**EY Banner Code KT DOC**

TCS PLANNED DOCUMENTATION

Date 23 April 2020

Contents

[1. Introduction 1](#_Toc38639962)

[2. Explanation step by step 2](#_Toc38639963)

[3. Account Used 5](#_Toc38639964)

[4. Flow 6](#_Toc38639965)

[5. Color Banner 8](#_Toc38639966)

[6. Steps to Execute the Utility 10](#_Toc38639967)

[7. How to create input CSV for Banner 12](#_Toc38639968)

1. Introduction

Purpose of Banner code is to inform Clients on which date we are going to migrate the source sites into online, by inserting Banner on top of the source sites by colour codes with date. Colour codes is mentioned in below sections.

Before/ After the migration when we get approval from client to insert banner on source sites then we need to execute this code. Banner code (Console) is written in SharePoint CSOM in visual studio 2019.

1. Explanation Step by Step

* **Program.cs – Main logic codes, options, events present in this file, Main function contains options to choose Actions.**

1. GetCredential – function to get Credentials entered from Console.
2. LoadConfig – Function will read config values from **EYOS\_TemplateArtefacts.js**
3. ReadCSV – Function will read the values from CSV file and stored the values in List.
4. AddPropertyBagValues – Function will check if site exist or not from function**CheckIfSiteExistsAndGetContext** present in **Helper.cs** file then it will pass values to the **AddPropertyBag** function present in Helper file.
5. DisplayPropertyBagValues – Function will check is site exist or not from function **CheckIfSiteExistsAndGetContext** present in **Helper.cs** file then it will pass values to the **DisplayPropertyBag** function present in Helper file.
6. RemovePropertyBagValues – Function will check is site exist or not from function **CheckIfSiteExistsAndGetContext** present in **Helper.cs** file then it will pass values to the **RemovePropertyBag** function present in Helper file.
7. AddSiteLevelCustomActions – Function will check if site exists or not from function **CheckIfSiteExistsAndGetContext** present in **Helper.cs** file then it will pass values to the **AddSiteLevelCustomActions** function present in Helper file.
8. ListSiteLevelCustomActions – Function will check is site exist or not from function **CheckIfSiteExistsAndGetContext** present in **Helper.cs** file then it will pass values to the **ListSiteLevelCustomActions** function present in Helper file.
9. RemoveSiteLevelCustomActions – Function will check is site exist or not from function **CheckIfSiteExistsAndGetContext** present in **Helper.cs** file then it will pass values to the **RemoveSiteLevelCustomActions** function present in Helper file.

* **Helper.cs** – **Helper file contains business logic functions.**

1. AddPropertyBag – Add property bag will add property to the source site.
2. RemovePropertyBag – Function will remove property from source site.
3. DisplayPropertyBag – Function will display all the property from source site.
4. AddSiteLevelCustomActions – Function will Add Custom Action Named **BannerJS** to the CDN path. It will insert **BannerUserCustomAction.js** and **BannerStyleSheet.css**.
5. ListSiteLevelCustomActions – Function will list all the Custom property Name, Location and Sequence.
6. RemoveSiteLevelCustomActions – Function will remove mentioned Custom Actions from the site.
7. ConvertToSecureString – Function will convert password into secure string password.
8. GetClientContext – Function will return context from site URL.
9. CheckIfSiteExistsAndGetContext – Function will check if Site context exist or not, if exist it will return context else return null.
10. GetSafeURL – Function will return URL by trimming the ‘/’.
11. GetRootWebURL – Function will return Root web URL by splitting the URL.
12. LogErrorMessage – Function will log error message in Red color in console from Common function.
13. LogInfoMessage – Function will log error message in White color in console from Common function.
14. LogWarningMessage – Function will log error message in Yellow color in console from Common function.
15. WriteToLog – Function will log all the Message into Current Directory/ Logging folder with yyyyMMdd.txt format.

