

Email Monitor Technote

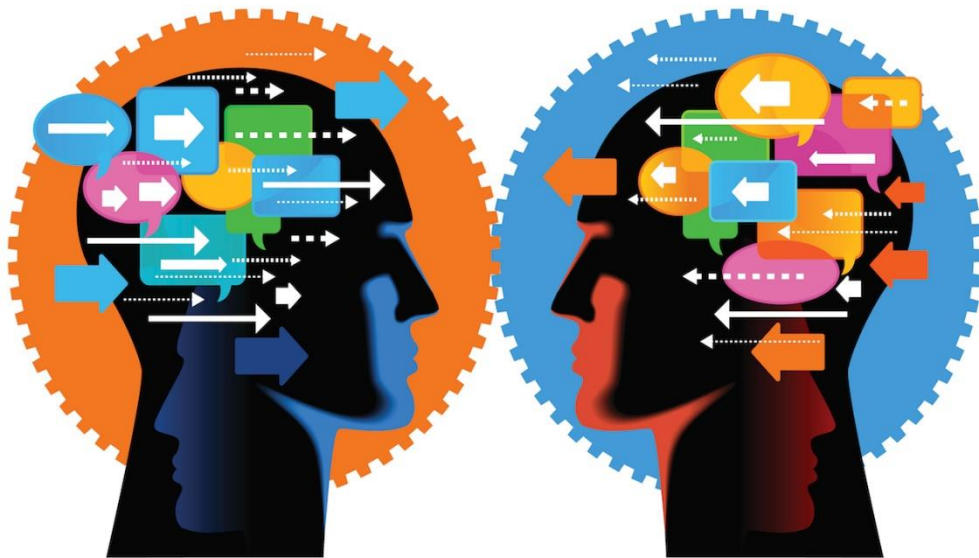


TABLE OF CONTENTS

MR Services (MR Agent)	2
Supported Azure DevOps(TFS) Server	2
Embedded Build Versions	2
Standalone Build versions.....	2
MR Services User Authentication	3
Manual identification of new project in the Azure DevOps Organization.....	6
Email Monitor	9
Timing configurations in AppSetting file.....	11
Email Monitor Configuraion	12
Deploying Email Monitor	18

MR SERVICES (MR AGENT)

MR Services (formerly called MR Agent) is one of the components of Modern Requirements4DevOps that is automatically installed with the main application. It's a framework that provides extensibility to Azure DevOps using triggers.

Please note that MR Services are only accessible with AZURE DEVOPS Services using LIVE/PUBLIC IP to communicate with VSTS(Azure DevOps services). If any machine has no public access than VSTS Azure DevOps services could not be used (as they require public access to communicate with machine). Users are advised to contact their Network Administrators to change the value to the live IP address of their machines including the relevant port.

Currently, MR Services (MR Agent) has the following three sub components:

1. Custom ID
2. Dirty Flag
3. Email Monitor

Proper user authentication is required before any of these components are configured. The config files of any of the components won't work unless the relevant organization (in Azure DevOps) or collection (in TFS) is registered using authentication.

SUPPORTED AZURE DEVOPS(TFS) SERVER

EMBEDDED BUILD VERSIONS

Standard

TFS 2015 update 2+ and higher

Single Sign-On

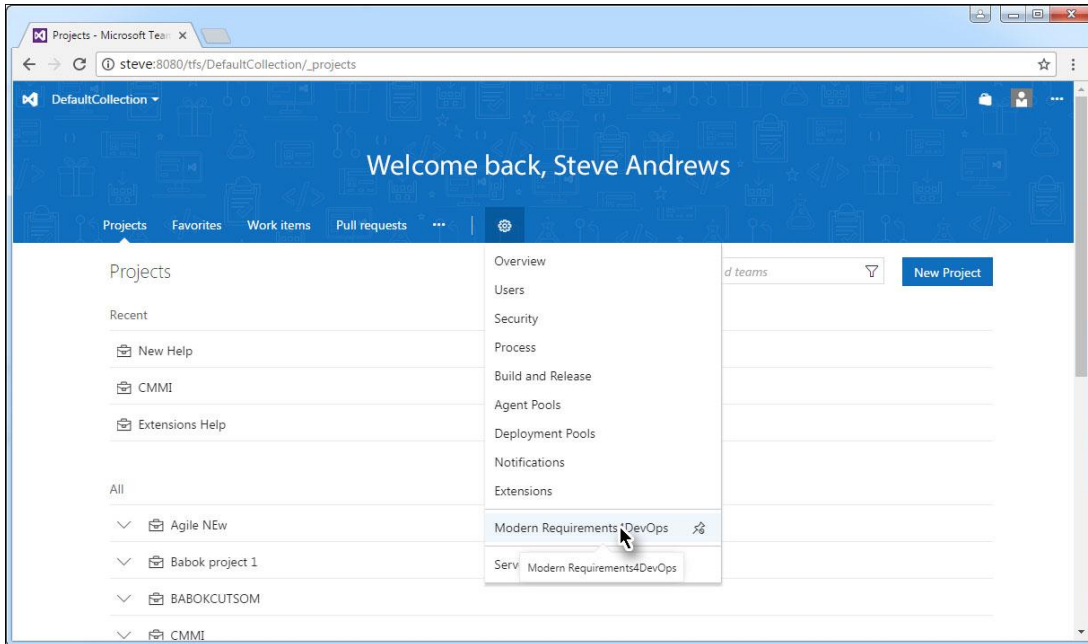
TFS 2017 update 3, TFS 2018 update 1 and higher

