Maximo/SCCD 7.5
Integration Framework
Application Import/Export

February 15, 2011
Revised: November 6, 2013 (page 23)
Revised: December 13, 2013 (page 4)

Summary 3
Application Import Processing Behavior 4
Enablement 6
Export/Import Execution 14
Helpful Hints 21
Enabling Import/Export in single-page application 23

Summary

This feature supports the import and export of data directly from applications, such as Asset or Work Orders. This allows users to perform import and export of data without having to have access to the I-F External System application where the Data Import/Export capability exists.

An example use case for this functionality would be where an application user has a need to provide mass updates across multiple rows of the same data. Although the user could go through each row of data (ie. each asset) via the application and update each record, this new functionality allows the user to export the data to a formatted file that can be maintained in an Excel spreadsheet. The user can apply changes across one or more rows of data in the spreadsheet and when completed, the data can be imported back into Maximo. The loading of this data into Maximo processes through the business objects to obtain the majority (not all) of the validations that the data would go through if entered through the application.

Although the implementation of this functionality relies on the integration framework, the intent is that the end user needs no understanding of the framework to import and export data. There is a requirement for an integration administrator to enable the import/export to a specific application.

The application-based import and export functionality relies on the object structure component of the integration framework to define the data content and any processing classes (Java) that are needed to facilitate the exporting and importing of data. No other integration framework components (such as publish channels, JMS queues & end points) need to be configured. An administrator will enable the integration for an application using a provided object structure or by creating a new object structure.

During the import and export processes, a user will have the option to save/pull data using a file. The file format can be either XML or a flat delimited file, such as a .csv file that can be maintained in Excel. All processing occurs in a synchronous manner such that the user will start the import or export process and wait until the process is completed before navigating further in the application.

Application Import Processing Behavior

Although the application import will load data through the business objects like the application UI does, there are cases where the processing behavior will vary. Implementers need to understand and account for these differences when appropriate.

Since application import uses the integration framework's Object Structure component, some of these object structures are designed based on the data coming from an external application such as an ERP system. The best example is the case of the Purchase Order object structure, MXPO. This object structure, when used for importing, is built on the design that the external application is the creator, or 'owner' of the PO and that external application can apply changes as needed. To allow this, when an update for an approved PO comes in through this object structure, the PO will get unapproved, all the updates from the external application will be applied and then the PO will be re-approved. This allows any/all updates to come in from the external application since it is considered to be the 'owner' of the PO. The processing behavior is controlled by the processing class defined on the object structure.

If you use this object structure (or a clone of it) for application import, then an application user will be able to import changes to an approved PO even though they are not allowed to make those same changes using the PO application.

A second processing difference is around Actions (found under the Select Action menu in an application). Integration object structures have support for a limited number of actions, the primary one being 'Change Status'. For those object structures that support 'Change Status', there is no check of the User's authorization, so all integration messages are allowed to change a status. Therefore a user who normally can't change the status of a record through the application, would be allowed to if that user imported data through application import.

In the cases above, both are dependent upon the status field being part of the object structure definition. One way to remove the status change capability in application import is to exclude the STATUS attribute of the object using the object structure application. This can be done by the person who performs the enablement step of application import. This approach can also be used to prevent users from updating any selected fields by application import users.

Other validations that are not supported in application import are those related to a field being 'read only'. An example of this is the Work Order Edit Rules that can be configured from the Organizations application. Importing work orders using application import will not adhere to the Edit Rules.

It is recommended that during the enablement and testing of the application import that you review the update capabilities the end users will have based on the configured object

structure and take necessary actions, such as excluding fields, to meet your business needs.

A final comment to emphasize that Application Import IS NOT a replacement for the queue-based data import feature that is provided by the integration framework. Application import is not intended to load thousands (even hundreds depending upon the data being loaded) of records, since it processes a file load as a single transaction. Attempting to load too large of a volume of records from a single file could lead to out of memory conditions.

Enablement

The enable application-based import and export is done by an administrator who is familiar with the integration framework. There are no applications pre-configured with import or export capabilities out of the box.

The first decision is to determine which users are in need of this functionality and from what applications does this functionality need to be available. Once that is determined, the administrator can begin the process of configuring an object structure to enable importing and/or exporting.

Defining the Object Structure content

Let's assume that there is a business user of the Asset application who has a need to be able to export and import Asset data to a spreadsheet. The first step is to identify an object structure for the application.

- The main (top) object of the object structure must be the same as the main object of the application. For the Asset application, you can verify that the main object is ASSET using the Application Designer application.

Next is the need to identify an object structure where the main object is ASSET. Only object structures with a Consumed By value of INTEGRATION can be used for exporting and importing. One object structure, MXASSET, is provided out of the box and it has ASSET as its main object.

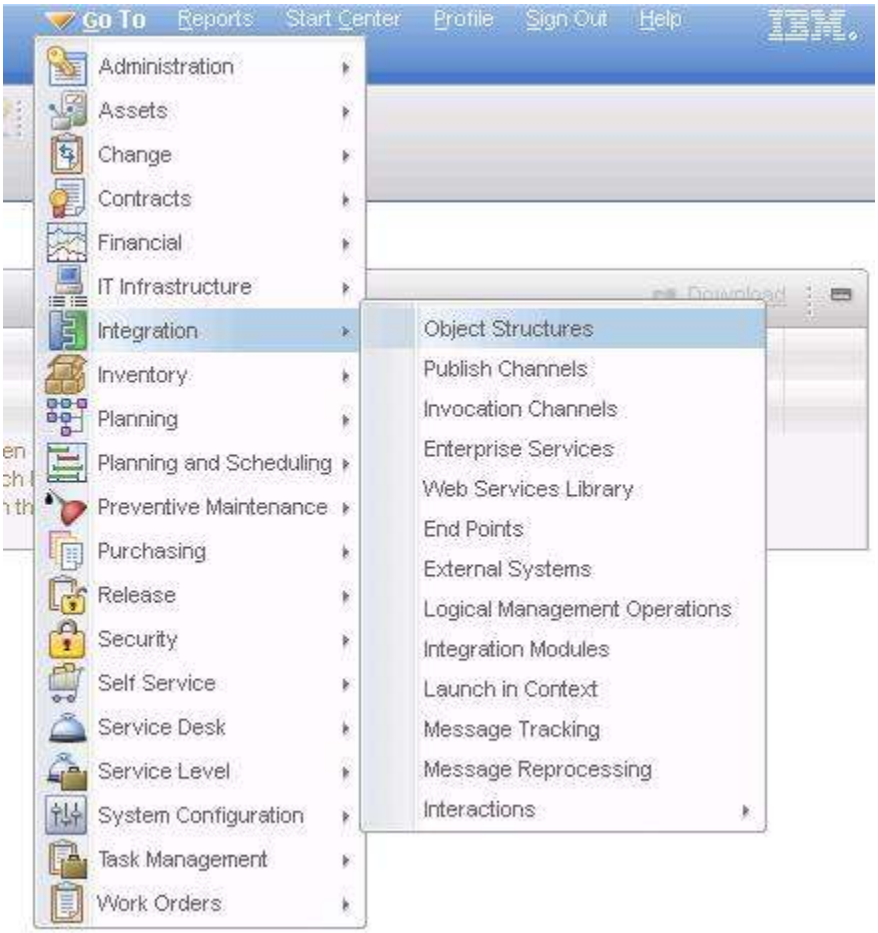
- Using an out of the box object structure is preferable especially when importing is needed as the object structure has been coded (has a processing class) to support importing. Not all object structures require a processing class for importing but many do have them.

Assuming MXASSET is the choice, the next step is to determine if it needs modifying to better suit your users needs. The MXASSET object structure includes the ASSET, ASSETMETER, ASSETUSERCUST and ASSETSPEC objects. If your user only wants data from the ASSET object, then you may want to duplicate this object structure through the Object Structure application.

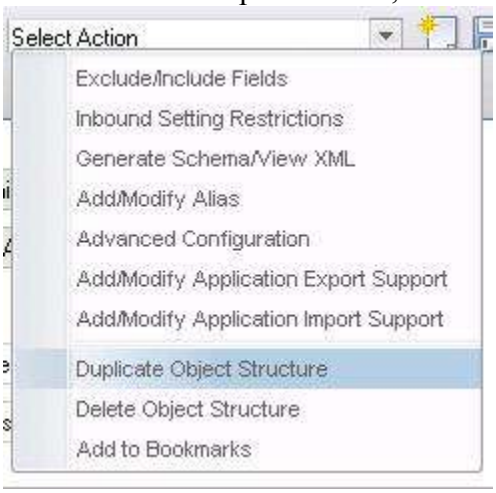
NOTE: If MXASSET is being used in integration scenarios with other external applications, any changes you make may impact those integrations (thus the suggestion to duplicate).

Use the following steps to duplicate the MXASSET object structure:

1. From the Maximo Admin Console select **'Go To' -> 'Integration' > Object Structures'**

a. 

2. Click on the **'Object Structure'** tab
3. Type **'MXASSET'** in the 'Find' dialog.
4. Press Return. The MXASSET Object Structure will be displayed.
5. In the **'Select Action'** drop down box, choose **'Duplicate Object Structure'**.

a. 

Assuming you duplicated MXASSET and created NEWMXASSET, based on your user's requirements, you could delete all the objects except ASSET. Once that is done your user may also want to limit the number of Asset fields that are available. Asset, by

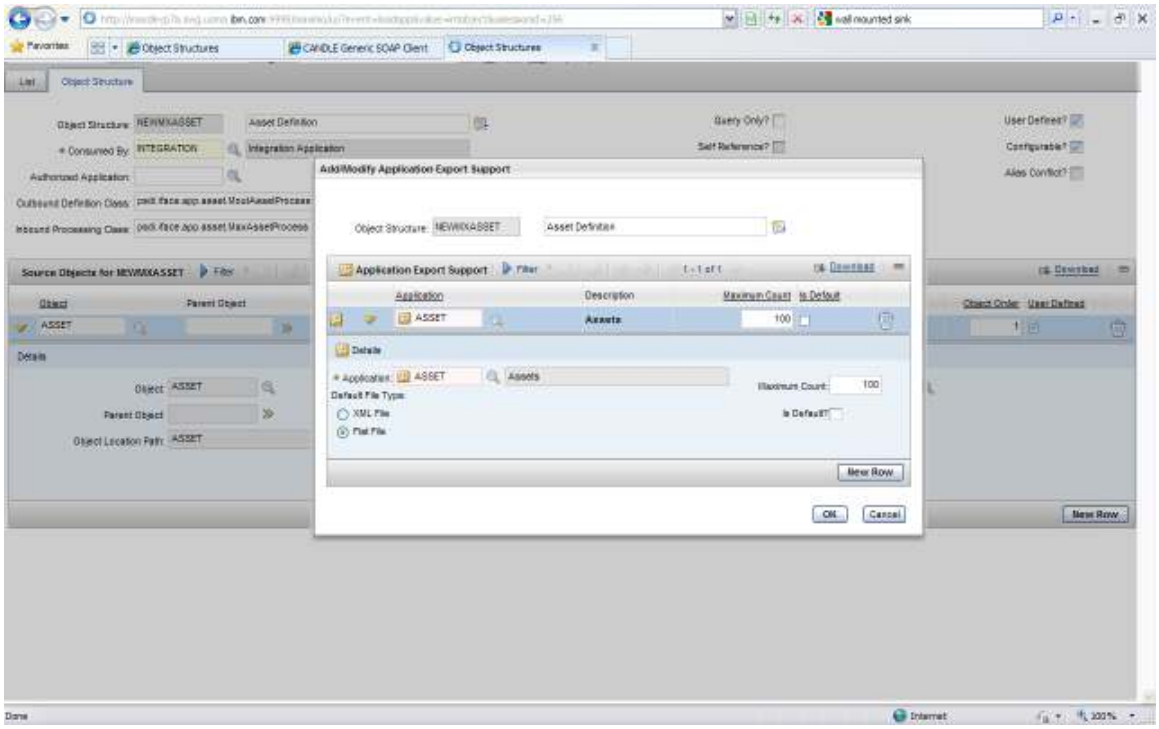
default has over 150 fields in the object and having a spreadsheet with that many columns can be difficult to maintain. Your user may only want/need to use up to 20 fields. To support this, the object structure application has a Select Action, Include/Exclude Fields, which allows the administrator to define which persistent and non-persistent fields are to be included in the data that is exported and imported. Use this option to limit the content to only what is needed by your end user.

