

# **VEhub Admin Release Process**

Aptean Ltd  
Copyright © 2011-2025.

# Contents

- 1 VEhub Admin Release Process.....1
  - 1.1 INFO:.....1
  - 1.2 Dev/Changes:.....1
  - 1.3 Build:.....1
  - 1.4 Release:.....2
  - 1.5 Switch Over.....4
  - 1.6 Tidy Up.....5

# 1 VEHub Admin Release Process

## VEHub Server Release Process

### 1.1 INFO:

1. VEHub code can be found in the repository: [https://dev.azure.com/Aptean/Calidus/\\_git/Vehub](https://dev.azure.com/Aptean/Calidus/_git/Vehub)
2. VEHub test web server is 172.183.45.55 (requires VPN, Runcorn, for access).
3. VEHub test database is on 172.164.45.205 (connect from local PC with VPN using MS SSMS).
4. VEHub live web server is the shared web (172.183.45.50)
5. VEHub live database is on 172.164.45.200 (connect from local PC with VPN using MS SSMS).
6. VS = Microsoft Visual Studio (currently using version 2019 for server dev, incl. builds)
7. MS SSMS = Microsoft SQL Server Management Studio
8. Anywhere in this document that refers to `_0.0.0.0_` this should be replaced by/read as the version number of the existing installation (i.e. the version on the target server before the release).
9. Anywhere in this document that refers to `_9.9.9.9_` this should be replaced by/read as the version number being built/released, i.e. the new version number (after it has been incremented in step 2 in the Dev/Changes section below).
10. Anywhere in this document that refers to `_env_vers_`, `env` should be replaced by/read as the environment (Test/Live) being released to and `vers` should be replaced by/read as the version being released in the format `xxxxxxx` with each portion of the version being padded to 2 digits (for example if releasing version 2.5.0.4 to test `env_vers` should be replaced by/read as `Test_02050004`).
11. Anywhere in this document that refers to `_env_versold_`, `env` should be replaced by/read as the environment (Test/Live) being released to and `vers` should be replaced by/read as the version being replaced in the format `xxxxxxx` with each portion of the version being padded to 2 digits (for example if releasing to test, replacing version 2.0.5.3, `env_versold` should be replaced by/read as `Test_02000503`).
12. Examples in this document will assume releasing version 2.0.5.4, replacing an existing version 2.0.5.3, into the test environment in the example.

### 1.2 Dev/Changes:

Before building/releasing the following should be completed

1. Any code changes should be completed and committed/merged into the `?VehubDev?` branch.
2. If the code changes did not include incrementing the version this should be incremented now by incrementing the following:
  1. Version number in `AssemblyVersion` and `AssemblyFileVersion` settings in the `AssemblyInfo.cs` files, found in the properties sub-folder, in the `?AdminTool > VEHub.admin?`, `?Core > VEHub.Core?`, `?Services > Emailer > Emailer?`, `?Services > TomTomLink > TomTomLink?`, `?Services > Poller > VEHub.WindowsService?`, `?WcfService > VEHub.service?` and `?WebPortal > VehubUI?` projects.
  2. The version number portion of the `publishUrl` value in the `?WcfService > VEHub.WindowsService > Properties > PublishProfiles > Local.pubxml?` file and the `?WebPortal > VehubUI > Properties > PublishProfiles > Local.pubxml?` file.
3. All of the required changes, including the version number increment should be merged from the `?VehubDev?` branch into the `?VehubQa?` branch.
  1. **Note** Any changes to `?config?` files are not included in a release but ideally, we should keep settings used for test/live servers in QA branch whereas we may have settings for dev environment in Dev branch.

### 1.3 Build:

These steps are performed on your local PC and need to be performed ahead of any release after all steps in the Dev/Changes section above have been completed.

1. Create a new folder called `?VEHub _9.9.9.9_?`. Suggest creating this in `C:\Vehub Builds`. This `VEHub _9.9.9.9_` folder is the **release folder**.
2. Publish the `VEHub.UI` project (from the QA branch) in VS.
  1. Publish the `WebPortal > VEHub.UI` project by right clicking the `VEHub.UI` project in the Solution Explorer section and selecting the Publish option. In the window this opens ensure the `Local.pubxml` file is selected (drop-down near the top) and click the Publish button. This publishes to the **UI Publish** folder (should be `C:\Vehub Builds\VEhub.UI 9.9.9.9`, the actual folder used is taken from the `Local.pubxml` file)



