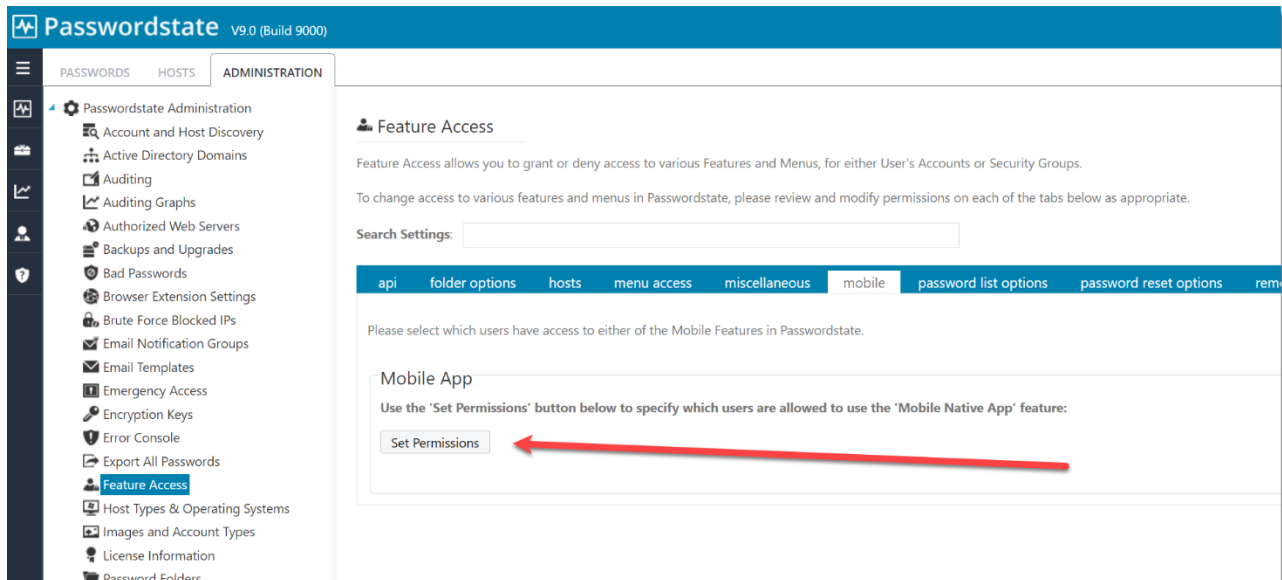
Passwordstate App Server's SSL Public Key: (This is a security feature to mitigate against SSL Certificate Man-in-the-middle attacks)

6 Mobile App Permissions

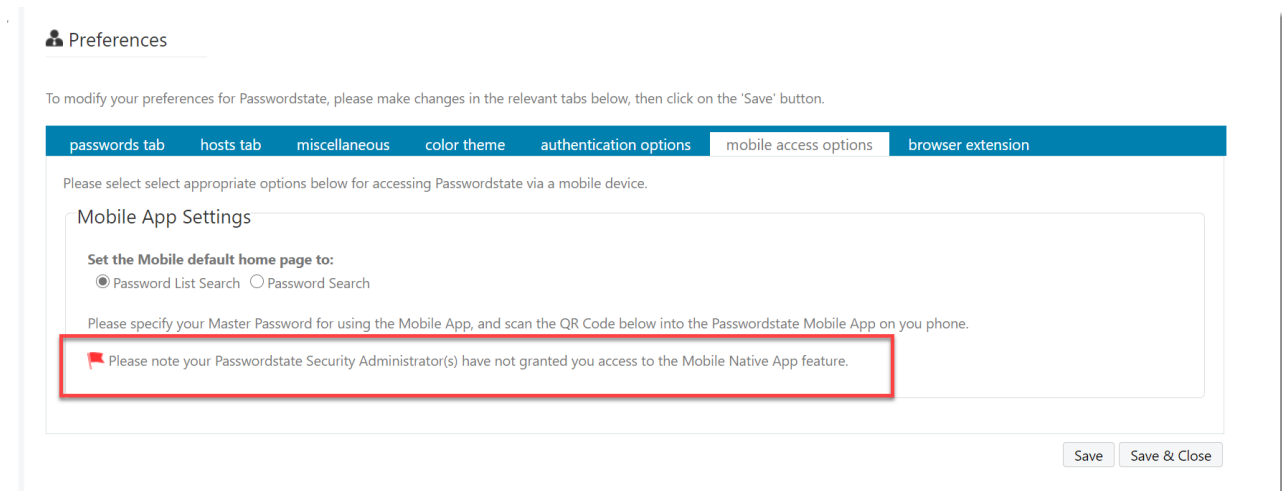
Passwordstate has the ability to enable or disable features for groups of users, and it's possible to prevent users from using the Mobile App. There are several places to consider when granting or denying these permissions.