* **Common.cs** – **Contains constants, variable used in entire code.**

1. Common – Common function class contains CSV file path, Source username, Destination Username etc.
2. Migrate – Migrate class contains get/set the values.
3. Config – Class contains get/set Config values.
4. ReportConfig – Does not contain any values.

* **EYOS\_TemplateArtefacts.json** – **Contains property values in json format.**
* **App.Config** – **Contains Configuration keys.**
* **Input file – Format is in CSV, Contains Header like** – **SourceUrl, DestinationUrl, MigrationDate, BannerColor.**
* **Logger.cs:** All success, Failure message will be logged to the text file with datewise in current system directory file bin\Debug\Logging\currentdate format (yyyymmdd), It will be overwritten every time so please check and delete if any current date log file exists before running. In Each function any success or failure information is writing to log file.

1. LogError – Function will write Error logs.
2. LogInformation – Function will write Information of logs.

* **BannerUserCustomAction.js –** Custom Action JavaScript file will add the HTML banner to the S4 div container based on Banner Color, also it will add redirect link of destination site URL according to the input CSV.

1. RegisterModuleInit **–** Function will register banner.js for MDS and it will call RemoteManager\_Inject function.
2. RemoteManager\_Inject **–** Function will wait for all the script to load then it will call ReadPropertyBag function.
3. ReadPropertyBag **–** Function will read all the property bags from source site it is success. If all the values are not empty it will call insertElement function.
4. insertElement **–** Function will insert banner based on date diff.
   * + Current date is equal to Migration date, Red banner will be inserted.
     + If Date difference is less than or equal to 7, i.e. 1 week before the migration, Orange banner will be inserted.
     + If Date difference is greater than 7 I. e before 2 weeks from migration date, yellow banner will be inserted.
     + Gray bag will be inserted once it is passed initial testing.

* **BannerStyleSheet.css –** It Contains all the styles, color for banner. It contains classes like redBg, yellowBg, orangeBg etc.

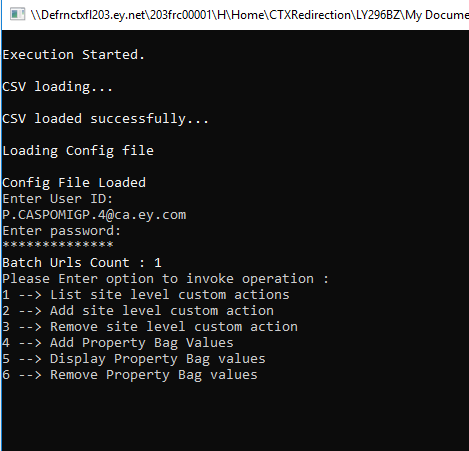
1. Account Used

EYDMZ\Migration Account for Source Sites.

1. Flow

### Mention CSV File path:

Before the **EY.SP.CommunicationBanner.exe** file is executed it is important to mention the CSV file path in **CSV\_FILE\_PATH** in **Common.cs** file, this is the input file for the Banner code. After executing the **EY.SP.CommunicationBanner.exe** file, provide source site credentials, once provided, console screen will be displayed as below:



According to the Task, User need to select options as above.

**Read CSV file** – once options are selected **ReadCSV** function read all the values from CSV file mentioned.

Before going into main functions, **CheckIfSiteExistsAndGetContex**t function will check if particular site is existing or not and it will log the message in bin\Debug\Logging\currentdate.

**Below flow will be changed based on requirement.**

### List Site level Custom Actions:

**ListSiteLevelCustomActions** function will list all the site level custom actions for sites mentioned in CSV like – Name, Location, Sequence.

### Add Site Level Custom Actions:

* + - Deploy a Site Level Custom action and will add property bag values in the Site Collection.
    - The Custom action is getting values from **BannerUserCustomAction.js** already deployed in the CDN of each site collection to display banner HTML on source sites created dynamically in the script. Custom Action name is **BannerJs**.