If your user plans to use delimited flat files (for use in a spread sheet), then you must configure the object structure to 'Support Flat Structure'. This provides a validation to ensure that every column across all the objects in your object structure has a unique name. If the validation fails, the Alias Conflict flag will be checked and you can use the Select Action, Add/Modify Alias, to assign unique field name where there are currently duplicates. For out of the box object structures, these have been pre-configured to ensure that there are no alias conflicts.

At this point your object structure content is configured and now, as an administrator, you can grant access to this object structure to an application.

Enabling the Object Structure for Exporting

In the object structure application there is a **Select Action, Add/Modify Application Export Support**, which allows you to enable exporting to an application (Save the object structure before using this Select Action). With the NEWMXASSET object structure selected you choose this select action.



In the dialog (above), select New Row and the application will default to an application that has a main object that is the same as ASSET. The Default File Type can be set so that your user will not have to select this object structure each time. So if your users will primarily use the data in a spreadsheet you should select Flat File. The Is Default is relevant when an application is configured with more than 1 object structure. The object structure marked as the default will be pre-filled for the user during execution.

- One scenario where multiple object structures may be desired is if you have multiple users of the Asset application but each wants to work with different sets of asset related data (different columns and/or objects). You may choose to configure different object structures for each of your users.

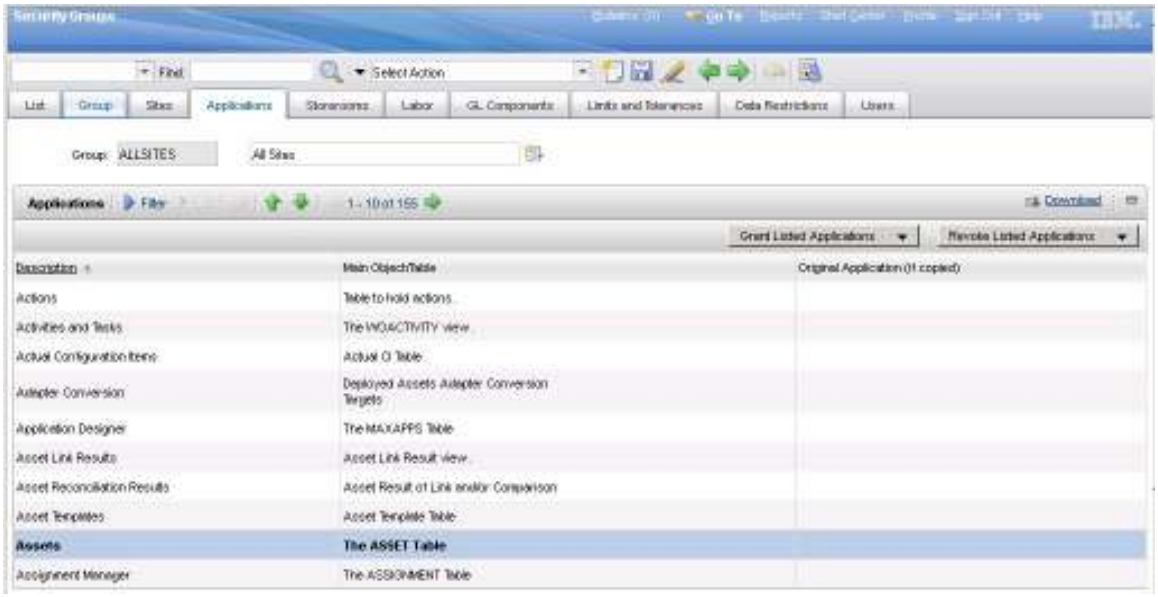
The Maximum Count is a limit to the number of rows that can be exported by the user during a single execution. Even if the user selects 1000 assets, if the limit is set to 100, then only 100 will be exported. This provides a way for the administrator to prevent a user from exporting a large volume that may have negative impact to the system. A value of 0 or null will allow an unlimited number of rows to be processed.

- Keep in mind the application export/import is done synchronously without the use of JMS queues (used in traditional integration scenarios). Excessive processing by a single user could impact other users (performance impact is more likely to come from the importing processing than the exporting process).

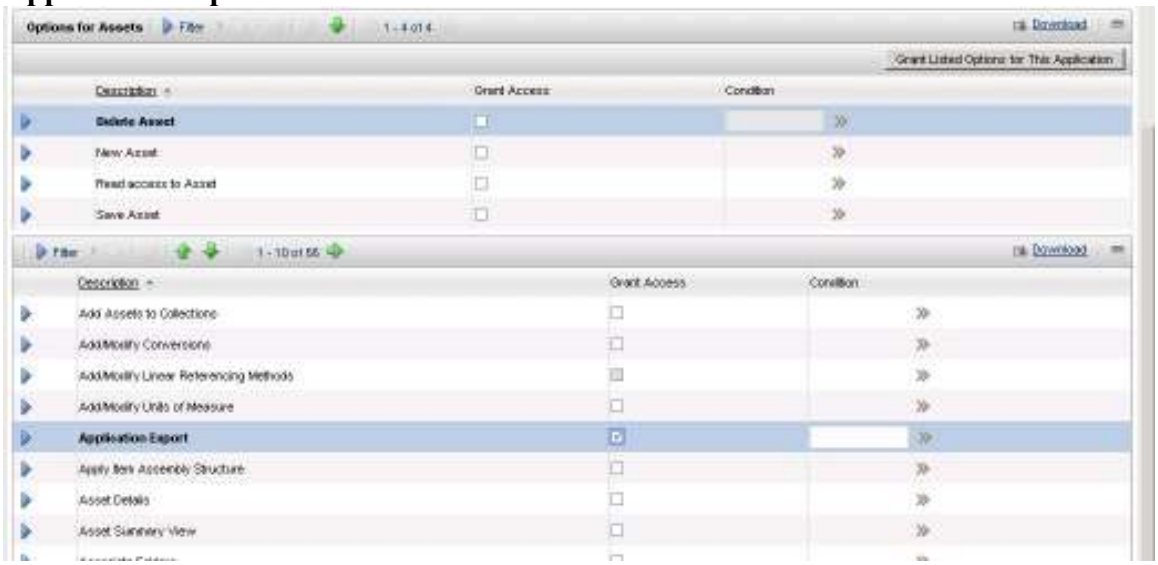
After the data is saved (by clicking the **OK** button), one last step is to Grant Security access through the Security Groups applications. There will be a Security option called Application Export for the Asset application that needs to be granted access.
At this point the configuration for Application Export is completed. If the user doesn't require support for importing then there are no other enablement steps.

Use the following steps to Grant Security access:

1. From the Maximo Admin console select **'Go To' -> 'Security Groups'**
2. Select **'Application'** tag
3. Select **'Asset'** Application

a. 

4. Select **'Application Export'**

a. 

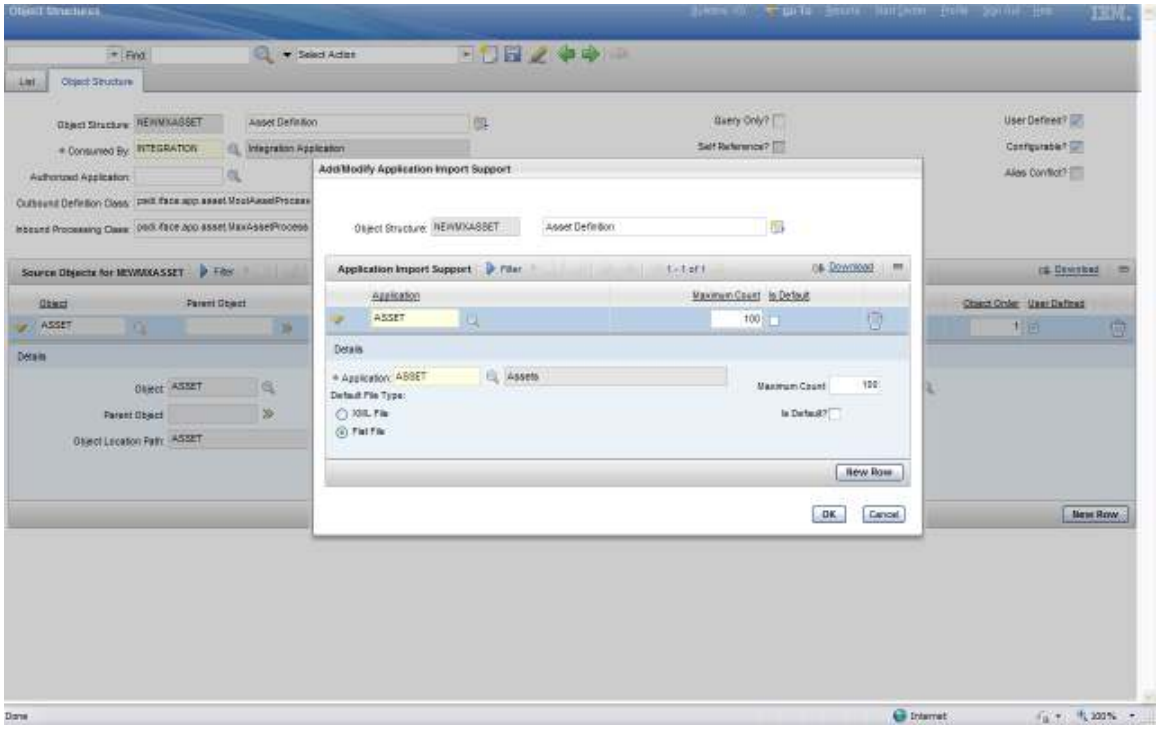
5. Check the **'Grant Access'** box
6. Select the **'Save'** icon

Enabling the Object Structure for Importing

Note: in order to get a sample file to have your users leverage for importing we suggest that you enable the export feature using the same format as is planned to be used for importing.

In the object structure application there is a **Select Action, Add/Modify Application Import Support**, which allows you to enable importing to an application (save the object

structure before using this Select Action). With the NEWMXASSET object structure selected you choose this select action.



In the dialog (above), select New Row and the application will default to an application that has a main object that is the same as ASSET. The Default File Type can be set so that your user will not have to select this object structure each time. So if your users will primarily use the data in a spreadsheet you should select Flat File. The Is Default is relevant when an application is configured with more than 1 object structure. The object structure marked as the default will be pre-filled for the user during execution.

- One scenario where multiple object structures may be desired is if you have multiple users of the Asset application but each wants to work with different sets of asset related data (different columns and/or objects). You may choose to configure different object structures for each of your users.

The Maximum Count is a limit to the number of rows that can be imported by the user during a single execution. Even if the user has an import file with 1000 assets, if the limit is set to 100, then only 100 will be imported. This provides a way for the administrator to prevent a user from importing a large volume that may have negative impact to the system. A value of 0 or null will allow an unlimited number of rows to be processed.

