Module 0 -- Workspace build and troubleshooting

Purpose

You will learn to build a workspace and setup the training materials for this course.
You'll also learn about common issues and how to resolve them.

General

COBOL Analyzer works by first loading the source repository, parsing it using the Micro Focus COBOL compiler, analyzing it using the COBOL Analyzer analysis engine and finally loading the analysis data into a database.

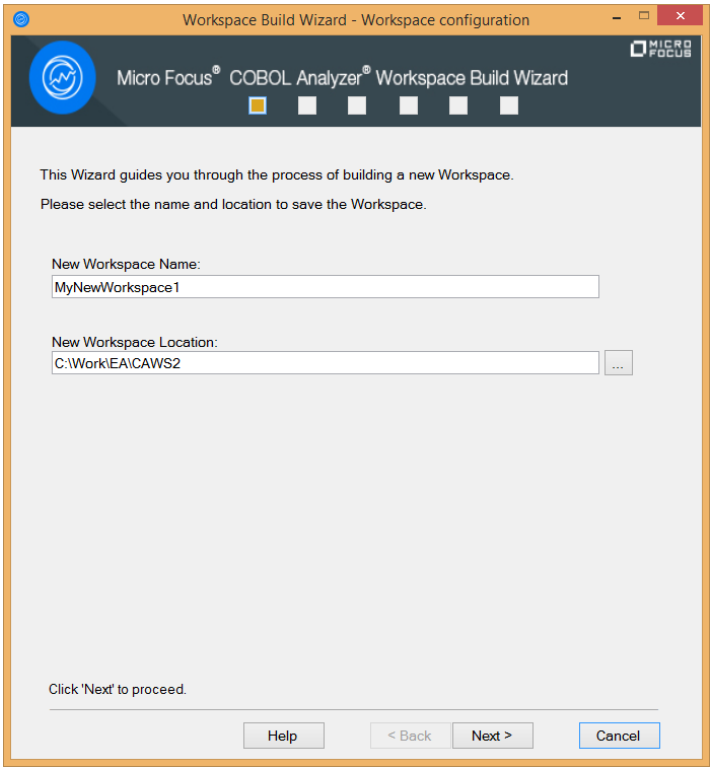
Once this has been completed, you'll get the full analysis information exposed in various different views using the COBOL Analyzer desktop client

Exercise 0-1: Getting familiarized with the workspace build wizard

This tutorial will utilize the Micro Focals sample code provided. The objective of the tutorial is to provide familiarity with the COBOL Analyzer and creating new workspaces.