STANDALONE BUILD VERSIONS

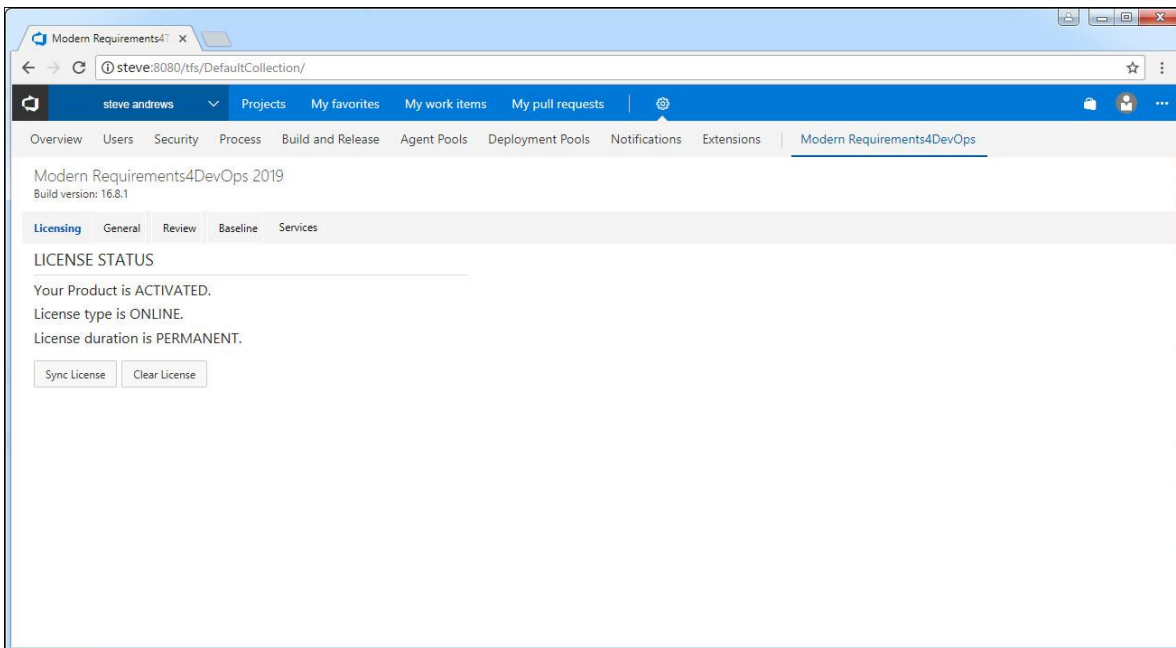
TFS 2013 and higher

MR SERVICES USER AUTHENTICATION

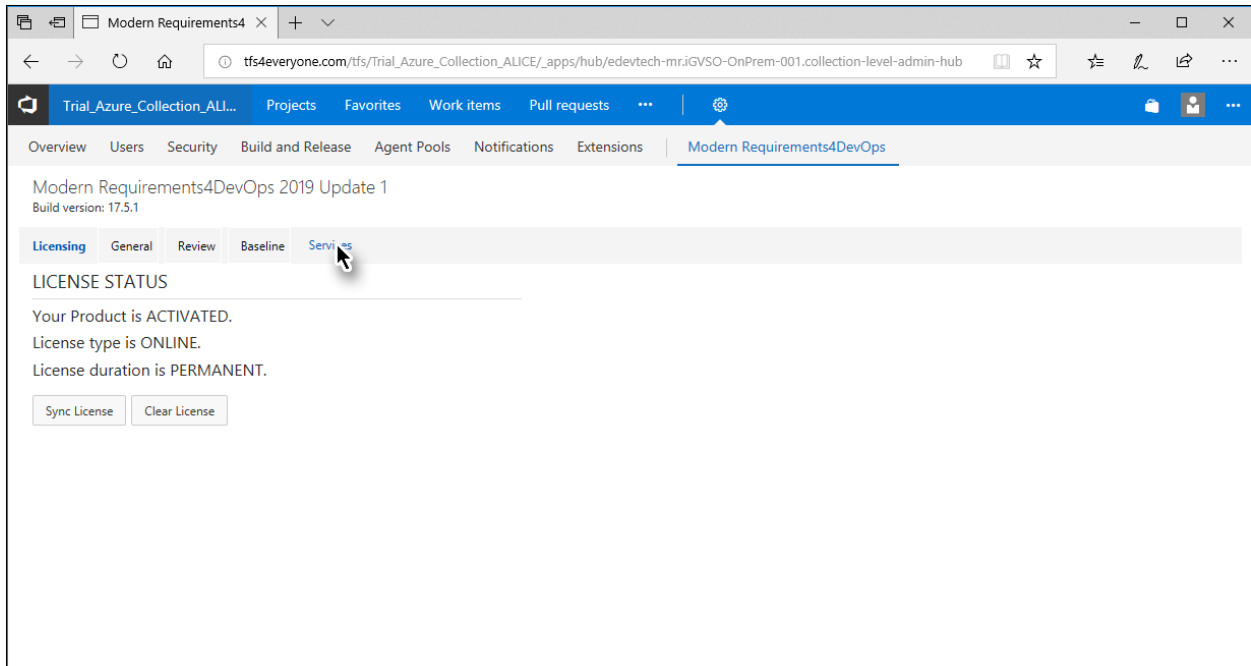
1. Launch the embedded version of the application and select the **Modern Requirements4DevOps** option under the **Settings** tab.



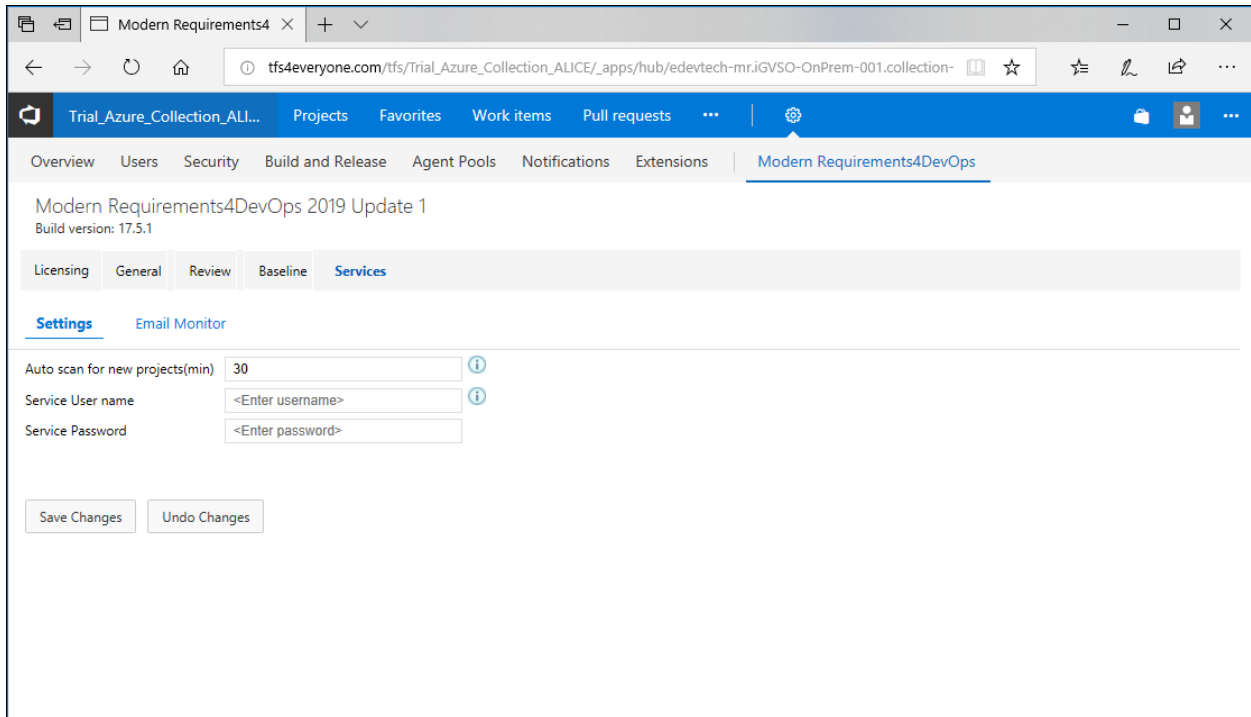
The Admin panel is displayed.



2. Click the **Services** Tab.



The options for Services Tab are displayed.



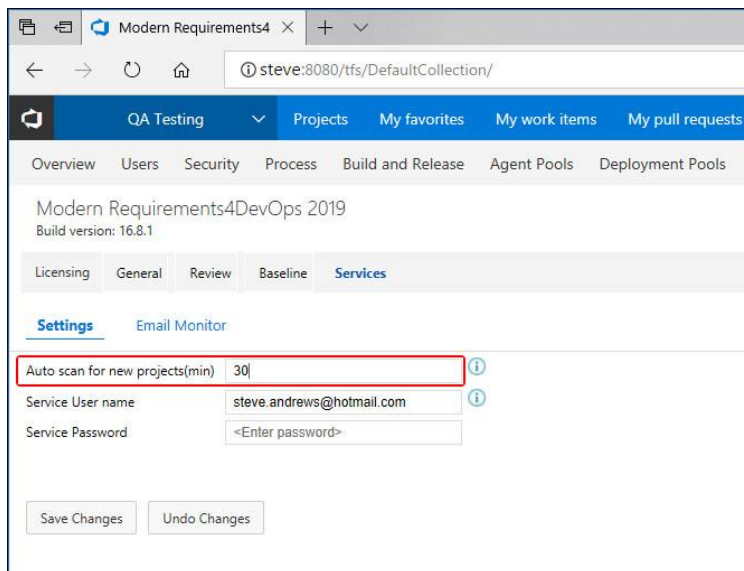
The **Settings** sub-tab deals with two options:

- Setting time interval to Scan the Azure DevOps organization (or TFS Collection) for new projects
- Registration of the current organization (Admin user credentials are required for this option)

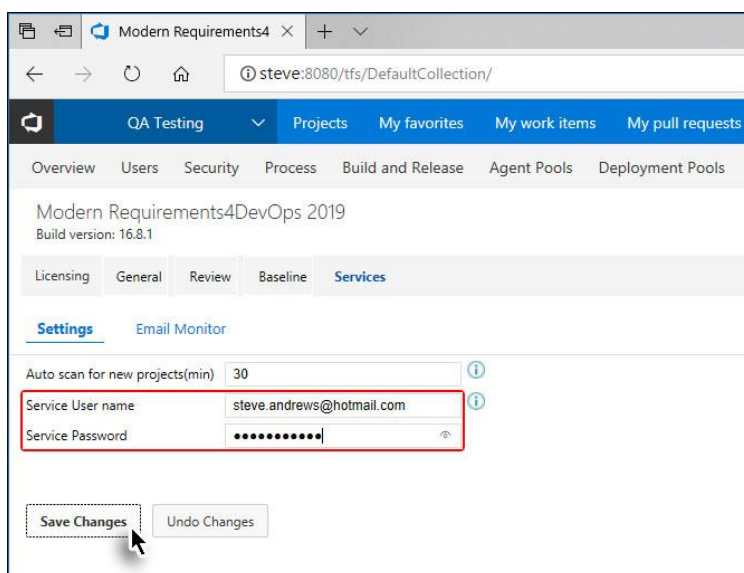
Note: Enter values for both of these settings in one go. Users can't choose to configure one setting while leave the other pending.

3. Enter time interval for Auto Scan (should be between 1 and 60).

This value determines the interval in minutes after which the registered Azure DevOps organization would be scanned for new projects.



