



# Modern Requirements **4** DevOps

## MR4DevOps Services

Technote

Version 6.0.0



**TABLE OF CONTENTS**

**MR AGENT/SERVICES.....2**

    Configuring AppSettings config file.....2

        Timing Configurations in AppSettings File .....4

    MR Services Collection / Organization Registration .....5

        For Azure DevOps .....5

        For Embedded version .....7

    Configuring general settings in the Settings tab.....9

    Custom ID Configuration .....11

    Suspect Links Configuration .....13

    Email Monitor.....15

        Configuring Email Monitor.....15

    MR MATCAL .....21

        Configuring the MATCAL XML File .....23

        MATCAL expression calculation examples .....24

## MR AGENT/SERVICES

Modern Requirements4DevOps 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:** 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 four sub components:

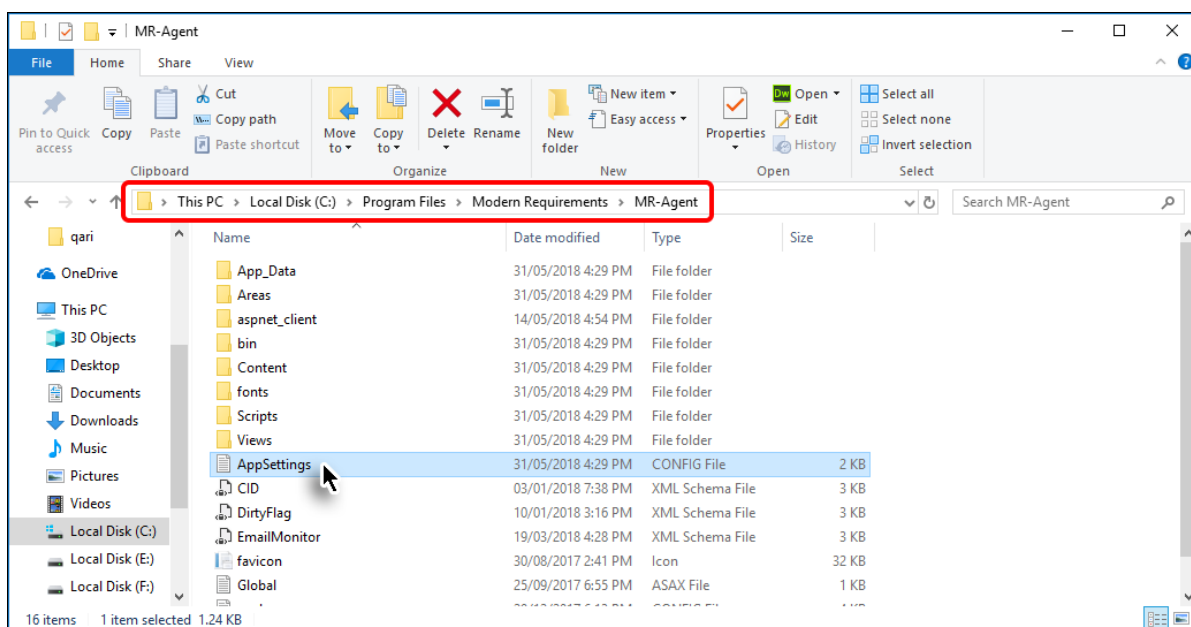
1. Custom ID
2. Suspect Links (formerly called Dirty Flag)
3. Email Monitor
4. MatCal

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.

