Modernization Platform as a Service (ModPaaS)

User Guide for Google Cloud Platform

Google Cloud

1. Overview

The following document summarizes some common tasks to perform using the Advanced Enterprise Application Viewer (eav) solution, a critical part of ModPaaS. Please consult the *ModPaaS Getting Started Guide* for instructions related to connecting to the ModPaaS system and performing the initial configuration related steps.

> The following information assumes that the ModPaaS system instance is started and accessible from your Google Cloud Platform account via a Remote Desktop session. Your Remote Desktop port should be secured at launch by your System Administrator per your network topology.

Common tasks covered by this document include:

- Loading source code[1]
- Scanning source code[1]
- Exploring scanned objects
- Reconciling your application assets
- Grouping, categorizing and analyzing applications
- Obtaining conversion related modernization estimates from Advanced

Please refer to the eav Help system (**Help > eav Help** from the main menu) for additional usage information or contact Advanced at modpaas.support@oneadvanced.com if you have any questions related to this document.

---

[1] Also covered in the ModPaaS Getting Started Guide

2. Loading source code

2.1 Define an Entity (container) to store one or more source code repositories

Eav permits select users to access selected source code locations. To simplify the definition process, eav allows the definition of Entities that are then associated to one or more users and one or more source code locations (Repositories). This type of security definition is often referred to as "role-based". See section 10 of the *ModPaaS Getting Started Guide* for a glossary of terms.

To define one or more Entities for the organization:

- Click F**unctions** in the top menu bar and then select **Entity Administration**
- Enter the information for each Entity that you wish to define:
    - Right-click anywhere in the main Entity Name pane
    - From the pop-up menu, select **Add** to add a new line to the end of the current Entity list, or select **Insert** to add a new line above the line that was highlighted when the insert option was selected

        Note: Upon initial setup this list will be blank

    - Enter the **Entity Name**. This can be any name (e.g., your team, agency, group, project, etc.) that represents the users and repositories that will be associated to the Entity
    - All new eav users will automatically be given access to the repositories associated to this Entity if the "Auto Assign" checkbox is checked. If it is not checked, new users must be associated to the Entity by an eav Administrator before they can see the source code associated to the Entity
    - Save the Entity Name by clicking on the **Save** icon in the upper left of the window
    - Associate the Entity with the Administrative User (see section 4.2 of the *ModPaaS Getting Started Guide*) by double clicking on the User name from the Users window in the bottom right on the eav interface. This will add the Administrative user to the Entity list
- Repeat three steps for each new Entity that you wish to define
- Click the **Save** icon in the upper left of the window
- Double click on the **Entity Administration** tab to close the window

2.2 Define an Entity (container) to store one or more source code repositories

It's possible to define multiple source code locations for the source code objects to be scanned, including Partitioned Datasets or externally managed locations (e.g., Endevor on a remote mainframe). For the purpose of this Getting Started Guide we will explain how to import source code locally from a zip file that has been loaded onto the ModPaaS image (or is accessible via the network). Please see the Frequently Asked Questions (FAQ) in the Getting Started Guide for a description of how the zip file should be created.

- Click **Functions** in the top menu bar and then select **Repository Administration**
- Click on the **Network Repository Wizard** icon (circled below, second from the right on the Repository Administration menu bar)

- From the Network Repository Wizard Window, click on the **Import from Zip File** button
- Navigate to the location of your zip file, then click **Open**
- The Browser for Folder window will appear. Select a local target directory to extract the zipped source code. **Tip:** Create a new directory for each zip file you wish to import, e.g., c:\Code\SampleApp
- A popup window will show the progress as the code is extracted. This window will automatically close
- Once complete, a list of source code repositories (which will map to sub directions in the provided zip file) will be displayed, as shown in the example below:
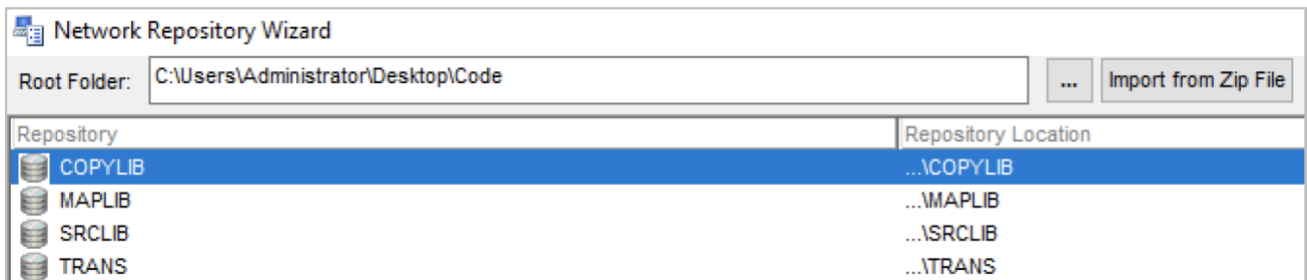

- Associate each repository with the Entity specified in the previous step by clicking in the white space below the **Entity** column for the first listed Repository. Select the Entity name from the drop-down menu
- If asked if you want to assign the Entity to all containers, click **Yes**
- Click **OK** to close the Network Repository Wizard window

2.3 Load each repository prior to scanning

Section 2.2 defined the Repository location (i.e., where the physical source objects reside) – however, before you can begin scanning the code, you must first register each object within eav and sync the objects to scan.

- From the Repository Administration window, register and syn each Repository by double clicking on a Repository name. The following window will appear – click **OK** to continue
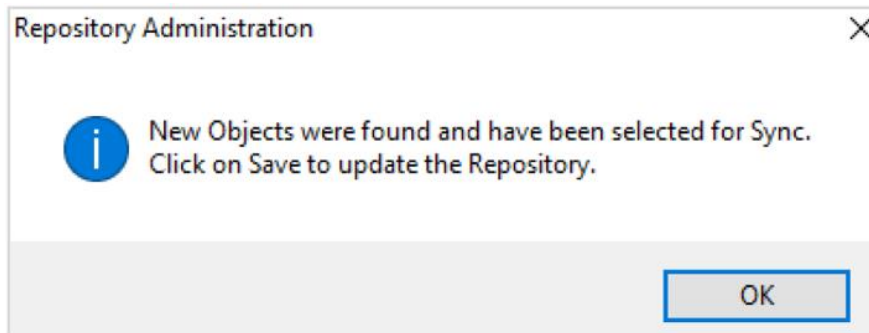
- The contents of the Repository will be displayed in the lower window panel

Note: The object Language type will be shown as Unknown unless you have appended a defined file extension to each object prior to registering with eav (unlikely when exporting from the mainframe). A list of defined language types supported by eav and the associated file extensions can be found in the File Associations tab.