4. Provide authorized login credentials. (with TFS admin rights)

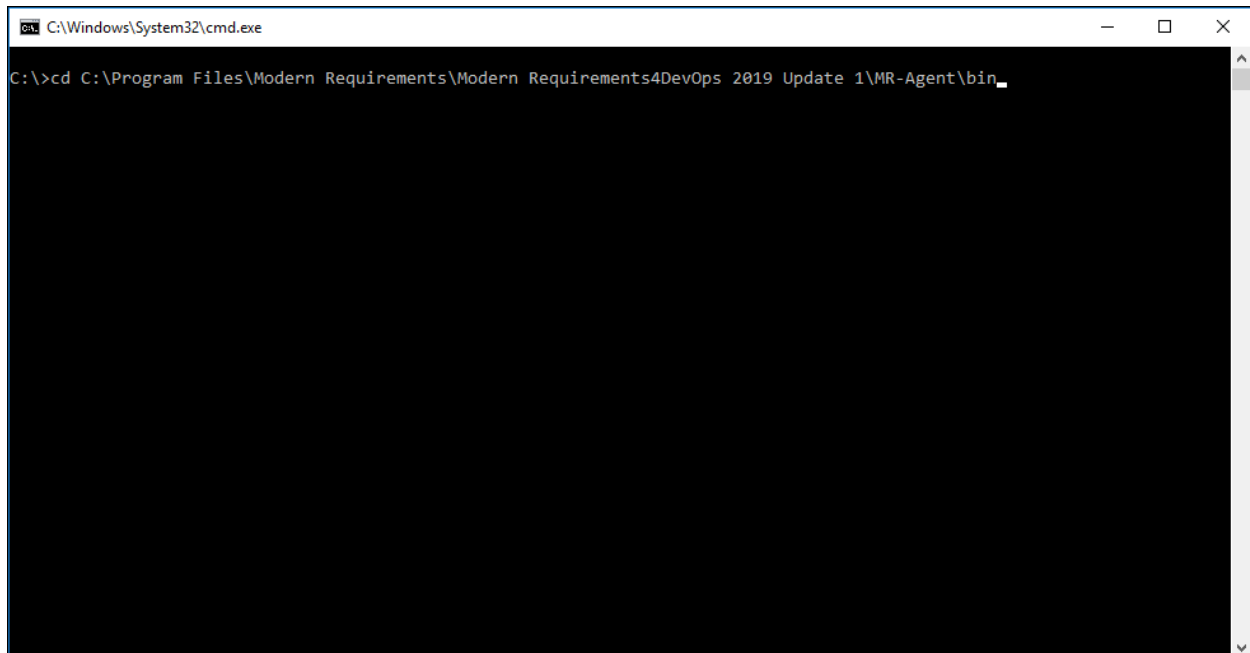


On successful authentication, current organization is registered and a confirmatory message is displayed.

MANUAL IDENTIFICATION OF NEW PROJECT IN THE AZURE DEVOPS ORGANIZATION

The above section elaborated the process to customize the automatic scanning time for the Azure DevOps Organization. The value shown in the above image means that the organization would be scanned every 30 minutes for the new project. However if the user has just created a new project and wants to work on it right away then he has to manually identify it (the project) in the Azure DevOps Organization. The following steps are required to do so:

1. Enter the following command on CMD: **cd <Installation Drive Letter>:\Program Files\Modern Requirements\MR-Agent\bin**



2. Once in the bin directory, enter the following command: **MRAgent**

A screenshot of a Windows command prompt window. The title bar reads "C:\Windows\System32\cmd.exe". The command prompt shows the current directory as "C:\Program Files\Modern Requirements\Modern Requirements4DevOps 2019 Update 1\MR-Agent\bin" and the command "MRAgent" has been entered, with a cursor at the end of the line.

The menu of options is displayed.

A screenshot of a Windows command prompt window. The title bar reads "C:\Windows\System32\cmd.exe - MRAgent". The command prompt shows the same directory as the previous screenshot. The command "MRAgent" has been executed, and a menu of options is displayed:

```

1 -> customid - Apply CustomId on existing WIs
2 -> emailmonitor - Email Registration
3 -> core - Collection Registration
4 -> core - Scan for new projects

Select Command _
  
```


3. Type 4 and click Enter.

```

C:\Windows\System32\cmd.exe - MRAgent

C:\Program Files\Modern Requirements\Modern Requirements4DevOps 2019 Update 1\MR-Agent\bin>MRAgent

1 -> customid - Apply CustomId on existing WIs
2 -> emailmonitor - Email Registration
3 -> core - Collection Registration
4 -> core - Scan for new projects

Select Command _ 4_
    
```

4. Enter the Azure DevOps organization value to scan for new projects.

```

C:\Windows\System32\cmd.exe - MRAgent

C:\Program Files\Modern Requirements\Modern Requirements4DevOps 2019 Update 1\MR-Agent\bin>MRAgent

1 -> customid - Apply CustomId on existing WIs
2 -> emailmonitor - Email Registration
3 -> core - Collection Registration
4 -> core - Scan for new projects

Select Command _ 4_

Enter Value For TFS Collection Url: http://steve:8080/tfs_
    
```

If no error message is displayed, the process has been successfully carried out for scanning new projects created after registering an Azure DevOps organization/or applying its config.

EMAIL MONITOR

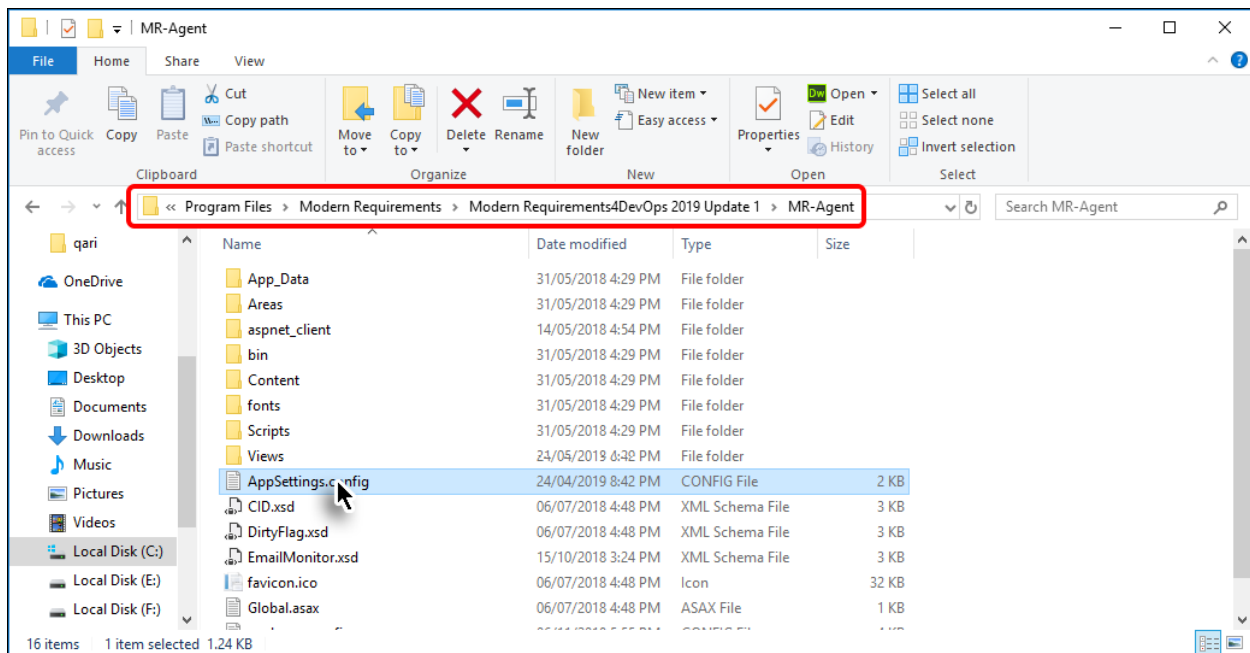
Email Monitor is a component of MR Services (MR Agent) that is used to automatically create work items from emails. A particular email address is configured for this purpose and on successful completion of the configuration process, all emails sent to this email address result in creating/updating work items. The process involves the following steps*:

- Configuring the Email Monitor Config file (placed at a particular location)
- Entering and verifying email settings

Each of these steps is elaborated further below.

*For local Azure DevOps (TFS) servers, MR Services (MR Agent) automatically adds the relevant location in the application settings file (**AppSettings.config**). However if Azure DevOps Services are involved then the user's machine should have a live IP address that the Azure DevOps Services can use to access/communicate. This IP address should be added in the **AppSettings config** file. The process to do it is elaborated in the following steps:

1. Go to the installation folder of MR Services (MR Agent) (highlighted in the image) and open the **AppSettings config** file in a text editor.