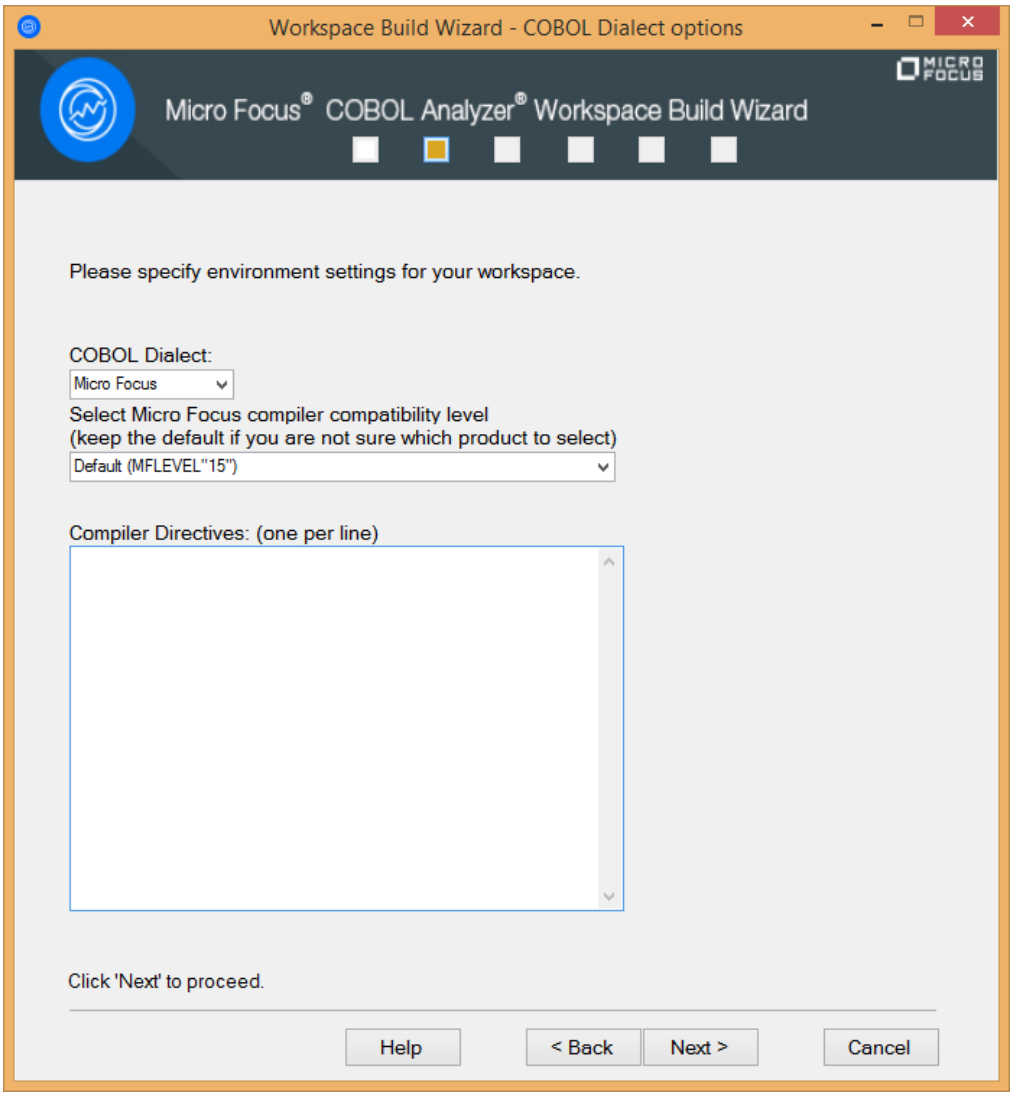
1. Open COBOL Analyzer
2. In the Open Existing Workspace dialog, click on "Create New"
3. **The tool will close, and a wizard will pop up, if this doesn't happen, re-open the tool as administrator and follow step 3**

4. In the first wizard page, you can name the workspace and choose its location:



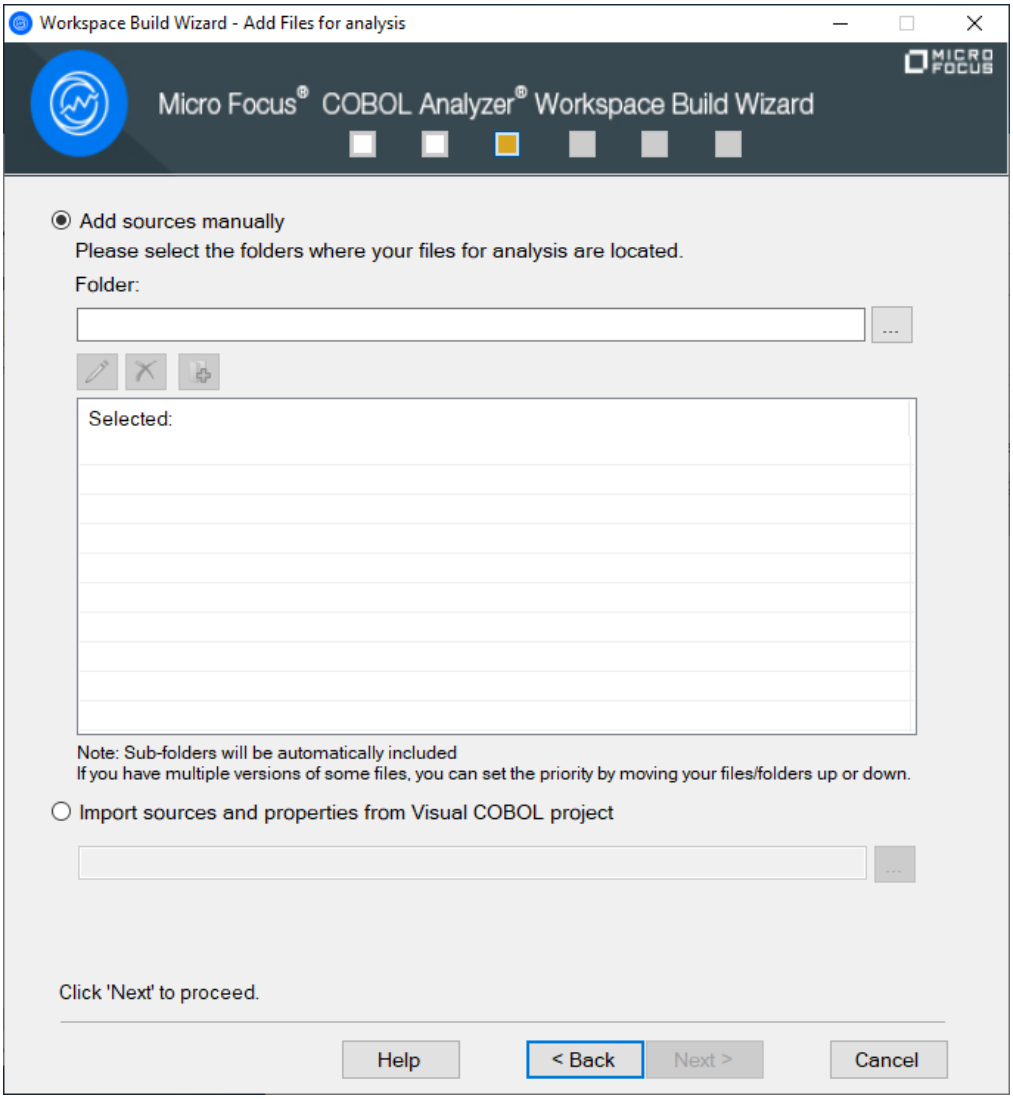
Click “Next”.