Phase 1 of the object scanning process (see section 3) aims to automatically detect the object language type based on known verb lists.

- Click on the **Save** icon, (circled below, left corner on the Repository Administration menu bar)
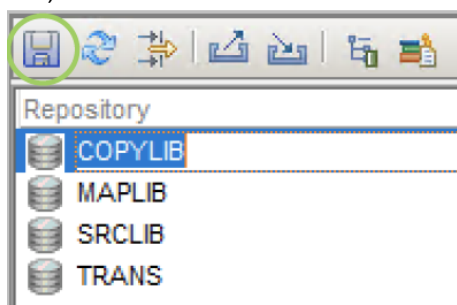
- Repeat the above steps for all remaining source code repositories

3. Scanning source code

It is now possible to run the Object Scanner to parse and analyze all source code objects and generate documentation for all of the repository objects that have been marked with Sync or Force Sync.

3.1 Scanning phases

The Object Scanner is a processing engine that executes in two key phases.

Phase 1

- Checks the settings for each object defined to eav
- Monitors the Repositories for updates to the objects marked for Sync
- Processes each object that is marked for Sync to determine the object dialect and total Lines of Code (LOC) to be scanned
- Uses the total LOC to be scanned to determine if your license key supports the required ModPaaS Base level (see section 7 of the *ModPaaS Getting Started Guide* for information regarding Base Levels), and if so proceeds to Phase 2

Phase 2

- Stores the total number of LOC scanned
- Processes each object that is marked for Sync to generate the documentation for each Sync Object
- Logs any issues encountered during the scan and document generation

3.2 Running the object scanner

- Click **Functions** in the top menu bar and then select **Object Scanner**
- You can optionally limit the processing to Repositories for a single Entity by selecting the desired Entity from the **Process Entity** pull-down at the top of the Object Scanner window. By default, objects for all loaded Entity Repositories (see section 2.3) are scanned.
- Start the Object Scanner by clicking the green Start flag (circled below, left corner on the Object Scanner window)

- You will be prompted with the following window notifying you of the two phase scanning process. Click **OK** to start Phase 1

Note: The time taken to complete the Phase 1 scan will depend on the total Lines of Code to be processed, in addition to the selected machine image memory and storage type
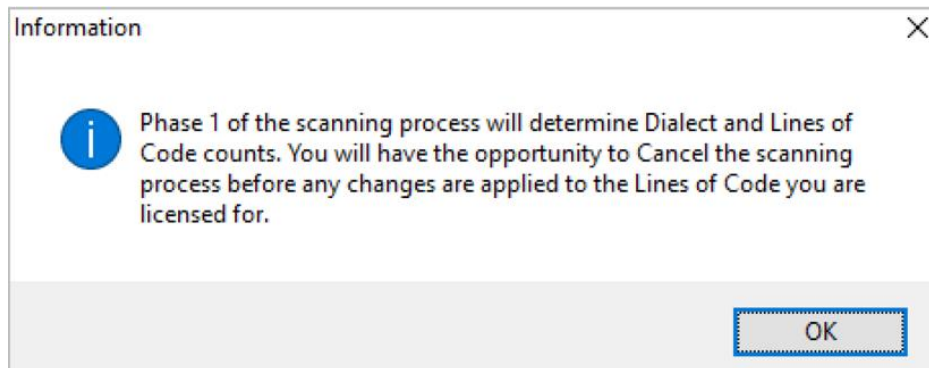
- Eav immediately begins providing scanning feedback. The bottom left of the eav main window shows the progression through the objects. To cancel the Object Scanner for any reason, click the red Stop icon in the upper left portion of the Scanner window
- Once the Phase 1 scan is complete, a popup ModPaaS Lines of Code Summary window will appear detailing language types, related lines of code, Current Base Level (which will default to "None" for the initial scan) and New Base Level should you proceed with Phase 2. You will also be notified whether the scan will exceed the licensed Base Level. If so, you can either request a new license or remove specific source code repositories to be scanned.
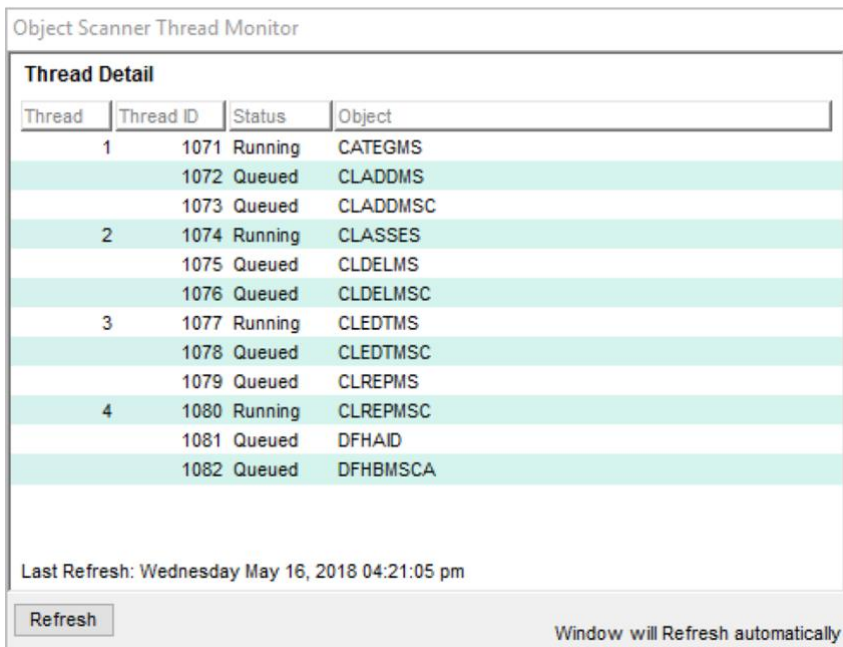

- At this point you have two options:
  - 1. Click **Cancel** to avoid the Phase 2 scan
    - You will be prompted one more time to confirm you wish to cancel Phase 2
    - After confirming you wish to cancel, you will need to perform step 2.3 again to register the source code objects to be scanned
    - No changes will be made to the Base Level
  - 2. Click **OK** to perform Phase 2 of the scan
    - You will be prompted one more time to confirm you wish to start Phase 2

Confirm

? You are about to start Phase 2 of the scanning process. This scan will be applied to the Lines of Code you are licensed for. Are you sure you want to continue?