The **ApplicationURL** is automatically set toward the local machine.

```

AppSettings.config
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <appSettings>
3   <add key="eDev.MR.Agent.Enabled" value="true"/>
4   <add key="eDev.MR.CustomId.HangfireLogIsEnabled" value="false"/>
5   <add key="eDev.MR.CustomId.MessageQueue.Redis" value="localhost:6379,DefaultDatabase=9,Password=,ssl=false,abortConnect=false"/>
6   <add key="eDev.MR.CustomId.Cache.Redis" value="localhost:6379,DefaultDatabase=9,Password=,ssl=false,abortConnect=false"/>
7   <add key="eDev.MR.Agent.CustomId.Folder" value="CustomId"/>
8   <add key="eDev.MR.Agent.DirtyFlag.Folder" value="DirtyFlag"/>
9   <add key="eDev.MR.Agent.EmailMonitor.Folder" value="EmailMonitor"/>
10  <add key="eDev.MR.Agent.ComponentFolder" value="~/bin/Component"/>
11  <add key="eDev.MR.Agent.ConfigurationRoot" value="~/App_Data"/>
12  <add key="eDev.MR.Agent.ApplicationUrl" value="http://STEVE:8029/" />
13  <add key="eDev.MR.Agent.CacheStore" value="Redis" />
14  <add key="eDev.MR.Agent.DirtyFlag.TagWithSource" value="true"/>
15  <add key="eDev.MR.Agent.DirtyFlag.InsertComments" value="true"/>
16  <add key="eDev.MR.Agent.Core.SubscribeSchedule" value="30"/>
17  <add key="eDev.MR.Agent.CustomId.ApplyAllSchedule" value="30"/>
18  <add key="eDev.MR.Agent.EmailMonitor.EmailCheckSchedule" value="15"/>
19 </appSettings>
20
  
```

2. Change the value (for Azure DevOps Services only) to the live IP address of your machine including the relevant port.*

*Contact your Network Administrator to get the live IP address and port information

```

AppSettings.config
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <appSettings>
3   <add key="eDev.MR.Agent.Enabled" value="true"/>
4   <add key="eDev.MR.CustomId.HangfireLogIsEnabled" value="false"/>
5   <add key="eDev.MR.CustomId.MessageQueue.Redis" value="localhost:6379,DefaultDatabase=9,Password=,ssl=false,abortConnect=false"/>
6   <add key="eDev.MR.CustomId.Cache.Redis" value="localhost:6379,DefaultDatabase=9,Password=,ssl=false,abortConnect=false"/>
7   <add key="eDev.MR.Agent.CustomId.Folder" value="CustomId"/>
8   <add key="eDev.MR.Agent.DirtyFlag.Folder" value="DirtyFlag"/>
9   <add key="eDev.MR.Agent.EmailMonitor.Folder" value="EmailMonitor"/>
10  <add key="eDev.MR.Agent.ComponentFolder" value="~/bin/Component"/>
11  <add key="eDev.MR.Agent.ConfigurationRoot" value="~/App_Data"/>
12  <add key="eDev.MR.Agent.ApplicationUrl" value="http://202.88.202.88:8029/" />
13  <add key="eDev.MR.Agent.CacheStore" value="Redis" />
14  <add key="eDev.MR.Agent.DirtyFlag.TagWithSource" value="true"/>
15  <add key="eDev.MR.Agent.DirtyFlag.InsertComments" value="true"/>
16  <add key="eDev.MR.Agent.Core.SubscribeSchedule" value="30"/>
17  <add key="eDev.MR.Agent.CustomId.ApplyAllSchedule" value="30"/>
18  <add key="eDev.MR.Agent.EmailMonitor.EmailCheckSchedule" value="15"/>
19 </appSettings>
20
  
```

3. Save and close the config file.

TIMING CONFIGURATIONS IN APPSETTING FILE

At the bottom of the **AppSettings** config file, there are three timing configurations available for users.

```
<add key="eDev.MR.Agent.ApplicationUrl" value="http://202.88.202.88:8029/" />
<add key="eDev.MR.Agent.CacheStore" value="Redis" />
<add key="eDev.MR.Agent.DirtyFlag.TagWithSource" value="true" />
<add key="eDev.MR.Agent.DirtyFlag.InsertComments" value="true" />
<add key="eDev.MR.Agent.Core.SubscribeSchedule" value="30" />
<add key="eDev.MR.Agent.CustomId.ApplyAllSchedule" value="30" />
<add key="eDev.MR.Agent.EmailMonitor.EmailCheckSchedule" value="15" />
</appSettings>
```

SUBSCRIBE SCHEDULE

1. Works for all components of MR Services
2. Used to check new projects/collections
3. Default value "30"* represents the number of minutes, after which MR Services (MR Agent) scans for new projects. Users can configure the value (in minutes) as per their requirements.

*This value can also be configured using Admin Panel.

APPPLYALL SCHEDULE

1. Works only for Custom ID
2. Used to apply Custom ID on newly created work items
3. Default value "30" represents the number of minutes, after which MR Services (MR Agent) scans for new work items and apply Custom IDs on them. Users can configure the value (in minutes) as per their requirements.

EMAILCHECK SCHEDULE

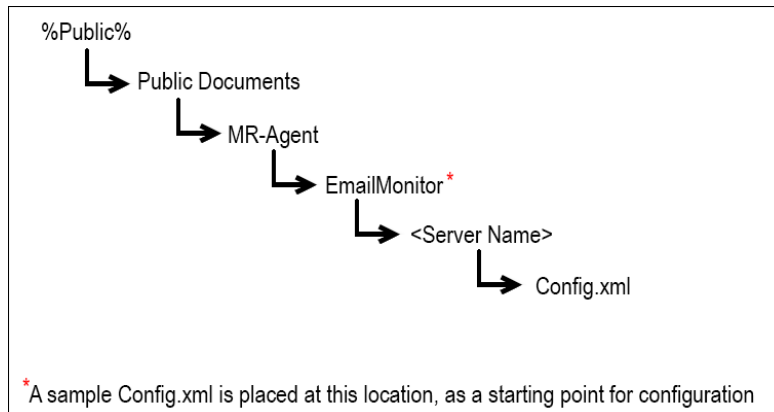
1. Works only for Email Monitor
2. Used to check if a new email has been received from which work items could be created/updated
3. Default value "15" represents the number of minutes, after which MR Services (MR Agent) scans for email. Users can configure the value (in minutes) as per their requirements.

EMAIL MONITOR CONFIGURAION

In order to make the Email Monitor work properly, users must manually create the following items:

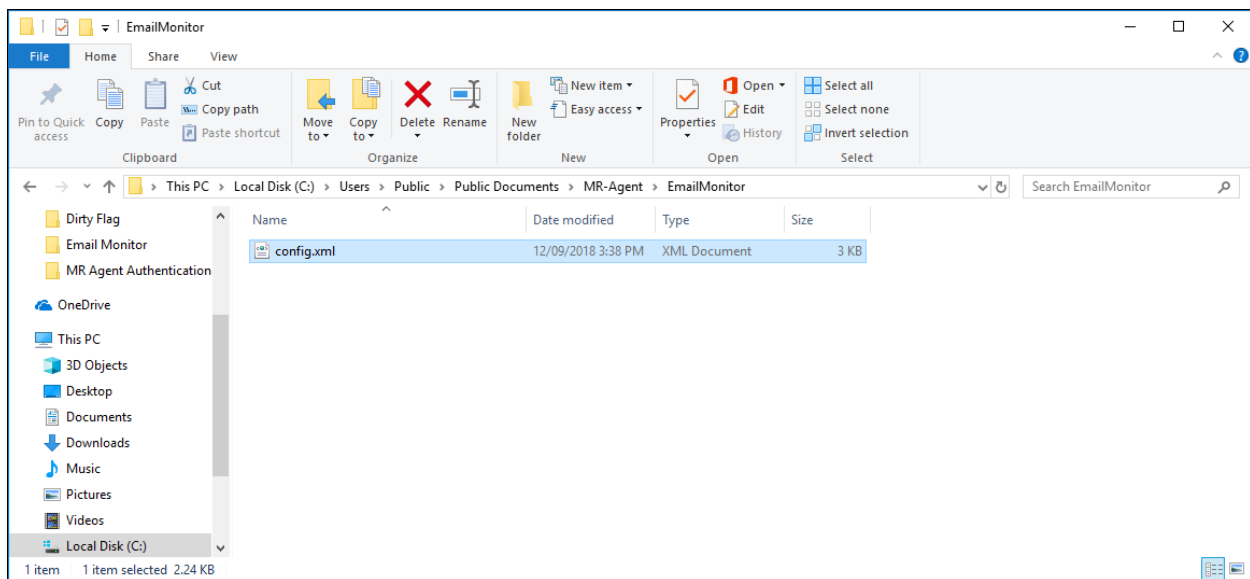
1. A folder that is named after the Azure DevOps (TFS) server name (on which the Email Monitor is required to apply).