- Keep in mind the application import/import is done synchronously without the use of JMS queues (used in traditional integration scenarios). Excessive processing by a single user could impact other users. Importing a large volume of

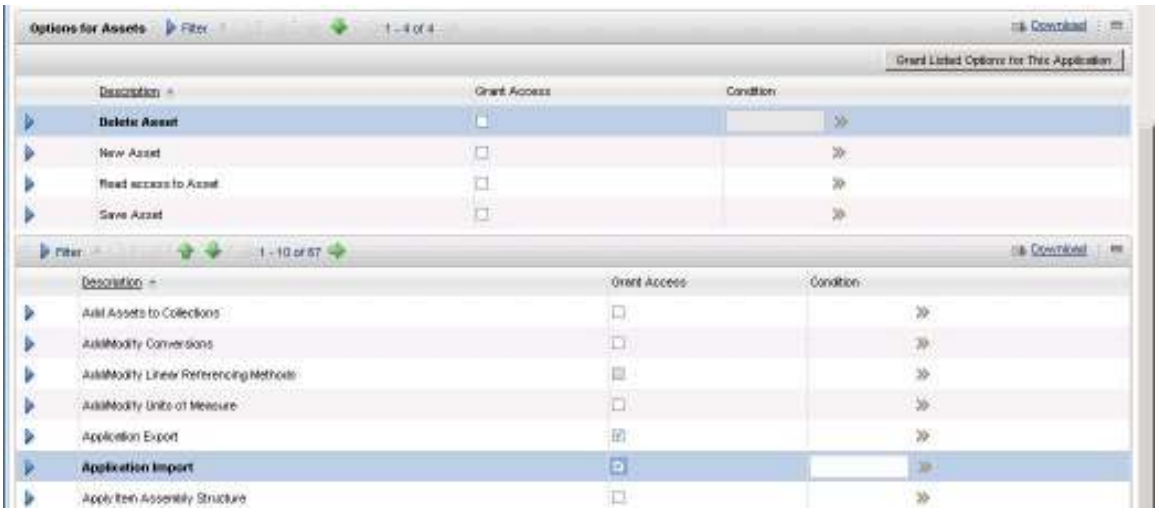
data via a file should be done using the standard Data Import feature of the integration framework. Controlling the volume of data that is imported through application-based importing will help avoid performance problems. Keep in mind the performance of objects can vary significantly as some objects (such as chart of accounts) are considered 'light' and don't have a large number of business validations and related object updates. Other objects (such as work orders) are considered 'heavy' and the number of rows imported during a single execution should be kept to a lower number.

After the data is saved (by clicking the **OK** button), one last step is to Grant Security access through the Security Groups applications. There will be a Security option called Application Import for the Asset application that needs to be granted access. At this point the configuration for Application Import is completed.

Use the following steps to Grant Security access:

1. From the Maximo Admin console select **'Go To' -> 'Security Groups'**
2. Select **'Application'** tag
3. Select **'Asset'** Application
   a. 

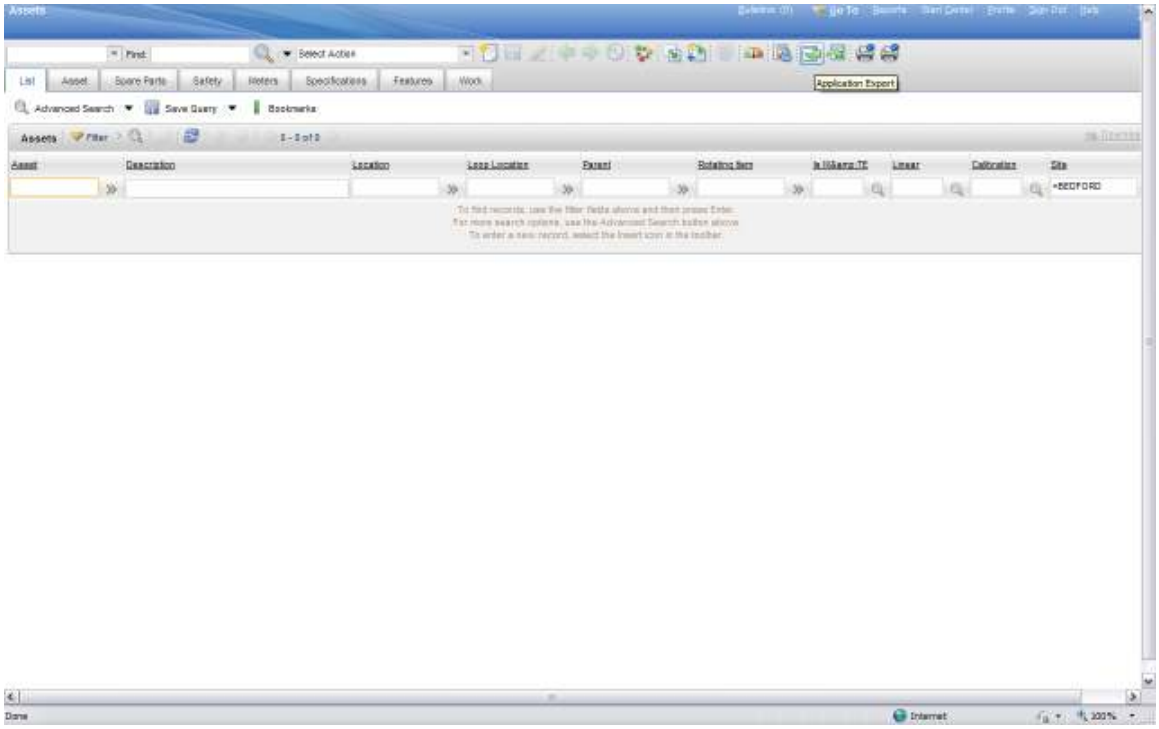
4. Select **'Application Import'**

a. 

5. Check the **'Grant Access'** box
6. Select the **'Save'** icon

Export/Import Execution

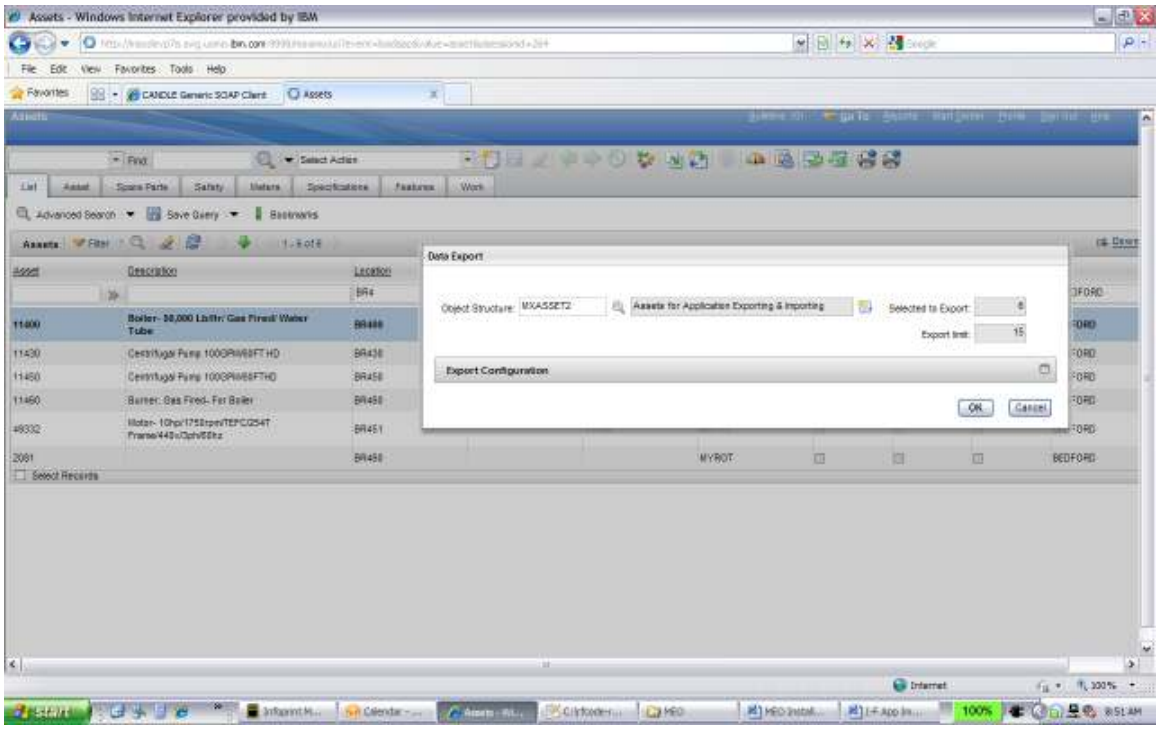
Once the enablement step is complete, the application users will now be able to perform exporting and/or importing of data directly from the enabled application. Continuing with the asset example from above, the Asset application will have two new icons on the tool bar once the security access is granted. **Note: Log off and (re)log in to the Maximo Admin console to display the two new icons.**



In the screen shot above, the two new icons on the tool bar are just before the last two printer icons.

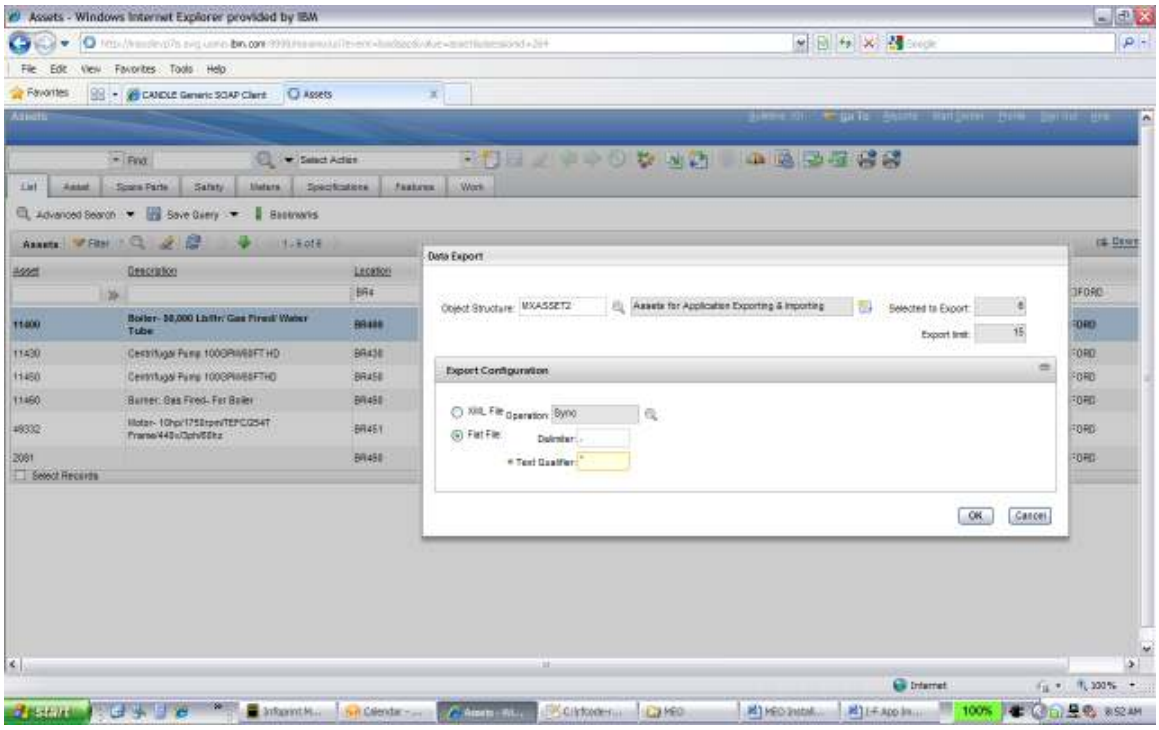
Exporting execution

When Exporting there are two options to drive what data is exported. When a user is on the List tab, the query that is current in the List tab will drive what content is exported. In the screen shot below, there is a filter on the Location field to display only locations that begin with 'BR4'. The list shows just those locations and the export will export just those locations. The second option is when the user is on any other tab in the application. This means the user has selected a specific asset and when export is selected it will export only the selected asset.