Yes   No
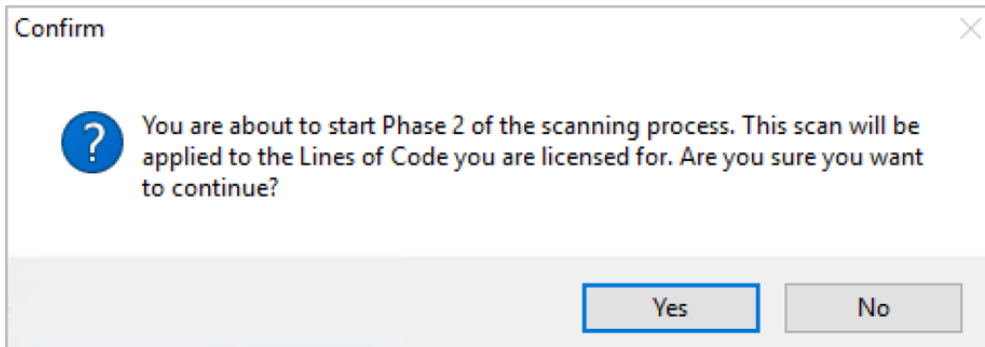
- After confirming (clicking Yes), a popup Object Scanner Threat Monitor window will appear, documenting the scan progress

Object Scanner Thread Monitor

Thread Detail

Thread	Thread ID	Status	Object
1	1071	Running	CATEGMS
	1072	Queued	CLADDMS
	1073	Queued	CLADDMSC
2	1074	Running	CLASSES
	1075	Queued	CLDELMS
	1076	Queued	CLDELMSC
3	1077	Running	CLEDTMS
	1078	Queued	CLEDTMSC
	1079	Queued	CLREPMS
4	1080	Running	CLREPMSC
	1081	Queued	DFHAID
	1082	Queued	DFHBMSCA

Last Refresh: Wednesday May 16, 2018 04:21:05 pm

Refresh   Window will Refresh automatically

- As with Phase 1, the time taken to complete the Phase 2 scan will depend on the total Lines of Code to be processed, in addition to the selected machine image memory and storage type. The Phase 2 scanning process will take significantly longer than Phase 1 due to the in-depth processing performed.
  - During our performance testing, approx. 600K LOC was scanned in 1 hour using 4 vCPUs, 2.5 GHz Intel Xeon® Platinum 8175, 16GB memory and 800 Provisioned IOPS SSD (with a Baseline Throughput of 200 MBs/sec)

Note: Once the Phase 2 scan is complete, all scanned objects will be available for analysis.

4. Exploring scanned objects

Scanned objects loaded into eav can be explored from the Object Browser, which is pinned to the main eav window.

- Click on the **Object Browser** tab and select one of the four available options for viewing, By Entity, By Type, Recent Objects or Custom (the latter is designed to work with categories)

- Click the **Blue arrow** (circled below) to refresh the contents each window
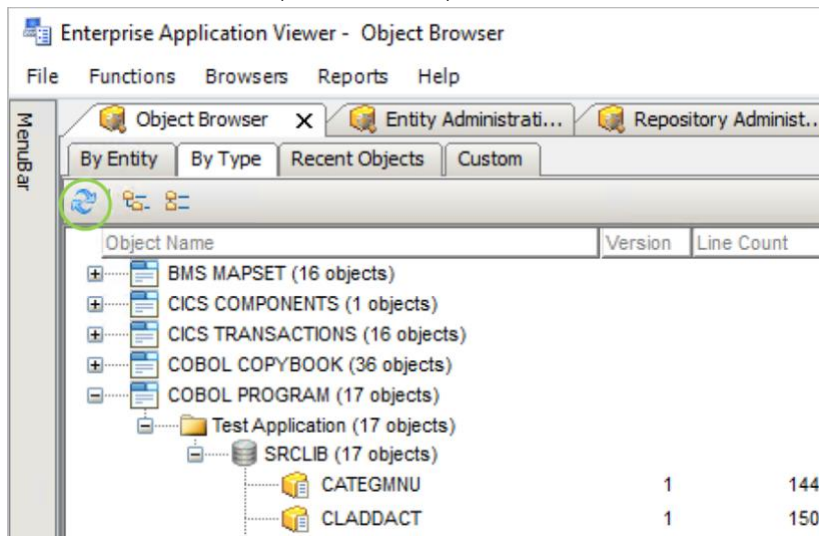
- Expand each tree list to locate a specific object to browse. Alternatively, it's possible to filter the view of objects based on an entered term. The filter input field and button are located on the top right of the Object Browser

- Objects are identified by the object icon: . Errors during the object scan will cause the object icon to be highlighted in red: . You can display the errors by selecting the Scanner Log tab from the Object Viewer, as described in the next step

- **Double click** a specific object to access the Object Viewer
  - The Object Viewer is used to explore the source code, business rules, flows and related reports generated for a specific object
  - The Object Viewer contains multiple tabs, as shown in the image below. The object type determines the available content tabs. The example image below is for a COBOL program

Note: Consult the eav Help system (select **Help > eav Help** from the main menu, then select **Object Viewer** from the TOC in the left column) for a description of each tab
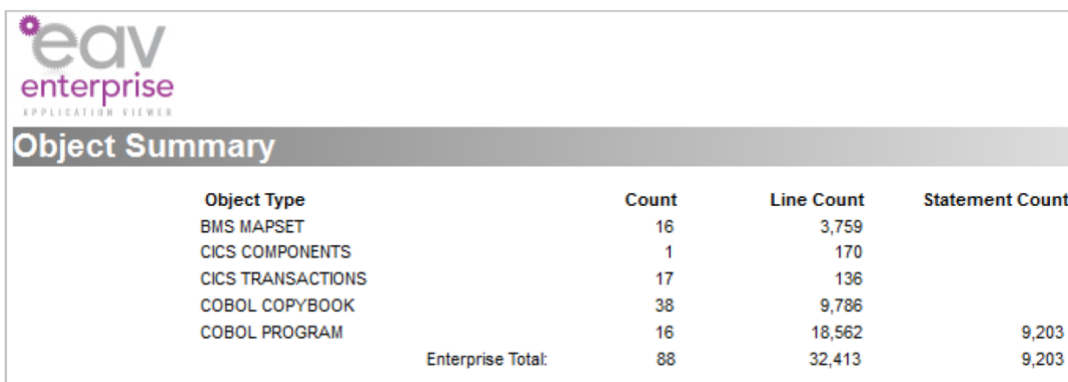
5. Reconciling your application assets

It's imperative that you reconcile your application assets before beginning to categorize and analyze the scanned source code. Eav can determine missing, unreferenced and unused objects. The following tasks will likely become part of an iterative process requiring multiple source code scans until an agreed asset scope is defined.