The relevant server folder should also include the **config.xml** file containing all configuration. The file and folder hierarchy should appear as shown below using the text pattern and relevant image:

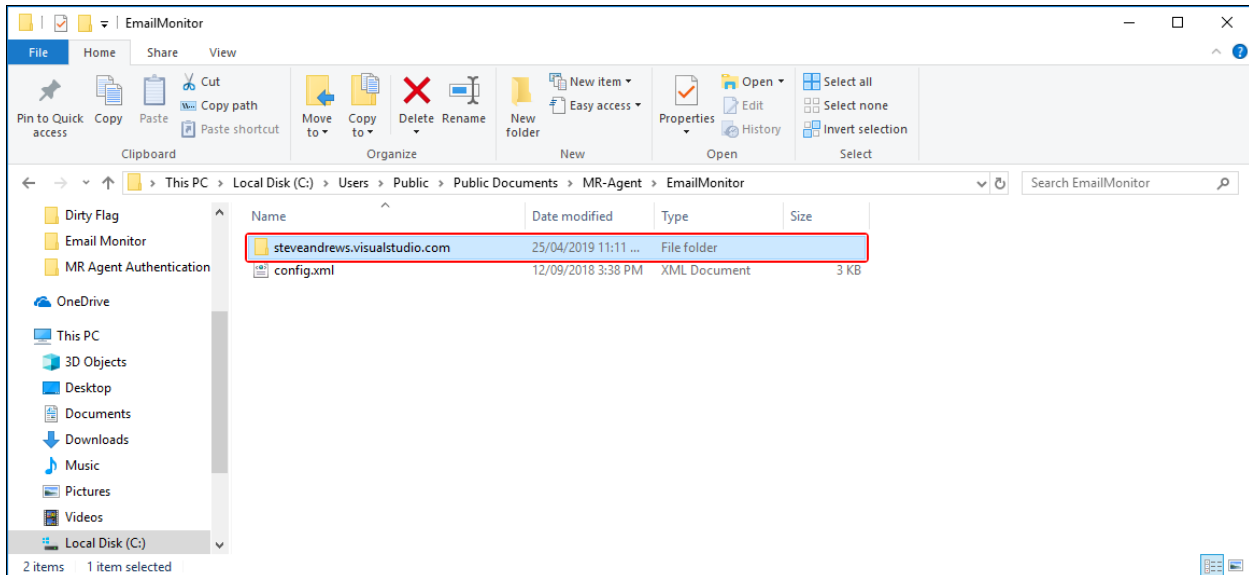


*** Note:** for the current versions of Email Monitor, the hierarchy stops at the server folder, and placing the file in that server folder. However for the future versions, the hierarchy would go up to the organization folder (like other components of MR Services (MR Agent)). Please consult with your administrator or contact Modern Requirements if any uncertainty persists in this matter.

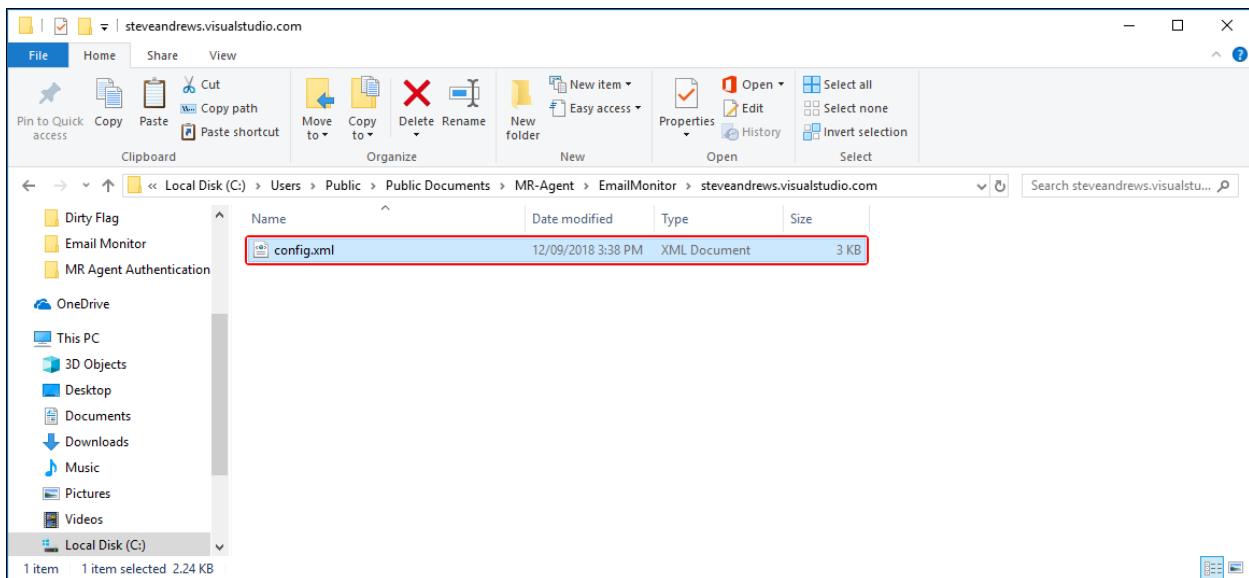
As described in the image above, a sample **Config.xml** file is placed in the **EmailMonitor** folder.



1. Create a folder named after the Azure DevOps (TFS) Server name (on which component is required to apply) at this location.



2. Enter the newly created folder and copy the **Config.xml** file (discussed earlier) into that folder i.e. Folder with Azure DevOps (TFS) Server name.



This file contains the blueprint for the desired configuration.

CONFIGURING THE EMAIL MONITOR XML FILE

1. Open the **config.xml** file in Notepad or any text editor.

```

config - Notepad
File Edit Format View Help
<?xml version="1.0" encoding="UTF-8"?>
<Root>
  <ServerUrl>http://tfsserver:port/tfs</ServerUrl>
  <Collection Url="http://tfsserver:port/tfs/Collection" Default="true">
    <DefaultProject>Cmmi</DefaultProject>
  </Collection>
  <AdminEmail>admin@orgemail.com</AdminEmail>
  <Email Address="tfs@orgemail.com">
    <WorkItemType CategoryReference="Microsoft.RequirementCategory">Requirement, User Story, Product Backlog Item</WorkItemType>
    <Field FieldReference="System.Description" OnCreate="true" OnUpdate="false">[SenderName]&lt;[SenderEmail]&gt;[NewLine][EmailBody]</Field>
    <Field FieldReference="System.History">[SenderName]&lt;[SenderEmail]&gt;[NewLine][EmailBody]</Field>
  </Email>
</Root>

```

2. Define the value of the **ServerURL** tag as per requirement, for example:

```

config - Notepad
File Edit Format View Help
<?xml version="1.0" encoding="UTF-8"?>
<Root>
  <ServerUrl>http://steve:8080/tfs</ServerUrl>
  <Collection Url="http://tfsserver:port/tfs/Collection" Default="true">
    <DefaultProject>Cmmi</DefaultProject>
  </Collection>
  <AdminEmail>admin@orgemail.com</AdminEmail>

```

3. Similarly define the value for **Collection Url** (including the **DefaultProject**)

```

config - Notepad
File Edit Format View Help
<?xml version="1.0" encoding="UTF-8"?>
<Root>
  <ServerUrl>https://steveandrews.visualstudio.com/tfs</ServerUrl>
  <Collection Url="http://steve:8080/tfs/steveandrews" Default="true">
    <DefaultProject>DefaultCmmiProject</DefaultProject>
  </Collection>
  <AdminEmail>admin@orgemail.com</AdminEmail>
  <Email Address="tfs@orgemail.com">

```

- The values for both the **ServerURL** and **Collection Url** should correspond to the folder structure described before.
- Make sure that the URL does not end with a forward slash “/”
- User can define multiple Collection URLs in the config file.

4. Provide the value for **AdminEmail**. This email address is used as a mitigation, in case the desired functionality can't be achieved using the address defined in **Email** tag (explained in the next step).

```

config - Notepad
File Edit Format View Help
<?xml version="1.0" encoding="UTF-8"?>
<Root>
  <ServerUrl>https://steveandrews.visualstudio.com/tfs</ServerUrl>
  <Collection Url="https://steveandrews.visualstudio.com/tfs/steve
    <DefaultProject>DefaultCmmiProject</DefaultProject>
  </Collection>

  <AdminEmail>admin@steveandrews.com</AdminEmail>

  <Email Address="tfs@orgemail.com">
    <WorkItemType CategoryReference="Microsoft.RequirementCa
  
```

Note: Only a single Admin email can exist in the config file.

5. **"Email"** tag is the main tag in this file that defines where the email would be sent. The emails sent to this address would be used to create desired types of work items. Configure the **Email** tag as required
 - e. **Email:** Provide the target email address where the email should be delivered for work items creation/update. In case some criteria does not match with the desired values, the warning email would be sent at the **Admin Email** defined above.

```

config1 - Notepad
File Edit Format View Help
<?xml version="1.0" encoding="UTF-8"?>
<Root>
  <ServerUrl>http://steve:8080/tfs</ServerUrl>
  <Collection Url="http://steve:8080/tfs/steveandrews" Def
    <DefaultProject>DefaultCmmiProject</DefaultProje
  </Collection>
  <AdminEmail>admin@steveandrews.com</AdminEmail>

  <Email Address="tfs@steveandrews.com">

  <WorkItemType CategoryReference="Microsoft.Requi
  
```

Note: Multiple email addresses can be defined in the config file.

