

Step-by-step guide

1. Add an Application Tab



Note. Copying following script as it is will create default sample tab.

1. Use following SQL script to create new tab in PME web banner.

-- Script to add a new application tab to PME.

-- Note: no spaces allowed in @id

DECLARE @id NVARCHAR(255)

DECLARE @name NVARCHAR(255)

DECLARE @description NVARCHAR(255)

DECLARE @order NVARCHAR(100)

DECLARE @target NVARCHAR(400)

DECLARE @userGroupBasedVisibility BIT

-- Edit the following values to define your custom application settings

-- Custom application's unique internal id. This isn't displayed, and should never be changed.

-- NOTE: This value must not have any spaces!

SET @id = 'MyApp'

-- Custom application tab name. This is what is displayed as the name of the tab.

SET @name = 'My Custom App Link'

-- Custom application tab description. This is not displayed. For informational use.

SET @description = 'A demonstration of how to add a custom app link to the system'

-- Custom application tab order; less than 10 will bring new tab first left position.
-- You can view the other order values in the settings table to determine the desired position

SET @order = 5

-- Target URL can be provided in single quote here e.g. '<http://www.google.com>'

-- If on the same machine as PME, should be the relative path. e.g., '/myApp/index.html'

SET @target = '<http://www.powerlogic.com/>'

-- The application can be shown or hidden based on which user groups a user is a member of.

-- If you want this functionality, set the following to 1 (true). Normally this is not

-- desired and can be left as 0 (false).

SET @userGroupBasedVisibility = 0

-- DO NOT MODIFY CONTENT BELOW

USE [ApplicationModules]

-- proper XML encoding

DECLARE @idXml NVARCHAR(MAX) = REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(@id,'&', '&'); '<', '<'); '>', '>'),

DECLARE @nameXml NVARCHAR(MAX) = REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(@name,'&', '&'); '<', '<'),

DECLARE @descriptionXml NVARCHAR(MAX) = REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(@description,'&', '&');

DECLARE @targetXml NVARCHAR(MAX) = REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(@target,'&', '&'); '<', '<'),

DECLARE @orderXml NVARCHAR(MAX) = REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(@order,'&', '&'); '<', '<'),

DECLARE @configurationValue NVARCHAR(MAX) =

'<FrameworkApplication' +

 'id="' + @idXml +

```

    "" displayName="" + @nameXml +
    "" description="" + @descriptionXml +
    "" resourceSet="" +
    ' target="" + @targetXml +
    "" privilege="" + @idXml + '.AccessApplication" ' +
    ' order="" + @orderXml +
    "" enabled="true" ' +
    'xmlns="uri:application-modules/power/framework/application#" />'

```

```
EXECUTE [Configuration].[WriteConfigurationValue]
```

```

    'ApplicationFramework'
    , 'Applications'
    , @id
    , @name
    , @configurationValue
    , 'custom'
    , 0

```

```
DECLARE @Privilege NVARCHAR(75) = SUBSTRING(@idXml, 1, 57) + '.AccessApplication'
```

```
DECLARE @PrivilegeId INT
```

```
DECLARE @ChangeResult TABLE (ChangeType VARCHAR(10), Id INTEGER)
```

```
MERGE [ApplicationFramework].[Privileges] AS target
```

```
USING (VALUES(@Privilege)) AS source(Privilege)
```

```
ON target.Privilege = source.Privilege
```

```
WHEN MATCHED THEN
```

```
    UPDATE SET IsTeamBased = @userGroupBasedVisibility, @PrivilegeId = target.Id
```

```
WHEN NOT MATCHED THEN
```

```
INSERT (Privilege, IsTeamBased) VALUES (@Privilege, @userGroupBasedVisibility)
```

OUTPUT

```
$action, inserted.Id INTO @ChangeResult;
```

```
IF @PrivilegeId IS NULL
```

```
BEGIN
```

```
    SET @PrivilegeId = SCOPE_IDENTITY();
```

```
END
```

```
MERGE [ApplicationFramework].[RolePrivilegeMap] AS target
```

```
USING (VALUES(@PrivilegeId)) AS source(PrivilegeId)
```

```
ON target.PrivilegeId = source.PrivilegeId AND RoleID = 1
```

```
WHEN NOT MATCHED THEN
```

```
    INSERT (RoleID, PrivilegeId) VALUES (1, @PrivilegeId);
```

```
MERGE [ApplicationFramework].[RolePrivilegeMap] AS target
```

```
USING (VALUES(@PrivilegeId)) AS source(PrivilegeId)
```

```
ON target.PrivilegeId = source.PrivilegeId AND RoleID = 8
```

```
WHEN NOT MATCHED THEN
```

```
    INSERT (RoleID, PrivilegeId) VALUES (8, @PrivilegeId);
```

```
MERGE [ApplicationFramework].[RolePrivilegeMap] AS target
```

```
USING (VALUES(@PrivilegeId)) AS source(PrivilegeId)
```

```
ON target.PrivilegeId = source.PrivilegeId AND RoleID = 9
```

WHEN NOT MATCHED THEN

```
INSERT (RoleID, PrivilegeId) VALUES (9, @PrivilegeId);
```

. New tab is successfully added and will appear on PME web page reload.