6.1 Feature Access

As a Security Administrator, under **Administration** -> **Feature Access** -> **Mobile**, you can **allow** or **deny** users in Passwordstate from using the Mobile App:



If the user, or a Security Group the user is in, has been granted access on the screen above, they will be able to scan the QR code under their own personal preferences. If you deny them access, then they will be presented with this warning under their own personal preferences, and will not be able to use the App:



6.2 Mobile Access to Password Lists

For a greater level of control as to what passwords and Password Lists will show in the mobile App, you can turn off or on Mobile access at a Password List level. By default, when granting a user access to a Password List, it will also enable Mobile access too:

Grant New Permissions

To grant additional permissions to the 'Banking Details' Password List, simply click on the three **Tabs** below to specify appropriate permissions and/or settings.

access permissions | **time based access**

Search for an appropriate user or security group (use * to search for all).

Site Location : Internal

Search : *

Search For : User Security Group

Search Results

View Permissions

Mobile Access
Enabled Mobile Access for these permissions:
 Yes No

Reason for Access

Status: Save Cancel

If you select the above option to **No** when granting a user access, that Password List will not show up in the Mobile App for that user.

There is also an option in the System Settings to set that above option to be **No** by default, instead of **Yes**:

System Settings

System Settings apply to all users in Passwordstate. To modify the system settings, please make changes within the appropriate tabs below, then click on the 'Save' button.

Search Settings:

account discoveries | active directory options | allowed ip ranges | api | auditing data | authentication options | branding | check for updates | email alerts & options | folder options | high availability options | hosts | miscellaneous | **mobile access options** | password list options | password options | password reset options | proxy & syslog servers | self destruct messages | usage tracking | user acceptance policy

Please specify appropriate settings for Mobile Access to Passwordstate.

Passwordstate App Server Installation Instructions

In order to use the Passwordstate Mobile App, you must first install the Passwordstate App Server. Typically this is installed in your DMZ, so users are able to access data when outside of the office. The Passwordstate App Server can also be used for the Self Destruct Message web site, and Browser Extensions as well.

Please use the buttons below to download and install the Passwordstate App Server.

Download App Server Installer App Server Install Guide

Mobile App Settings

Protect against brute force dictionary authentication attempts by locking out an active session after the following number of failed login attempts: (Blocked IP Addresses can be removed on the screen Administration -> Brute Force Blocked IPs)

3

When adding new permissions to Password Lists, enabled Mobile Access by default: (Permissions can also be changed in bulk on the page Administration - Passwordstate Administration -> Password Lists)

Yes No

Select whether you would like Passwords Masked or Visible:

Mask Visible

7 Mobile App System Settings

Under **Administration** -> **System Settings** -> **Mobile Access Options**, there are also a number of other settings you can configure:

7.1 Brute Force Protection

If a user tries to log into the Mobile App, there is a brute force setting you can configure to prevent them from logging in again if they failed entering their **Master** password. Three is the default setting but is configurable:

Mobile App Settings

Protect against brute force dictionary authentication attempts by locking out an active session after the following number of failed login attempts: (Blocked IP Addresses can be removed on the screen Administration -> Brute Force Blocked IPs)

If a user fails to log in and the Brute Force feature locks them out of the Mobile App, the user will be presented with an error message on their Mobile App screen, and a Passwordstate Administrator must unblock them from this screen:

The screenshot shows the Passwordstate Administration interface. The left sidebar contains a navigation menu with items like Passwordstate Administration, Account and Host Discovery, Active Directory Domains, Auditing, Auditing Graphs, Authorized Web Servers, Backups and Upgrades, Bad Passwords, Browser Extension Settings, Brute Force Blocked IPs (highlighted), Email Notification Groups, Email Templates, Emergency Access, Encryption Keys, Error Console, and Export All Passwords. The main content area is titled 'Brute Force Blocked IPs' and includes a table with columns for Actions, Blocked Date, and IP Address. A red arrow points to the 'Remove Blocked IP Address' button in the Actions column for the entry with IP Address 10.0.0.120 and Blocked Date 15/12/2020 12:43:50 PM.

Actions	Blocked Date	IP Address
Remove Blocked IP Address	15/12/2020 12:43:50 PM	10.0.0.120

7.2 Mask Passwords or make them Visible

By default, your passwords in the Mobile App will be masked, and your users must press their finger on the masked password on the phone or tablet and choose **Toggle Mask** to see the password. You can make the password visible by default by changing this system setting option to **Visible**

Select whether you would like Passwords Masked or Visible:

Mask Visible

7.3 Password Strength Policy

When a user attempts to use the Mobile App for the first time, they must create a Master Password under their own personal preferences. When the users need to authenticate into their Mobile App, this is the

password that they need to enter. This system setting can force the user to adhere to a specific Password Strength Policy of your choice:

Select the Password Strength Policy the user's Master Password for Authentication must adhere to:

Default Policy ▼

7.4 Offline Cache

The Mobile App will sync with Passwordstate and download an offline cache of all Password Lists and Passwords user has access to. This feature improves performance dramatically and also allow your users to have access to their passwords when their device does not have access to the internet.

This System Setting below can be configured to require a manual authentication into the App more frequently, or less frequently, depending on your company policy.

Specify the number of days the user can access their offline cache before they need to re-authenticate again to the Passwordstate App Server (this setting can also be overwritten per user on the User Accounts screen in Administration):

7 Days ▼

By default, this setting is set to 7 days, and this value can be overwritten on a per user basis under **Administration -> User Accounts**, on the **Mobile Access Options** tab on each user in the system.

8 Encrypting the Database Connection String in the Web.config file

It is recommended you encrypt the database connection string within the web.config file, so the SQL Account credentials used to access the Passwordstate database is unreadable from anyone who can read the file system on your web server.

To encrypt the database connections string, please follow these instructions:

Encrypt Connection String

- Open a command prompt and change to the folder C:\Windows\Microsoft.NET\<Framework or Framework64>\v4.0.30319
- Type the following:
 - `aspnet_regiis.exe -pef "connectionStrings" "c:\inetpub\PasswordstateAppServer"` (change the path if you've installed the App Server to a different location)

Decrypt Connection String

- Open a command prompt and change to the folder C:\Windows\Microsoft.NET\<Framework or Framework64>\v4.0.30319
- Type the following:
 - `aspnet_regiis.exe -pdf "connectionStrings" "c:\inetpub\PasswordstateAppServer"` (change the path if you've installed the App Server to a different location)

Note: If you intend to rename your server host name, or move your Passwordstate App Server install to a different server, you should decrypt these settings first.

9 Encrypting the AppSettings Section within the Web.config file

It is recommended you encrypt the appSettings section within the web.config file, as this section of the file stores half of your split encryption keys.