5.1 Object summary report

Begin by reviewing the object summary report. This will show the summarized results of the object scan, including the object type count and line count within each object type.

- Select **Reports** > **Inventory Statistics Reports** > **Object Summary**
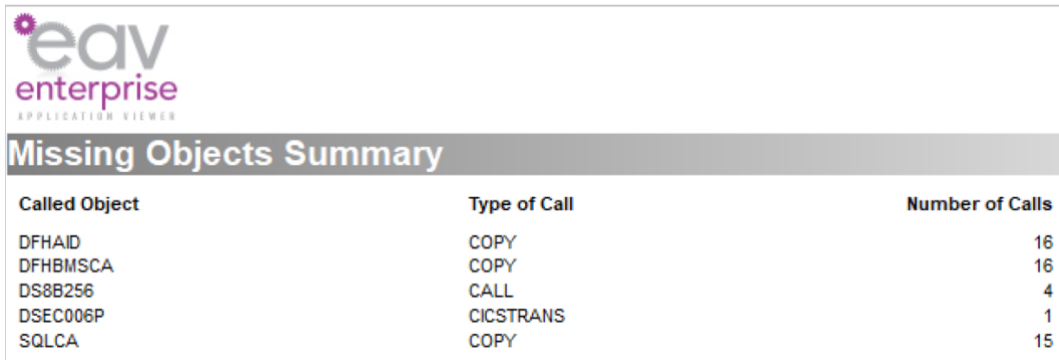
5.2 Missing object reports

Review the contents of the missing objects reports, starting with a high-level summary report – then progressing to the detailed report.

- Select **Reports** > **Inventory Error Reports** > **Missing Objects** > **Summary** to display the Missing Objects Summary report or **Reports** > **Inventory Error Reports** > **Missing Objects** > **Detail** to display the Missing Objects Detail report
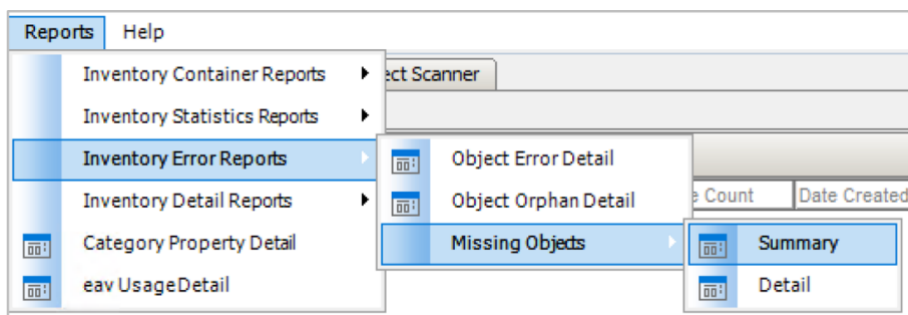
- The **Missing Objects Summary** report lists each missing object, the type of call, and the number of times it's called. There will likely be several known missing objects which can be ignored from future reports (see the next point for details of how this can be accomplished), such as JCL utilities or CICS supplied copybooks such DFHAID or DFHBMSCA, as shown in the following image.
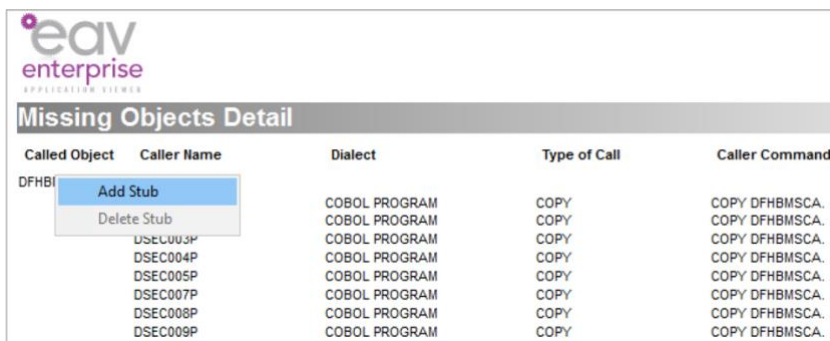
Missing Objects Summary

Called Object	Type of Call	Number of Calls
DFHAID	COPY	16
DFHBMSCA	COPY	16
DS8B256	CALL	4
DSEC006P	CICSTRANS	1
SQLCA	COPY	15

- The **Missing Objects Detail** report shows the caller name, dialect, type of call and caller command for each object referenced in the Objects Summary report. Here is where you have the ability to "stub" any known missing objects.

5.2.1 Adding a stub

Stub objects provide the ability to remove reference to known missing objects in the Missing Objects reports. This is helpful when your code references CICS supplied copybooks or JCL utilities. To add a stub for a missing object:

- Right-click on the name of the missing Called Object you wish to stub in the Missing Objects Detail report
- Select **Add Stub** from the popup window, as shown below

Missing Objects Detail

Called Object	Caller Name	Dialect	Type of Call	Caller Command
DFHBI	Add Stub			
	Delete Stub	COBOL PROGRAM	COPY	COPY DFHBMSCA.
		COBOL PROGRAM	COPY	COPY DFHBMSCA.
	DSEC003P	COBOL PROGRAM	COPY	COPY DFHBMSCA.
	DSEC004P	COBOL PROGRAM	COPY	COPY DFHBMSCA.
	DSEC005P	COBOL PROGRAM	COPY	COPY DFHBMSCA.
	DSEC007P	COBOL PROGRAM	COPY	COPY DFHBMSCA.
	DSEC008P	COBOL PROGRAM	COPY	COPY DFHBMSCA.
	DSEC009P	COBOL PROGRAM	COPY	COPY DFHBMSCA.

