Introduction:
The DocuSign Sample plug-in demonstrates how you can send documents from an IBM Enterprise Content Management (ECM) repository DocuSign. Users can then use DocuSign to sign and approve the documents digitally from anywhere and on any device.

The plug-in provides an IBM Content Navigator menu action to get the document signature status from DocuSign systems. In addition, the plug-in uses IBM Content Navigator task manager to create asynchronous tasks that periodically update signature status automatically updates or automatically check-in signed documents and update signature status.

DocuSign prerequisites:
To use the DocuSign Sample plug-in, you must have a DocuSign developer account. If you don't have a developer account already, you can create it at https://secure.docusign.com/signup/develop

From the sandbox, you can find the integrator key or application key that is used to connect to your DocuSign account from any application. For information about your integrator key, log in to https://admindemo.docusign.com/api-integrator-key

IBM Content Navigator prerequisites:

1. Make sure the ANT is installed.
2. Copy all the dependencies into the lib directory.

    navigatorAPI.jar

    com.ibm.ws.prereq.commonj-twm.jar

    ICNTask.jar

    j2ee.jar

    log4j.jar

    navigatorAPI.jar

    taskManagerAPI.jar

    taskServiceClient.jar

FileNet P8 prerequisites:
Before you deploy and use the plug-in, make the following changes to the document metadata in FileNet P8 using Administrative Console for Content Engine (ACCE):

a. Copy the following JAR file into the lib directory
    Jace.jar
b. Create an integer choice list with following settings:
    Symbolic Name: DSSignatureStatusChoiceList
    Type : Integer
    Options : None: 0
    Draft: 1
    Sent: 2
    Completed: 3
    Checkedin: 4
    Voided: 5
c. Create a property definition called DSSignatureStatus with the following settings:
    Display Name : Signature Status

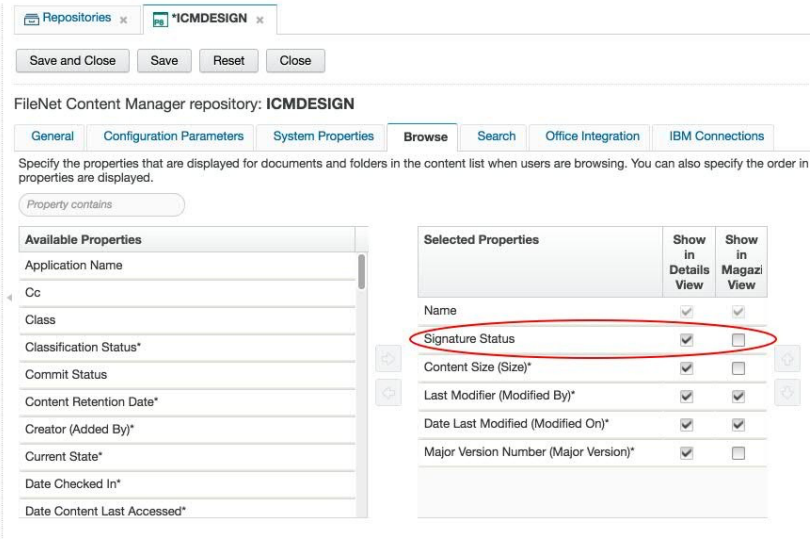
Symbolic Name : DSSignatureStatus
Type : Integer

Associate the DSSignatureStatus property definition with the DSSignatureStatusChoiceList choice list and set the default value to 0.

d. Create a property definition called DSEnvelopeID with the following settings:
   Display Name : Envelope ID
   Symbolic Name : DSEnvelopeID
   Type : String
e. Add the DSSignatureStatus property definition to the Document class.
f. Add the DSEnvelopeID property definition to the Document class as a hidden property.
g. Copy DocuSignPlugin.jar to /opt/IBM/ECMClient/configure/explodedformat/taskManager/taskManagerWeb/WEB-INF/dropins
   The above mentioned path will be different for Windows.
h. Restart WebSphere.