5. In the next page you can choose the COBOL dialect (Micro Focus, ACUCOBOL-GT(R) or RM/COBOL) and compiler level (MFLEVEL) depending on the product that is used to compile the code.
You also need to add any compiler directives that are needed in order to compile the code. Linker related directives are not needed. You can also use a directives file by using the USE compiler directive (for example: *use(console.dir)*)



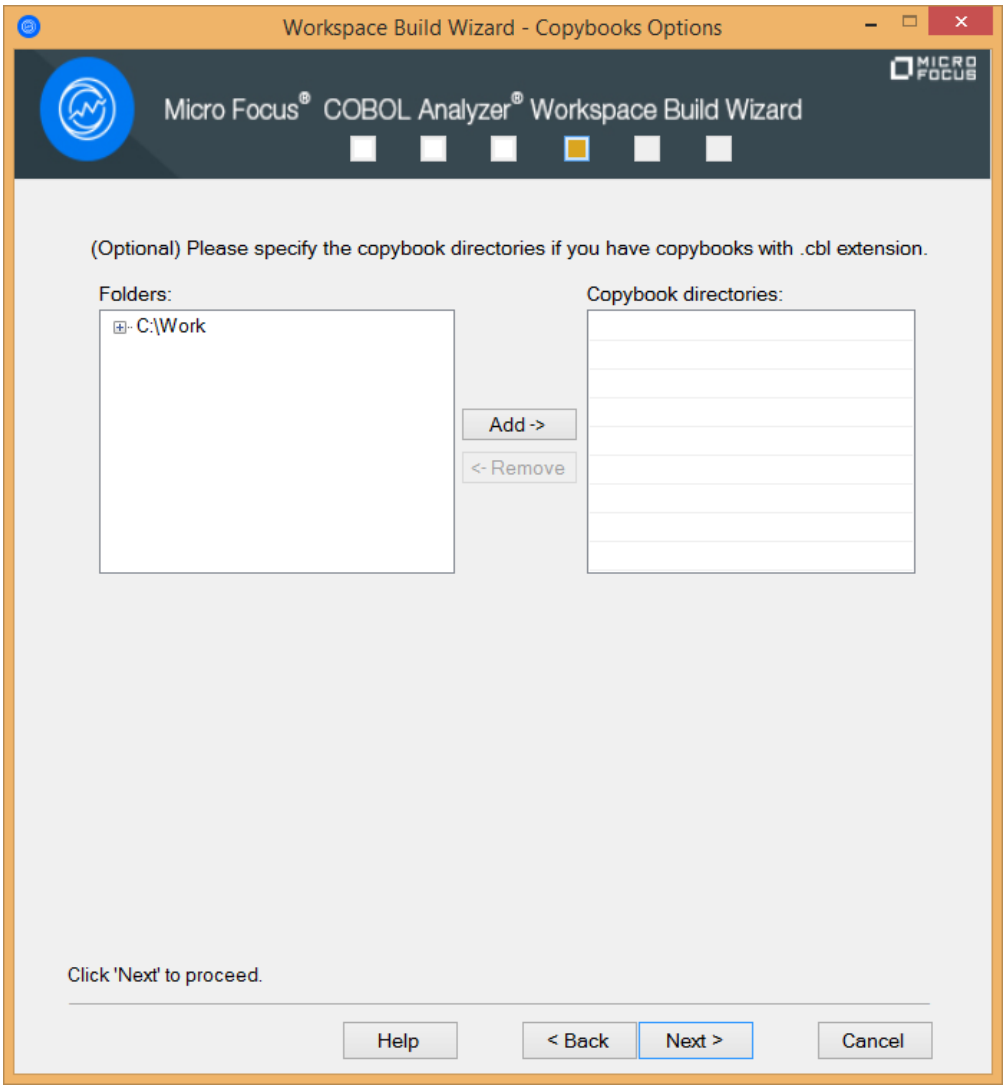
Click “Next”