- Click the Refresh button on the toolbar. The missing object will be removed from the Missing Objects Detail report (along with the Missing Objects Summary report once refreshed)

5.2.2 Deleting a stub

A stub can be deleted if you replace it with the real object. Delete a stub as follows:

- From the **Repository Administration** window, double click on the specific Repository to view the Repository Contents
- The object that has been stubbed will be shown as language type "STUB", as shown below

Repository Contents	Source Viewer	File Associations	User Defined File Associations			
Repository				Object Name	Ext	Language
📔 Security				DFHBMSCA		STUB

- Right-click on the relevant object that has been stubbed
- From the popup window select **Delete Object**, then click **OK** to confirm the deletion
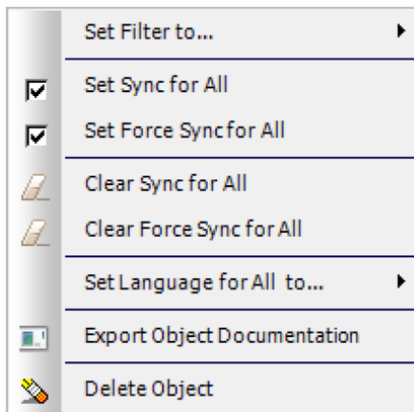
5.3 Additional error reports

There are two additional inventory error reports which aid in reconciling the application asset.
- Right-click on the relevant object that has been stubbed
- Select **Reports** > **Inventory Error Reports** > **Object Error Detail** to display a list of each object containing an error (i.e. has a scan error) and all the messages from the most recent scan

- Select **Reports** > **Inventory Error Reports** > **Object Orphan Detail** to display a list of all objects that have no 'called' components and no 'calling' components. At this point these objects could either be considered orphaned or standalone – however this might change as additional missing code is scanned

---

Object Orphan Detail

Object	Dialect	Ver	Repository
@CICSSEC	CICS COMPONENTS	1	Security
DCLSYSV	SAS	1	Security
DCLUSRD	SAS	1	Security
DSEC014I	COBOL COPYBOOK	1	Security
DSS6	CICS TRANSACTIONS	1	Security
Total: 5			

Note: Each report can be filtered by Entity and also by Category (see section 6 for additional information related to categories). Reports contents can be saved in multiple format types, including PDF, Excel, CSV, XML, HTML, etc.

5.4 Reloading missing source code

At this point you should have enough information to compile and send a list of missing components to your relevant subject matter expert(s). See Section 9 of the *ModPaaS Getting Started Guide* for instructions related to reloading and scanning addition source code in order to resolve discrepancies.

6. Grouping, categorizing and analyzing applications

Eav uses categories to group related objects. Once created, categories can aid with validating defined modernization disposition strategies and ultimately support strategic modernization planning efforts. Example category uses include:

- Understanding the conversion impact if a specific piece of functionality is replaced with a COTS application
- Supporting reengineering projects by validating the results of a top-down driven assessment to identify possible candidate microservices
- Identifying all impacted objects if a specific field value is modified
- Determining the possibility of supporting a phased production cutover, or whether all applications need to go-live "big bang" due to objects being tightly integrated
- Estimating the cost and duration of implementing different modernization disposition strategies

Note: The first step in creating a category is to determine the (one or many) "entry points" into the application or specific application feature. For an online application, an entry point might be the first execution point in a production transaction thread, e.g., a CICS-based transaction or specific screen. For a batch application, an entry point might be any JCL job executed as a production business process. Make sure to gather all objects marked as entry points prior to creating a new category.

6.1 Creating a new category

- Click **Functions** in the top menu bar and then select **Category Administration**
- From the Category Administration window, right-click anywhere on the blank window and select **Add**
- Enter a **Name**, select a **Color**, and provide a **Description**. If you want other users to see this category, click the Public checkbox. The following image shows an example category called Order Entry.

- Click **Update**
- Finish off the category by clicking on the **Property Values** ellipsis (shown on the far right of the Category Administration window). From the popup Category Property Values window, set relevant property values for the **Disposition Strategy** and **Target Mapping** (if known) – these will be required requesting modernization estimates from Advanced (see section 7). An example is shown below for the Order Entry category.

- Click OK to accept the values, then Close to exit Category Administration

6.2 Adding custom category properties

You have the ability to define up to six (6) additional custom category properties.

- Click on the **Define Properties** button to open the Category Properties window.
- Enter the new **Property** name and select the **Type Id** (input field, drop down selection or currency field). You have the ability to define **Property Values** (comma delimited) for the "Drop Down" Type Id.

6.3 Manually applying a category to scanned objects

A category can be manually applied to one or more objects from within the Object Browser. See section 6.4 for instructions related to automatically applying a category to a number of defined entry point objects using Lists.

- Click on the **Object Browser** tab
- Locate the object(s) identified as the entry points to apply the category on. Choose, **By Entity** or **By Type** by clicking the relevant Object Browser tab, or enter an object name in the Filter input field (located top right) and click the **Filter** button
- Select the identified entry point object(s) (hold the CTRL key to select multiple) to apply the category to. Once selected they will be highlighted blue. Right-click, select either **Add Category by Chain** or **Add Category by Need** from the popup window, followed by the category name. "By Chain" will include all dependencies in the specified chain direction (you will have the option to specify the chain direction, Up, Down or Both) and "By Need" will include only the objects that have a direct dependency
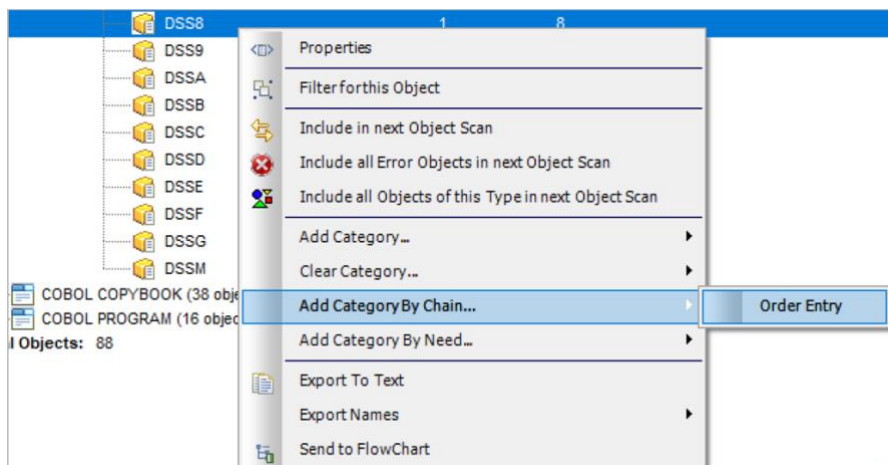
- To display the results showing the objects referenced by applying the category, click on the **Custom** tab in the Object Browser, then click on **Category**


- From the popup Category Select window, select the relevant category then click **OK** to display referenced objects

