
DocuPro

Version 80000

**Installation
And
User's Guide**

© 2021-2021 Mpowered Ventures Ltd.
301 – 1555 Fir St
White Rock, BC V4B 4B6
Canada
<http://www.mpowered.biz>

1. Installing DocuPro

System Requirements

Web App (External):

- .NET: Requires an IIS server, running .NET Framework v4.5

Web Services (Internal):

- .NET: Requires an IIS server, running .NET Framework v4.5

Tempest Licences:

- Prospero
- Land

Create database user MpoweredWeb

Create a user named MpoweredWeb in each Tempest database (usually LIVE and TEST) that you wish to access with DocuPro. (You may already have created MpoweredWeb during the install of other Mpowered products.)

Grant database user MpoweredWeb database access permissions

Grant the table permissions found in \Docs\dbgrants.txt to the database user MpoweredWeb using your database management console. The table permissions need to run in each database (for example, Test and Live) that you will run DocuPro against. It's important to run the grants provided in the dbgrants.txt for each new version of DocuPro, as they can change – as well as after every Tempest patch/update.

Download the Install package

Go to www.mpowered.biz and click on Downloads. Here you will find links to various setup ZIPs that match recent versions of Tempest. Click on the Download link for the most recent version. This will download the ZIP package, which you can then extract into a working directory on your web servers (external and internal will need to be installed at the same version).

Contents of the ZIP package

Once the ZIP package is extracted to a working directory, you will find this structure:

```
\Docs  
\Dotnet
```

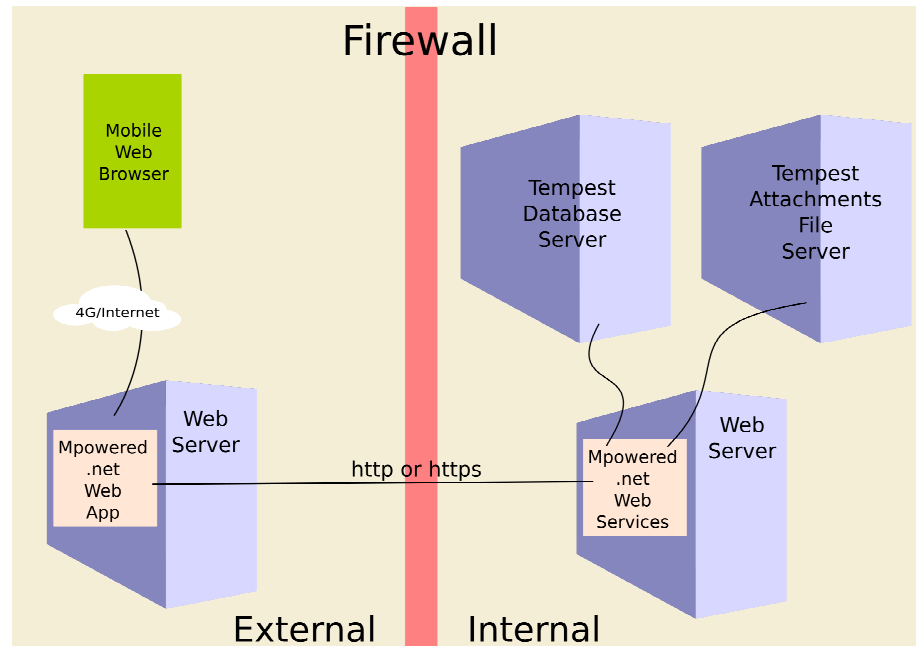
Mobile first philosophy

DocuPro is designed as a “mobile first” application. This means that the app is first designed for mobile devices, but it will also work beautifully on other devices - such as desktop browsers. The rest of this document uses the “mobile” terminology, but it applies equally to desktop browsers.

Install the .NET web app

The \Dotnet directory contains the .NET web app and web services required for DocuPro.

Examine the figure below:



On the mobile device, a browser (Chrome, Safari, etc) is used to load up the Mpowered .net web app on the external web server outside the main firewall, making its requests to an internal web server for anything to do with the Tempest database or Tempest attachments. The internal web server does all the heavy lifting and then simply hands the result back to the external web server, which then hands the web page back to the mobile browser.

The external web server and internal web server must be running Mpowered DocuPro software at the same version.