- and is displayed in the Output window).
- 2. Delete the web.config from the **\*\*UI publish\*\*** folder.
- 3. Copy/move the **\*\*UI publish\*\*** folder into the release folder.
- 3. Publish the VEHub.service project
  - 1. Publish the WcfService > VEHub.Service project by right clicking the VEHub.Service project in the Solution Explorer section and selecting the Publish option. In the window this opens ensure the Local.pubxml file is selected (drop-down near the top) and click the Publish button. This publishes to the Service Publish folder (should be C:\Vehub Builds\VEHub.Service \_9.9.9.9\_, the actual folder used is taken from the Local.pubxml file and is displayed in the Output window).
  - 2. Delete the web.config file from the service publish folder.
  - 3. Copy/move the Service publish folder into the release folder.
- 4. Build the Emailer project
  - 1. Create a local emailer build folder called VEHub.Emailer \_9.9.9.9\_ (e.g. C:\Vehub Builds\VEHub.Emailer 2.0.5.4). This is the **\*\*Emailer Build\*\*** folder.
  - 2. Build the Service > Emailer > Emailer project by right clicking the Emailer project in the Solution Explorer section and selecting the Build option. Take note of the configuration (Release or Debug) used (shown in drop-down on toolbar). Open the folder in file explorer (right click the Emailer project and select the ?OpenFolder in File Explorer? option), go into the bin folder. From there go into the Release or Debug folder (depending on which configuration was used, you can check the date/time of the Emailer.application or Emailer.exe files to make sure this is the correct folder), this folder is also shown in the output window. Copy all of the contents of this Release/Debug folder into the **\*\*Emailer Build\*\*** folder.
  - 3. Delete the Emailer.exe.config file in the **\*\*Emailer Build\*\*** folder.
  - 4. Copy/move the **\*\*Emailer Build\*\*** folder into the release folder.
- 5. Build the Vehub.WindowsService project (i.e. the poller project)
  - 1. Create a local poller build folder called VEHub.Poller 9.9.9.9 (e.g. C:\Vehub Builds\VEHub.Poller 2.0.5.4). This is the **\*\*Poller build\*\*** folder.
  - 2. Build the Services > Poller > VEHub.WindowService project by right clicking the VEHub.WindowService project in the Solution Explorer section and selecting the Build option. Take note of the configuration (Release or Debug) used (shown in drop-down on toolbar). Open the folder in file explorer (right click the Emailer project and select the ?OpenFolder in File Explorer? option), go into the bin folder. From there go into the Release or Debug folder (depending on which configuration was used, you can check the date/time of the VEHub.WindowsService.exe file to make sure this is the correct folder), this folder is also shown in the output window. Copy all of the contents of this Release/Debug folder into the **\*\*Poller build\*\*** folder.
  - 3. Delete the VEHub.WindowsService.exe.config file in the **\*\*Poller build\*\*** folder.
  - 4. Copy/move the **\*\*Poller build\*\*** folder into the release folder.
- 6. Zip the release folder to create a VEHub \_9.9.9.9\_.zip file, this is the **\*\*release Zip file\*\***.
- 7. Copy the **\*\*release Zip file\*\*** into the \\DGA1FS01OBS\Projects\Product\VEHUB\5. Releasing\Server\Releases directory on the network drives.

## 1.4 Release:

These steps are mainly performed on the target (test or live) web server and need to be performed ahead of

- 1. Copy the **\*\*release ZIP file\*\*** for the version to be released onto the target server. There is a ?Release working folder? on the same level as the IIS folders that can be used. Note you may have to copy the zip file onto the desktop then move it to this directory.
- 2. Extract the release Zip file to create a **\*\*release folder\*\***.
- 3. Release the UI programs
  - 1. Create a directory called VEHub.UI.\_env\_vers\_ at the same level as the existing UI site. To do this right click the existing UI site in IIS manager and select the Explore option to open the existing directory in file explorer. Go up one directory and create the new VEHub.UI.\_env\_vers\_ directory there, i.e. alongside the existing directory. For example create a new C:\inetpub\wwwroot\VEHub.UI.Test\_02050004 directory.
  - 2. Copy the Web.config from the existing UI directory into the newly created UI directory. For example copy the web.config file from the existing C:\inetpub\wwwroot\VEHub.UI.Test\_02050003 directory to the new C:\inetpub\wwwroot\VEHub.UI.Test\_02050004 directory.
  - 3. Copy all contents of the VEHub.UI folder (in the **\*\*release folder\*\***) into to the new VEHub.UI.\_env\_vers\_ folder.
- 4. Create the UI application pool
  - 1. In the IIS manager right-click on the Application Pools and select the ?Add Application Pool? option.
  - 2. In the opened pop-up set the Name to VEHUB.UI.\_env\_vers\_
  - 3. Click the OK button