- To show relationships between all objects in the category, select all objects, right-click, select **Send to FlowChart**
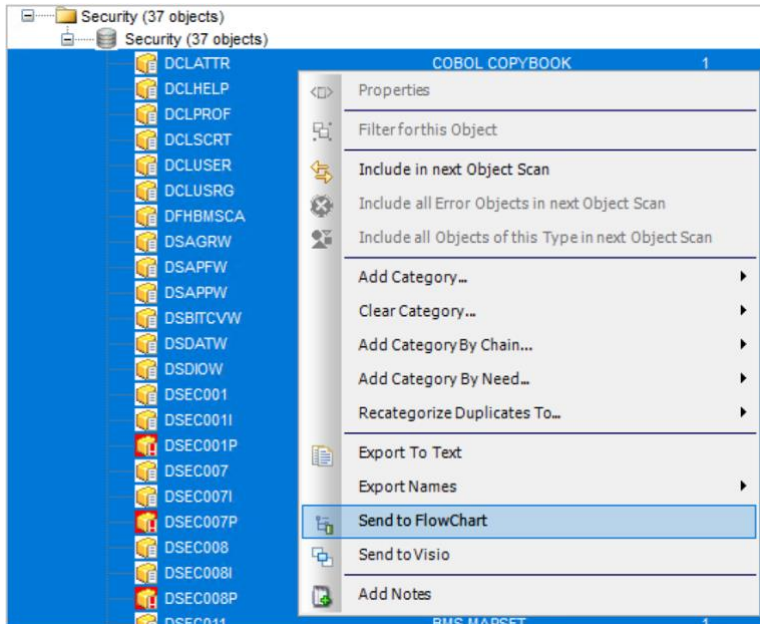
- A Custom Flowchart will be generated
- The flowchart can be printed 🖨, or saved as a PDF file 📄 using the buttons on the toolbar. The view can be zoomed 🔼 in and out 🔽, and can be changed to use 'Spring View' 🔲. The contents of the flowchart can be searched using the search field on the toolbar.
- Right-click a specific object to "Explore" the object
- Objects shown in grey are known as "Ghost" objects. These objects are just references to real objects used elsewhere in the flowchart. Ghost objects are used for visual reasons so you can cleanly see how objects interact without overcomplicating the layout. **Double clicking** on a specific ghost object will reference the real object
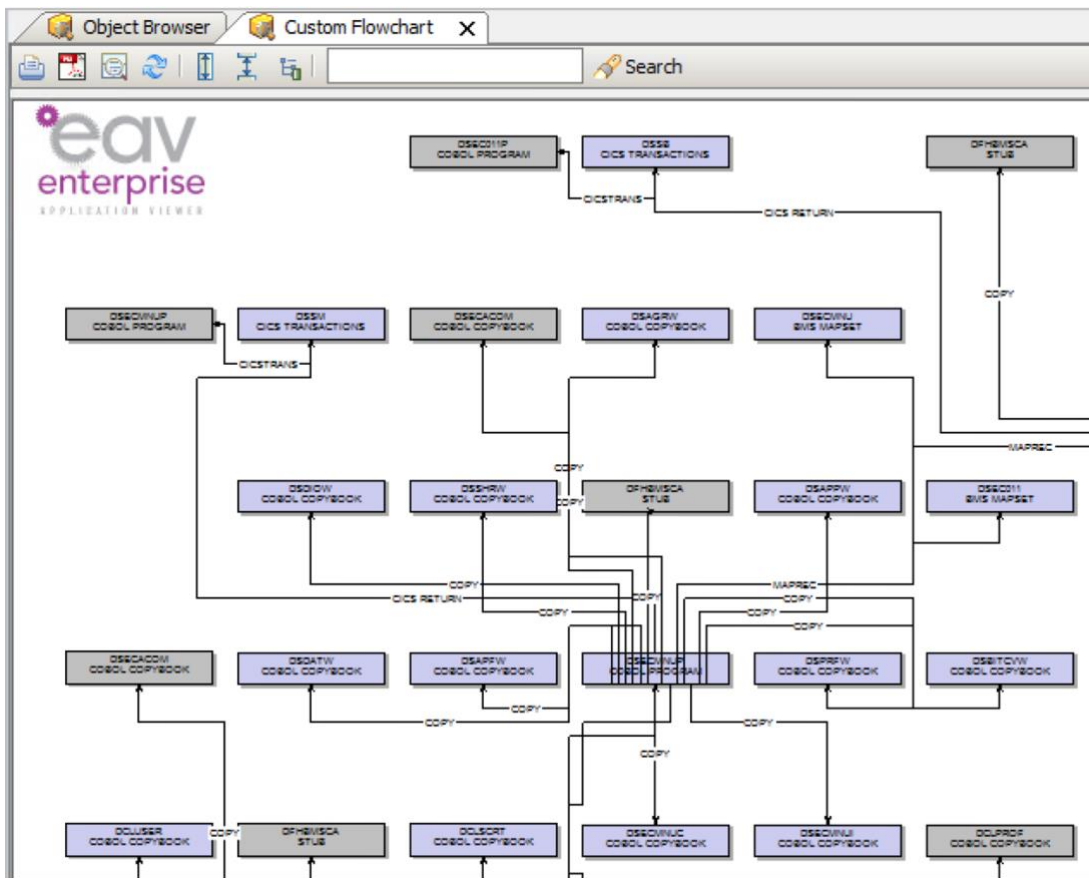
6.4 Automatically applying a category to scanned objects using lists

It's possible to automate the steps described in section 6.3 by creating and managing lists containing the defined entry point object names. Lists can be modified to include additional entry points, and automatically processed to regenerate the category results. Create a list as follows:

- Click on the **Object Browser** tab
- Locate the object(s) identified as the entry points to apply the category on. Choose, **By Entity**, **By Type** by clicking the relevant tab, or enter an object name in the Filter input field (located top right) and click the **Filter** button
- Select the identified entry point object(s) to apply the category to (they will be highlighted blue), right-click, and select **ExportNames** > **Clipboard**
- Click **Functions** in the top menu bar and then select **List Administration**
- Right-click anywhere on the main List Administration window
- Select **Add** from the popup window
- Enter the **List Name**, select the **Action** to perform and the **Category** to process against. Also select the **Direction**

- Finally, click on the **List** button (displayed to the far right of the field) to open the Edit List window
- **Paste** the contents of the clipboard (copied from a previous step), then click **OK**
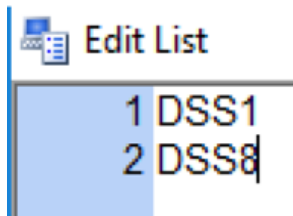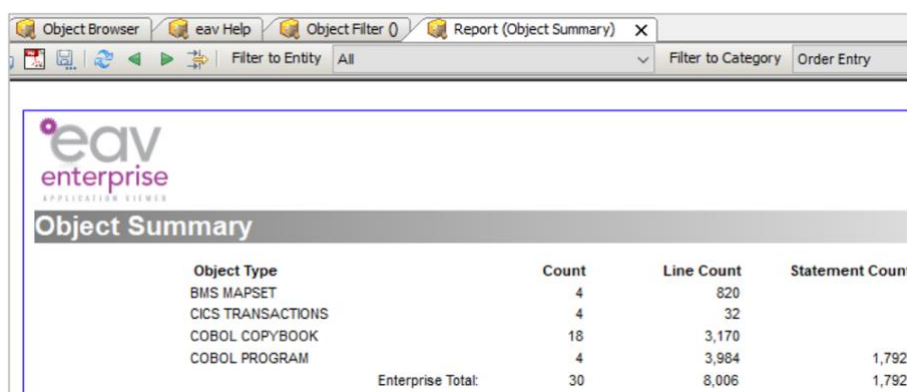