To encrypt the appSettings section, please follow these instructions:

Encrypt appSettings Section

- Open a command prompt and type `CD C:\Windows\Microsoft.NET\Framework64\v4.0.30319`
- Type the following:
 - `aspnet_regiis.exe -pef "appSettings" "c:\inetpub\PasswordstateAppServer"` (change the path if you've installed the App Server to a different location)

Decrypt appSettings Section

- Open a command prompt and type `CD C:\Windows\Microsoft.NET\Framework64\v4.0.30319`
- Type the following:
 - `aspnet_regiis.exe -pdf "appSettings" "c:\inetpub\PasswordstateAppServer"` (change the path if you've installed the App Server to a different location)

Note: If you intend to rename your server host name, or move your Passwordstate App Server install to a different server, you should decrypt these settings first.

10 Browser Extension and Self Destruct Web Site Usage

The Passwordstate App Server can also be used for the Browser Extensions, and the Self Destruct Message web site if required.

Browser Extensions

If you want your user's browser extensions to communicate via the URL of your Passwordstate App Server, instead of your normal Passwordstate instance, you can configure this below. Users will need to log out of their extension, and log back into Passwordstate to pick up this change.

Passwordstate v9.0 (Build 9050)

PASSWORDS HOSTS ADMINISTRATION

Browser Extension Settings

Use each of the appropriate Tabs below to indicate various Settings, which URLs are ignored by the Browser Extension, which users are allowed to use

browser extension settings ignored urls allowed to use the extension prevent users from saving logins

Please specify general settings below for the behavior of the Browser Extension.

Automatically log the user out of their Browser Extension when they close the browser:
 Yes No

Attempt to form fill web sites if fields on the 'Browser Form Fields' tab for password records are blank: (this only applies to Legacy extensions, and not Chrome)
 Yes No

Automatically log the user out of their Browser Extension when their computer has been idle for (x) minutes:
0 (Setting to 0 disables this feature)

If you need your browser extensions to communicate to a different URL compared to your main Passwordstate URL, please specify it here: (in the format of https://mypasswordstate.com). This must also be the same database you're communicating with, otherwise encryption/decryption will not work with different encryption keys.

Various menus in the Browser Extension navigate users back to your Passwordstate Web Site. Please select below which URL you would like to use for this purpose:
 Alternative URL Specified Above Base URL Specified on the System Settings page

Save

Self Destruct Message Web Site

By default, the Self Destruct Message web site is embedded with your normal Passwordstate web site, but it can also be used with the App Server as well if required i.e. if you install the App Server is installed in the DMZ.

To use the Self Destruct web site with the App Server, simple specify your App Server URL below in the following screenshot.

- System Settings
- Account and Host Discovery
- Active Directory Domains
- Auditing
- Auditing Graphs
- Authorized Web Servers
- Backups and Upgrades
- Bad Passwords
- Browser Extension Settings
- Brute Force Blocked IPs
- Email Notification Groups
- Email Templates
- Emergency Access
- Encryption Keys
- Error Console
- Export All Passwords
- Feature Access
- Host Types & Operating Systems
- Images and Account Types
- License Information
- Password Folders
- Password Generator Policies
- Password Lists
- Password List Templates
- Password Strength Policies
- Privileged Account Credentials
- PowerShell Scripts
- Remote Session Management
- Reporting
- Security Administrators
- Security Groups
- User Accounts
- User Account Policies
- Remote Site Administration
- Remote Site Locations
- Password Reset Portal Administration
- Active Directory Domains
- Auditing
- Auditing Graphs

System Settings

System Settings apply to all users in Passwordstate. To modify the system settings, please make changes within the appropriate tabs below, then click on the 'Save' button.

Search Settings:

- account discoveries
- active directory options
- allowed ip ranges
- api
- auditing data
- authentication options
- branding
- check for updates
- email alerts & options
- folder options
- high availability options
- hosts
- miscellaneous
- mobile access options
- password list options
- password options
- password reset options
- proxy & syslog servers
- self destruct messages
- usage tracking
- user acceptance policy

Please specify settings below as appropriate for the Self Destruct Message Site.

