# Office AMS: Deploy Pre-Configured Web PArts to Web Part Gallery

|  |  |
| --- | --- |
| Summary: | Applies to: |
| This sample shows deploy a pre-configured Content Editor Web Part to the Web Part Gallery of the host web | * Office 365 Multi Tenant (MT) * Office 365 Dedicated (D) * SharePoint 2013 on-premises |
| Solution: | Core.UploadWebPart, version 1.0 |
| Author: | Richard diZerega (Microsoft) |
| //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  // THIS CODE IS PROVIDED \*AS IS\* WITHOUT WARRANTY OF  // ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING ANY  // IMPLIED WARRANTIES OF FITNESS FOR A PARTICULAR  // PURPOSE, MERCHANTABILITY, OR NON-INFRINGEMENT.  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* | |

# General comments

The sample shows a technique for exporting pre-configured Content Editor Web Parts (w/ scripts for calling SharePoint APIs) and deploying them to the Web Part Gallery in the host web via apps. This provides a powerful alternative to App Parts, which are hosted in IFRAME elements and are more challenging to deliver a responsive UX.

# Scenario 1: Deploying Pre-Configured Web Parts to Web Part Gallery

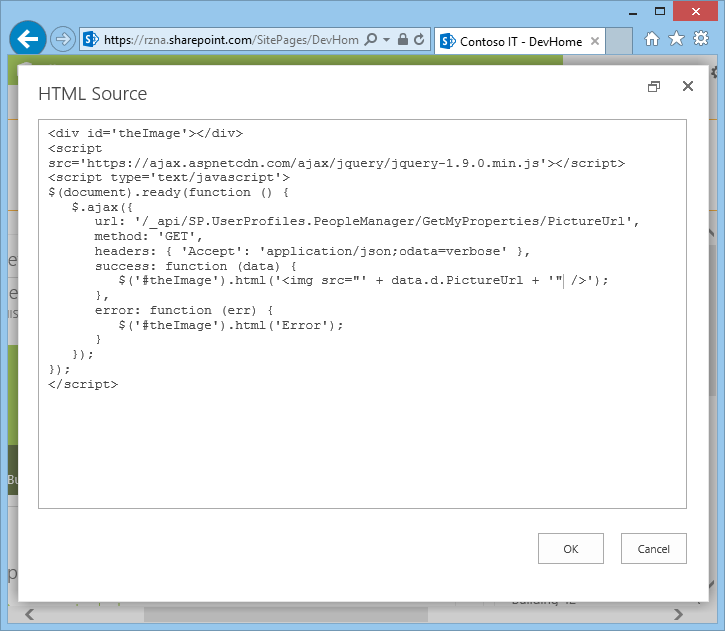
Deploying a pre-configured Content Editor Web Part requires the following steps that are details below:

* Configured desired presentation/logic in Content Editor Web Part via HTML and scripts
* Export the configured Content Editor Web Part with the page in edit mode
* Add the web part file to an app project and deploy it to the Web Part Gallery via CSOM

We use the Content Editor instead of the Script Editor because the former has the option to be exported. You could just as easily use a Script Editor Web Part, but you would have to manually configure the XML instead of using the SharePoint UI to export.

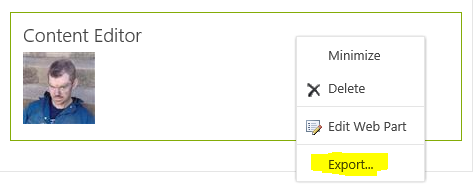
## Configure The Content Editor

With a Content Editor Web Part on your page, you can use the “Edit Source” ribbon button (in the “Markup” group of the “Format Text” tab) to paste your scripts/markup. Remember to reference any script references you might need (ex: JQuery). Here is an example of script to display the user’s profile picture:



## Export the Configured Content Editor

Now that the Content Editor Web Part is configured, we can export it with the page in edit mode:



## Deploying the Configured Content Editor via CSOM

Next, we can add the exported Content Editor Web Part file to our app project and deploy it to the host web programmatically with CSOM. You cannot deploy the web part as a module to the host web. Notice we are using client context of the HOST web:

var spContext = SharePointContextProvider.Current.GetSharePointContext(Context);

using (var clientContext = spContext.CreateUserClientContextForSPHost())

{

var folder = clientContext.Web.Lists.GetByTitle("Web Part Gallery").RootFolder;

clientContext.Load(folder);

clientContext.ExecuteQuery();

//get the local webpart and upload to web part gallery

using (var stream = System.IO.File.OpenRead(Server.MapPath("~/MyPicWebPart.dwp")))

{

FileCreationInformation fileInfo = new FileCreationInformation();

fileInfo.ContentStream = stream;

fileInfo.Overwrite = true;

fileInfo.Url = "MyPicWebPart.dwp";

folder.Files.Add(fileInfo);

clientContext.ExecuteQuery();

}

}