In the export dialog above the object structure field is defaulted with the object structure name that was configured as the default during the enablement step. The user may select a different object structure that was also enabled for the application. The Selected to Export field identifies the number of rows selected to be exported. When on the List tab, this will be driven by the query selection, while the value will be 1 for any other tab that the user may be current on. The Export Limit is the configured value entered during the enablement step and will limit the rows exported to this value.

The screen shot below shows the same dialog but with the Export Configuration section maximized. Depending upon how the enablement step is done, a user may not need to open this section. If default File Type is the one the user wants, then there would be no need for updates in this section of the dialog.

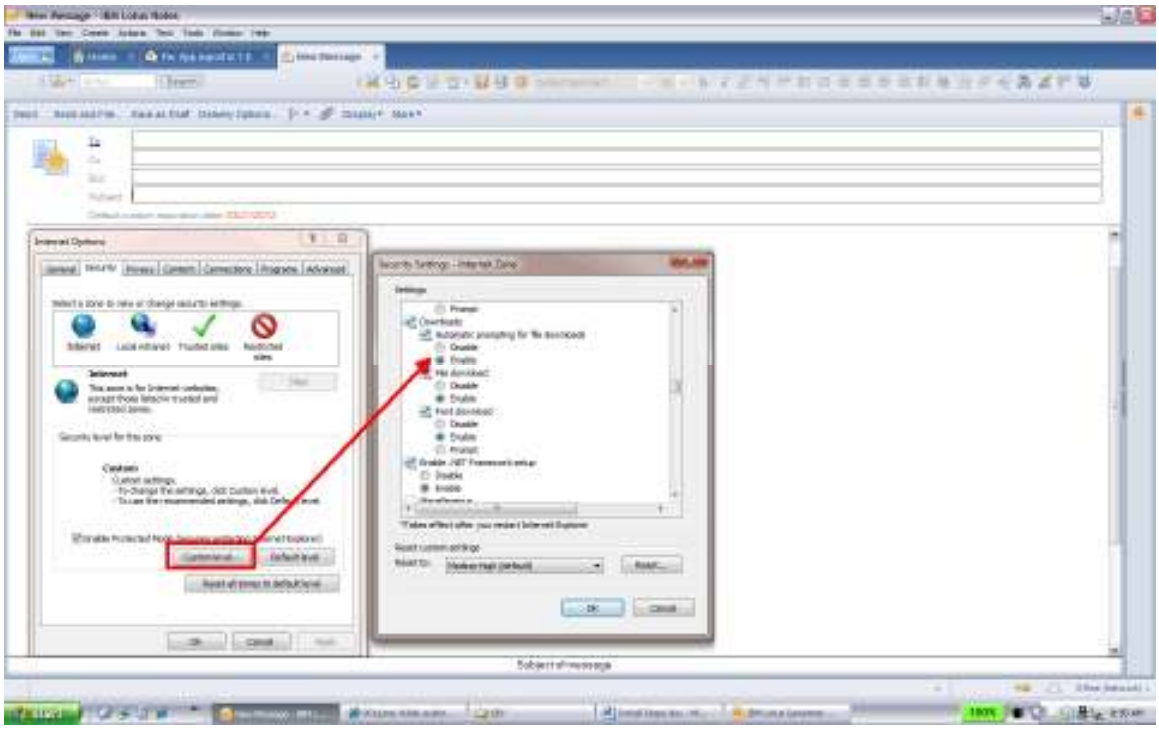


This section does allow the user to choose a file type, XML or Flat when the default is not desired. When XML is selected the user can choose an Operation that will be part of the top and bottom XML tags. The primary purpose of this selection is when there is a plan to export data and then re-import the data back into Maximo. If that is the case the user can select an operation that is valid for importing, such as Sync. The default value is populated from the system property, mxe.int.defaultoperation.

When Flat File is selected the user has the option to select delimiter value such as a comma. The default value is populated from the system property mxe.int.flatfiledelimiter. The text qualifier is used to wrap any occurrences of the delimiter within the actual fields. The default value is populated from the system property mxe.int.textqualifier.

Once the user clicks OK to start the export, a dialog window will appear to allow the user to save the file to a folder of their choice. For a flat file to be opened in Excel, you can change the file name to use the .csv suffix.

NOTE: when you export, you should get a Save dialog, if you do not it could be related to your browser's configuration. The following IE configuration is needed to get the save dialog (else it will open/close immediately)

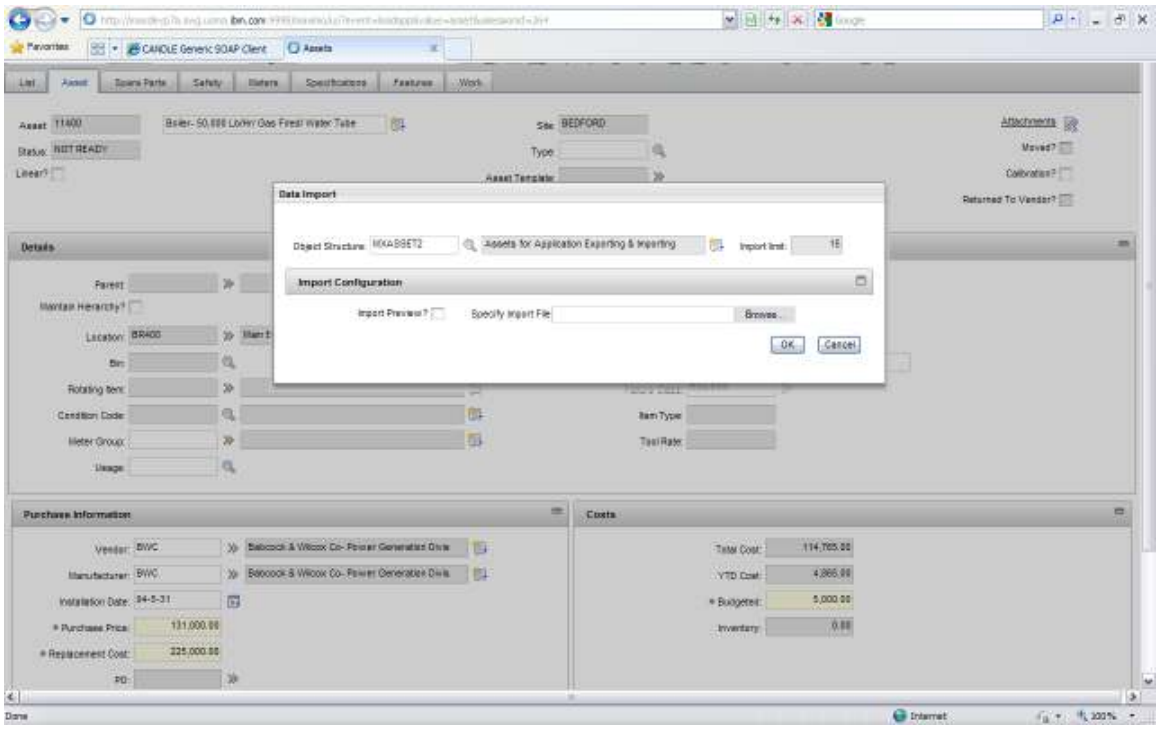


Under Downloads, Enable automatic prompting for file downloads.

If no save dialog is presented (i.e. the browser is not configured to prompt for a location) the exported file is saved to the default application server directory for your J2EE server within the 'flatfiles' or 'xmlfiles' directory depending on the file type that was chosen. If you have configured the Integration Global Directory property, the file will be saved in a subfolder of that directory. Importing execution

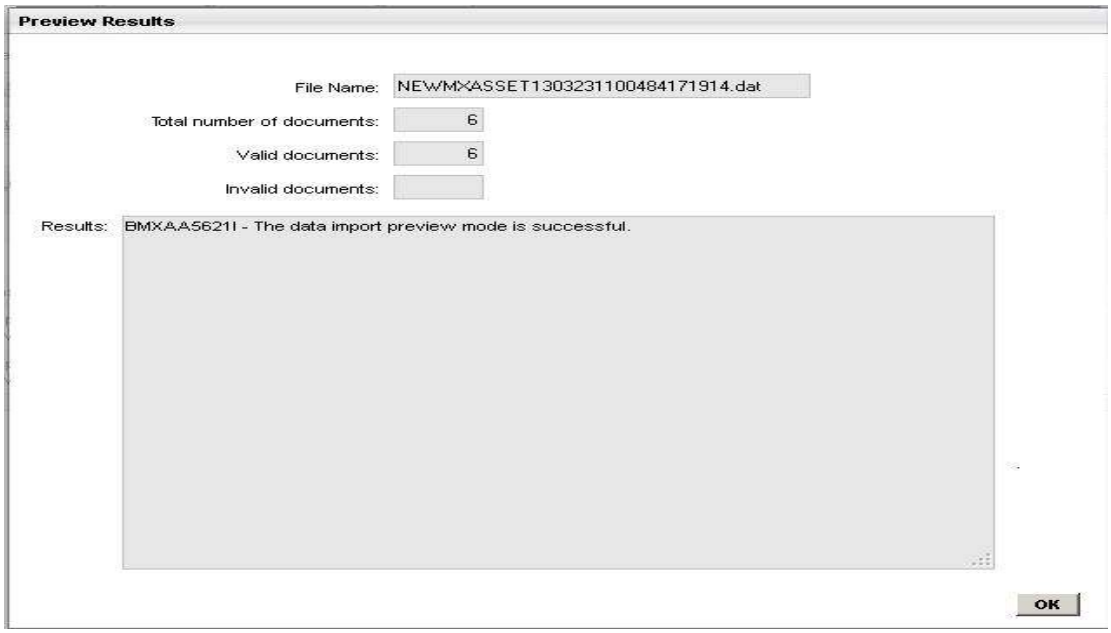
When Importing there is no impact on the processing based where the user currently is on the application (differs from exporting) since the data in the input file is going to drive what is loaded to Maximo.

On Firefox: Use Tools - Options – Content tab – unchecked Block pop-up windows. This will enable the download dialog to open.