### CONFIGURING APPSETTINGS CONFIG FILE

For local Azure DevOps servers (TFS-OnPrem), 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>
```

### APPPLYALLSCHEDULE

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.

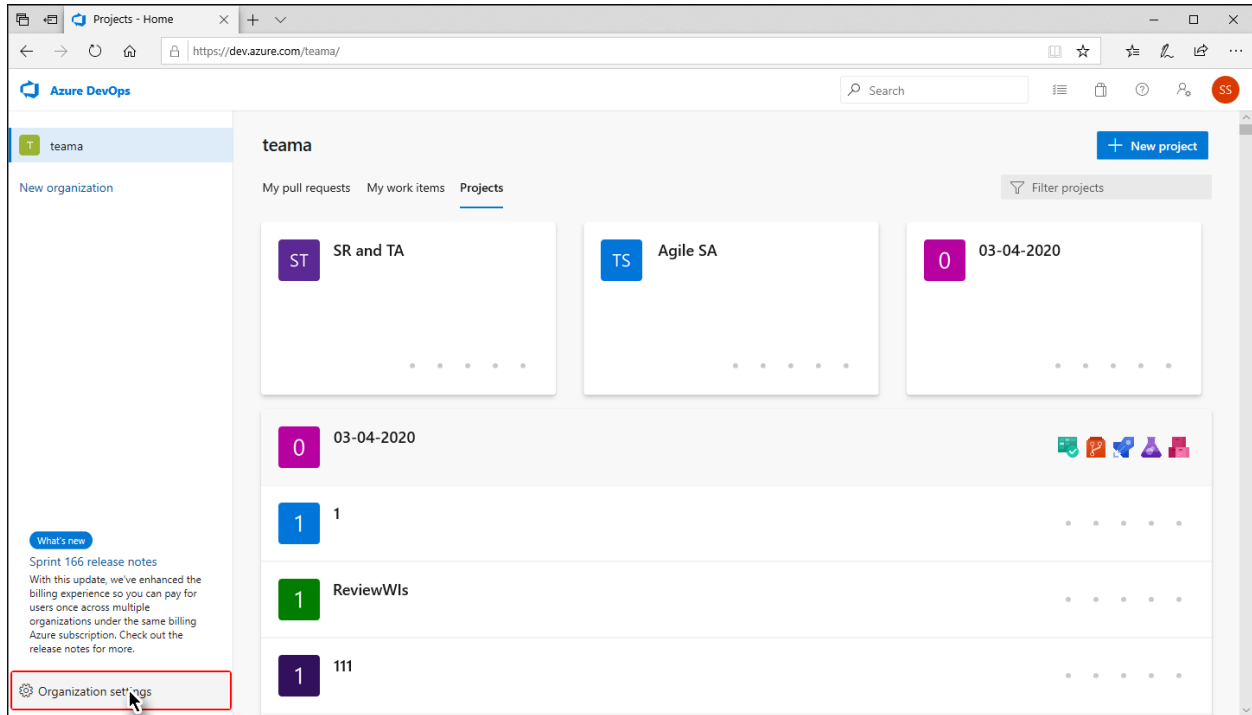
### EMAILCHECKSCHEDULE

1. Works only for Email Monitor
2. Used to check if a new email has arrived 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.

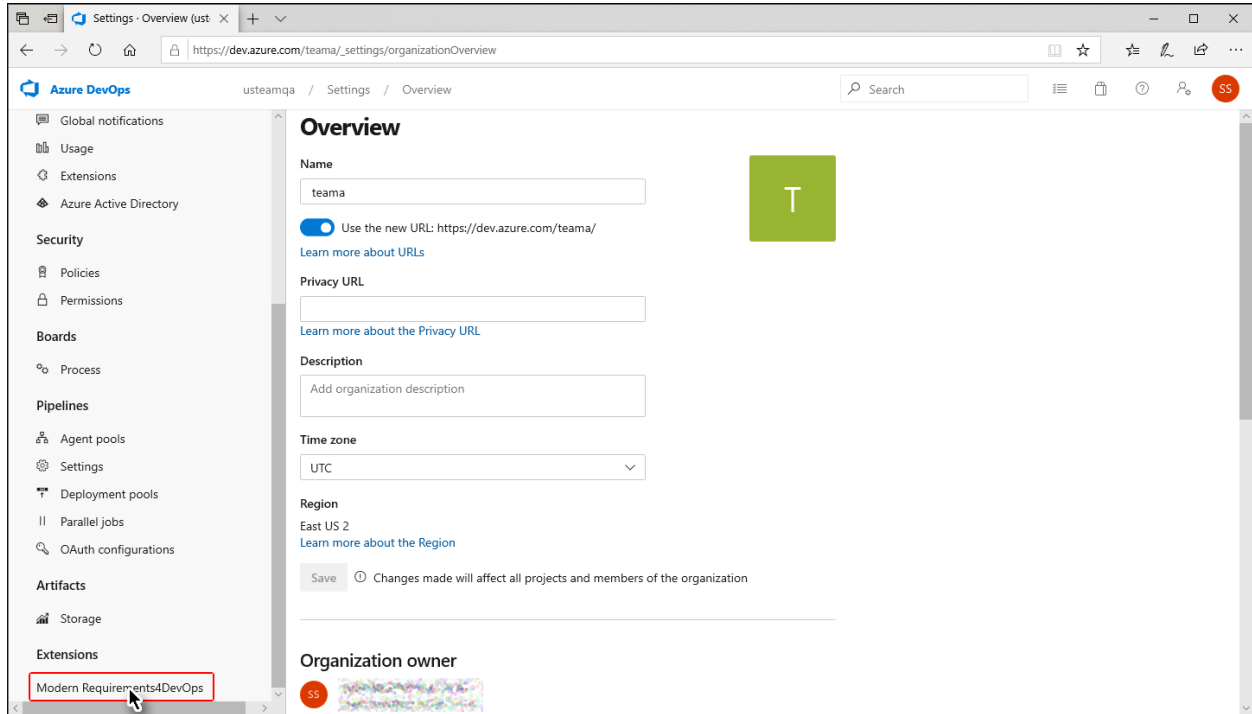
**MR SERVICES COLLECTION / ORGANIZATION REGISTRATION**

**FOR AZURE DEVOPS VERSION**

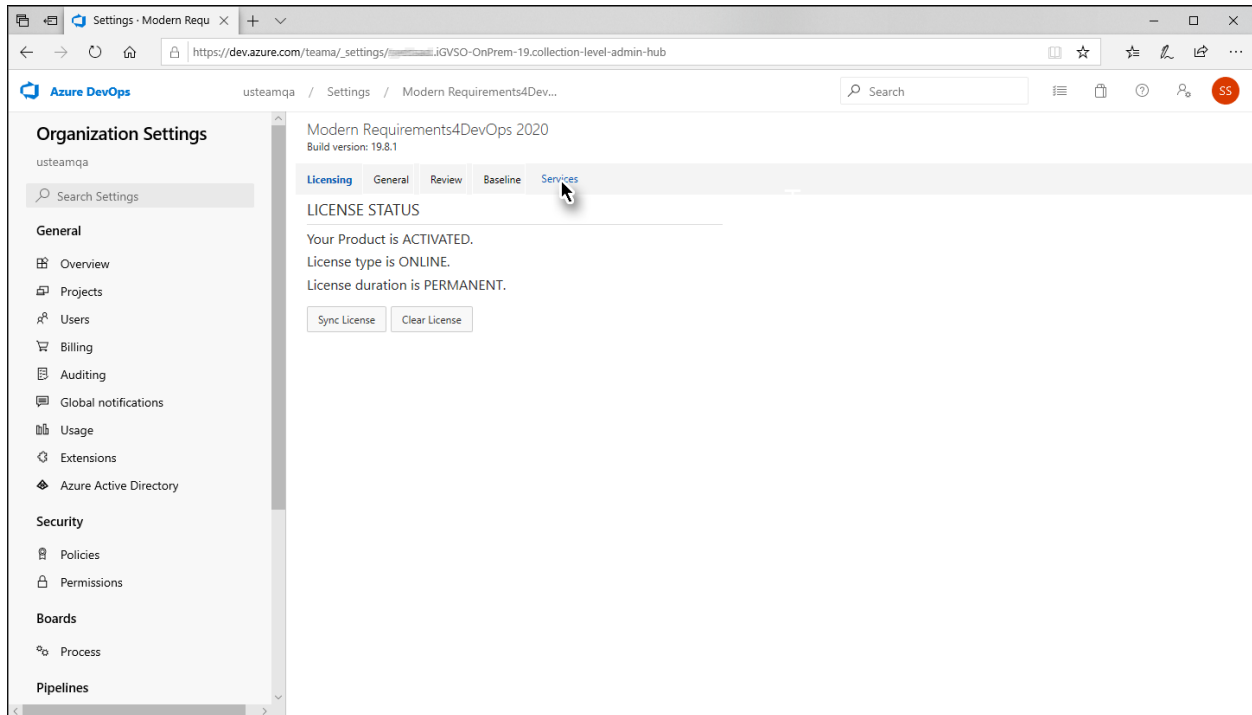
1. Logon to the relevant Azure DevOps Organization and click the **Organization settings** option at bottom left.



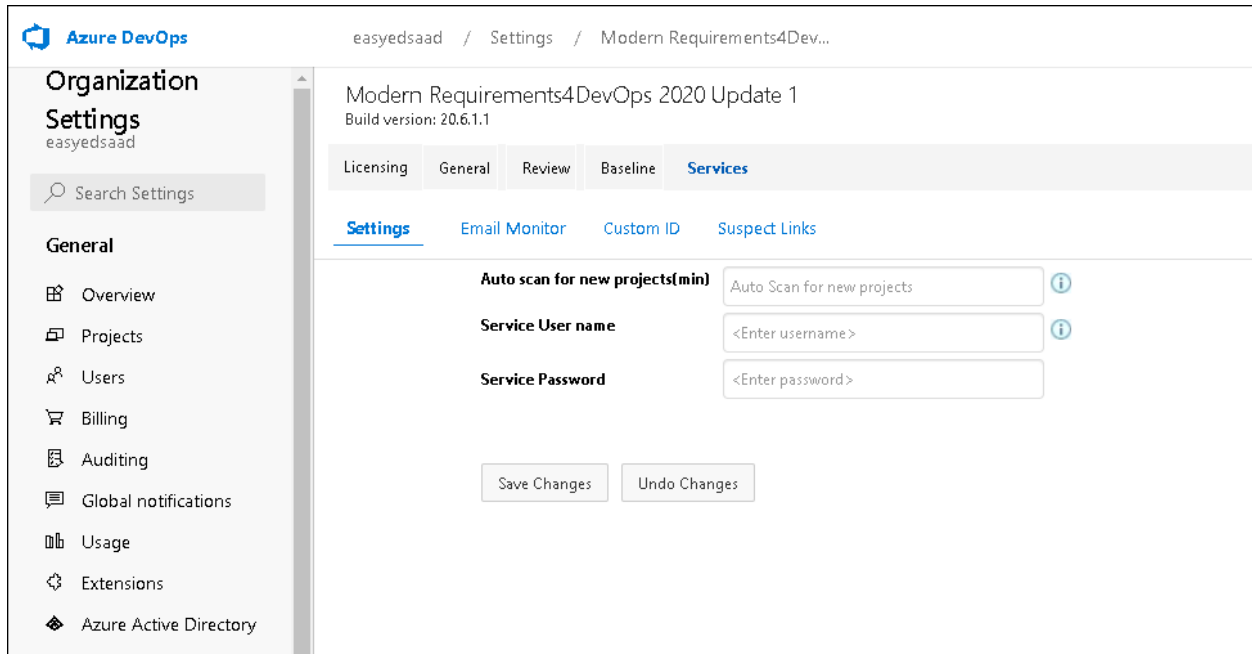
2. Scroll down the left side panel and click the **Modern Requirements4DevOps** option at the bottom.



3. Click the **Services** option in the Admin Panel.

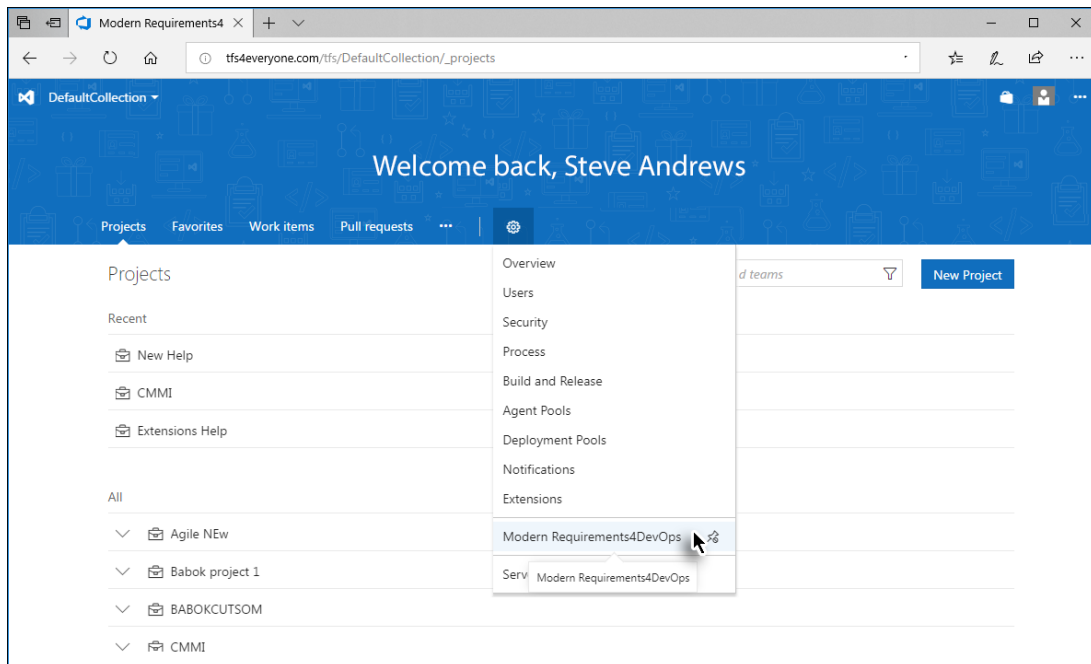


The options for **Services** Tab are displayed. Further steps are defined under [Configuring general settings in the Settings Tab](#) section



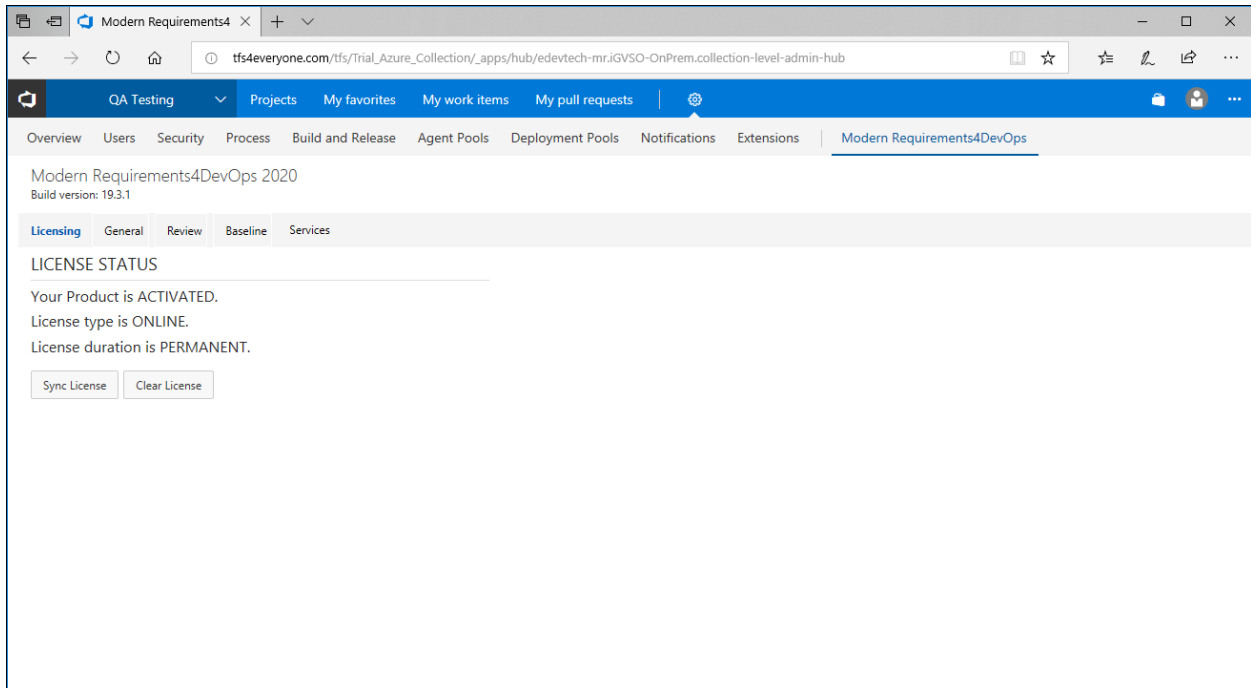
### FOR EMBEDDED VERSION

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

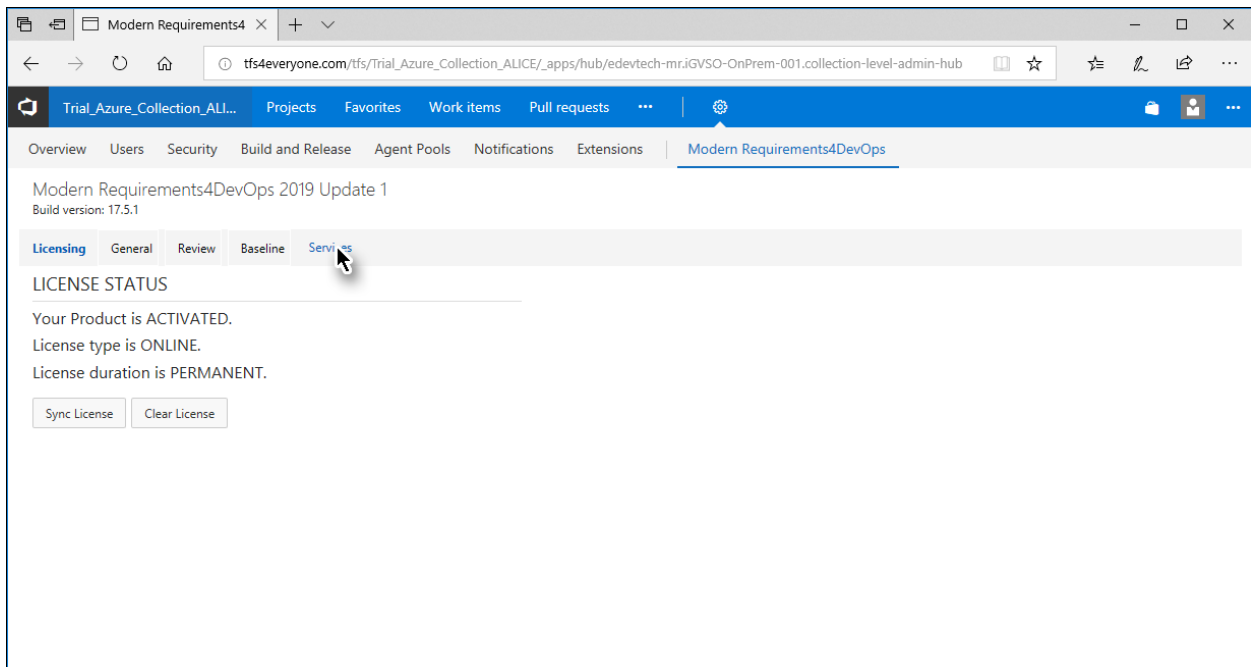




The Admin panel is displayed.



2. Click the **Services** Tab.



The options for **Services** Tab are displayed. Further steps are defined in the next section.

### CONFIGURING GENERAL SETTINGS IN THE SETTINGS TAB

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)

It is recommended to define both settings i.e. scan interval (as per requirements) along with collection/Organization registration.

1. 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 (or TFS Collection) would be scanned for new projects.

- Provide authorized login credentials. (with DevOps ID having Admin rights on Organization/TFS). In the case of Azure DevOps only **Personal Access Token (PAT)** is allowed to authenticate into Azure DevOps.

For Azure DevOps, the **'service username'** would be user's login email of Azure DevOps account that has admin rights, whereas, the **'service password'** will always be **PAT (Personal Access Token)**. However, in the case of Azure DevOps server the user will be required to enter the username and password of Collection / Organization Administrator.

To know how to create, use, modify, and revoke PATs for Azure DevOps, visit this [link](#).

Modern Requirements4DevOps 2020 Update 1  
Build version: 20.6.1.1

Licensing General Review Baseline **Services**

**Settings** Email Monitor Custom ID Suspect Links

**Auto scan for new projects(min)** 30 i

**Service User name** steve.andrews@hotmail.com i

**Service Password** .....

Save Changes Undo Changes

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

Modern Requirements4DevOps 2020 Update 1  
Build version: 20.6.1.1

Licensing General Review Baseline **Services**

**Settings** Email Monitor Custom ID Suspect Links

**Auto scan for new projects(min)** 30 i

**Service User name** steve.andrews@hotmail.com i

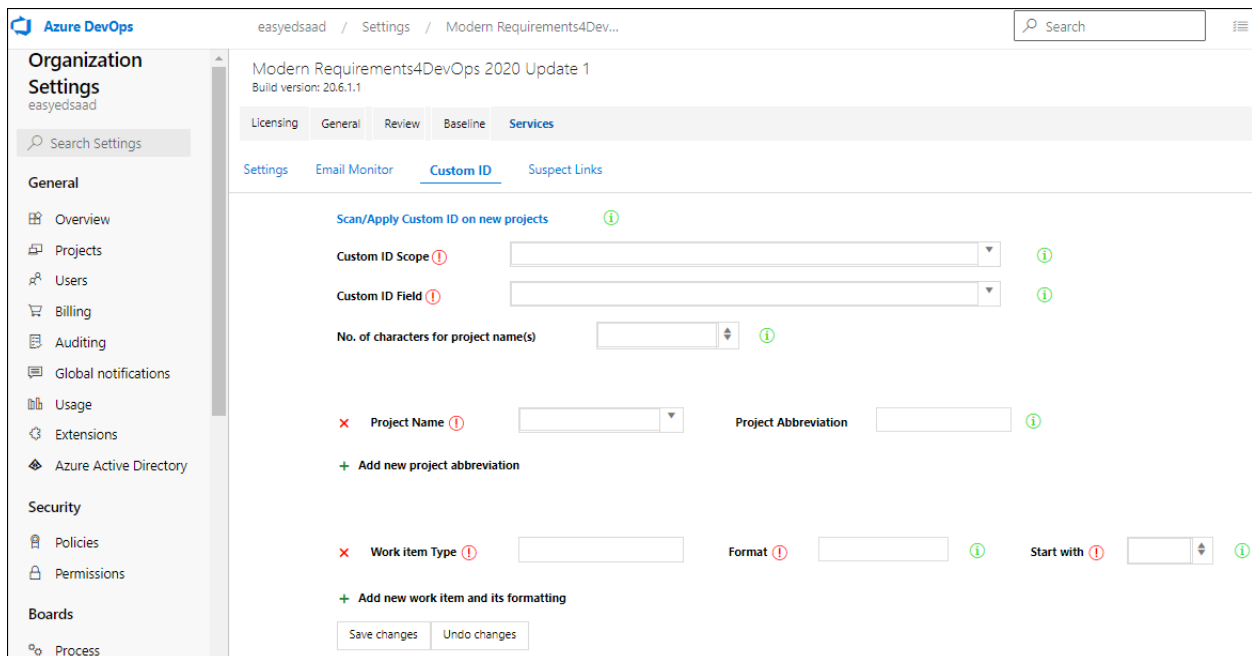
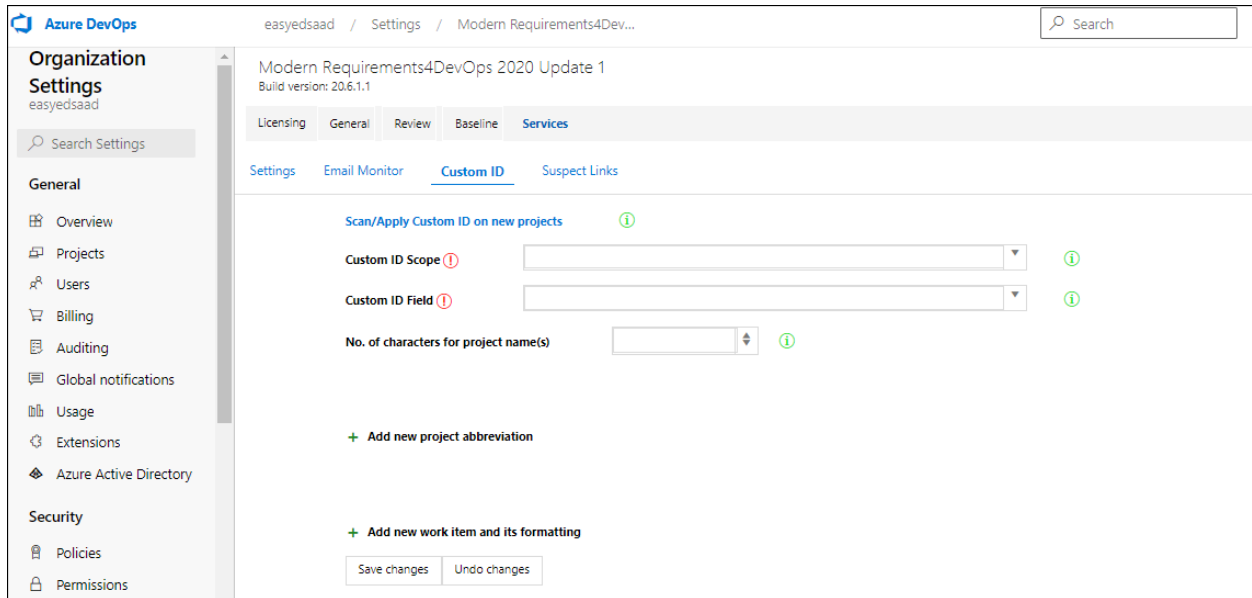