The huge benefits to this method are **security, security, security!** The external web server knows absolutely nothing about your internal network structure and configuration. No configuration files on the external web server contain any sensitive information, thereby removing any potential for malicious hacking from the outside. It is the internal web server (protected by your firewall and internal network security policies) that knows where your Tempest database and attachments servers are, and has the sensitive information about how to connect to your database servers.

Internal Web Server

On your internal (behind the firewall) web server, create a home directory for the Mpowered .NET web services if you don't already have one... something like:

```
C:\inetpub\wwwroot\Mpowered\DocuPro-80000
```

Copy the entire \Dotnet\WebServices\Redmond directory from the download here. Now on your internal web server, you should have this structure:

```
...\wwwroot\Mpowered\DocuPro-80000\  
    bin\  
        DP80000WS.dll  
        DocuPro.asmx  
        Web.config.internal.txt
```

Now edit the Web.config.internal.txt file and look for a section with the tag <connectionStrings> near the bottom. Here you will see a sample connection string for SQL Server named "MpoweredSQL".

With the connection string you will use, edit it so that YOURHOST becomes the server name where the Tempest database lives, and INSTANCE becomes the name of the database instance. Also, change the Password= to the MpoweredWeb password you created earlier. (NOTE: the password is entered in clear text here – this file should be secured so that only people with proper permissions can view this file. Contact Mpowered for more info about encrypting the config file if you wish additional security.) If you don't know the server name or password values, you may have to talk with your Database Administrator.

Note: you can have multiple connection strings in this file, for example you could have an MpoweredSQLProd and an MpoweredSQLTest connection string each pointing to the Production and Test Tempest databases. When you set up the external web server below, you will choose which DSN (connection string) to use.


♣ Very important!

Save and exit. Rename ♣ the Web.config.internal.txt file to **Web.config**

If you are upgrading from a previous version, MpoweredApps will already be there.

Now we need to fire up IIS Manager on the internal web server. Browse into Application Pools, and right-click and choose Add Application Pool. Create a new pool named "MpoweredApps" using .NET CLR Version v4.0.30319 (if you do not have this version, you will need to install MS .NET Framework 4.5 on this machine), Integrated, Start application pool immediately ON. Click on the newly created

pool, and browse to Advanced Settings on the right side menu. Make sure that Enable 32-Bit Applications is set to True, and click OK.

Now on the left tree, browse down to Sites > Default Web Site > Mpowered and right-click on DocuPro-80000. Choose Convert to Application. Keep the Alias as DocuPro-80000, but select Application pool MpoweredApps, and click OK. This should change the icon in the tree to: .

Now right-click on DocuPro-80000 again, and choose Manage Application > Browse. The default browser should appear with the DocuPro .NET services listing, containing links for AA_ServiceInfo, AC_DatabaseTest, etc. Click on AC_DatabaseTest, and just hit Invoke. You should get an XML page that says "SUCCESS: Found nnnn rows in the cd_folders table". This means that the DSN was set up correctly, and we are getting a connection to the Tempest database.

If you get the message "Timeout expired. The timeout period elapsed prior to completion of the operation or the server is not responding." you may be able to solve the issue by running "exec sp_updatestats" on the database.

That completes the set-up of the internal web server.

External Web Server

On your external (outside the firewall) web server, create a home directory for the Mpowered .NET web app... something like:

```
C:\inetpub\wwwroot\Mpowered\DocuPro-80000
```

Copy the entire \Dotnet\Client directory from the download here. Now on your external web server, you should have this structure:

```
...\wwwroot\Mpowered\DocuPro-80000\  
    bin\  
        DP80000.dll  
    favicon.ico  
    GetBlade.aspx  
    ...etc  
    Web.config.external.txt
```

If you are upgrading from a previous version, just copy the Web.config from there.


Now edit the Web.config.external.txt file and look for a section with the tag <appSettings> near the bottom. Here you will see a “webservice” key. It is the value that you must edit to point to the web services location on the internal web server (through the firewall). You may need to get your firewall expert to help you figure this one out. In most cases, you will simply need to change {ip} to the ip address of the internal web server (as seen from outside the firewall). You may also need to change the value of the “dsn” key to use the name of the connectionstring you entered in the setup of the external web server above.