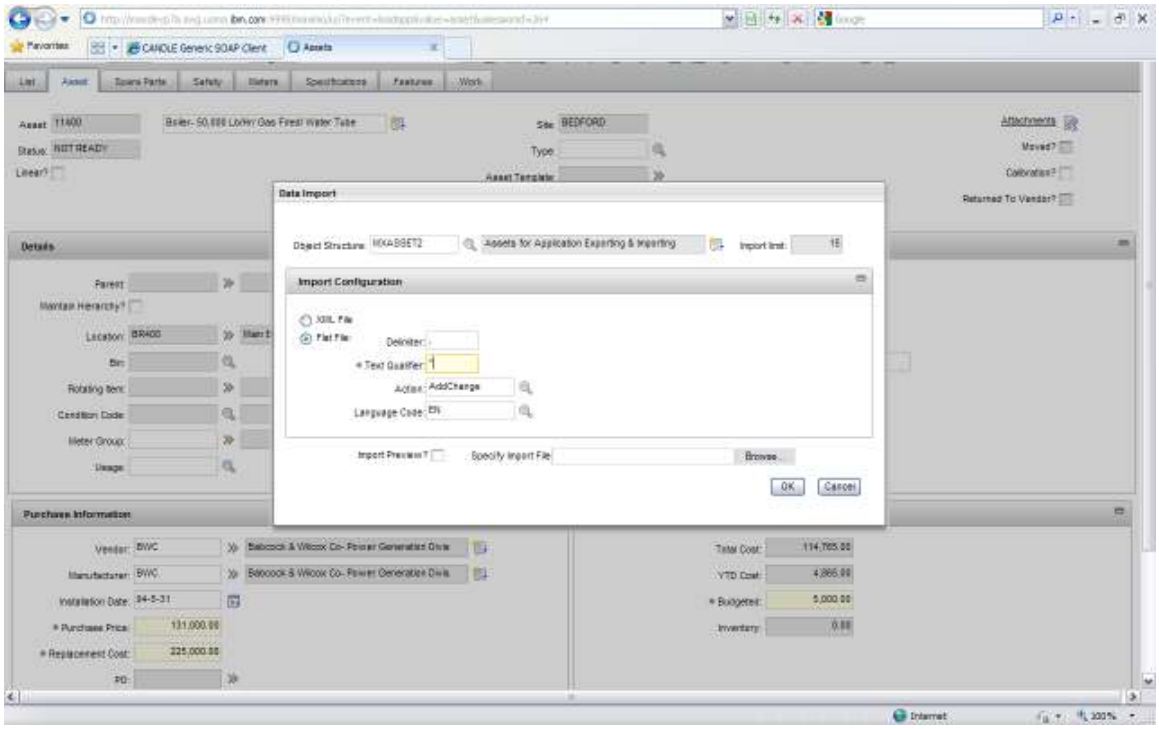


In the import dialog above the object structure field is defaulted with the object structure name that was configured as the default during the enablement step. The user may select a different object structure that was also enabled for the application. The Import Limit is the configured value entered during the enablement step and will limit the rows imported to this value. If the selected input file has 200 rows but the import limit is 100 then only the first 100 rows are imported.

The user needs to specify the file to import. The Preview Mode will process the data through the business objects but will not save the data. This allows the user to 'test' their data load to validate that there are no errors prior to loading. If errors are hit, they will be displayed and the user can apply corrections to their input file and attempt to re-import. When loading data, keep in mind that the processing is synchronous so the user will 'wait' in the UI until the load is complete and a confirmation message appears. If the load encounters errors, no data is loaded as the file is processed as a single transaction with a single commit. The screen shot below shows a successful import preview:



The screen shot below shows the same dialog but with the Import Configuration section maximized. Depending upon how the enablement step is done, a user may not need to open this section. If default File Type matches the file format to be loaded then there would be no need for updates in this section of the dialog.



This section does allow the user to choose a file type, XML or Flat, when the default is not desired. When XML is selected there are no other fields to provide as the message operation, language code and action code are provided as part of the xml file.

When Flat File is selected the user has the following fields to populate:

- Delimiter value such as a comma. The default value is populated from the system property mxe.int.flatfiledelimiter.
- Text Qualifier is used to wrap any occurrences of the delimiter within the actual fields. The default value is populated from the system property mxe.int.textqualifier.
- Action value such as a AddChange. The default value is populated from the system property mxe.int.defaultaction.
- Delimiter value such as a comma. The default value is populated from the system property mxe.int.flatfiledelimiter.
- Language Code value such as a EN (English). The default value is populated from the base language of the application.

Helpful Hints

XML file format – The XML file format provides the means to identify the operation, action and language code directly in the XML, thus no need to enter via the application. In the example below the operation is Sync, the transaction language code is defined using the transLanguage attribute and the action attribute is provided on the main object (asset).

SyncNEWMXASSET xmlns="http://www.ibm.com/maximo" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" creationDateTime="2010-12-21T10:26:36-05:00" **transLanguage="EN"** baseLanguage="EN" messageID="1292945196843485728" maximoVersion="7 1 20100522-0325 V7117-47" event="0">
<NEWMXASSETSet>
<ASSET **action="AddChange"**>

Valid operations are Sync, Create, Update and Delete. Sync is most common as it will support create, update and delete based on the provided Action code. The transaction language code must be a valid language code and this provides support for non-base language values for fields that are configured for multi-language support.

Action Codes (apply to both XML and Flat files) are only applicable when the operation is Sync. Valid action codes include Add, Change, Delete, Replace, AddChange and null. A null value is comparable to a Replace. The two most commonly used are Replace and AddChange. Both of these will Add the data if it doesn't already exist or update it if it does. However there is one significant difference. Replace will delete any child objects that are not provided in the input data. For example, if currently in Maximo there is a PO with 5 POLINES (child object) and a PO message comes in for that same PO with an action of Replace and only provides lines 1, 3 and 5; the PO and lines 1,3,5 will be updated while lines 2 and 4 will be deleted. For the AddChange action, the processing will be the same except lines 2 and 4 will not be deleted (left untouched). Providing an Action code on a child object (not the top object of the object structure) is only supported using XML files (not flat files. An action code on a child object is only valid (and used during processing) when the top object has an action code of Change.

Key Fields - The key fields of an object is used to 'find' the record currently in the database. The key fields for an object are the primary key columns of that object (as defined through DB configuration). If an object has an alternate key defined, then the fields of that alternate key index will be used as the key fields when 'finding' the record. An alternate key can be defined at the object level (within DB configure) or at the object level within the object structure (via the object structure application). Keep in mind, updating the alternate key at the object level would impact all integration scenarios using that object.

AutoKey Fields

The first step is to configure the Auto Key capability through the Maximo applications. For the data loading:

XML - When wanting to use the AutoKey feature to populate a key field, such as using an autokey value when creating a new asset, you should not provide the Asset Number in the XML input file.

FLAT – The loading framework is XML-based, thus when loading flat data the first step by the framework is to convert the flat structure to a hierarchical XML structure. To do this requires that the key fields be provided to properly form the XML. So even if you are planning on using the Auto Key to create your asset number, your input flat file must contain a unique asset number for each row to support the Flat to XML conversion. To then restrict the load process from using the value in the file (instead use the autokey value), the key field (assetnum) needs to be configured as Set Restricted. The Set Restriction can be defined at the object level (within DB configure using the Restrict Attributes select action) or at the object level within the object structure (via the object structure application using the Inbound Setting Restrictions select action). Keep in mind, updating the setting restriction at the object level would impact all integration scenarios using that object.

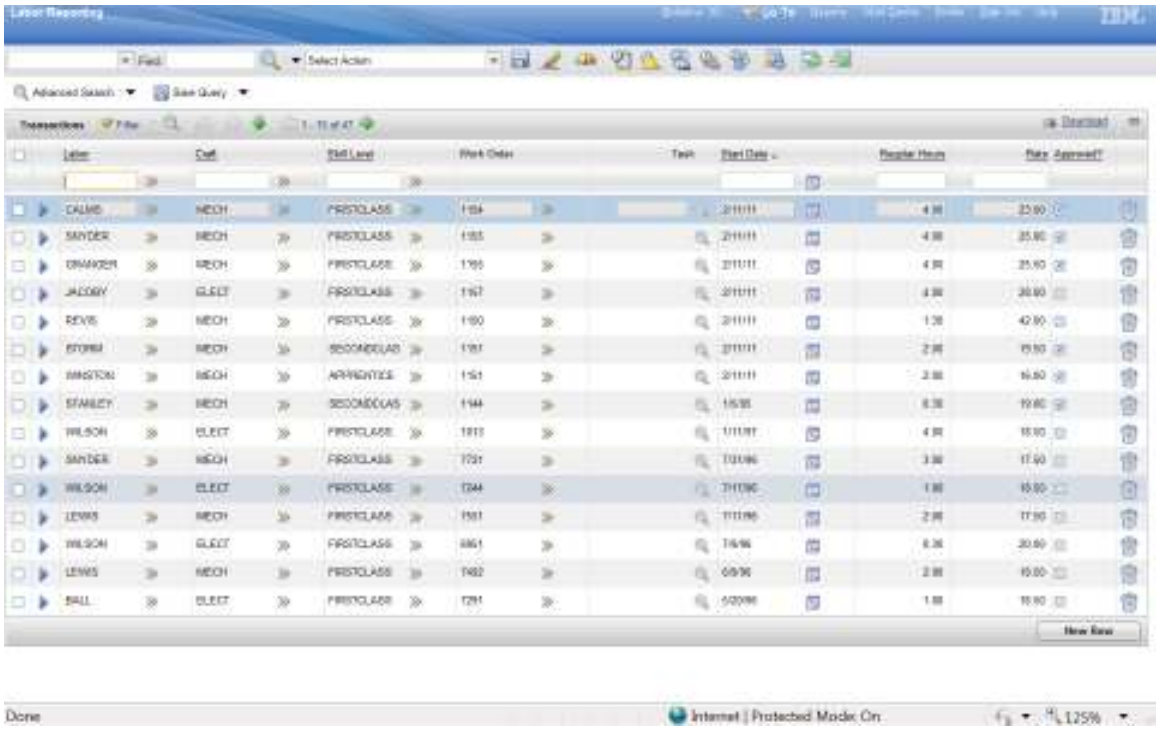
Trailing Commas (Flat File only)

When using Excel and saving data to be imported, there are times where Excel doesn't retain a comma delimiter for each column when the column does not have any value. When this occurs you will get errors due to a mismatch in the number of columns provided for a row of data. A simple way to avoid this is to make the last column of your spreadsheet be a column that has a value for every row.

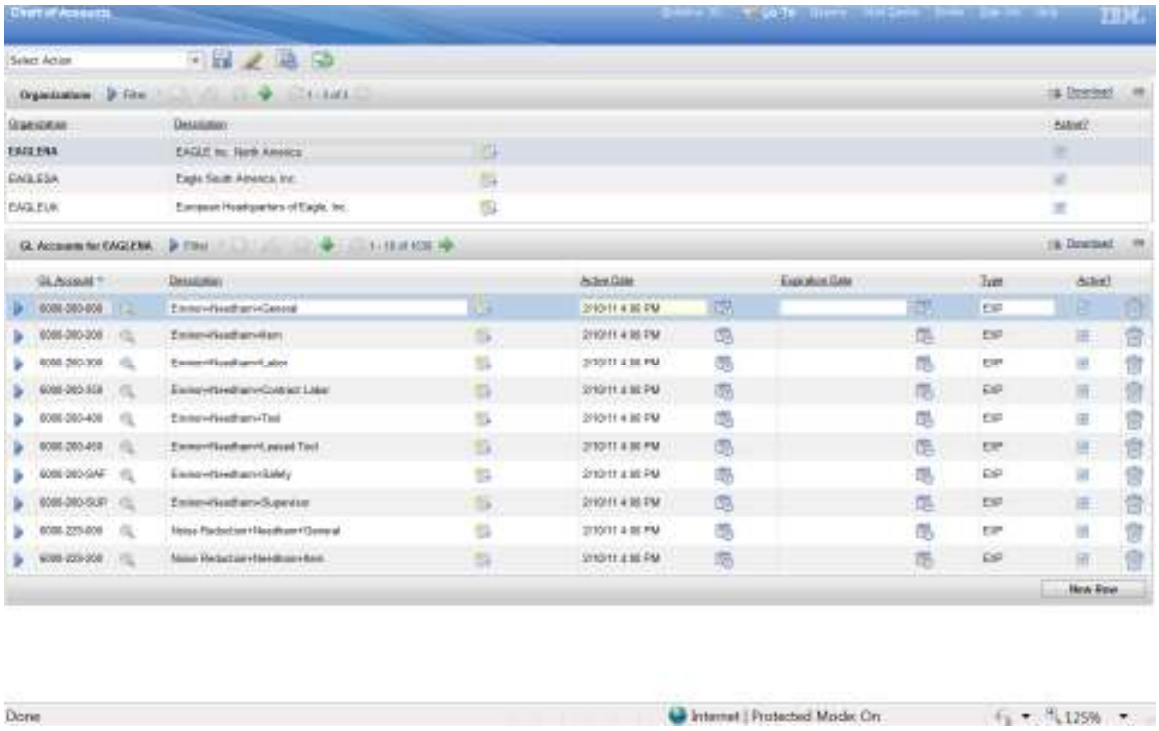
Enabling Import/Export in single-page application

The enablement described above covers the majority of applications that follow the model of having a List tab for the viewing of multiple rows of data and then a ‘main’ tab that displays an individual row of data.