6. In the next page you need to specify the folders that contains your source code files, **the recommended setup is to have one root folder that contains all the folders and sub folders that contain your sources.**
   There are multiple ways to add source code in the next pane:
   a. You can type the path in the “folder:” text box
   b. Use the browse button to navigate to where your source code is located or
   c. Drag and drop a folder from Windows Explorer. (Support depends on your Windows OS version and User Account Control settings)
   d. Import a Visual COBOL project file



Add a folder and click “Next”.

7. The next page is an optional extra settings page.
   In case you have dedicated folders that contain copybooks only (libraries folders), and specifically when those copybooks do not have an identifying file extension such as “.cpy” etc., specifying those folders would help COBOL Analyzer to identify the required copybooks and analyze your source code correctly and quickly.
   You can expand the folders tree on the left and add the libraries folders to the Copybook directories list by clicking the ‘Add’ button.



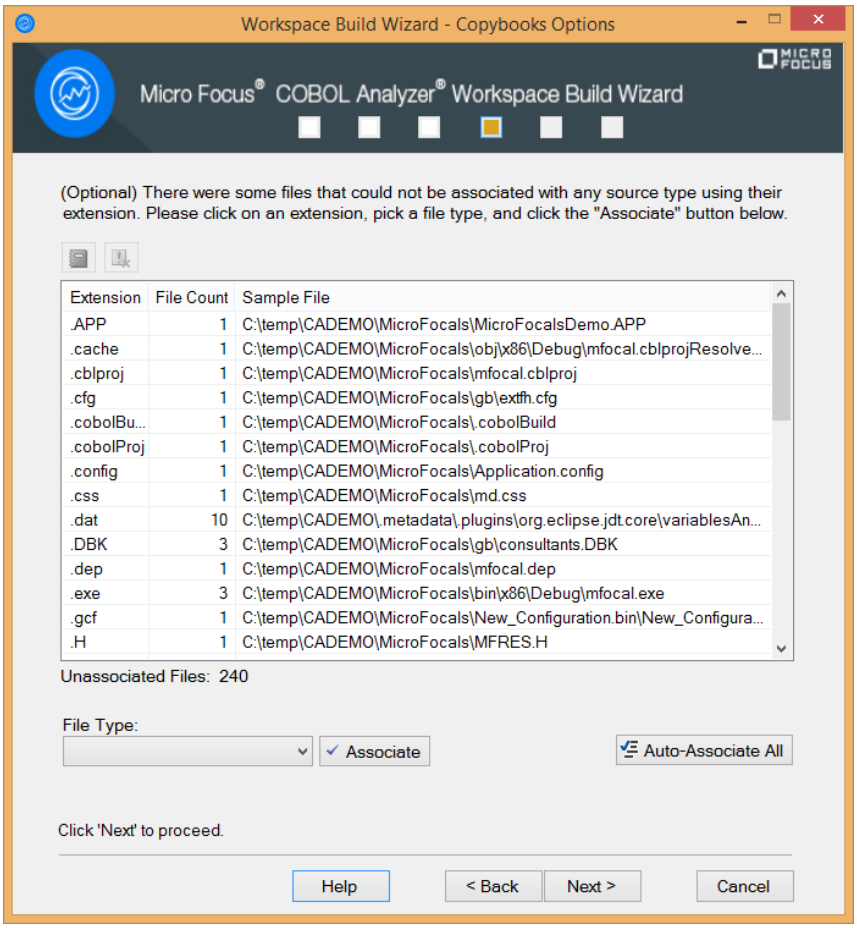
For our example we have no settings to update, so click “Next”.