### Remove site level custom action:

**RemoveSiteLevelCustomActions** function willRemove the Custom Actions named **BannerJs.** This function will loop thru all the custom actions and it will delete once it finds the matching Custom Action.

### Add Property Bag Values:

**AddPropertyBagValues** function will add property to the source URL of MigrationDate, BannerColor, DestinationUrl. Once Custom Actions are inserted Property bags can be inserted.

### Display Property Bag values:

**DisplayPropertyBagValues** function will display all property bags for sites mentioned in CSV.

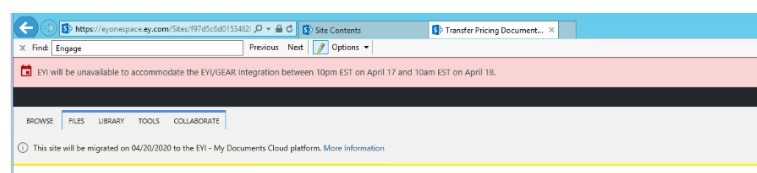
### Remove Property Bag values:

**RemovePropertyBagValues** function will remove property bags of DestinationUrl, MigrationDate, BannerColor.

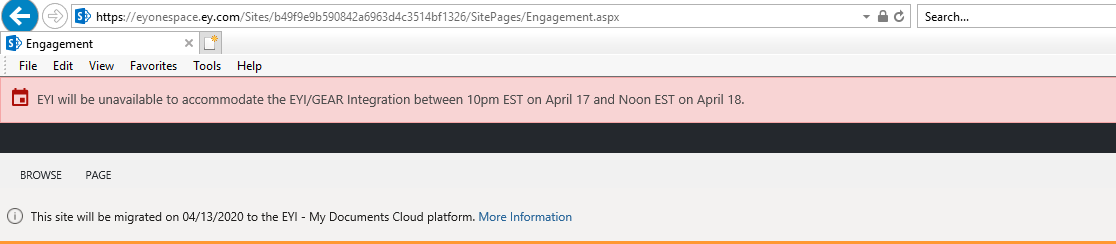
1. Color Banner

Color Code will be mentioned in CSV based on Migration Process.

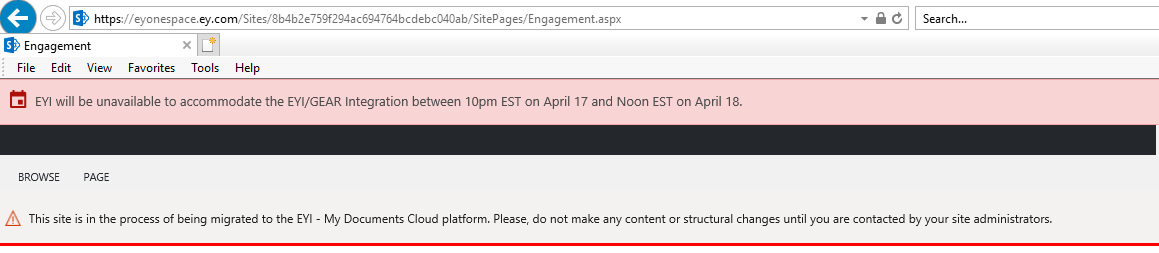
* YELLOW – **2 weeks before** Migration Yellow banner will be inserted.



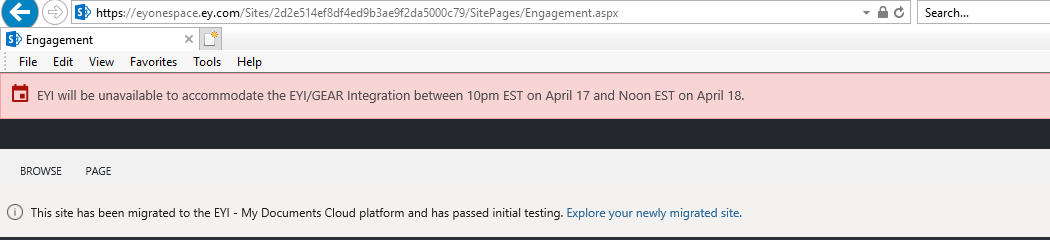
* ORANGE – **1 week before** Migration Orange banner will be inserted.