**Service Password** .....

Save Changes Undo Changes

Collection registered Successfully

## CUSTOM ID CONFIGURAION

**Custom ID** is a component of MR Services (MR Agent) that is used to provide Customized IDs to work items in addition to their default work item IDs. The Custom IDs do not replace the original IDs, instead they complement them. The customized IDs can be used to keep track of the work item origins (i.e. which team created a particular work item).



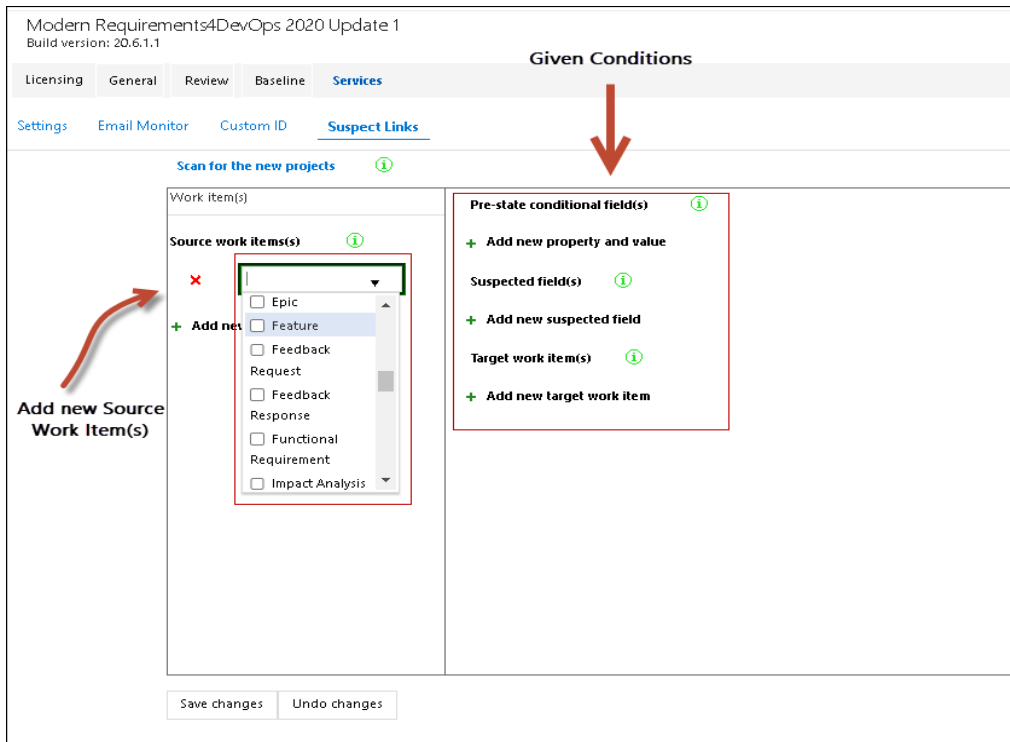
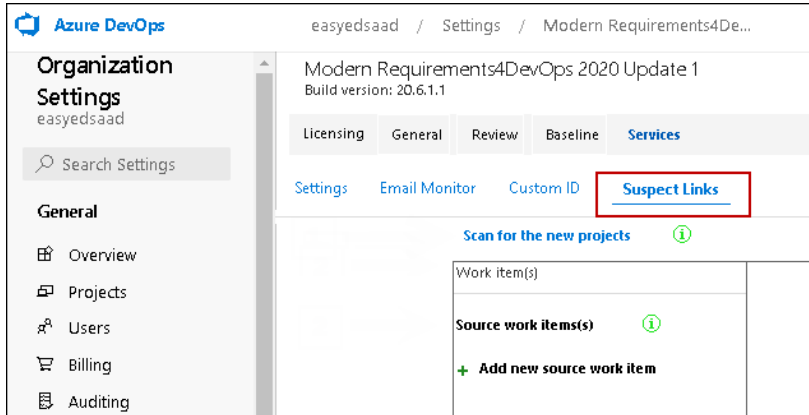
**UI FUNCTIONALITIES OF CUSTOM ID**

	UI Options	Functionality
1	Scan / Apply Custom ID on new projects	Links to execute Scan and Apply command manually. Now a user can SCAN and Apply at the same time from this option. The option helps in scanning for any new project that has been created or registered on organization/collection, and on the same time it applies Custom ID configuration on configured work item type(s) on which Custom ID is not applied yet.
2	Custom ID Scope	This field defines boundary of Custom ID i.e. <b>Collection</b> applies counter scope to Collection level, <b>Project</b> applies counter scope to Project Level and <b>Team</b> applies counter scope to the Team Level.
3	Custom ID Field	Select reference name of the field on which Custom ID is required to apply in Work Item.
4	No. of Characters for Project name(s)	This field defines the by-default no. of characters to be taken from the Project Name(s) in Organization to be used as Project Prefix (if not defined explicitly) in Custom ID formatting (Using a tag [PN])
5	Project Abbreviation	Defines prefix for the selected project name. The field defines prefix for the selected project (that will be used in Custom ID formatting by tag [PN])
6	Work item type	Denotes the type of work items with which the Suspect Links will work. This field defines list of work item types(s) to select on which Custom ID is required to apply. Multiple work item types could be selected (using “,” as a separator) to create a group on which same formatting is required.
7	Format	Defines the format of the ID
8	Start with	Defines starting sequence of Custom ID counter

## SUSPECT LINKS CONFIGURATION

'Suspect Links' is a component of MR Services (MR Agent) that is used to mark particular work items as **suspect** (due to changed requirements) so that relevant stakeholders may review these work items once instead of proceeding with the outdated requirements.

Suspect Links option can be found under the **Services** tab.



	UI Options	Functionality
1	Scan for the new projects	Applies Suspect Links configuration on new created projects in collection
2	Add new source work item	This field defines source work item(s) which is / are if modified then their associated work item(s) would be marked dirty or flagged when the given conditions are met.
3	Pre-state conditional fields	This section defines pre-state fields criteria of Source Work Item, which needs to be satisfied first in order to go and watch for suspected field(s)
4	Add new property and value	Sets the pre-state condition of Source Work Item