- f. **Work Item Type:** Set the desired type of work item to be created. In the following example **CategoryReference** represents the internal category type of work items. Multiple values in this tag means that the relevant type of work item would be created depending upon the template of team project. e.g. If the team project is using CMMI template then the email would create a Requirement work item. Similarly, for Agile based project a User Story would be created and for a Scrum based project a Project Backlog Item would be created.


```

config - Notepad
File Edit Format View Help
<?xml version="1.0" encoding="UTF-8"?>
<Root>
  <ServerUrl>http://steve:8080/tfs</ServerUrl>
  <Collection Url="http://steve:8080/tfs/steveandrews" Default="true">
    <DefaultProject>DefaultCmmiProject</DefaultProject>
  </Collection>
  <AdminEmail>admin@steveandrews.com</AdminEmail>
  <Email Address="tfs@steveandrews.com">

    <WorkItemType CategoryReference="Microsoft.RequirementCategory">Requirement, User Story, Product Backlog Item</WorkItemType>

  <Field FieldReference="System.Description" OnCreate="true" OnUpdate="false">[SenderName]&lt;[SenderEmail]&gt;[NewLine][EmailBody]</Field>
  <Field FieldReference="System.History">[SenderName]&lt;[SenderEmail]&gt;[NewLine][EmailBody]</Field>
</Email>
</Root>

```

Denotes the category that would be used to create the work item

Denotes the actual work item type that would be created from emails sent to the designated address

- c. **FieldReference="System.Title"**: Tells what would be the title of the work Item to be created. The following example shows that the subject of the email would become the title of the work item.
 - i. **OnCreate="true"** means that the title would be set from the email's subject only for new Work Items.
 - ii. **OnUpdate="false"** means that for existing Work Items, the Title field would not be updated.

```

config - Notepad
File Edit Format View Help
<?xml version="1.0" encoding="UTF-8"?>
<Root>
  <ServerUrl>http://steve:8080/tfs</ServerUrl>
  <Collection Url="http://steve:8080/tfs/steveandrews" Default="true">
    <DefaultProject>DefaultCmmiProject</DefaultProject>
  </Collection>
  <AdminEmail>admin@steveandrews.com</AdminEmail>
  <Email Address="tfs@steveandrews.com">
    <WorkItemType CategoryReference="Microsoft.RequirementCategory">Requirement, User Story, Product Backlog Item</WorkItemType>
    <Field FieldReference="System.Title" OnCreate="true" OnUpdate="false">[Subject]</Field>
    <Field FieldReference="System.Description" OnCreate="true" OnUpdate="false">[SenderName]&lt;[SenderEmail]&gt;[NewLine][EmailBody]</Field>
    <Field FieldReference="System.History">[SenderName]&lt;[SenderEmail]&gt;[NewLine][EmailBody]</Field>
  </Email>

```

- d. **FieldReference="System.Description"**: Tell where to put the information from the incoming emails (i.e. in which property/field of the work item). The following example uses the **Description** field for this purpose.
 - i. **OnCreate="true"** means that for new Work Items, the content of the email (described next) would be used to populate the Description field of the work item.
 - ii. **OnUpdate="false"** means that for existing Work Items, the Description field would not be overwritten. Instead the update would go to the comments/history section of the work item.

The later part of this field tells the composition of the description field. The following information would compose the Description field:

- i. Sender Name shown inside <> e.g. <Alice Dukas>
- ii. Sender Email also shown inside <> e.g. <alice.ducas@steveandrews.com>
- iii. Email body added in a new line

```

config - Notepad
File Edit Format View Help
<?xml version="1.0" encoding="UTF-8"?>
<Root>
  <ServerUrl>https://steveandrews.visualstudio.com/tfs</ServerUrl>
  <Collection Url="https://steveandrews.visualstudio.com/tfs/steveandrews" Default="true">
    <DefaultProject>DefaultCmmiProject</DefaultProject>
  </Collection>
  <AdminEmail>admin@steveandrews.com</AdminEmail>

  <Email Address="steveandrews@hotmail.com">
    <WorkItemType CategoryReference="Microsoft.RequirementCategory">Requirement, User Story, Product Backlog Item</WorkItemType>
    <Field FieldReference="System.Title" OnCreate="true" OnUpdate="false">[Subject]</Field>
    <Field FieldReference="System.Description" OnCreate="true" OnUpdate="false">[SenderName]&lt;[SenderEmail]&gt;[NewLine][EmailBody]</Field>
    <Field FieldReference="System.History">[SenderName]&lt;[SenderEmail]&gt;[NewLine][EmailBody]</Field>
  </Email>
</Root>

```

- g. The final **FieldReference="System.History"** is used for discussion emails that come after a work item has been created. Instead of overwriting the **Description** field, the subsequent email information is stored in the **Comments** field (internally called **History**). The composition of the **History** field is more or less the same as of **Description** field discussed above. Users are advised to keep the original settings for this tag.

```

config - Notepad
File Edit Format View Help
<?xml version="1.0" encoding="UTF-8"?>
<Root>
  <ServerUrl>https://steveandrews.visualstudio.com/tfs</ServerUrl>
  <Collection Url="https://steveandrews.visualstudio.com/tfs/steveandrews" Default="true">
    <DefaultProject>DefaultCmmiProject</DefaultProject>
  </Collection>
  <AdminEmail>admin@steveandrews.com</AdminEmail>

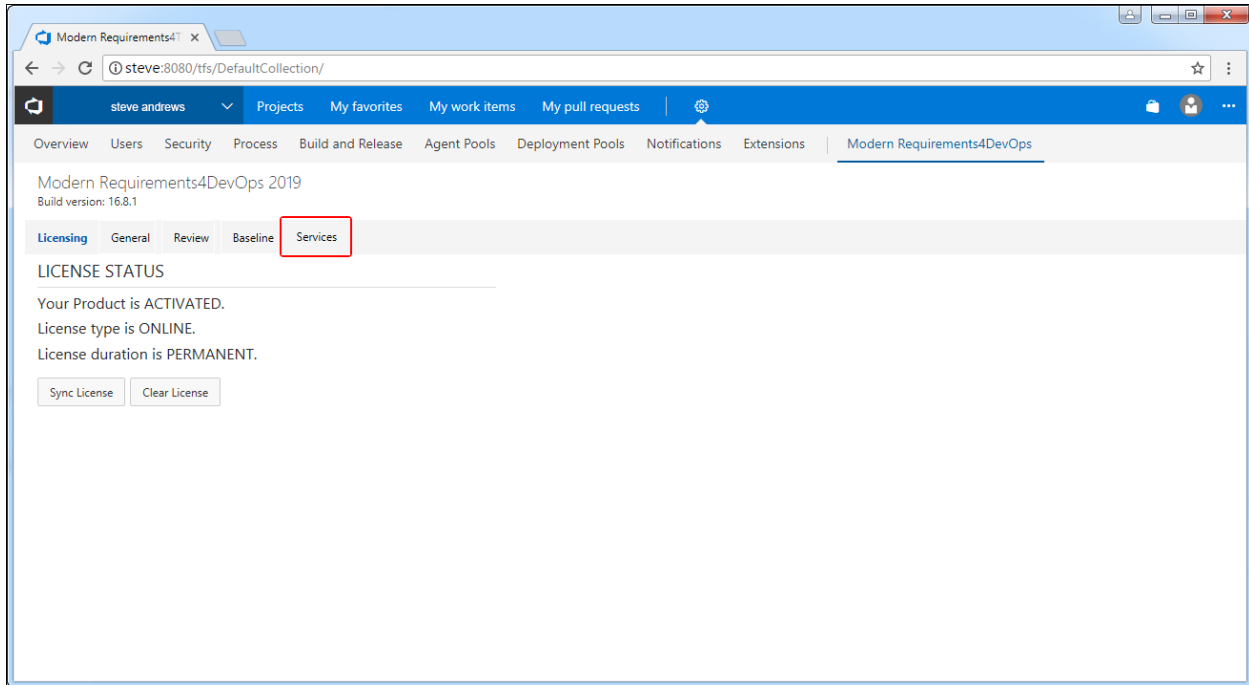
  <Email Address="steveandrews@hotmail.com">
    <WorkItemType CategoryReference="Microsoft.RequirementCategory">Requirement, User Story, Product Backlog Item
    <Field FieldReference="System.Title" OnCreate="true" OnUpdate="false">[Subject]</Field>
    <Field FieldReference="System.Description" OnCreate="true" OnUpdate="false">[SenderName]&lt;[SenderEmail]&gt;
    <Field FieldReference="System.History">[SenderName]&lt;[SenderEmail]&gt;[NewLine][EmailBody]</Field>
  </Email>
</Root>