8. Next you get a list of all the files that COBOL Analyzer didn’t identify by their extension.

   You need to review those files, select the extension and use the file type drop down and the Associate button to associate them correctly.

   Another option is to click the Auto-Associate All button and let COBOL Analyzer parsers decide by the files content.

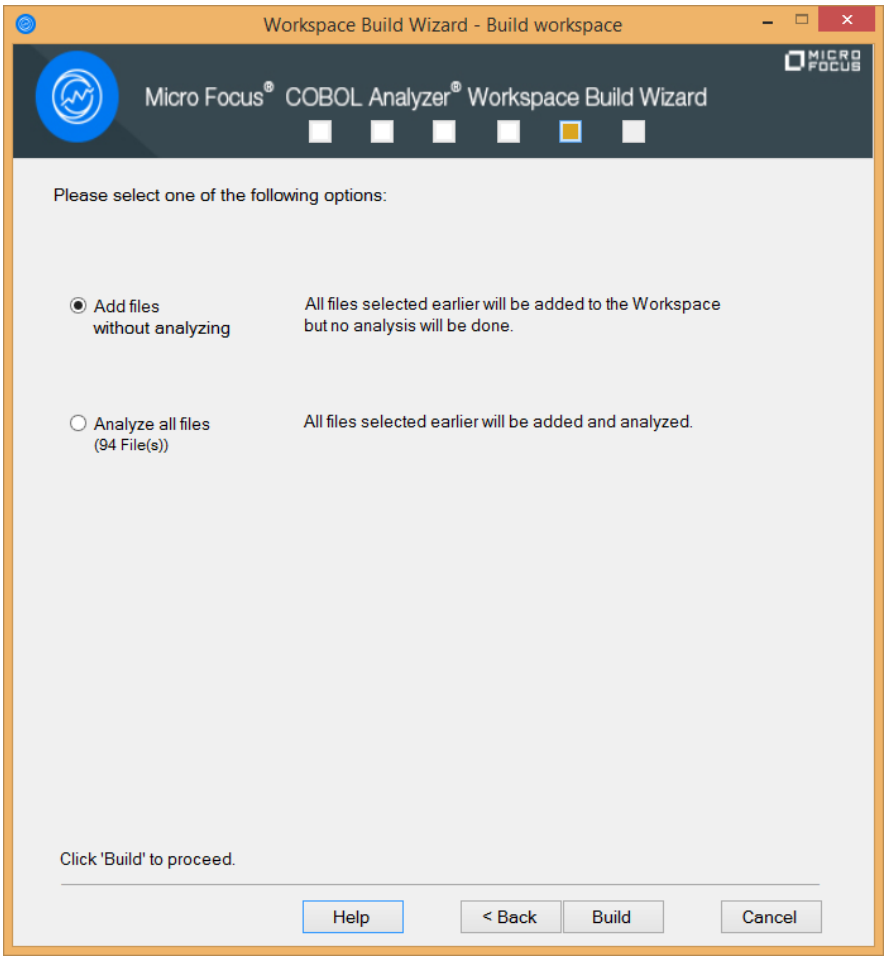
   Most of the cases when we encountered unassociated files they are either files that are not part of the sources (project files, binaries, document etc.) or copybooks.



9. In the last page you'll have 2 options, creating the workspace but not analyzing the files yet or creating the workspace and analyzing all the files.