2. Hide an Application Tab



There are two ways to hide a tab in web application, this specific section describes only through database:

- Through Database (This will hide tab for everyone including supervisors).
- Through User Manager (This will hide tab from all users of the system, except supervisors), see PME help for details.

1. Run following SQL statement in SQL management studio on PC where PME is installed or DB Server in case of distributed system.

```
SELECT *
```

```
FROM [ApplicationModules].[Configuration].[ConfigurationSettings]
```

```
WHERE ItemType ='ApplicationFramework' AND Item = 'Applications'
```

All applications (Dashboards, Diagrams, Trends, Alarms, Reports) will be listed, copy “Value” column from any item (e.g., "Alarms") and paste it in notepad below data will be available.

```
'<FrameworkApplication id="Alarms" displayName="AlarmViewerApplication_Title"
description="AlarmViewerApplication_Description" resourceSet="ApplicationFrameworkResources"
target="/Alarms/" privilege="AlarmViewer.AccessApplication" order="35" enabled="true"
xmlns="uri:application-modules/power/framework/application#" />'
```

2. "Enabled" value will decide to display applications tab or not, change it to “false” and place it in following SQL script.

```
UPDATE [ApplicationModules].[Configuration].[ConfigurationSettings]
SET Value = 'Copy entire data from notepad with modified enabled here'
WHERE [key]= 'type application name here'
```

3. The tab is successfully hidden and will dis-appear on PME web page reload

3. Re-order an Application Tab



To change the tab order for a Web application:

1. Run following SQL statement in SQL management studio on PC where PME is installed.

```
SELECT *
```

```
FROM [ApplicationModules].[Configuration].[ConfigurationSettings]
```

```
WHERE ItemType ='ApplicationFramework' AND Item = 'Applications'
```

All applications (Dashboards, Diagrams, Trends, Alarms, Reports) will be listed, copy the “Value” column content of an application (e.g., "Alarms") and paste it into notepad.

Example:

```
'<FrameworkApplication id="Alarms" displayName="AlarmViewerApplication_Title"
description="AlarmViewerApplication_Description" resourceSet="ApplicationFrameworkResources"
target="/Alarms/" privilege="AlarmViewer.AccessApplication" order="35" enabled="true"
xmlns="uri:application-modules/power/framework/application#" />'
```

2. Change the "order" value to the number you want, place it in following SQL script and execute the script. The "order" property determines the tab order in Web Applications. The tab order is assigned based on increasing order number, from left to right. The numbers for the default tabs are Dashboards (10), Diagrams (20), Trends (30), Alarms (40), Reports (50). The Settings tab is always in the right most position.

```
UPDATE [ApplicationModules].[Configuration].[ConfigurationSettings]
```

```
SET Value = 'Copy entire data from notepad with modified order here'
```

```
WHERE [key]= 'type application name here'
```

3. The tab order has been changed and will be visible in PME after a web page reload

4. Delete an Application Tab



Note. Using following script as it is will work if default sample tab was created in “Add” section.

You need to provide Name (e.g., MyApp) in following both delete scripts. The Name is the @id specified in the script in #1 above.

```
DELETE FROM [ApplicationModules].[Configuration].[ConfigurationValue]
```

```
WHERE ConfigurationKeyId = (Select id from [ApplicationModules].[Configuration].[ConfigurationKey]
where Name='MyApp')
```

```
DELETE FROM [ApplicationModules].[Configuration].[ConfigurationKey]
```

```
where Name='MyApp'
```

Icon

A deleted factory tab will be re-instated on an upgrade. If you wish the factory tab to be permanently removed, follow the steps for 'Hide an Application Tab'.

Icon

Deleting factory tabs is not recommended. Hiding them should be sufficient for all use cases.

5. Enable/Disable an Application Tab's Visibility Based on User Groups

By default,

- factory tabs' visibility can be based on which user groups the user is a member of.
- custom tabs' visibility is not based on user groups, unless it was specified in the script in step 1.

To alter this functionality, you must update the ApplicationFramework.Privileges table in the ApplicationModules database.

You simply change the IsTeamBased value from true to false for the corresponding **ApplicationName.AccessApplication**. For example, the RealtimeTrend application (a.k.a., "Trends") tab's visibility is not based on user groups when configured this way, as the value was changed from True to False.

Id	Privilege	IsTeamBased
25	AlarmConfiguration.AccessApplication	False
11	AlarmViewer.AccessApplication	True
5	Dashboard.AccessApplication	True
16	Diagrams.AccessApplication	True
20	Hierarchy.AccessApplication	False
26	RateEditor.AccessApplication	False
21	RealtimeTrend.AccessApplication	False
27	UserManager.AccessApplication	False
3	WebReach.AccessApplication	False
4	WebReporter.AccessApplication	False

This works for custom tabs as well. If you don't see the custom app listed you can manually add it to this table.

- Id: Automatically added by the system.
- Privilege: **ApplicationName.AccessApplication**. The application name must match the *@ConfigurationKey* used in the script in step 1, and should appear already if you ran the script in step 1.
- IsTeamBased: **True/False**

This causes the application to show up when configuring a User Group through the User Manager.