### 1. Create the UI IIS site

1. In IIS Manager right-click the Sites folder and select the ?Add website?? option.
2. Set the Site Name to VEHub.UI.\_env\_vers\_
3. Check the application pool is set to the VEHub.UI.env\_vers application pool created in step 3 above.
4. Set the Physical path to the directory created in step 2a above
5. In the binding section set the Type to http, set the IP Address to All Unassigned and set the Port to any unused port number (suggest 2099).
6. Uncheck the ?Start Website immediately? checkbox.
7. Click OK

### 1. Release the Service programs

1. Create a directory called VEHub.Service.\_env\_vers\_ at the same level as the existing Service site. To do this right click the existing Service site in IIS manager and select the Explore option to open the existing directory in file explorer. Go up one directory and create the new VEHub.Service.\_env\_vers\_ directory there, i.e. alongside the existing directory. For example create a new C:\inetpub\wwwroot\VEHub.Service.Test\_02050004 directory.
2. Copy the Web.config from the existing Service directory into the newly created Service directory. For example copy the web.config file from the existing C:\inetpub\wwwroot\VEHub.Service.Test\_02050003 directory to the new C:\inetpub\wwwroot\VEHub.Service.Test\_02050004 directory.
3. Copy all contents of the VEHub.Service folder (in the \*\*release folder\*\*) into to the new VEHub.UI.\_env\_vers\_ folder.

### 2. Create the Service application pool

1. In the IIS manager right-click on the Application Pools and select the ?Add Application Pool? option.
2. In the opened pop-up set the Name to VEHub.Service.env\_vers
3. Click the OK button

### 1. Create the Service IIS site

1. In IIS Manager right-click the Sites folder and select the ?Add website?? option.
2. Set the Site Name to VEHub.Service.\_env\_vers\_
3. Check the application pool is set to the VEHub.Service.\_env\_vers\_ application pool created in step 6 above.
4. Set the Physical path to the directory created in step 5a above
5. In the binding section set the Type to http, set the IP Address to All Unassigned and set the Port to any unused port number (suggest 45499).
6. Uncheck the ?Start Website immediately? checkbox.
7. Click OK

### 1. Create Emailer service

1. Create a folder called VEHub.Emailer.\_env\_vers\_ within the C:\VEHub Services directory
2. Copy the Emailer.exe.config from the existing Service directory into the newly created Service directory. For example copy the Emailer.exe.config file from the existing C:\VEHub Services\VEHub.Emailer.Test\_02000503 directory to the new C:\VEHub Services\VEHub.Emailer.Test\_02000504 directory.
3. Copy all contents of the VEHub.Emailer folder (in the \*\*release folder\*\*) into to the new VEHub.Emailer.\_env\_vers\_ folder.
4. Run the following command (using an elevated command prompt, i.e. run command prompt as administrator):

```
`sc.exe create VEHub.Emailer.env_vers binpath= "C:\VEHub Services\VEHub.Emailer.env_vers\Emailer.exe" `
```

### 1. 1. Again using an elevated command prompt, run the following command:

```
`sc.exe description VEHub.Emailer.env_vers "Sends VEHub emails using release env_vers" `
```

1. 1. Open the services app and check the service for name VEHub.Emailer.env\_vers has been created and has the Startup Type set to Manual and is not running (Status is blank).
2. Open the services app and check the service for name VEHub.Emailer.\_env\_vers\_ has been created and has the Startup Type set to Manual and is not running (Status is blank).
3. Create Poller service



1. Create a folder called VEHUB.Poller.\_env\_vers\_ within the C:\VEHub Services directory
2. Copy the VEHUB.WindowsService.exe.config from the existing Service directory into the newly created Service directory. For example copy the VEHUB.WindowsService.exe.config file from the existing C:\VEHub Services\VEHub.Poller.Test\_02000503 directory to the new C:\VEHub Services\VEHub.Poller.Test\_02000504 directory.
3. Copy all contents of the VEHUB.Poller folder (in the release folder) into to the new VEHUB.Poller.\_env\_vers\_ folder.
4. Run the following command (using an elevated command prompt, i.e. run command prompt as administrator):

```
`sc.exe create VEHUB.Poller._env_vers_ binpath= "C:\VEHub Services\VEHub.Poller._env_vers_\VEHub.WindowsService.exe"
```