♣ Very important!

Save and exit. Rename ♣ the Web.config.external.txt file to **Web.config**

If you are upgrading from a previous version, MpoweredApps will already be there.

Now we need to fire up IIS Manager on the external web server. Browse into Application Pools, and right-click and choose Add Application Pool. Create a new pool named “MpoweredApps” using .NET CLR Version v4.0.30319 (if you do not have this version, you will need to install MS .NET Framework 4.5 on this machine), Integrated, Start application pool immediately ON. Click on the newly created pool, and browse to Advanced Settings on the right side menu. Make sure that Enable 32-Bit Applications is set to True, and click OK.

Now on the left tree, browse down to Sites > Default Web Site > Mpowered and right-click on DocuPro-80000. Choose Convert to Application. Keep the Alias as DocuPro-80000, but select Application pool MpoweredApps, and click OK. This should change the icon in the tree to:  (you may have to refresh to see the icon).

A generic alias is highly recommended to save time in the future when new releases come out.

Now we are going to additionally create a “generic” alias that will point to this version, and can point to new versions (as they are released in the future) so that anyone who has bookmarked the application’s Index.aspx will not need to do anything to keep running the most-recent version of the application.

In IIS, right-click on the Mpowered node, and choose “Add Application...”. Make the Alias “DocuPro”, set the Application pool to “MpoweredApps”, and under Physical path use the [...] button to browse to the ...\\wwwroot\\Mpowered\\DocuPro-80000 directory used above. You should now see a node like this:



DocuPro

under the Mpowered node (you may have to refresh to see it).

Releasing new versions to users.

As you upgrade in the future, and after you have tested the new version using the DocuPro-nnnnn application, you can edit this DocuPro alias to point to the new version’s directory. Edit the alias once you are ready to “release” the new version to all users without having to manage their devices. All devices that have a bookmark to the alias will just simply start using the new version, next time they visit the bookmark.

Test to make sure external browsers can access DocuPro. You should be able to browse to:

`http://{yourserver}/mpowered/docupro/`

substituting https if on a secure server and your actual server name for {yourserver}. You should get the DocuPro start page.

The Web.config files

The Web.config files for both the external and internal servers contain all of the information about the way the application looks and operates. The external Web.config file contains a few settings which are needed when you want to turn the application offline, for example. Another config you will want to change is the app-000.010 which is a URL to the banner graphic DocuPro will use, and allows you to brand the app.

The bulk of the application is contained in the **internal** Web.config. This file is heavily documented with comments, and can guide you if you are going to DIY the install. Mpowered is always available for consultation, as this file can be tricky to understand and get right. The Web.config lets you configure almost everything about the app, and that makes it extremely flexible, but also requires you to be extremely careful about any changes to the Web.config. The most important advice here is to ALWAYS have a Test environment for DocuPro, and test any changes to the Web.config in Test first. A common error is to receive nothing but an Error 500, and this usually means that the Web.config.xml is not properly formed. Use a good xml editor/validator as you are making changes. Because the web.config file is actually xml, you need to be aware that you cannot use xml control characters like < and " within attributes, you need to xml-quote them as < and " to have a valid xml file. Another piece of standard advice: make a backup copy of the working Web.config file before you start making changes! One more thing: use the AD_ConfigAndDatabaseTest operation to check your internal Web.config for common issues with some of the most important app settings (see next section).

Once you are 100% sure that the Test site is working as expected, then promote the Web.config to Live. Note that **as soon as** you put a new Web.config in place, IIS will begin to use it immediately, and if there are issues or errors in the file, users will get those immediately as well – which obviously is undesirable.

AD_ConfigAndDatabaseTest and AE_EmailTest operations

There are two helpful Web.config checkers for your internal Web.config - AD_ConfigAndDatabaseTest and AE_EmailTest. You can access these operations while on the Internal server's IIS Manager, right-clicking on DocuPro-80000, and choosing Manage Application > Browse.

DocuPro

The following operations are supported. For a formal definition, please review the [Service Description](#).