```

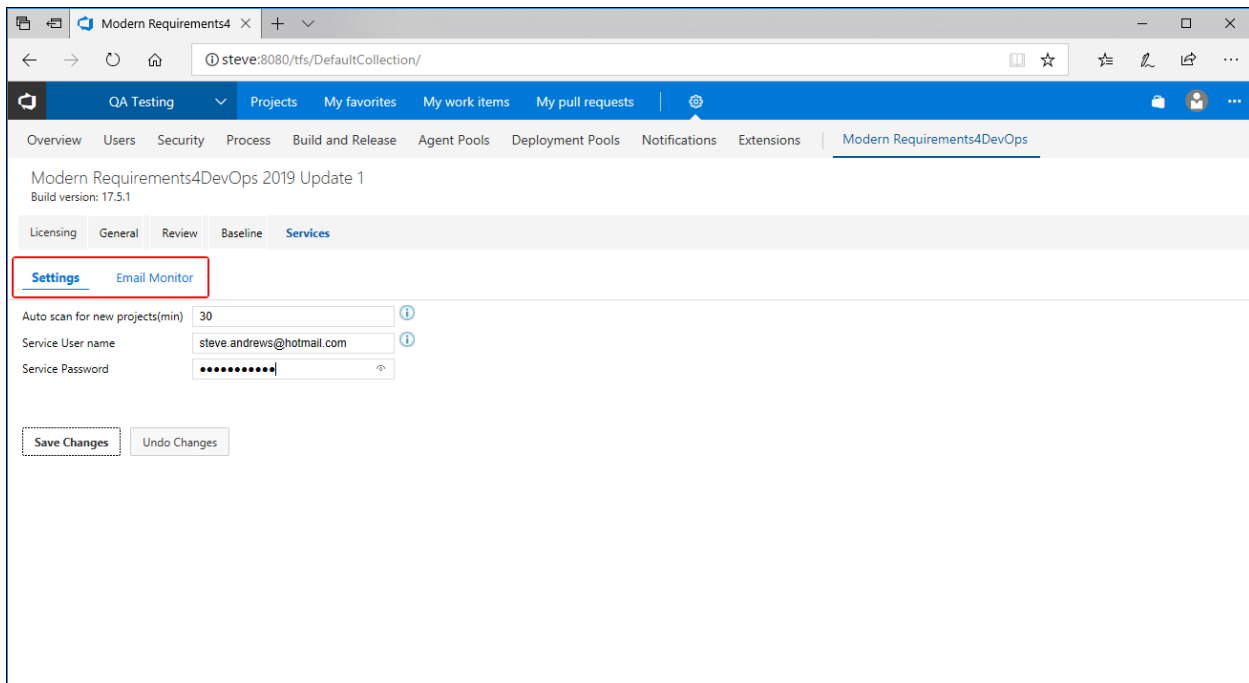
- 2. Following the successful completion of the configuration file, save and close the file.

DEPLOYING EMAIL MONITOR

Email Monitor can be deployed by configuring relevant settings under the Admin Panel. These settings can be accessed using the Services Tab.



The Services section of the Admin panel currently has two Tabs: **Settings & Email Monitor**



Settings Tab deals with 1) User Authentication/Organization Registration 2) Scanning for new projects.

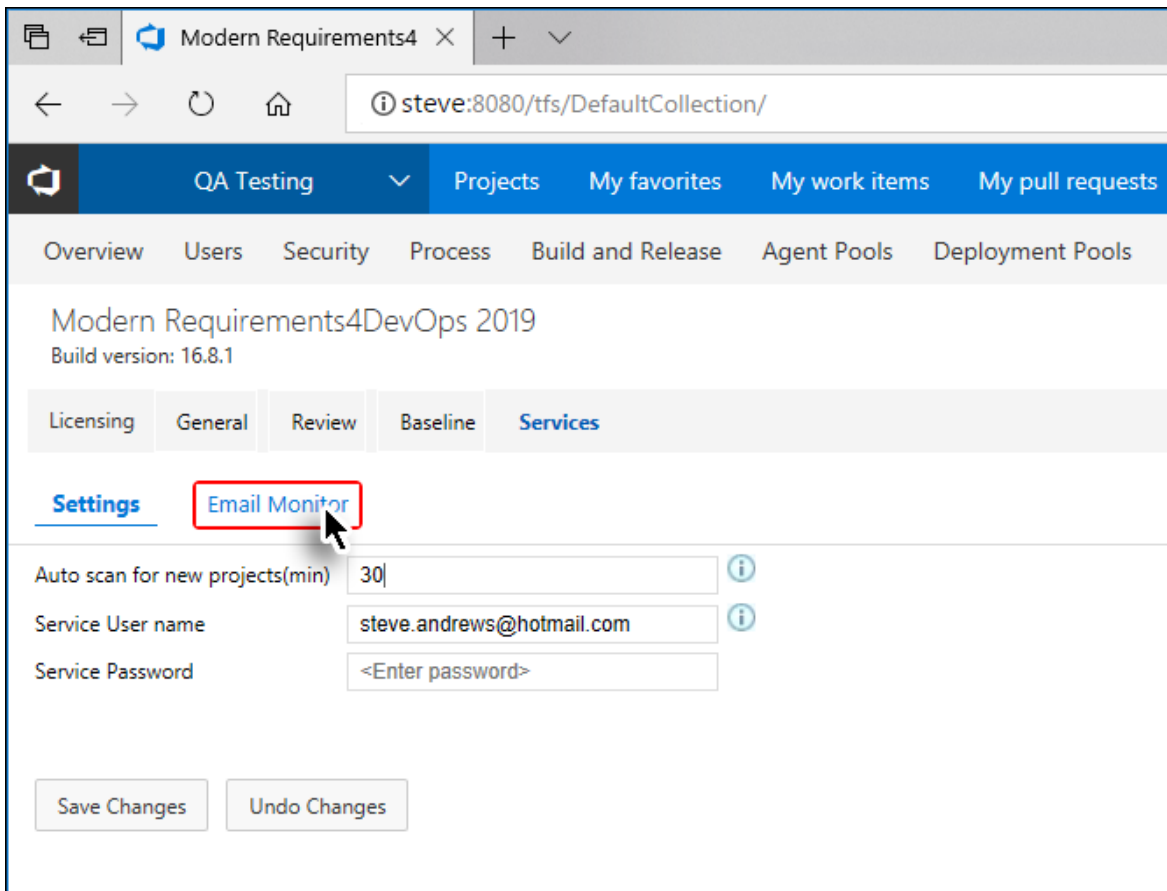
Email Monitor deals with all the email related options discussed earlier in the command line section.

ACCESSING ADMIN PANEL (AND THE SERVICES TAB)

- Refer to MR Services User Authentication for accessing Admin Panel and Settings sub-tab options.

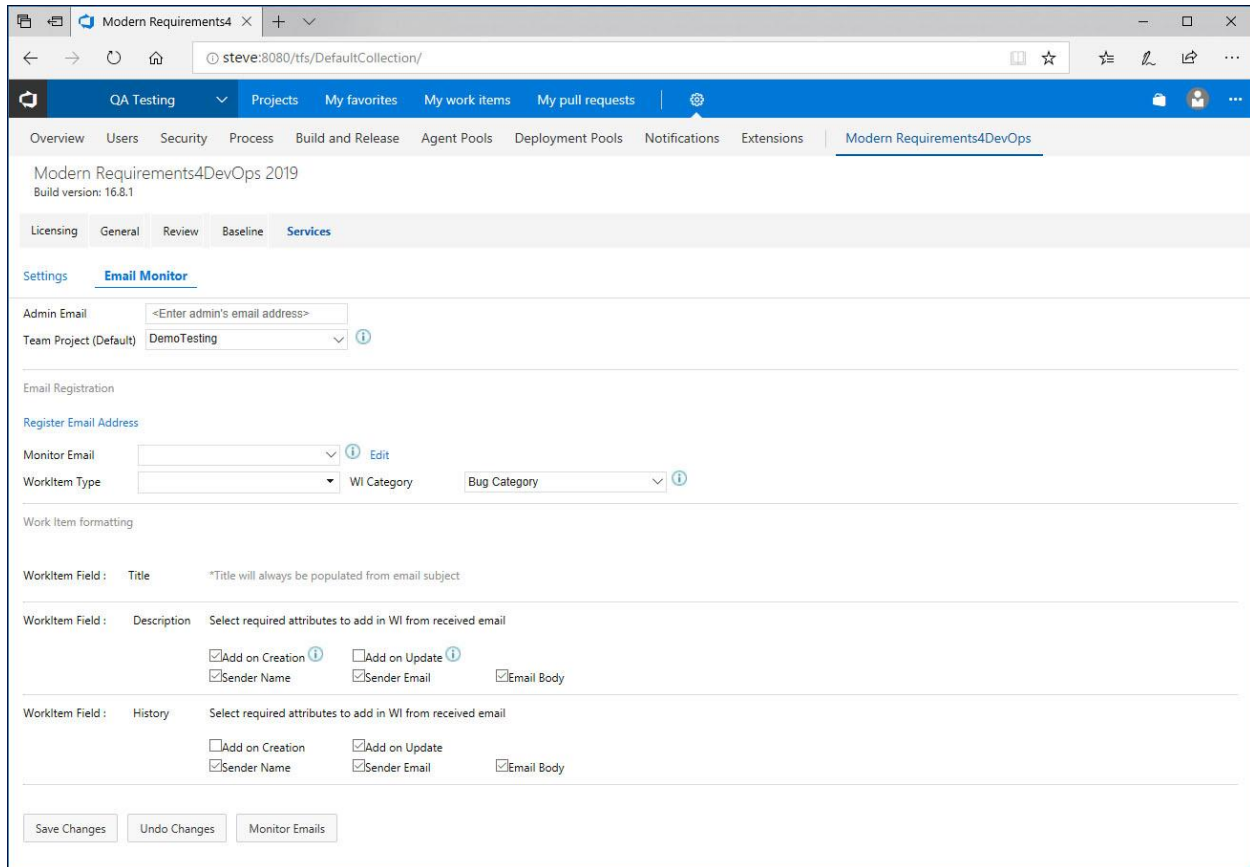
CONFIGURING EMAIL MONITOR OPTIONS

- Email Monitor tab under the Services section is used to configure email settings.
- The options can be accessed by clicking the Email Monitor tab as shown in the following image.

