Encompass Loan Message Controller

The Encompass Loan Message Controller is an Encompass plugin that creates a window on the right side of the work pane to display messages to the user. These messages are passive and will not interrupt the users workflow unlike the Macro.Alert() or MsgBox() functions that require the user to click the “OK” button before they can continue with their work. Because these messages are passive they will help reduce popup fatigue, this occurs when the user stops paying attention to the popup messages and just clicks through them with out reading the message. These messages are also stored on the loan, so when the loan is closed and re-opened the messages will remain until they are dismissed by the user or programmatically. These messages can be added anywhere in Encompass where you can update a Database Field.

**Please read the entire guide before making any changes to the config files to familiarize yourself with what everything does.**

# Install the plugin

* Open the Input Form Builder (IFB) application.
* On the top menu bar select Tools -> Manage Customizations
* In the Manage Customizations popup window select the middle tab “Plugins”
* Click the “Upload Button”
* Navigate to where you have the Encompass.Loan.MessageController.dll stored.
* Select the dll and press the “Open” button.
* A popup message will appear confirming your choice to upload the dll.
* The plugin is now installed, close the “Manage Customizations” window but leave the IFB open. We will be returning here shortly.

# MessageConfig.json File

This file will allow you to customize certain aspects of the Message Controller.

**Create the Config File**

* If Encompass is open, close it.
* Open and log into Encompass.
* Once the Home page has loaded close Encompass.
* The MessageConfig.json file has been created and placed in the Custom Data Objects (CDO) repository.

**Modifying the Config File**

* If the IFB is not open, open it now.
* On the top menu bar select Tools -> Manage Customizations
* In the Manage Customizations popup window select the last tab “Custom Data Objects”
* In the list of data objects find the MessageConfig.json file and select it.
  + If you can find the MessageConfig.json file in the data objects list, try closing the Manage Customizations popup window and re-open it following the directions above.
* Press the Download Button and save the file either on your local machine or network drive.
* Using Windows Explorer, navigate to where you saved the MessageConfig.json file.
  + When you open the file it will look something like this:
    - { "HeaderTitle": "Message Center", "MessageColors": [{"Key": 0, "BackColor": -2631681,"TextColor": -16777184,"TitleColor": -16777120, "TitleTextColor": -1},{"Key": 1,"BackColor": -4128832,"TextColor": 16769024,"TitleColor": -16752640,"TitleTextColor": -1}, {"Key": 2, "BackColor": -64, "TextColor": -14671872,"TitleColor": -10461184, "TitleTextColor": -1}, {"Key": 3, "BackColor": -16192,"TextColor": -4680064,"TitleColor": -10485760, "TitleTextColor": -1}], "TriggerField": "CX.Update.Message"}
    - If you have NotePad++, there is a plugin that will format the json so it is easier to read.
      * NotPad++ is a free application that can be downloaded from here:
        + <https://notepad-plus-plus.org/>
    - If you don’t have NotePad++ there are several websites that will format the json for you.
      * Here is one that does a pretty good job:
        + [https://jsonformatter.curiousconcept.com/#](https://jsonformatter.curiousconcept.com/)
    - Once you have the json formatted, save it to the MessageConfig.json file,
      * The plugin doesn’t care whether or not the json is formatted so we will save it formatted so it is easier to read and understand.
    - Your MessageConfig.json should look something like this now:

{

"HeaderTitle": "Message Center",

"MessageColors": [

{

"Key": 0,

"BackColor": -2631681,

"TextColor": -16777184,

"TitleColor": -16777120,

"TitleTextColor": -1

},

{

"Key": 1,

"BackColor": -4128832,

"TextColor": -16769024,

"TitleColor": -16752640,

"TitleTextColor": -1

},

{

"Key": 2,

"BackColor": -64,

"TextColor": -14671872,

"TitleColor": -10461184,

"TitleTextColor": -1

},

{

"Key": 3,

"BackColor": -16192,

"TextColor": -14680064,

"TitleColor": -10485760,

"TitleTextColor": -1

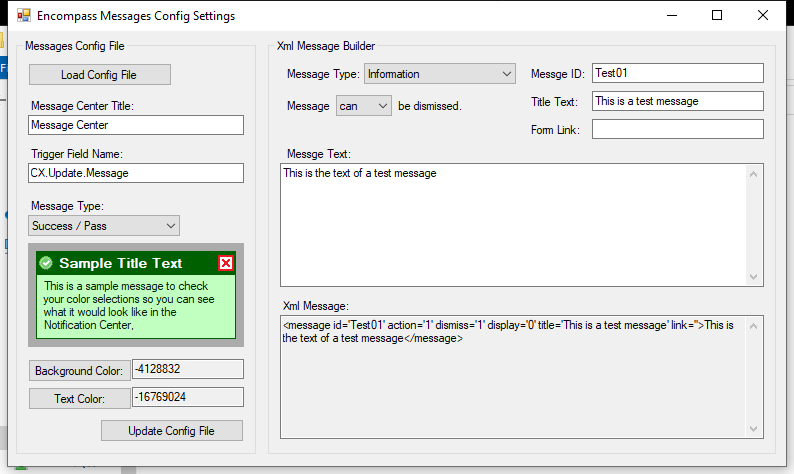
}

],

"TriggerField": "CX.Update.Message"

}

# Encompass Message Config Settings



# Upload MessageConfig.json file changes to Encompass

* If IFB is not open, Open it.
* On the top menu bar select Tools -> Manage Customizations
* In the Manage Customizations popup window select the last tab “Custom Data Objects”
* Click on the “Upload Button” and navigate out to where you save the “MessageConfig.json” file.
* Select the file and press the “Open” button.
* You will get a popup to confirm this action. Press the “OK” button.
* Important note, The config file is read in when Encompass first opens up. Any changes to the config file will not be seen until Encompass is closed and re-opened.
* If at any time you want to revert the config file back to its default, delete it from the Custom Data Objects List. Close and re-open Encompass, the default config file will be re-created.

# Create the Message Center Trigger Field

* If the IFB is not open, open it now.
* On the top menu bar select Tools -> Custom Field Editor
* On the Encompass Loan Custom Field Editor select the add button located in the upper right corner of the screen:
  + 
* Setup your new custom field like below, be sure to match the spelling from the TriggerField property in the MessageConfig.json file.
  + 