In 7.5.0.5, support was added for a single-page app where this is just a simple List of rows of data, an example of this is the Labor Reporting application:



However there are other variations of a single-page application that may not be supported. Below is a description of how to enable import/export support for the Chart of Accounts (COA) application. The COA application is similar to the Labor Reporting app with one exception, the list of COA data is dictated by the selection of an Organization that is displayed at the top of the page:

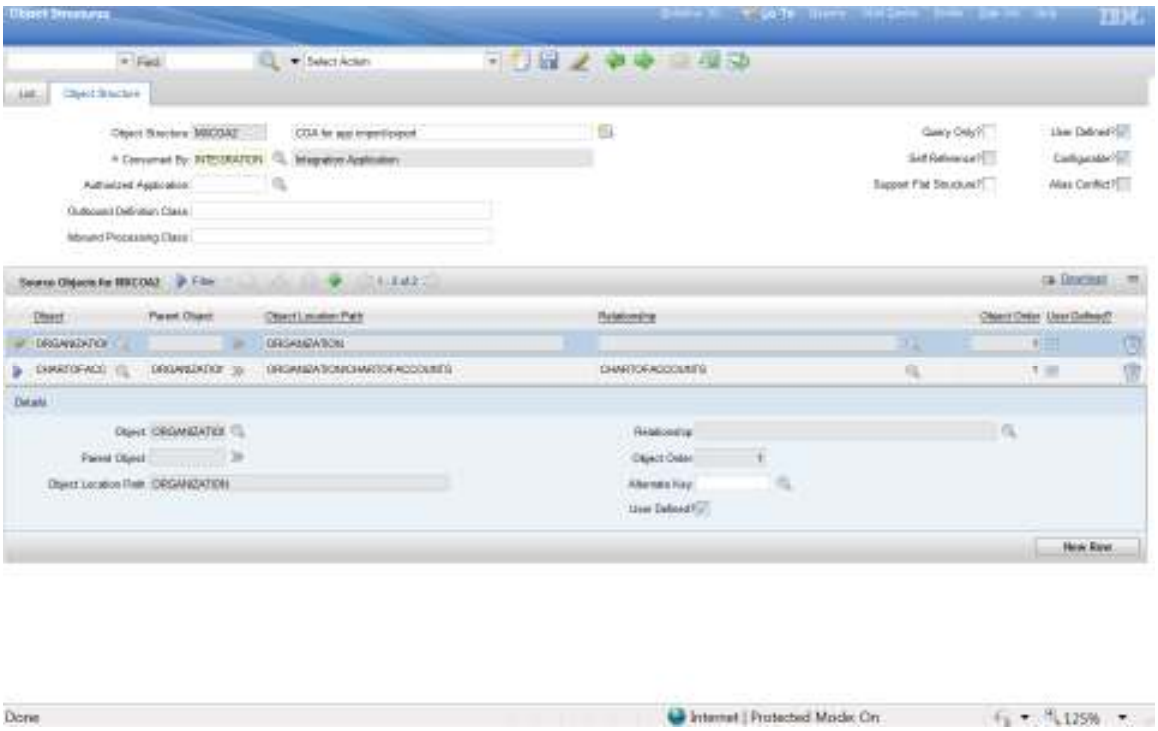


In the above screen, the top table is a list of Organizations and when one of those rows if selected, the child window is populated with the list of chart of accounts for that organization.

In order to support application-based import/export, varying configurations are needed between supporting Export versus Import. The steps below cover both.

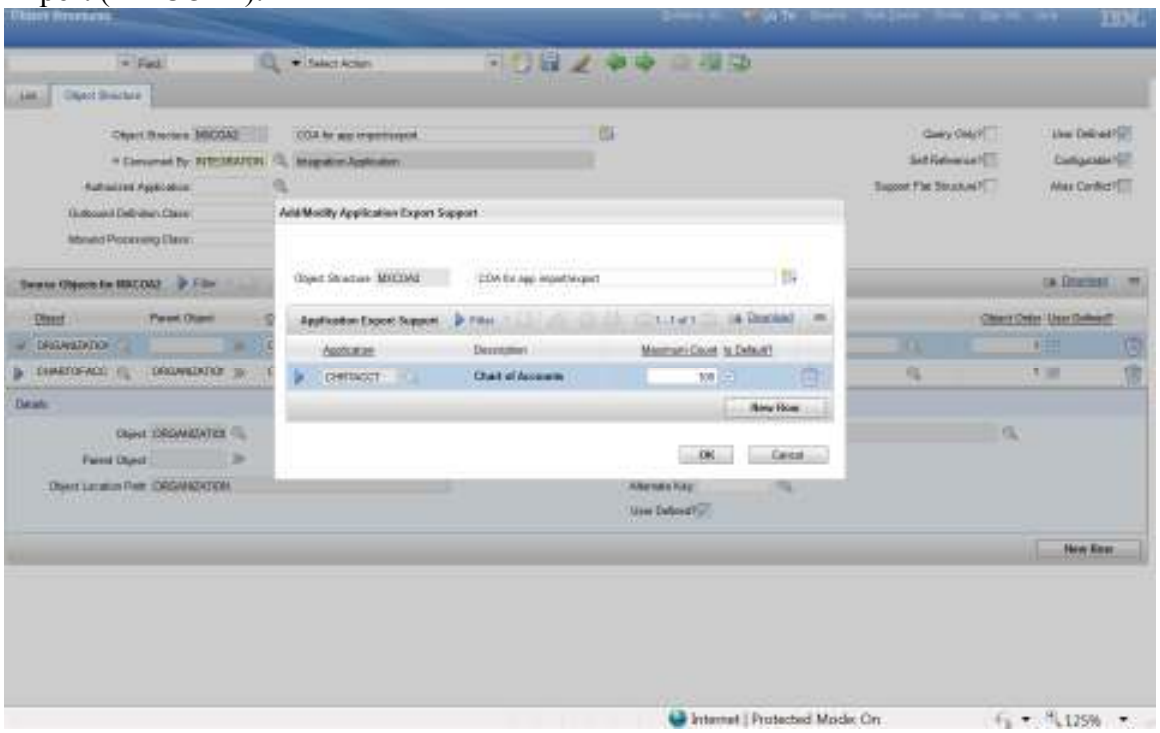
App Export

1. There is an existing object structure, MXCOA, which supports the import/export of COA data. However this object structure does not include the Organization object. In order to support exporting of multiple chart of accounts you will need to create a new object structure with both the organization and chart of account objects. Below is an example of the new object (MXCOA2) structure required. . If you plan to export the data into a delimited file format (.csv), you will need to check the Support Flat Structure checkbox

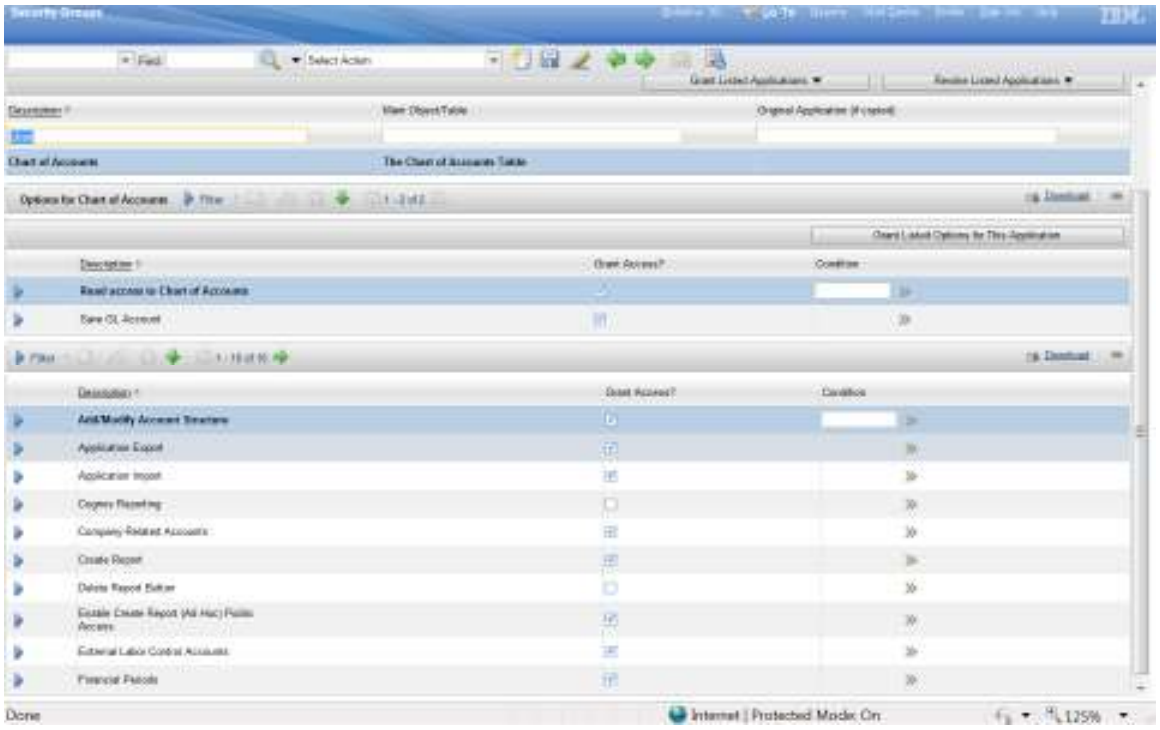


2. With your object structure defined, you can use the Select Action options in the object structure application to enable Exporting:

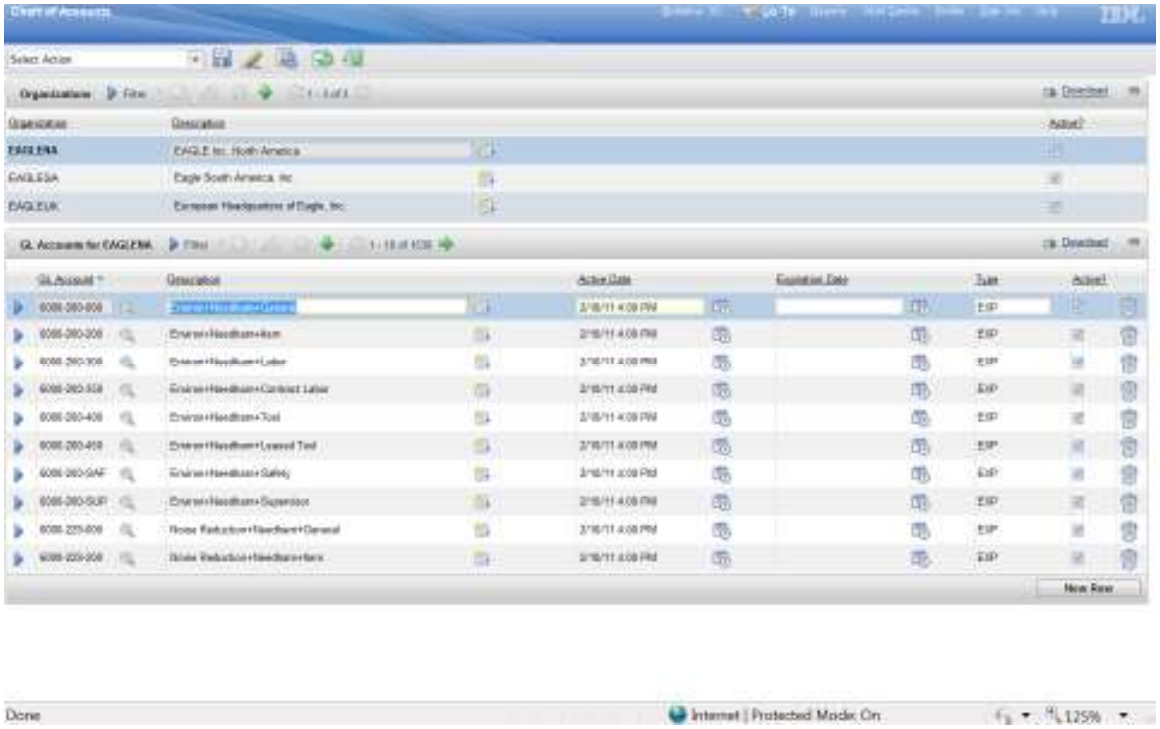
   Export (MXCOA2):



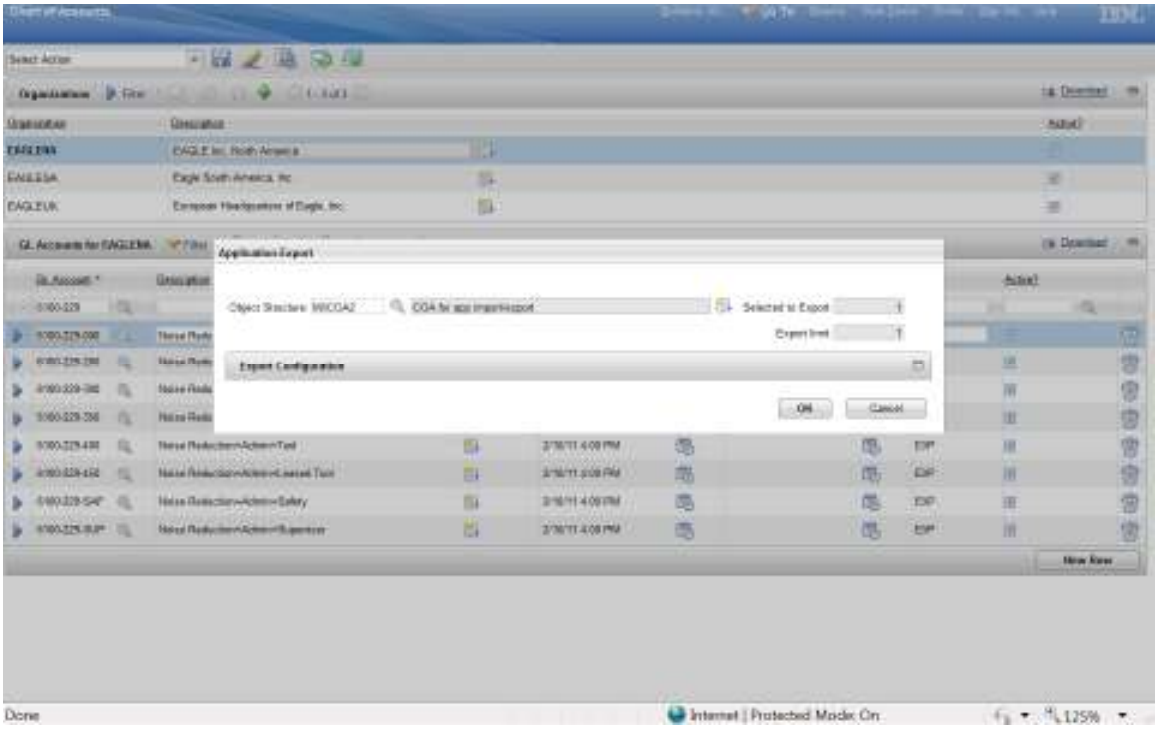
3. As with other applications, the next step is to grant access to these new options in the appropriate group using the Security Groups application:



When you access (after logging out /logging back in) the Chart of Accounts application you will see the import and export icons on the tool bar.



4. When you Export, this behaves a bit different than other applications. You need to focus on a single Organization and then you have the option to filter down the list of COAs displayed in the bottom window (to export just those accounts you want). When you select the Export icon, the following dialog displays. Note that the **Export Limit is set to 1** – this is tied to the Organization being selected. The export limit **does not apply** to the chart of account data. It is important that you filter the COA list to only the data that you want. The Export Limit configuration that you can do during enablement is not applicable, so it is the **responsibility of the user** to limit the result set. Selecting too much data in one export could result in long-running queries and a possible application timeout.



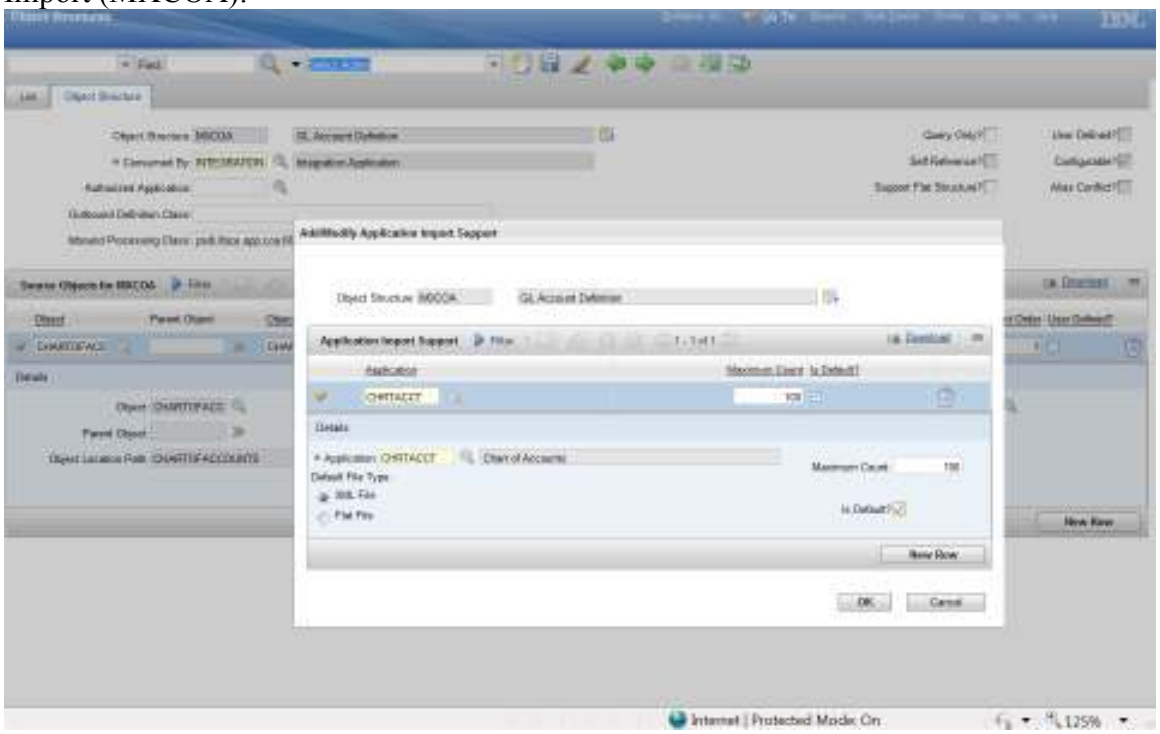
App Import

When you Import, you can use the out of the box MXCOA object structure and the import will behave in a similar manner to all other applications (as Importing is described earlier in this document). You do not have to select an Organization like you do with exporting. The content being imported is driven strictly by the content in the file being loaded. The Import Limit will apply based on the configuration during enablement (in the object structure application).

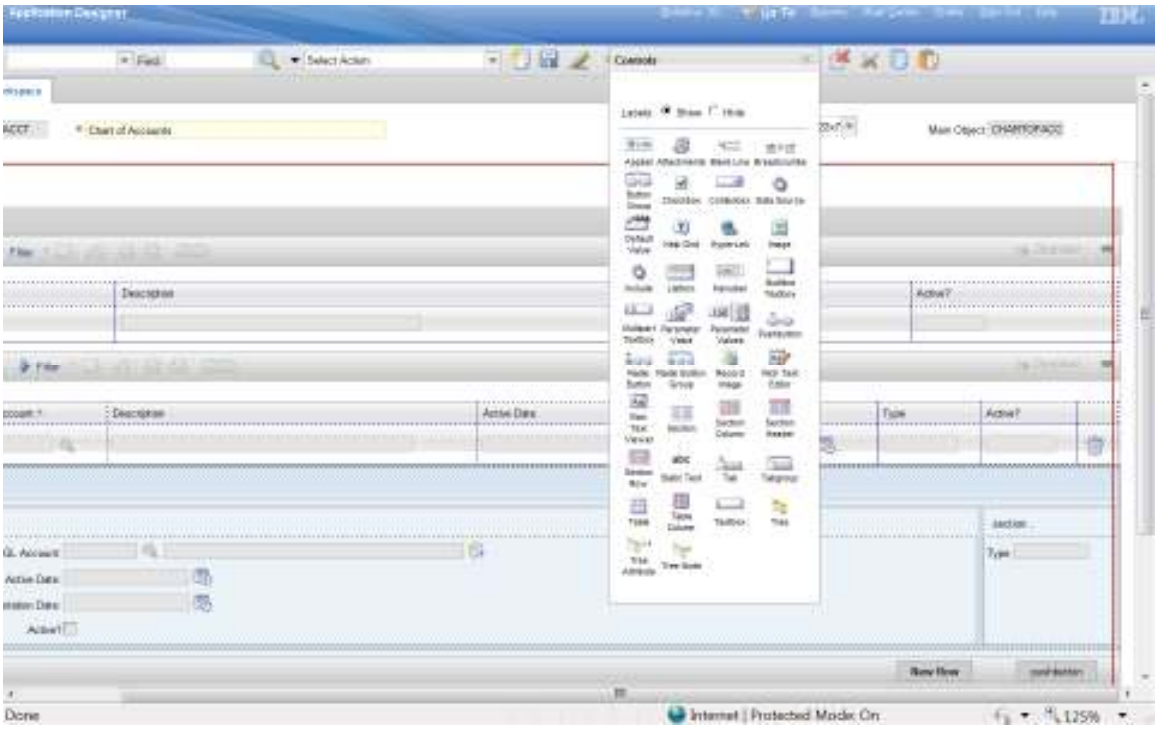
Since the Chart of Accounts application has the Organization at the top of the app, you will need to do some manual configuration to get import to work. The primary change is to implement the Import Icon on the table window where the COA data resides, rather than on the toolbar at the top of the application. To do this follow these steps:

1. In Integration Object Structures application add application import support. This will add the Import tool bar button which will not be used and will be deleted later. The reason we do this step is because it also adds the underlying sigoptions and applicationauth records needed for this functionality.