5	Suspected Field(s)	This section defines suspected fields to watch, after satisfying defined criteria for pre-conditional field(s) of source Work Items(s).
6	Target Work Item(s)	It tells (MR Services) MR Agent which type of work items will be tagged as dirty or flagged in case of a trigger.

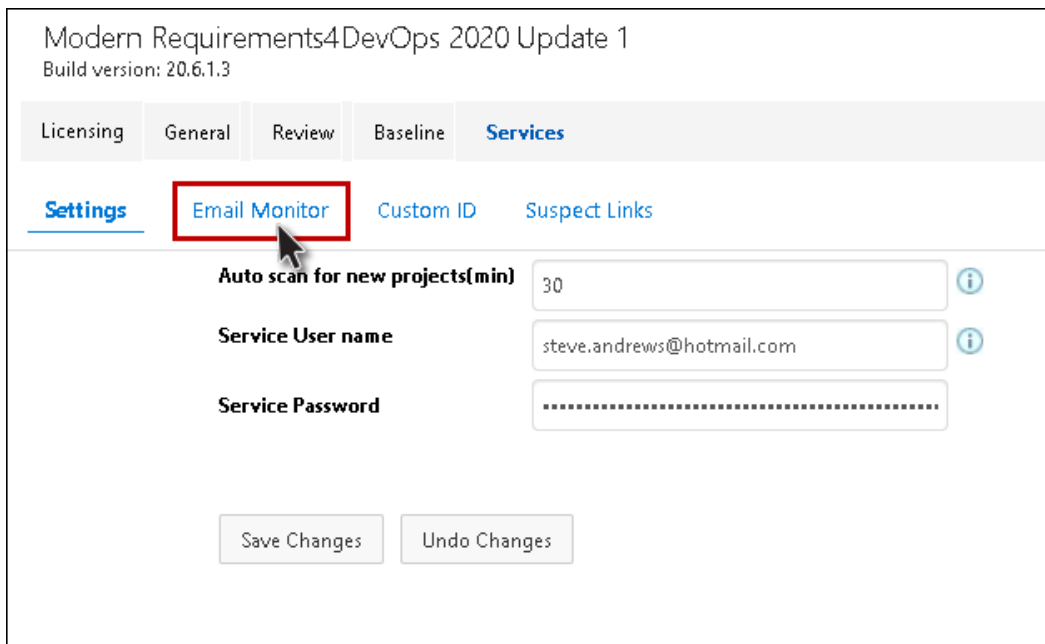
**EMAIL MONITOR**

**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.

**CONFIGURING EMAIL MONITOR**

Make sure to carry out the steps defined in [MR Services User Authentication](#) section for registration of Azure DevOps organization (or TFS Collection). Once you are done with that, follow these steps to configure Email Monitor settings:

1. Click the Email Monitor option.





The **Email Monitor settings** are divided into sections, where each section is used to configure a particular setting. However, all necessary settings are configured once. Users can not configure certain settings and leave other pending.

Modern Requirements4DevOps 2020 Update 1  
Build version:20.6.1.3

Licensing General Review Baseline **Services**

Settings **Email Monitor** Custom ID Suspect Links

---

**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

Add on Creation  Add on Update ⓘ

Sender Name  Sender Email  Email Body

---

**Workitem Field : History** Select required attributes to add in WI from received email

Add on Creation  Add on Update

Sender Name  Sender Email  Email Body

---

2. In the first section, configure the default project and the admin email address.

Modern Requirements4DevOps 2020 Update 1  
Build version: 20.6.1.3

Licensing General Review Baseline **Services**

Settings **Email Monitor** Custom ID Suspect Links

**Admin Email**

**Team Project (Default)**  ⓘ

Email Registration

[Register Email Address](#)

**Monitor Email**  ⓘ Edit

**WorkItem Type**  **WI Category**  ⓘ

	UI Options	Functionality
1	Admin Email	This email address is used as a mitigation; in case the desired functionality can't be achieved using the address defined in Email tag. In case some criteria do not match with the desired values, the warning email would be sent at the Admin Email.
2	Team Project (Default)	It defines the default team project for this collection

- In the second section, configure the email address (that would be used for Email Monitoring) along with the relevant Work Item type and category. Relevant work item type which is required to be created against email in Azure DevOps along with its category, so that if a particular work item is not available, other work item from that category would be created against the email.

Modern Requirements4DevOps 2020 Update 1  
Build version: 20.6.1.3

Licensing General Review Baseline **Services**

Settings **Email Monitor** Custom ID Suspect Links

**Admin Email**

**Team Project (Default)**

---

Email Registration

[Register Email Address](#)

**Monitor Email**  [Edit](#)

**WorkItem Type**  **WI Category**

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.

Settings **Email Monitor** Custom ID Suspect Links

**Admin Email**

**Team Project**

**Monitor Email**  [Edit](#)

**WorkItem Type**  **WI Category**

---

**Email Registration**

Use SSL