1. Again using an elevated command prompt, run the following command:

```
`sc.exe description VEHUB.Poller._env_vers_ "Checks for overdue VEHUB vehicle checks using release env_vers_"`
```

1. Open the services app and check the service for name VEHUB.Poller.\_env\_vers\_ has been created and has the Startup Type set to Manual and is not running (Status is blank).
2. Prepare DB Scripts incl. Rollback script
  1. Create a new folder, called Release Scripts inside the release folder. This new folder is the release scripts folder.
  2. Find any update script folders for any versions included by this release. For our example of releasing version 2.0.5.4, replacing 2.0.5.3, this would include any folder for version 2.0.5.4. This should include folders for any version higher than the existing version up to and including the release version. So if the existing version was 2.0.5.1 (instead of 2.0.5.3, but still releasing version 2.0.5.4) then this would include any folders for versions 2.0.5.2, 2.0.5.3 and 2.0.5.4. These folders can be found in the bin/Scripts/Updates folder within the UI folder (either in the release ZIP file, the release file or the VEHUB.UI.\_env\_vers\_ folder from step 2 above). Copy all SQL scripts from these folders into the release scripts folder. If there are no update scripts folders/update scripts you can skip the rest of this step 13.
  3. Create an empty SQL file called Rollback to V0.0.0.0.sql in the release scripts folder.
  4. Go through each of the update scripts found/copied in step 13.b above and add SQL to undo the changes to the rollback SQL file. Note If there are multiple changes to the same records/file/stored procedures the Rollback SQL only needs to set these back to their original state (i.e. before the update scripts are applied). The original state of the database records/file/stored procedures can be found on the target database. Note 2 The test and live databases can both be accessed remotely using MS SSMS (Microsoft SQL Server Management Studio).

## 1.5 Switch Over

These steps are performed on the target (test or live) web server and need to be performed after all steps in the Release section and release prep above have been completed. Note: These will require the target system being taken down and so for a release to the live system clients should be warned of the planned down-time.

1. In IIS manager stop the VEHUB.UI.\_env\_versold\_ site.
2. Swap the port numbers used on the VEHUB.UI.\_env\_versold\_ and VEHUB UI.\_env\_vers\_ sites. As both sites should be stopped at this point it will allow you to set the same port on both but will issue a warning.
3. In IIS manager stop the VEHUB.Service.\_env\_versold\_ site.
4. Swap the port numbers used on the VEHUB.Service.\_env\_versold\_ and VEHUB Service.\_env\_vers\_ sites. As both sites should be stopped at this point it will allow you to set the same port on both but will issue a warning.
5. In the windows services app stop the VEHUB.Emailer.\_env\_versold\_ service. Set the startup type to Disabled (by double-clicking on the service and changing the startup type drop-down)
6. In the windows services app stop the VEHUB.Poller.\_env\_versold\_ service. Set the startup type to Disabled (by double-clicking on the service and changing the startup type drop-down)
7. Apply all Update scripts (not the Rollback script) from the release scripts folder (created in step 3.b above) to the target database.
8. Start the VEHUB.UI.\_env\_vers\_ site in IIS manager.
9. Start the VEHUB.Service.\_env\_vers\_ site in IIS manager.
10. Start the VEHUB.Emailer.\_env\_vers\_ service.
11. Start the VEHUB.Poller.\_env\_vers\_ service.



## 1.6 Tidy Up

These steps are intended to prevent the build-up of old IIS sites and windows services over time and should only be performed after we can be sure everything is OK and no Rollback will be required (e.g. after a test release has been passed or a live release has been used by clients with no issues for some time).

1. Delete the old VEHub.Poller.\_env\_versold\_ service using:

```
`sc.exe create VEHub.Poller.env_versold`
```

1. Delete the old VEHub.Emailer.\_env\_versold\_ service using:

```
`sc.exe create VEHub.Emailer.env_versold`
```

1. Delete the old VEHub.UI.\_env\_versold\_ site in IIS manager, making sure to remove the directory it points to.
2. Delete the old VEHub.Service.\_env\_versold\_ site in IIS manager, making sure to remove the directory it points to.