Self Destruct Site Installation Instructions

By default, the Self Destruct Message web site is accessible as part of an "embedded" URL within your main Passwordstate web site i.e. with /selfdestruct appended to the end of your URL.

It also is available as part of the Passwordstate Application Server install, which can be installed in your DMZ for external access. Please click on the 'Mobile Access Options' tab above, for instructions on how to install the Passwordstate App Server.

Self Destruct Settings

Please specify settings for the Self Destruct Message Web Site below as appropriate.

Enforce the use of Passphrase protection for every Self Destruct Message sent:

Yes No

Allow users to see the value of Passphrases when composing Self Destruct Messages:

Yes No

Allow users to send Self Destruct Messages via the API:

Yes No

Specify which users are allowed to use the 'Send Self Destruct Message' Actions menu item for Password records:

Default Passphrase:

If Passphrase protection is mandatory, and the Passphrase is associated with the contact you are sending a message to, then this Passphrase will be used to protect the message.

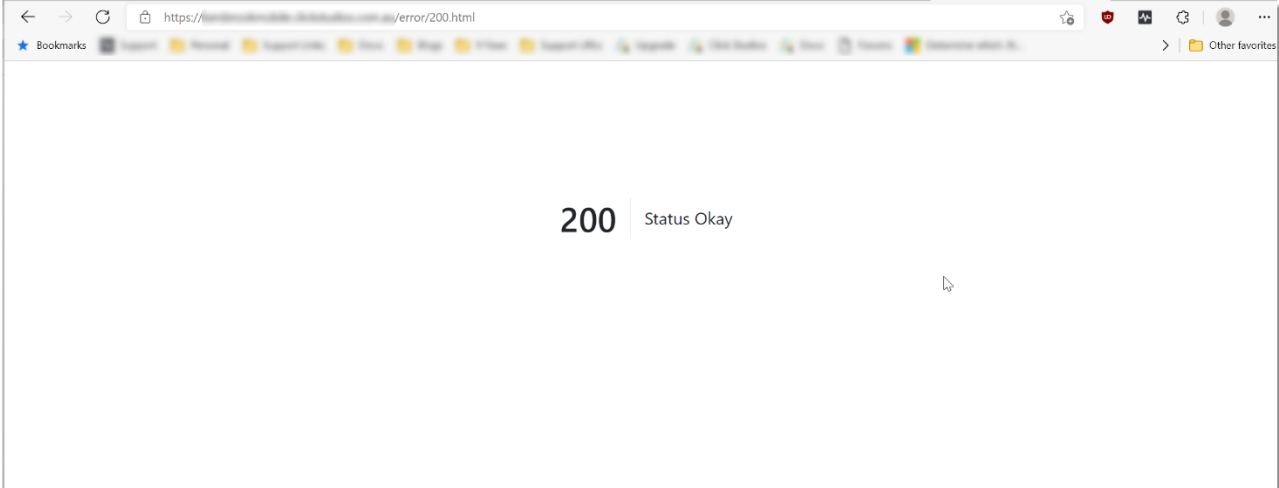
Separate Site URL:

Specify your Passwordstate Application Server URL here if you wish Self Destruct Messages to be accessed from there, as opposed to the embedded 'Self Destruct' site in your main Passwordstate install.

11 AppServer Troubleshooting Guide

In this section we'll troubleshoot the App Server not working

First test to do is try browsing to your App Server URL and if the website is working correctly, you will get a **200 | Status Okay** message as per the below screenshot. This is the URL you have set under **Administration -> System Settings -> Mobile Access Options**



If you see an Error like this screenshot below, log directly into your App Server and try browsing to your URL again. You should get a more verbose error message when logged directly into the server which can help troubleshoot the issue:

Server Error in '/' Application.