Setting up the DocuSign Sample plug-in IBM Content Navigator

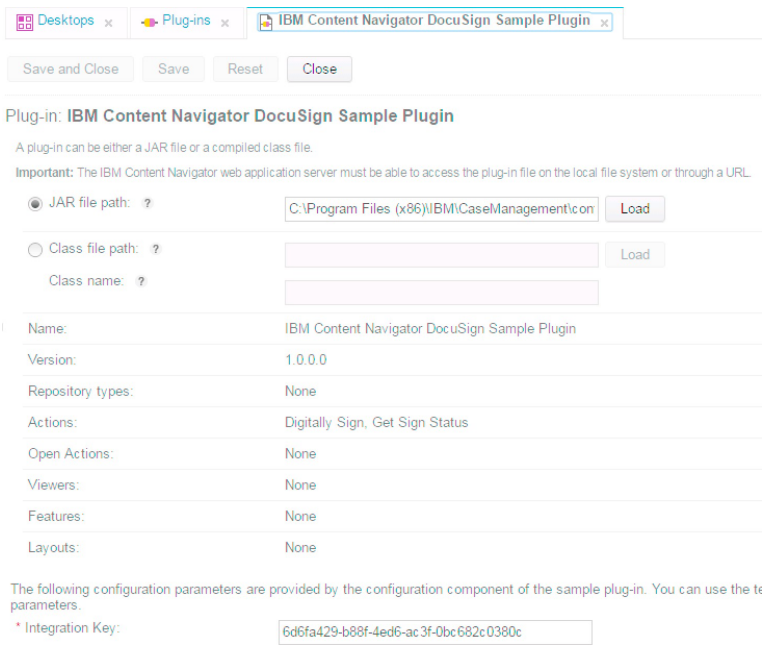
1. In the IBM Content Navigator, open the administration tool.
2. Configure your repository to show the Signature Status property in the Details View as shown in the following screen capture:



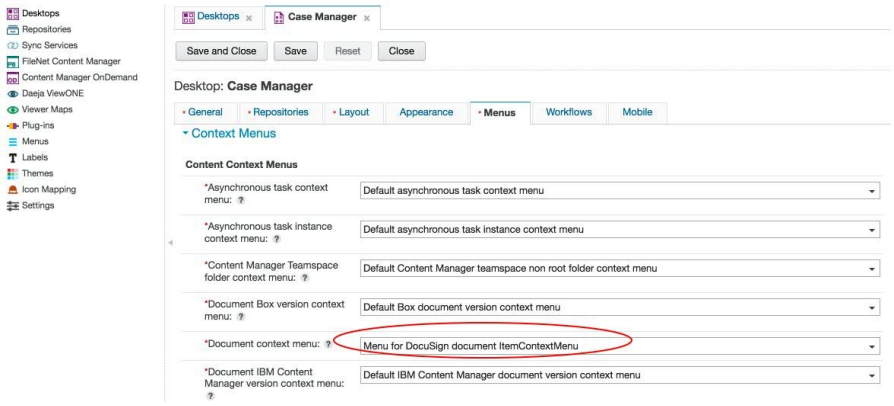
2. Log in to your DocuSign developer account (https://secure.docusign.com/signup/develop).