- [AA_ServiceInfo](#)
Returns basic service info (no forwarding), indicating that the web service is registered on the machine correctly. Fo
- [AC_DatabaseTest](#)
Simple database connection test using server-side config 'dsn'. Returns the number of rows in the cd_folders table.
- [AD_ConfigAndDatabaseTest](#)
Checks appSetting 'app-740.200' to make sure that all the cdcat/cdtype folder type combinations exist.
- [AE_EmailTest](#)
Checks appSetting 'app-000.090' to test if emails are going through. Requires a valid AC_DatabaseTest first.
- [dp_procmain](#)

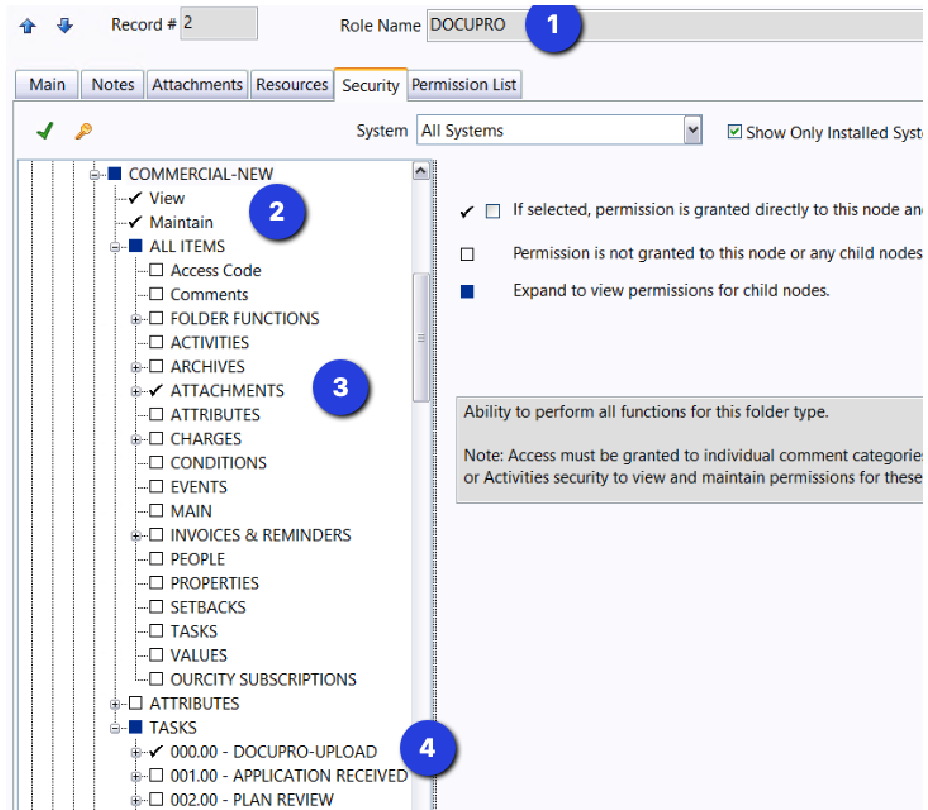
AE_EmailTest will check that an email can be sent to the email address defined in app setting app-000.090. It is really important to make sure that emails are going through on the Live system! The most common issue with not receiving emails is that the Internal server does not have a firewall opening to the server/port defined in the Web.config.

AD_ConfigAndDatabaseTest does an extensive crosscheck of a few key settings (app-740.200 and related) to make sure that they are valid, and that the Tempest resources specified in those settings have the correct security, etc. Use this operation frequently when changing or adding to those settings. See Security Nodes for DocuPro below for some key security the DocuPro user needs in Tempest.

Security Nodes for DocuPro

The Internal Web.config setting app-000.020 (usually DOCUPRO) defines the Tempest user the DocuPro app will use for all of its Tempest database operations, and it requires Tempest security in order to function. This allows you to restrict the DocuPro app to only folder types that you wish to make available.

In the example below, the DOCUPRO user has been granted Maintain security for the COMMERCIAL-NEW folder type, as well as Attachments and the task DOCUPRO-UPLOAD.



The AD_ConfigAndDatabaseTest discussed above will check for the correct security, and so using it as you add/change the Web.config file will be useful.

And In The End... Using DocuPro

When you browse to:

<https://{yourserver}/mpowered/DocuPro>

substituting your actual External server name for {yourserver}, you should see the DocuPro start page. When you are connected to a database that is defined as a Test database in Tempest, you will see an info block at the top notifying you of this.