Runtime Error

Description: An application error occurred on the server. The current custom error settings for this application prevent the details of the application error from being viewed remotely (for security reasons). It could, however, be viewed by browsers running on the local server machine.

Details: To enable the details of this specific error message to be viewable on remote machines, please create a <customErrors> tag within a "web.config" configuration file located in the root directory of the current web application. This <customErrors> tag should then have its "mode" attribute set to "Off".

<!-- Web.Config Configuration File -->

```
<configuration>
  <system.web>
    <customErrors mode="Off"/>
  </system.web>
</configuration>
```

Notes: The current error page you are seeing can be replaced by a custom error page by modifying the "defaultRedirect" attribute of the application's <customErrors> configuration tag to point to a custom error page URL.

<!-- Web.Config Configuration File -->

```
<configuration>
  <system.web>
    <customErrors mode="RemoteOnly" defaultRedirect="mycustompage.htm"/>
  </system.web>
</configuration>
```

Error 1:

"A network-related or instance-specific error occurred while establishing a connection to SQL Server. The server was not found or was not accessible. Verify that the instance name is correct and that SQL Server is configured to allow remote connections"

What does this mean?

Your App Server needs to be able to communicate directly to your Passwordstate database in order for it to work successfully. There's a few possible reasons why your App Server cannot communicate to your database server, and below are some possible fixes for this:

Fix 1:

First confirm that you have copied the Connection String out of your web.config file from your core Passwordstate webserver across to the App Server web.config file. The database server name, SQL user and password need to be the same on both servers.

Please note if your web.config file is encrypted, you cannot copy the encrypted text across from the primary web.config file to the App Server web.config. You will need to decrypt it first and copy across the clear text you see. A good tutorial on what an encrypted web.config file looks like can be found here:

<https://forums.clickstudios.com.au/topic/2699-encrypting-and-decrypting-the-webconfig-file/>

An example of a clean Connection String to copy across can be seen in screenshot below:



```
web.config - Notepad
File Edit Format View Help
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <configSections>
    <sectionGroup name="telerik.web.ui">
      <section name="radScheduler" type="Telerik.Web.UI.RadSchedulerConfigurationSection" allowDefinition="MachineToApplication" requirePermission="false" />
      <section name="radCompression" type="Telerik.Web.UI.RadCompressionConfigurationSection" allowDefinition="MachineToApplication" requirePermission="false" />
    </sectionGroup>
  </configSections>
  <connectionStrings>
    <add name="PasswordstateConnectionString" connectionString="Data Source=webserver01;Initial Catalog=passwordstate;User ID=passwordstate_user;Password=MgDj$e9P$N5Sar4EucCs" providerName="System.Data.SqlClient" />
  </connectionStrings>
  <appSettings>
    <add key="SetupStage" value="Setup Complete" />
    <add key="Secret1" value="d71-1-98b40c0f4f827f1c1ab82cce9feb99e78c4d5f67bc1162348f49161b80c687c:d4339be71074a5d8330e9263d318e9294adae021bb97b3757c6ddaecb7f5eca" />
    <add key="Secret2" value="371-1-d1dca6ce731882e7a11e39c72e2786d213b4ce462c6bec70d99cb26ba9016988dd2cb2e84f2c0078e60cf9d063afec9e07d7f4480e6b180dce8d986533d5713c3f" />
  </appSettings>
  <system.web>
    <customErrors mode="On" defaultRedirect="/error/generalerror.aspx" />
  </system.web>
</configuration>
```

Fix 2:

Remote Directly into the App Server and perform a database connectivity test to you Passwordstate database server. An example of this in Powershell using the connection string above is:

Test-NetConnection webserver01 -Port 1433

1433 is the default port used for SQL, but is configurable. If you have changed the port, it will be referenced in your Connection String as per below screenshot:



```
web.config - Notepad
File Edit Format View Help
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <configSections>
    <sectionGroup name="telerik.web.ui">
      <section name="radScheduler" type="Telerik.Web.UI.RadSchedulerConfigurationSection" allowDefinition="MachineToApplication" requirePermission="false" />
      <section name="radCompression" type="Telerik.Web.UI.RadCompressionConfigurationSection" allowDefinition="MachineToApplication" requirePermission="false" />
    </sectionGroup>
  </configSections>
  <connectionStrings>
    <add name="PasswordstateConnectionString" connectionString="Data Source=webserver01,9000;Initial Catalog=passwordstate;User ID=passwordstate_user;Password=MgDj$e9P$N5Sar4EucCs" providerName="System.Data.SqlClient" />
  </connectionStrings>
  <appSettings>
    <add key="SetupStage" value="Setup Complete" />
    <add key="Secret1" value="d71-1-98b40c0f4f827f1c1ab82cce9feb99e78c4d5f67bc1162348f49161b80c687c:d4339be71074a5d8330e9263d318e9294adae021bb97b3757c6ddaecb7f5eca" />
    <add key="Secret2" value="371-1-d1dca6ce731882e7a11e39c72e2786d213b4ce462c6bec70d99cb26ba9016988dd2cb2e84f2c0078e60cf9d063afec9e07d7f4480e6b180dce8d986533d5713c3f" />
  </appSettings>
  <system.web>
    <customErrors mode="On" defaultRedirect="/error/generalerror.aspx" />
    <xhtmlConformance mode="Strict" />
  </system.web>
</configuration>
```

If this Open Port test returns a failure, then most likely a firewall is blocking access between your App Server and Database server.

If this test fails it could also possibly be DNS for the Database server name is not resolvable from your App Server. A quick test to check DNS working when logged into your app server is as follows, either in a command prompt, or in Powershell:

NSLookup webserver01

If this test fails, please investigate DNS to confirm it is set up correctly.

Fix 3:

If you are using SQL Express you may need to enable TCP/IP on your database server, and possibly set a static Port. SQL Express by default uses a dynamic port and setting a static port can ensure firewall rules are working correctly. An example of what a SQL Express instance looks like in you web.config file is as per below screenshot:



```
web.config - Notepad
File Edit Format View Help
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <configSections>
    <sectionGroup name="telerik.web.ui">
      <section name="radScheduler" type="Telerik.Web.UI.RadSchedulerConfigurationSection" allowDefinition="MachineToApplication" requirePermission="false" />
      <section name="radCompression" type="Telerik.Web.UI.RadCompressionConfigurationSection" allowDefinition="MachineToApplication" requirePermission="false" />
    </sectionGroup>
  </configSections>
  <connectionStrings>
    <add name="PasswordstateConnectionString" connectionString="Data Source=webservice01\sqlexpress;Initial Catalog=passwordstate;User ID=passwordstate_user;Password=Mg0j$e9PSH5Sar4Euc5" providerName="System.Data.SqlClient" />
  </connectionStrings>
  <appSettings>
    <add key="SetupStage" value="Setup Complete" />
    <add key="Secret1" value="d71-1-98b48c0f4f827f1c1ab82ce9feb99e78c4d5f67bc1162348f49161b80c68f7cd4339be71074a5d8330e9263d318e9294adae021bb97b3757c6ddaecb7f5eca" />
    <add key="Secret2" value="371-1-d1dcad6ce731882e7a11e39c72e2786d213b4ce462c6bec70d99cb26ba9016988dd2cb2e84f2c0078e68cf9d063afec9e07d7f4480e6b180dce8d986533d5713c3f" />
  </appSettings>
</system.web>
```

To learn how to fix this issue, please see Section 3 in this document:

https://www.clickstudios.com.au/downloads/version9/Installation_Instructions.pdf

Fix 4:

If you are using the High Availability Module of Passwordstate, it's possible you are also using SQL Replication with a "SQL Listener". If using one of these you may need to open port **1434 UDP** on your database server, unless you have specially have changed this port.