3. Create a DocuSign template as described in Creating a DocuSign template.

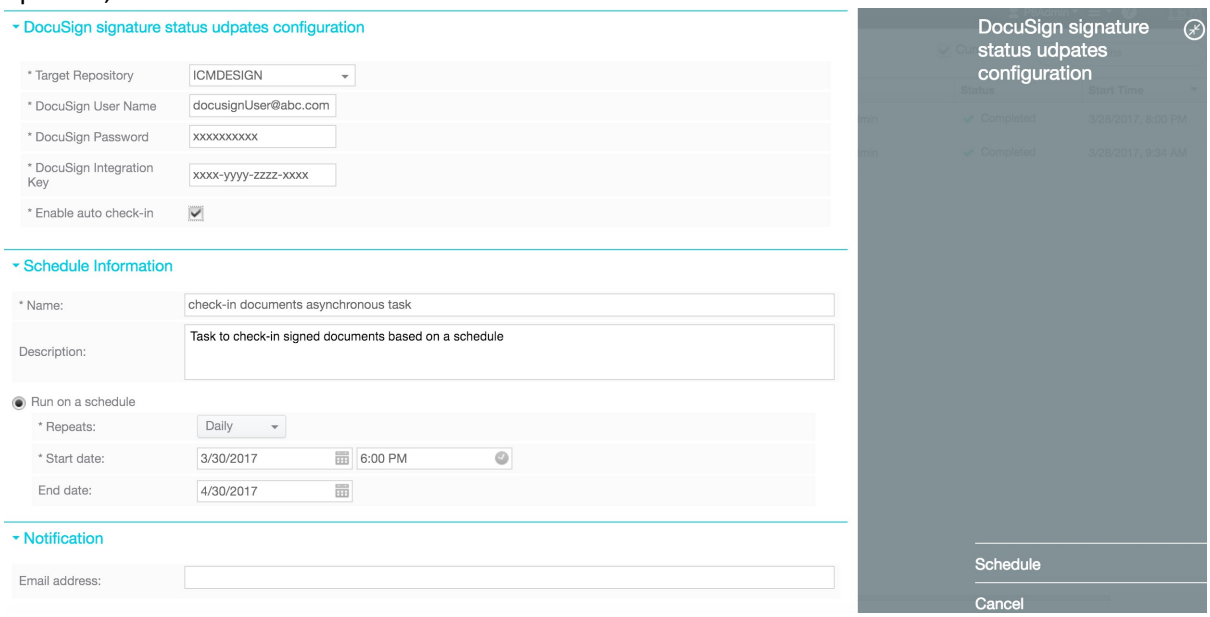
4. In the IBM Content Navigator administration tool, register and configure DocuSign Sample plug-in. You must provide the DocuSign Integration Key as part of the configuration as shown in the following screen capture:



5. Configure the IBM Content Navigator menu actions:
    a. In the IBM Content Navigator administration tool, open the desktop that you want to configure.
    b. On the Menus tab, change the value for the Document Context Menu field to “Menu for DocuSign document itemContextMenu” as in the following screen capture:

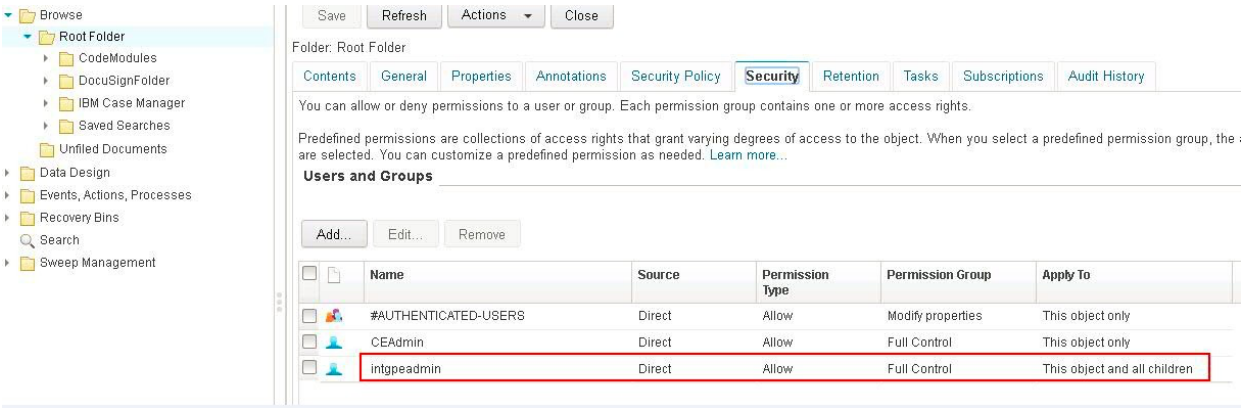


6. Click ‘Open Asynchronous Tasks View’ on Navigator features pane and create a new task ‘DocuSign signature status updates configuration’ from ‘Schedule Task’ drop down list. You can schedule this task by providing the following details. If you want auto check-in of signed documents, select ‘Enable auto check-in’ checkbox. If you want only the signature status to be updated, then do not ‘Enable auto check-in’ checkbox.