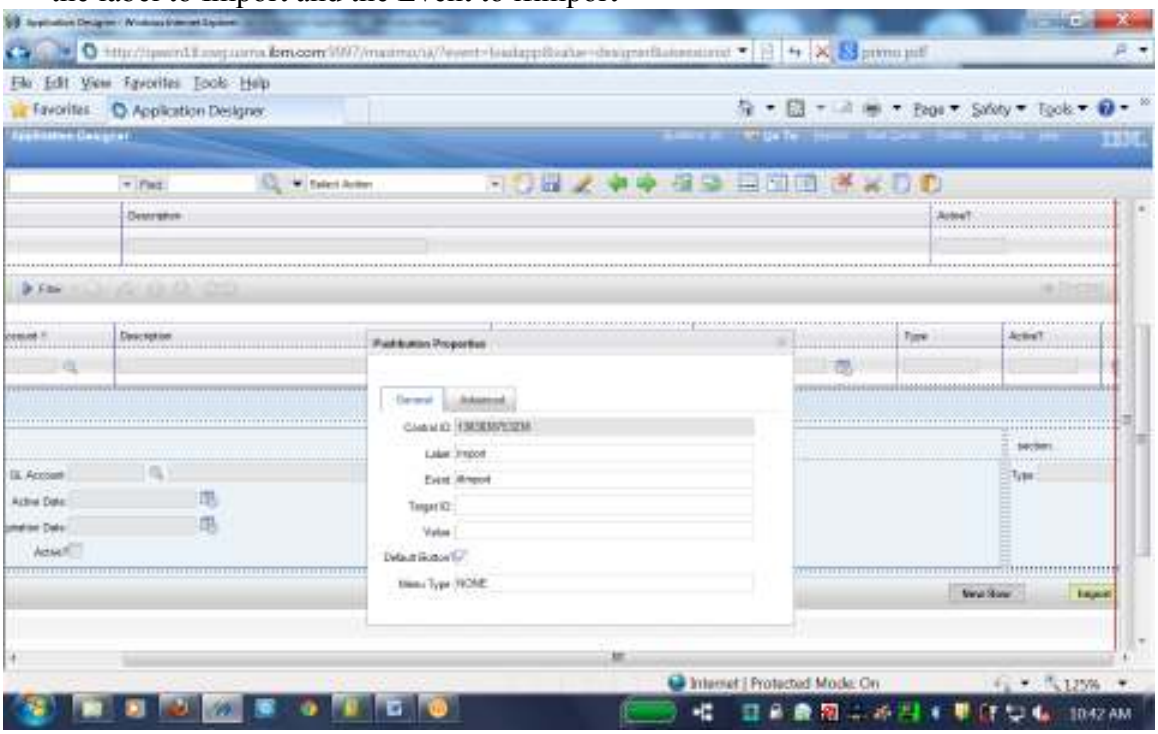
   Import (MXCOA):



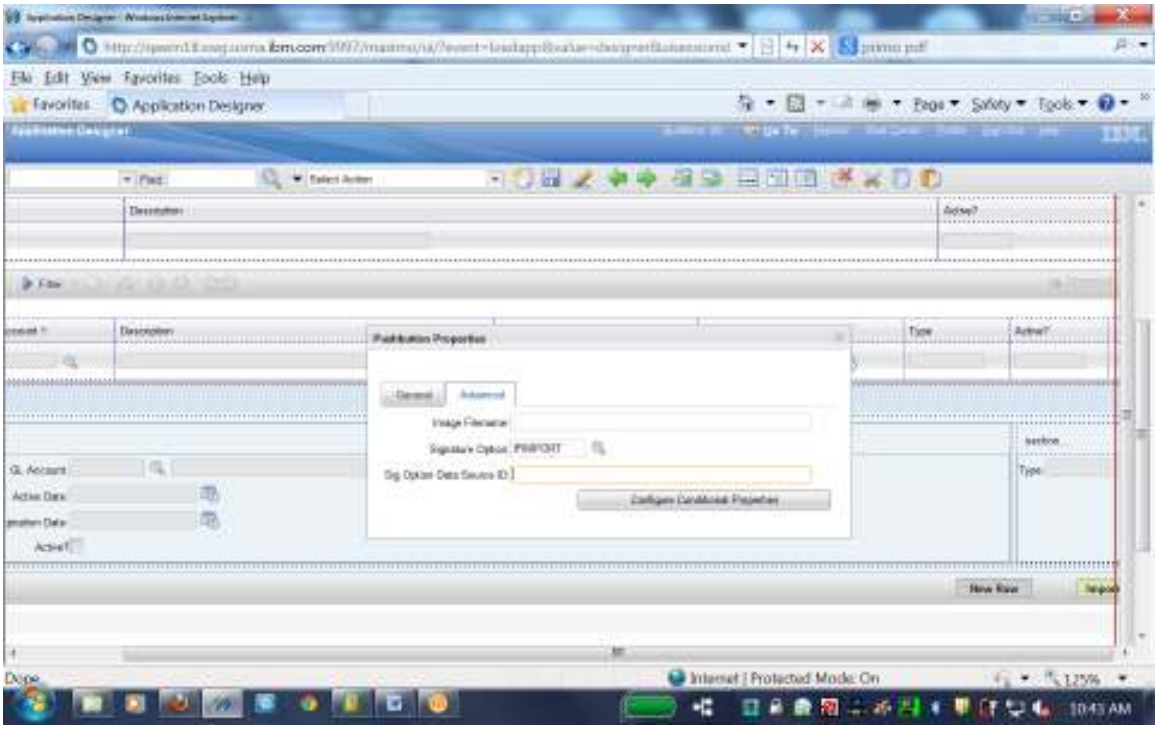
2. Next you need to go to the Application Designer application and add a new Pushbutton to support Importing. In App Designer select the CHRTACCT application and using the Control Pallette, select Pushbutton and place it next to the New Row button near the bottom of the application page.



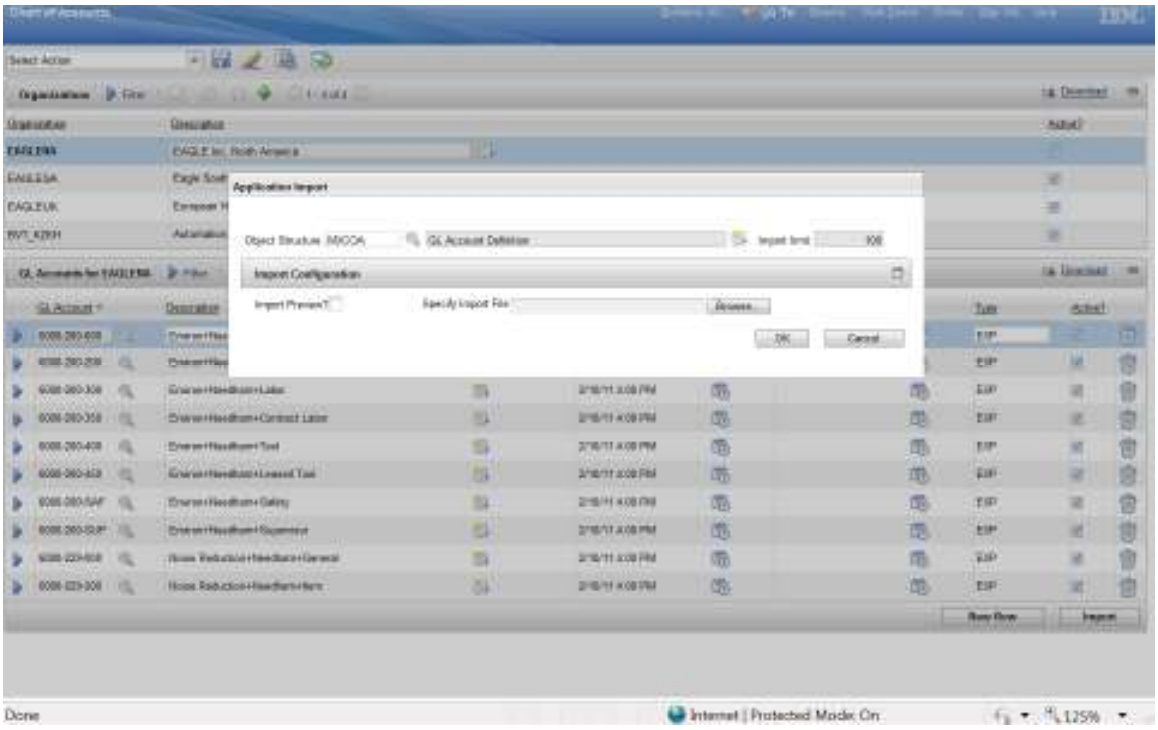
3. Right click on the new pushbutton and select Properties. On the General tab set the label to Import and the Event to ifimport



On the Advanced tab set the signature option to IFIMPORT



4. Under the Select Action, select Add/Modify Toolbar Menu. Delete the row for IFIMPORT. This will remove the import icon from the toolbar at the top of the application.
5. Save the Application and Exit.
6. If you have not already done so, go to the Security Groups applications and for the appropriate Groups, select the Chart of Accounts application and grant access to Application Import.
7. For users to see the new pushbutton in the chart of accounts application, they would need to log out and log back in to Maximo.
8. When you go to the Chart of Accounts application you should see the new import button and when pushed the App Import dialog should appear:



When you import you do not have to select an Organization like you do with exporting. The content being imported is driven strictly by the content in the file being loaded. The Import Limit will apply based on the configuration during enablement (in the object structure application).

One additional point is that you cannot use the XML format of the exported file to then import back into maximo, without change the file format (since the object structures differ). The exported XML file will have different outer tags as well as the content of the Organization object. If you are going to use an exported file as the basis to build a file for importing, you we will need to remove the Organization-related tags and change the outer tags of your xml to match the MXCOA object structure.
This is an example of what the exported XML would look like (the Chart of account sections of the xml are collapsed). In the object structure configuration, the ORGID attribute of the ORGANIZATION object was configured as included, all other attributes were excluded.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <SyncMXCOA2 xmlns="http://www.ibm.com/maximo"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    creationDateTime="2013-11-07T08:18:47-05:00"
    transLanguage="EN" baseLanguage="EN"
    messageID="1383830331581803907" maximoVersion="7 5
    20131101-1515 V7510-69">
```

```
- <MXCOA2Set>
- <ORGANIZATION>
  <ORGID>EAGLENA</ORGID>
+ <CHARTOFACCOUNTS>
+ <CHARTOFACCOUNTS>
+ <CHARTOFACCOUNTS>
+ <CHARTOFACCOUNTS>
+ <CHARTOFACCOUNTS>
+ <CHARTOFACCOUNTS>
+ <CHARTOFACCOUNTS>
+ <CHARTOFACCOUNTS>
    </ORGANIZATION>
    </MXCOA2Set>
    </SyncMXCOA2>
```

Below is an example of what the imported XML would need to look like – the values in **bold** have been changed and the Organization related tags have been removed..

```
    <?xml version="1.0" encoding="UTF-8" ?>
- <SyncMXCOA xmlns="http://www.ibm.com/maximo"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    creationDateTime="2013-11-07T08:18:47-05:00"
    transLanguage="EN" baseLanguage="EN"
    messageID="1383830331581803907" maximoVersion="7 5
    20131101-1515 V7510-69">
-  <MXCOASet>
+    <CHARTOFACCOUNTS>
+    <CHARTOFACCOUNTS>
+    <CHARTOFACCOUNTS>
+    <CHARTOFACCOUNTS>
+    <CHARTOFACCOUNTS>
+    <CHARTOFACCOUNTS>
+    <CHARTOFACCOUNTS>
+    <CHARTOFACCOUNTS>
   </MXCOASet>
  </SyncMXCOA>
```

If you are using the delimited file format (.csv), you will need to remove any columns that are from the organization object, leaving only those needed for the chart of account data.