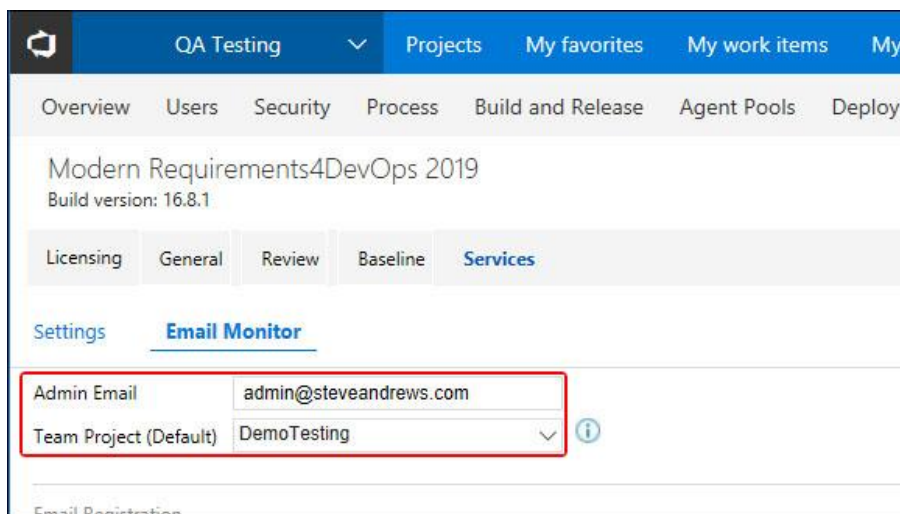


- If the user hasn't registered their organization (by providing required details in the Settings sub-tab), then upon clicking the Email Monitor sub-tab, the user is sent back to the Settings sub-tab, unless the desired information is entered.

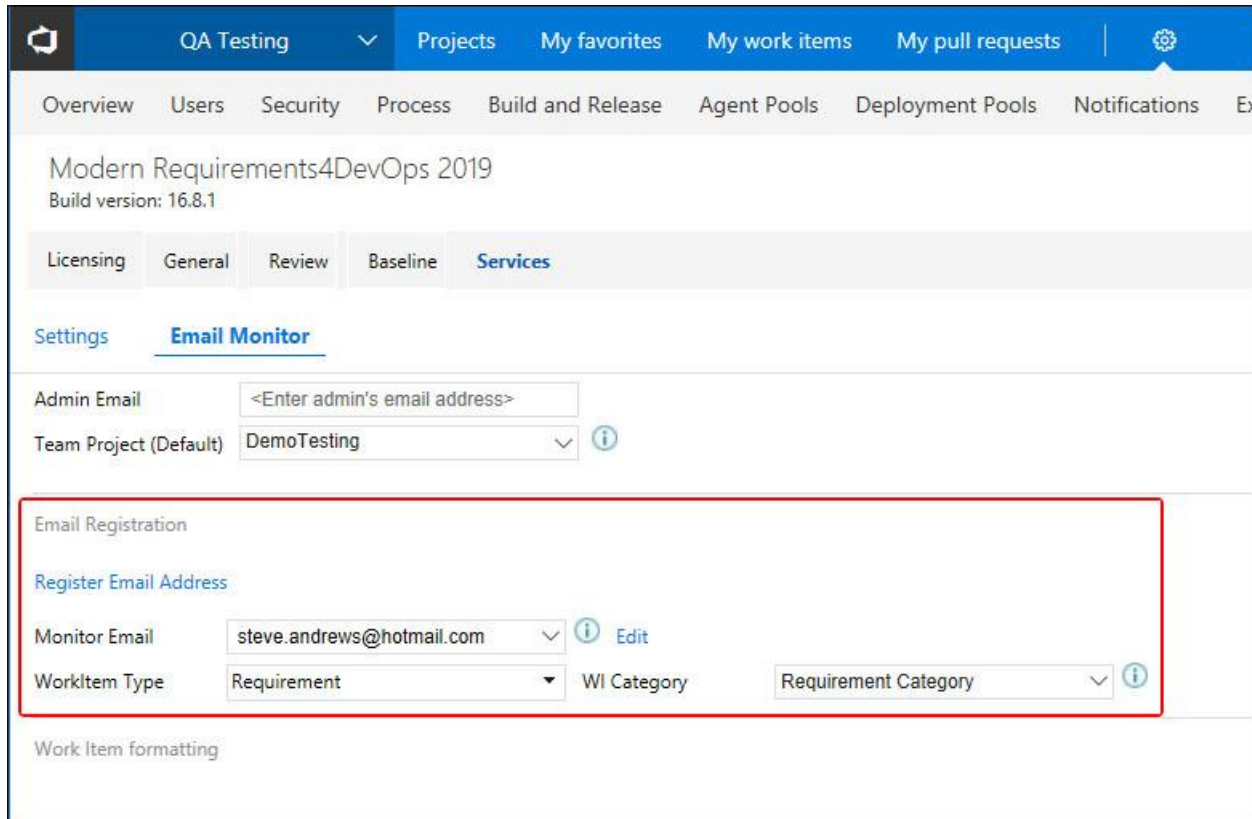
- The Email Monitor settings are divided into sections, where each section is used to configure a particular setting.



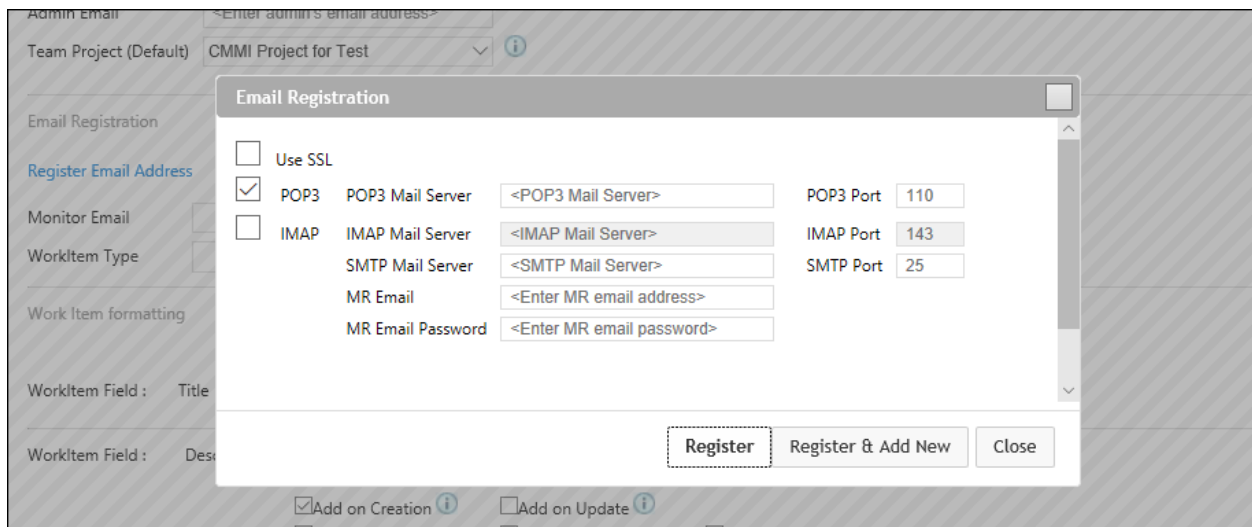
- All necessary settings are configured once. Users can not configure certain settings and leave other pending.
- The first section is used to configure the default project and the admin email address.



- The second section is used to configure the email address that would be used for Email Monitoring.



Clicking on **Register Email Address** would open a popup window where the network settings for the email (e.g. SSL, POP3, IMAP etc) can be configured.



- The third section is used for settings (that will be used to) to extract Work Item content from the emails sent to the Registered Email Address.

QA Testing Projects My favorites My work items My pull requests

Overview Users Security Process Build and Release Agent Pools Deployment Pools Notifications

Modern Requirements4DevOps 2019
Build version: 16.8.1

Licensing General Review Baseline **Services**

Settings **Email Monitor**

Admin Email

Team Project (Default)

Email Registration

[Register Email Address](#)

Monitor Email [Edit](#)

WorkItem Type WI Category

Work Item formatting

WorkItem Field :	Title	*Title will always be populated from email subject		
WorkItem Field :	Description	Select required attributes to add in WI from received email		
		<input checked="" type="checkbox"/> Add on Creation	<input type="checkbox"/> Add on Update	
		<input checked="" type="checkbox"/> Sender Name	<input checked="" type="checkbox"/> Sender Email	<input checked="" type="checkbox"/> Email Body
WorkItem Field :	History	Select required attributes to add in WI from received email		
		<input type="checkbox"/> Add on Creation	<input checked="" type="checkbox"/> Add on Update	
		<input checked="" type="checkbox"/> Sender Name	<input checked="" type="checkbox"/> Sender Email	<input checked="" type="checkbox"/> Email Body

Clicking on **Save Changes** button after configuring all the settings would deploy the Email Monitor.