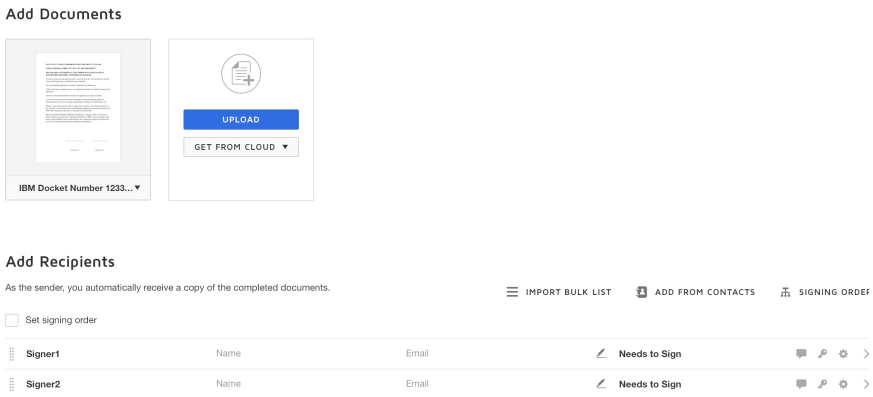
7. Provide appropriate security access to the hidden “DocuSignFolder” folder that the plugin creates for staging the documents. Provide Full Control (or File-in/Un-file) security access permissions on root folder level using ACCE for all the users who would be sending the signature requests to DocuSign.

   Below snapshot shows the security configuration to add "intgpeadmin" user.

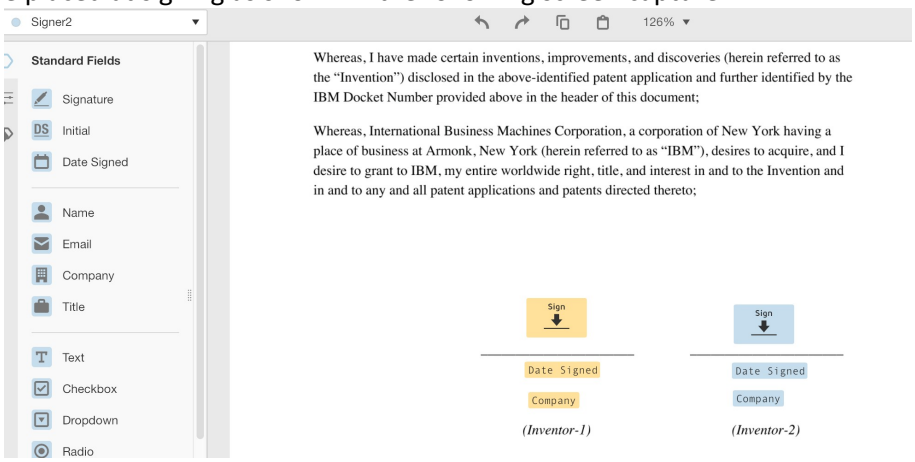


Creating a DocuSign template:

1. On the TEMPLATES tab, select the New option to create a new template.
2. Provide a template name and upload the sample document IBM Docket Number 123344 Assignment Declaration.pdf from the Box share folder for the sample plug-in.
3. Provide a placeholder Signer1 and Signer2 as shown in the following screen capture and click Next:



4. Drag and drop the required Sign tabs from the palette onto the document where the signatures are to be placed at signing as shown in the following screen capture:



5. Click Save and Close the template