* RED – **On day of** Migration RED banner will be inserted.



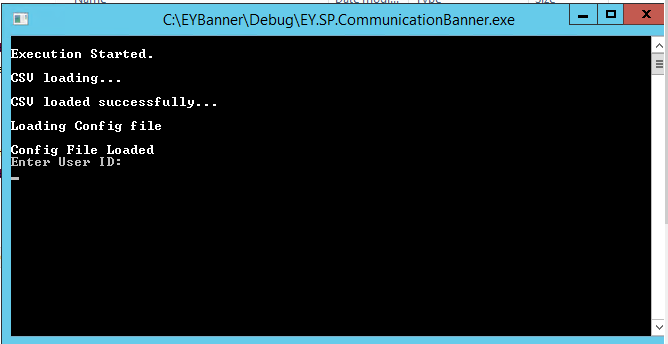
* GRAY – Once site has been migrated to the EYI – My Documents Cloud platform and has passed initial testing, Gray banner will be inserted. **Destination site link** will be inserted in Gray Banner.



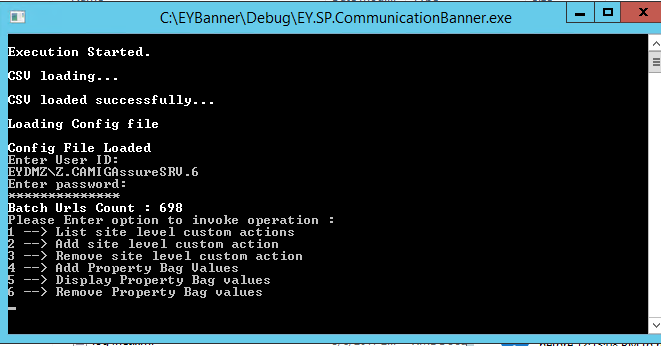
1. Steps to Execute the Utility

* Copy the “inputFile.csv” in the same location as that of the console application “EY.SP.CommunicationBanner.exe”. (The path of the input csv should be same as that of the console application “**EY.SP.CommunicationBanner.exe**”)
* Open the Console Application “EY.SP.CommunicationBanner.exe”.

After opening the console will look as shown in the figure below.



* Now enter the source credentials.



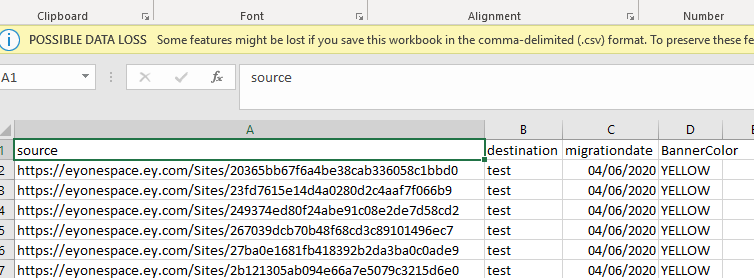
* After entering the credentials, the number of source sites in csv will be displayed as” Batch URLs count” followed by the six options as shown in the figure above.
* Choose the option as per the requirement.
* After the execution is completed a log file will be generated in the Log folder which is present at the same location as that of the Console Application **EY.SP.CommunicationBanner.exe**.

1. How to create input CSV for Banner

The input csv should contain 4 columns with the header names in the order shown below: -

1. Source
2. Destination
3. Migrationdate
4. BannerColor

Save the csv with the name as “inputFile.csv” in current project path EYBanner\_EXE\EYBanner\Debug.



**Notes:**

**Property bag** is basically a hash table of key-value pair options. ... **Property bag** helps you to store meta data as key-value pairs example Connection Strings, file paths, server names and other settings in SharePoint application.