POP3 POP3 Mail Server  POP3 Port

IMAP IMAP Mail Server  IMAP Port

SMTP Mail Server  SMTP Port

MR Email

MR Email Password

**WorkItem Field** : Title \*Title will always be populated from email subject

- The third and last section provides user options to define what information is required from the email to be added in work item fields and on what events.

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

**Add on Creation** ⓘ  **Add on Update** ⓘ

**Sender Name**       **Sender Email**       **Email Body**

---

**WorkItem Field :** History Select required attributes to add in WI from received email

**Add on Creation**       **Add on Update**

**Sender Name**       **Sender Email**       **Email Body**

Save Changes
Undo Changes
Monitor Emails

	UI Options	Functionality			
1	<table border="1" style="width: 100%;"> <tr><td>Add on Creation</td></tr> <tr><td>Add on Update</td></tr> </table>	Add on Creation	Add on Update	These are the events for the user to define on which event they would like below information to be added.	
Add on Creation					
Add on Update					
2	<table border="1" style="width: 100%;"> <tr><td>Sender Name</td></tr> <tr><td>Sender Email</td></tr> <tr><td>Email Body</td></tr> </table>	Sender Name	Sender Email	Email Body	These are the options provided to the user to define what information is required from Email to be added in work item.
Sender Name					
Sender Email					
Email Body					

5. Click the **Save Changes** option to complete the Email Monitor configuration process.

Modern Requirements4DevOps 2020 Update 1  
Build version: 20.6.1.3

Licensing General Review Baseline Services

Settings **Email Monitor** Custom ID Suspect Links

Admin Email: admin@steveandrews.com

Team Project (Default): Test Project

Email Registration

Register Email Address

Monitor Email: steve.andrews@hotmail.com Edit

Workitem Type: Requirement WI Category: Requirement 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

Add on Creation  Add on Update

Sender Name  Sender Email  Email Body

Workitem Field: History Select required attributes to add in WI from received email

Add on Creation  Add on Update

Sender Name  Sender Email  Email Body

Save Changes Undo Changes Monitor Emails

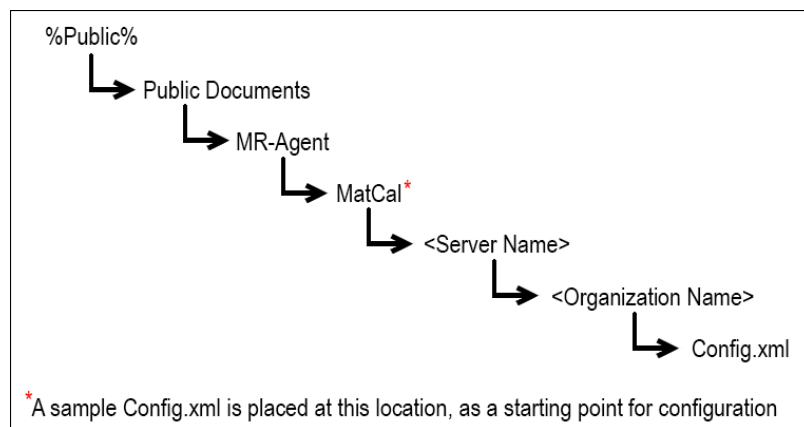
Once the configuration process is completed, users can click on the **Monitor Emails** button to initiate the process of monitoring the emails, as per settings done.

**MR MATCAL**

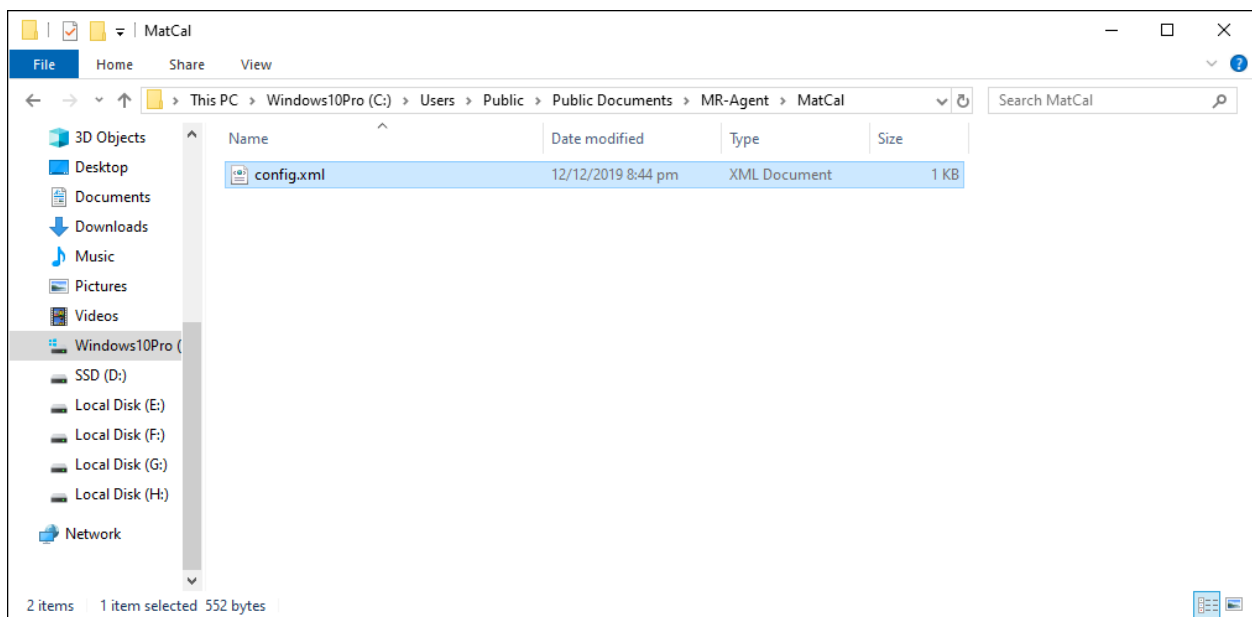
**MR MatCal** is a component of MR Agent that is used to perform/execute mathematical expressions on work item(s) field(s) and reflect its output on defined field(s) of same work item. The MatCal allows user to define nested expressions for calculation as well in which output of one expression could be input for another.

In order to make the MatCal work properly, users must manually create the following two items:

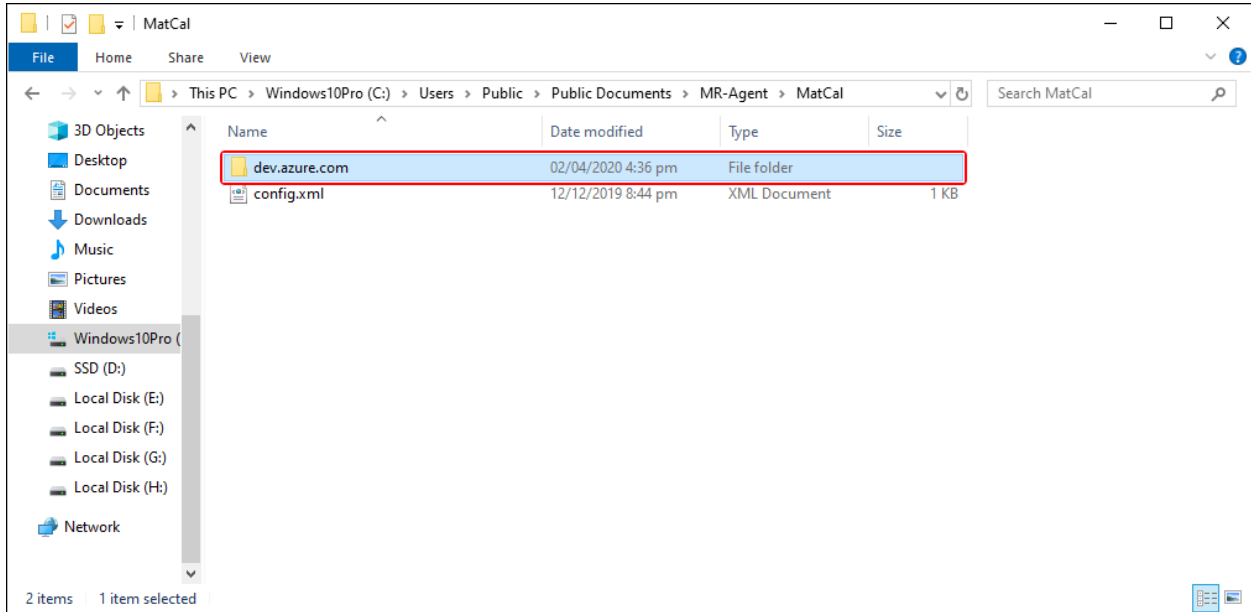
1. Azure DevOps server name on which the MatCal is required to apply.
  2. Azure DevOps Collection / Organization name (the Collection / Organization on which the Custom ID is required) under the TFS folder name.