- Click **Update** to save the list, or Process to execute the list

You can come back to List Administration at any time in order to modify, delete or process the list again. You can process one or multiple lists at the same time as follows:

- Click the **Process** checkbox for each list to process
- Click the **Process** button
- From the Object Browser Custom tab, click on **Category**
- From the popup Category Select window, select the category then click **OK** to display the referenced objects

6.5 Filtering reports by specific category

You now have the ability to filter the reports by the category you created. For example, the following image shows the Object Summary report referenced in section 5.1 (Reports > Inventory Statistic Reports > Object Summary) filtered for the Order Entry category.


6.6 Determining category dependency interactions

An important analysis task is understanding the object dependencies between multiple categories. This is critical when determining the possibility of supporting a phased production cutover, as it highlights any "bridges" that will need to be developed during the testing phase of the project.
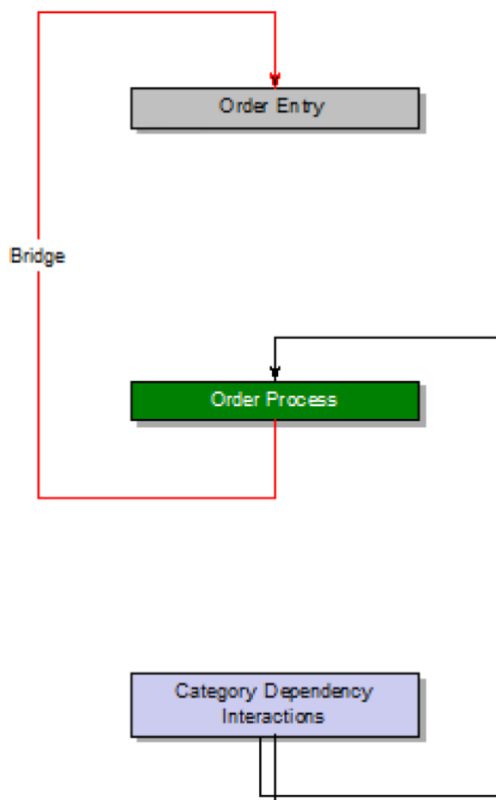
To determine category dependency interactions:

- Select the **Category** option from the Object Browser, Custom tab
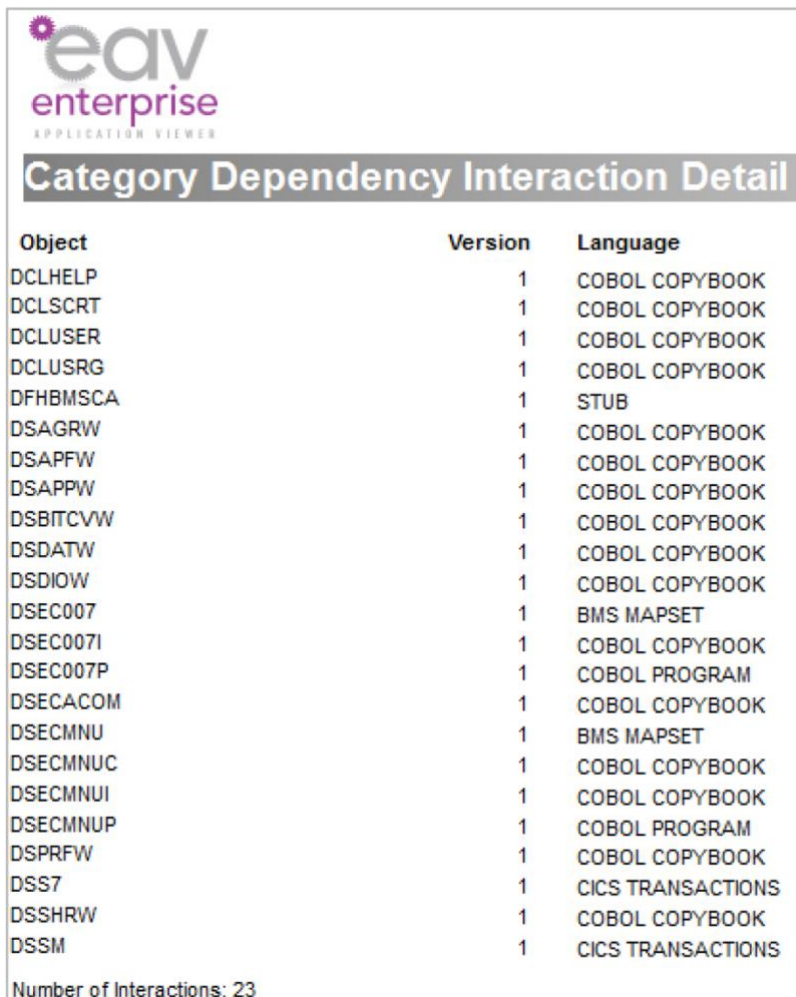
- From the **Category Select** window, select two or more categories and click on the **CDI** button
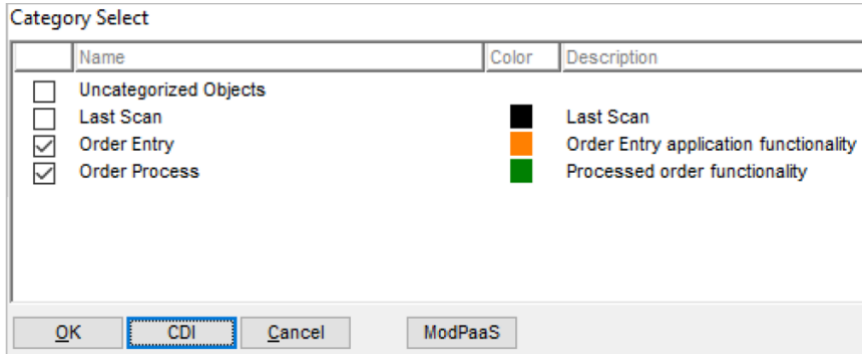
- A Custom Flowchart will be created for the selected categories. To see the specific object interaction details, **double click** on the **Bridge** link between the categories (an example is shown in red below)