1. The relevant Collection / Organization folder should contain the **config.xml** file containing all configuration. The file and folder hierarchy should appear as shown below using the text pattern and relevant image:



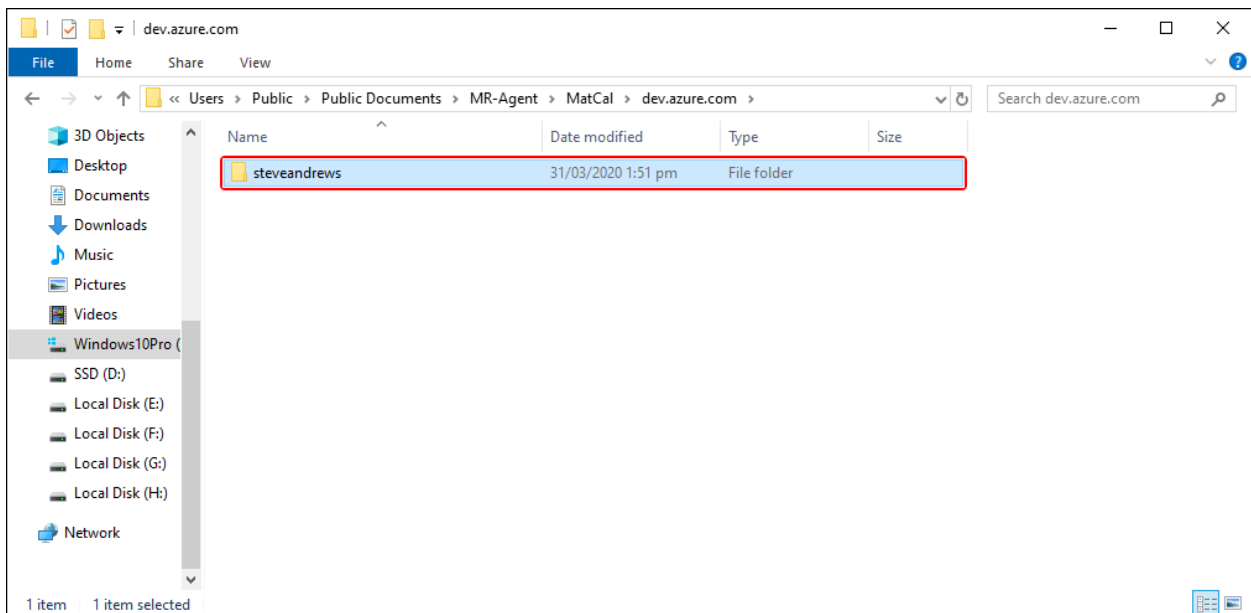
2. As shown in the image below, a sample **config.xml** file is placed in the **MatCal** folder.



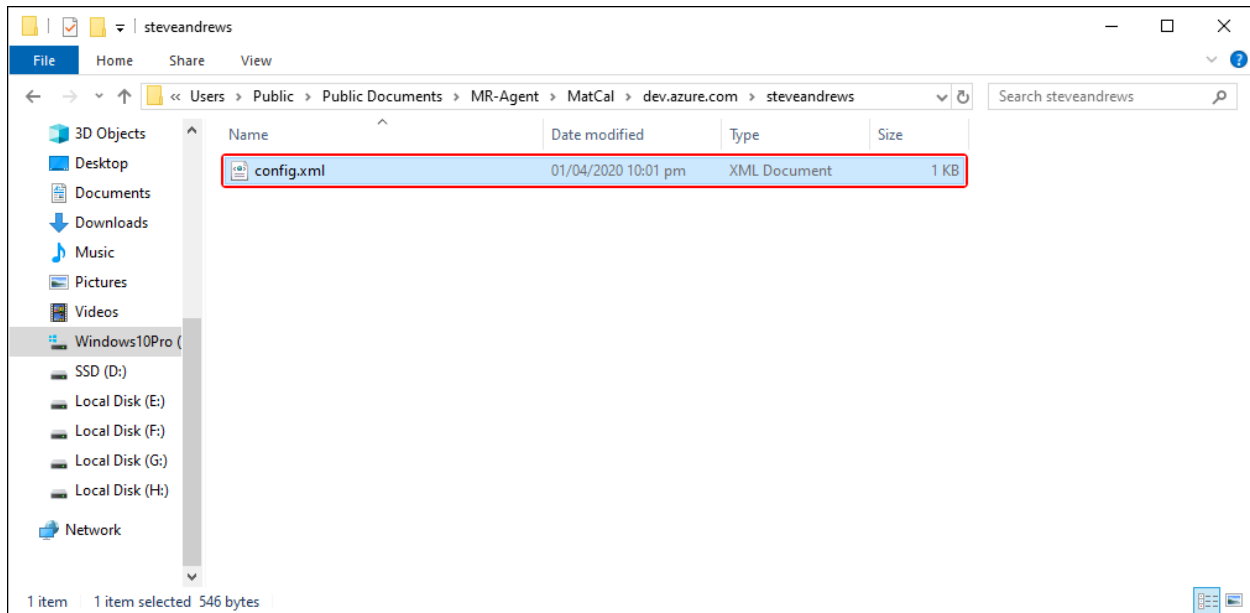
3. Create a folder named after the Azure DevOps (TFS) Server name (on which component is required to apply) at this location.



4. Enter the newly created folder and again create another folder here with the name of Azure DevOps Organization (or TFS Collection) (on which component is required to apply).



- Copy the **config.xml** file (discussed earlier) into the newly created folder i.e. Folder with TFS collection name.



This file contains the blueprint for the desired configuration.

## CONFIGURING THE MATCAL XML FILE

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



- Define the tags with their required values:
- “CollectionUrl”** tag requires the URL for the Azure DevOps Organization (or TFS Collection) on which the MatCal is required to apply. (Note: Please make sure that the URL should not end on ‘\’)
- “TeamProjects”** tag helps to define list of Team Projects on which user want to apply MatCal, Team Project names could be added using tags **“ProjectName”**.
- Within tag **“Math Expression”** define work item type in value of **“WIType”** tag, name of the field OR reference name of the field of the same work item on which output is required to add and events on which



user wanted MatCal to perform work (by setting value of tags “**OnCreate**” and “**OnUpdate**” to “**true**” or “**false**”)

6. **OnCreate**: It is the attribute at work item level, which allows user to configure whether MatCal calculation should be applied at the time of work item creation only. If it is set as "false" then it will not apply MatCal configuration to a work item at the time of creation.
7. **OnUpdate**: Similarly, it is the second attribute which allows user to configure if they want MatCal configuration to be applied every time when work item is updated. So it is set as "false" then MatCal calculation/configuration would be applied at the time of any update on a work item.
8. “**Expression**” tag contains mathematical expression required to perform and calculate required functions.
9. Following the successful completion of the configuration file, save and close the file.

### MATCAL EXPRESSION CALCULATION EXAMPLES

Following are the examples of expressions that could be configured in MatCal:

**Please Note:** Copy pasting the configuration from the Word/PDF document might create issues. It may change content like “-” and it doesn’t match with the values that apparently seem same on the UI. Therefore, it is recommended that the **user add field values by himself again**. This applies to all the examples below.

#### EXAMPLE 1 (FOR APPLYING CALCULATION)

For Risk WI:            Original Estimate = (Priority + Severity) \* Probability

MatCal configuration for the above expression would be:

```
<MathExpression WType="Risk" OutputField="Microsoft.VSTS.Scheduling.OriginalEstimate"
    OnCreate="true" OnUpdate="true">
    <Expression> (Priority + severityVal) * Probability </Expression>
    <MathExpression OutputField="severityVal">
        <Expression>If (Severity = "1 - Critical", 1, If (Severity = "2 - High", 2,
            If (Severity = "3 - Medium", 3, 4))) </Expression>
    </MathExpression>
</MathExpression>
```

**EXAMPLE 2 (FOR APPLYING MULTIPLE CONDITIONS)**

For Risk WI:            If Probability > 10 than Severity = '4 – Low'  
                               If Probability > 40 & <= 10 than Severity = '3 – Medium'  
                               If Probability > 80 & <= 40 than Severity = '2 – High'  
                               Otherwise Severity = '1 – Critical'

As per the structure of MatCal config file, following are the details required:

- **WIType**= Work Item Type on which expression is required to perform/execute. = **Risk**
- **OutputField**= Field reference name or Field name on which resultant of expression calculation is required to added. = **Severity**

So, the complete configuration of MatCal for the above expression would be:

```
<MathExpression WIType="Risk" OutputField="Severity" OnCreate="true" OnUpdate="true">
  <Expression> If (Probability &lt; 10, "4 – Low", If (Probability &lt; 40 AND Probability &gt;=
    10, "3 – Medium", If (Probability &lt; 80 AND Probability &gt;= 40, "2 – High",
    "1 – Critical")))</Expression>
</MathExpression>
```

**EXAMPLE 3 (FOR APPLYING CALCULATION AND CONDITION)**

For Risk WI:            If Blocked = NO than Probability = Priority \* Severity  
                               If Blocked = YES than Probability = 50

As per the structure of MatCal config file, following are the details required:

- **WIType**= Work Item Type on which expression is required to perform/execute. = **Risk**
- **OutputField**= Field reference name or Field name on which resultant of expression calculation is required to added. = **Probability**
- **Expression**= mathematical Expression and condition required for calculation.  
 i.e.  
**If Blocked = NO than Probability = Priority \* Severity** *<this is part of our mathematical expression>*  
**If (Blocked.ToLower() = "no", (Priority \* severityVal), 50)** *<this is the line of code used to implement the mathematical expression>*

Now to convert values of Severity to numeric values,

If Severity value is set to "1-Critical" then set value =1 in "severityVal",  
 Else if Severity value is set to "2-High" then set value =2 in "severityVal",  
 Else if Severity value is set to "3-Medium" then set value =3 in "severityVal",  
 Else set value= 4 in "severityVal".

OutputField="severityVal"

If (Severity = "1 - Critical", 1, If (Severity = "2 - High", 2, If (Severity = "3 - Medium", 3, 4))

So, the complete configuration of MatCal for the above expression would be:

```
<MathExpression WType="Risk" OutputField="Probability" OnCreate="true"
OnUpdate="true">
  <Expression> If (Blocked.ToLower() = "no", (Priority * severityVal), 50) </Expression>
  <MathExpression OutputField="severityVal">
    <Expression> If (Severity = "1 - Critical", 1, If (Severity = "2 - High", 2, If(Severity = "3 -
      Medium", 3, 4)))</Expression>
  </MathExpression>
</MathExpression>
```

#### EXAMPLE 4 (FOR APPLYING CONCATENATION)

For Risk WI: Description = concatenation ('priority'=Priority, 'Severity'=severity)

MatCal configuration for the above expression would be:

```
<MathExpression WType="Risk" OutputField="Description" OnCreate="true"
  OnUpdate="true">
  <Expression>"Priority: " + Priority + " Severity: " + Severity </Expression>
</MathExpression>
```

#### EXAMPLE 5 (FOR APPLYING SAME RULE ON MULTIPLE WORK ITEMS)

For Risk & Task WI: if Priority = 1, 2, or 3 than Blocked = Yes  
if Priority = 4 than Blocked = No

MatCal configuration for the above expression would be:

```
<MathExpression WType="Risk,Task" OutputField="Blocked" OnCreate="true"
OnUpdate="true">
  <Expression> If (Priority = 1 OR Priority = 2 OR Priority = 3, "Yes", "No") </Expression>
</MathExpression>
```

**EXAMPLE 6 (FOR APPLYING SEPARATE RULES ON SEPARATE WORK ITEMS)**

For Feature:            If Effort > 50 Mark Priority = 1  
                                  Else If Effort < 50 Mark Priority = 4

For Bug:                If Size > 10 Mark Blocked = Yes  
                                  Else If Effort <10 Mark Blocked = No

As per the structure of MatCal config file, following are the details required:

- **WIType**= Work Item Types on which expression is required to perform/execute in above scenario are = **Feature & Bug**
- **OutputField**= Field reference name or Field name on which resultant of expression calculation required to added = In Feature its **Priority** & In Bug its **Blocked**

So, the complete configuration of MatCal for the above expression would be:

```
<MathExpression WIType="Feature" OutputField="Priority" OnCreate="true"
OnUpdate="true">
    <Expression> if ( Effort > 50 , 1 , 4) </Expression>
</MathExpression>
<MathExpression WIType="Bug" OutputField="Blocked" OnCreate="true" OnUpdate="true">
    <Expression> if ( Size > 10 , "Yes" , "No" )</Expression>
</MathExpression>
```