- This will generate a custom Category Dependency Interaction report showing the object name and language types

Category Dependency Interaction Detail

| Object	Version	Language
DCLHELP	1	COBOL COPYBOOK
DCLSCRT	1	COBOL COPYBOOK
DCLUSER	1	COBOL COPYBOOK
DCLUSRG	1	COBOL COPYBOOK
DFHBMSCA	1	STUB
DSAGRW	1	COBOL COPYBOOK
DSAPFW	1	COBOL COPYBOOK
DSAPPW	1	COBOL COPYBOOK
DSBITCVW	1	COBOL COPYBOOK
DSDATW	1	COBOL COPYBOOK
DSDIOW	1	COBOL COPYBOOK
DSEC007	1	BMS MAPSET
DSEC007I	1	COBOL COPYBOOK
DSEC007P	1	COBOL PROGRAM
DSECACOM	1	COBOL COPYBOOK
DSECMNU	1	BMS MAPSET
DSECMNUC	1	COBOL COPYBOOK
DSECMNUI	1	COBOL COPYBOOK
DSECMNUP	1	COBOL PROGRAM
DSPRFW	1	COBOL COPYBOOK
DSS7	1	CICS TRANSACTIONS
DSSHRW	1	COBOL COPYBOOK
DSSM	1	CICS TRANSACTIONS

Number of Interactions: 23

# 7. Obtaining estimates from Advanced

7.1 Obtaining conversion-related modernization estimates from Advanced

Estimating the cost and duration of implementing different modernization disposition strategies is a logical next step once your applications are grouped into categories. Strategic options include automatically converting COBOL to Java or C#, rehosting COBOL to COBOL, reengineering to Java-based microservices, etc. Estimates for the specific modernization options can be obtained from Advanced using eav as follows:

- Click on the **Custom** tab in the Object Browser, then click on **Category**
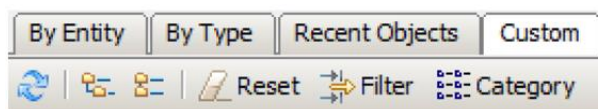
- From the Category Select window, select each category that you want to generate an estimate for, then click on the **ModPaaS** button
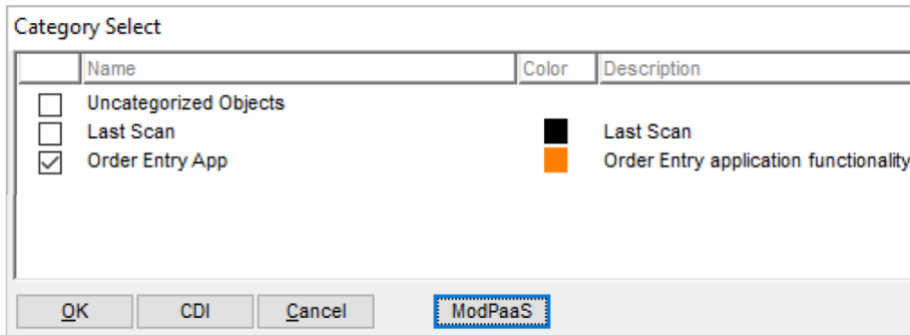
- From the ModPaaS window, complete the **Email Address** and **Company Name** fields then select **Send selected Category details to Modern Systems for ROM modernization estimate**
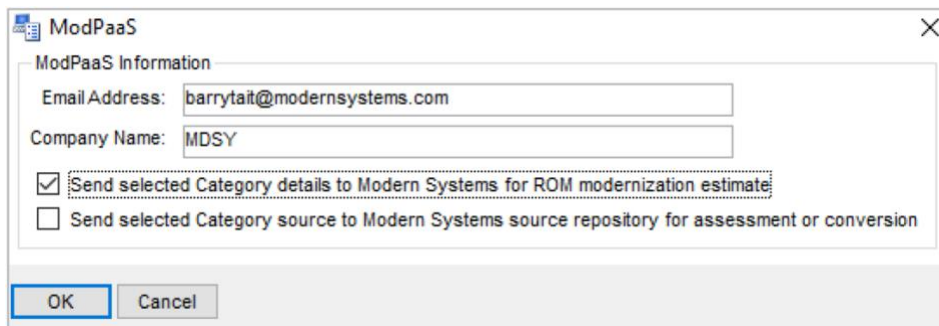
- Click **OK** then allow a little time to dynamically generate the report and send the email to Advanced
- When complete you'll receive a conformation message that the information was successfully sent

Note: Your Email Address will be retained after sending. In addition, you can set the Company name by selecting **Options Administration** from the **Functions** menu.

7.2 Sending selected category source code to Advanced
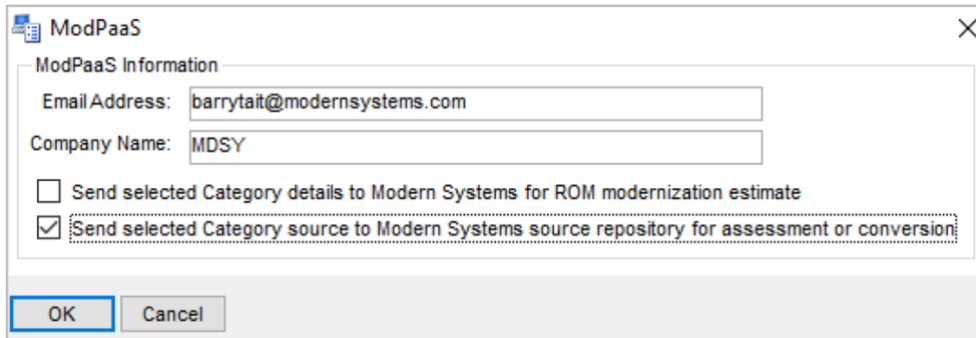
In order to obtain a fixed-bid proposal for a modernization project you must first extract the relevant category source code from eav into an Advanced GitLab repository. Advanced will then extract and analyze the code to determine conversion effort and delivery options with fixed-bid pricing. The same process is also followed when converting selected source code.

Note: This feature requires additional configuration, including the creation of a local SSH key on your ModPaaS system, and an account in our GitLab repository. To configure this capability please contact Advanced at modpaas.support@oneadvanced.com for assistance.

Once configured, follow the steps below to extract code:

- Follow the same steps as section 7.1 to select each category to send to Advanced

- From the ModPaaS window, complete the **Email Address** and **Company Name** fields then select **Send selected Category source to Modern Systems source repository for assessment or conversion**

- Click **OK** and allow a little time to process the request