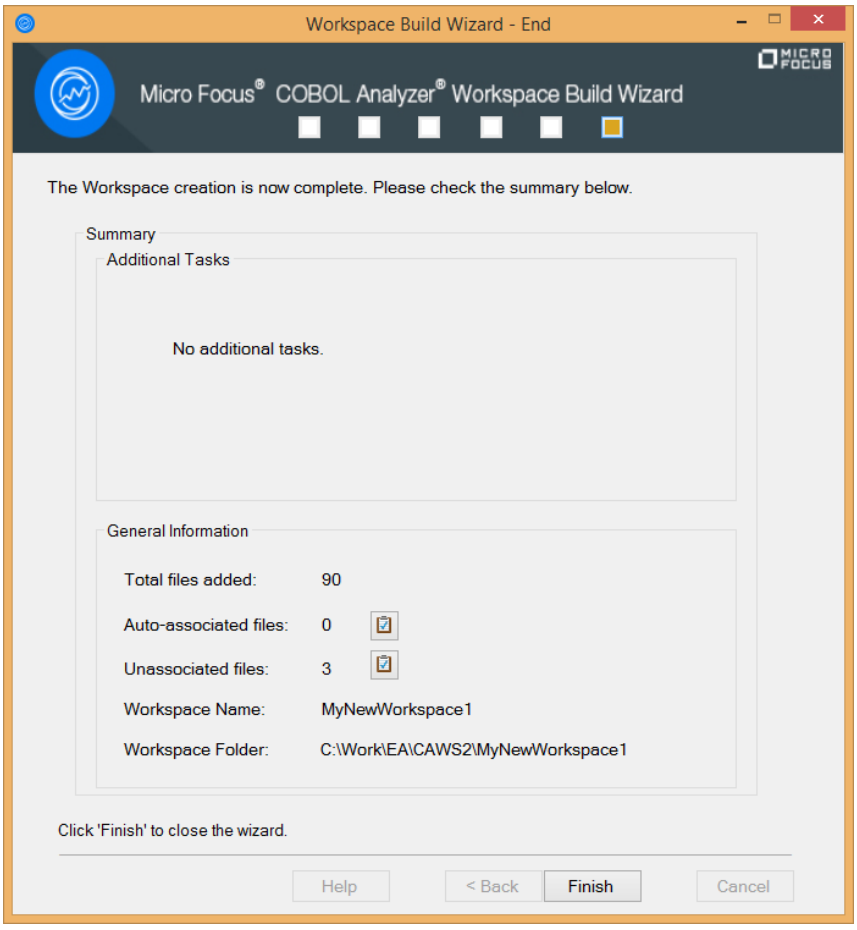
   The analysis could be time consuming when loading a large amount of code (millions of lines of code), so in that case and if this is a first trial of analyzing the code, the recommended course of action is to add files without analysis and analyze several files from within the tool before kicking off the analysis of all the source code.

   If however this is a small source code repository or you are confident that the analysis will pass without issues, choose "Analyze all files". Our sample is relatively small in size so we will select "Analyzer all files".

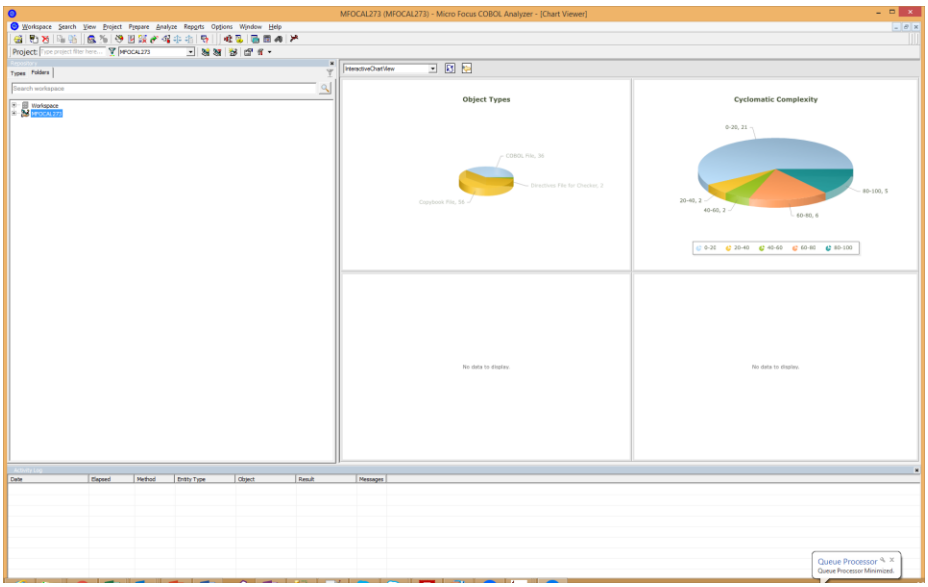

Workspace Build Wizard - Build workspace
Micro Focus® COBOL Analyzer® Workspace Build Wizard
Please select one of the following options:
Add files without analyzing
All files selected earlier will be added to the Workspace but no analysis will be done.
Analyze all files (94 File(s))
All files selected earlier will be added and analyzed.
Click 'Build' to proceed.
Help
< Back
Build
Cancel

Click “Build”

10. The Workspace build process will start. When it is finished, the newly built workspace will open automatically and the workspace build summary page will appear:



Click “Finish” and the wizard will close and you will see the COBOL Analyzer Dashboard.



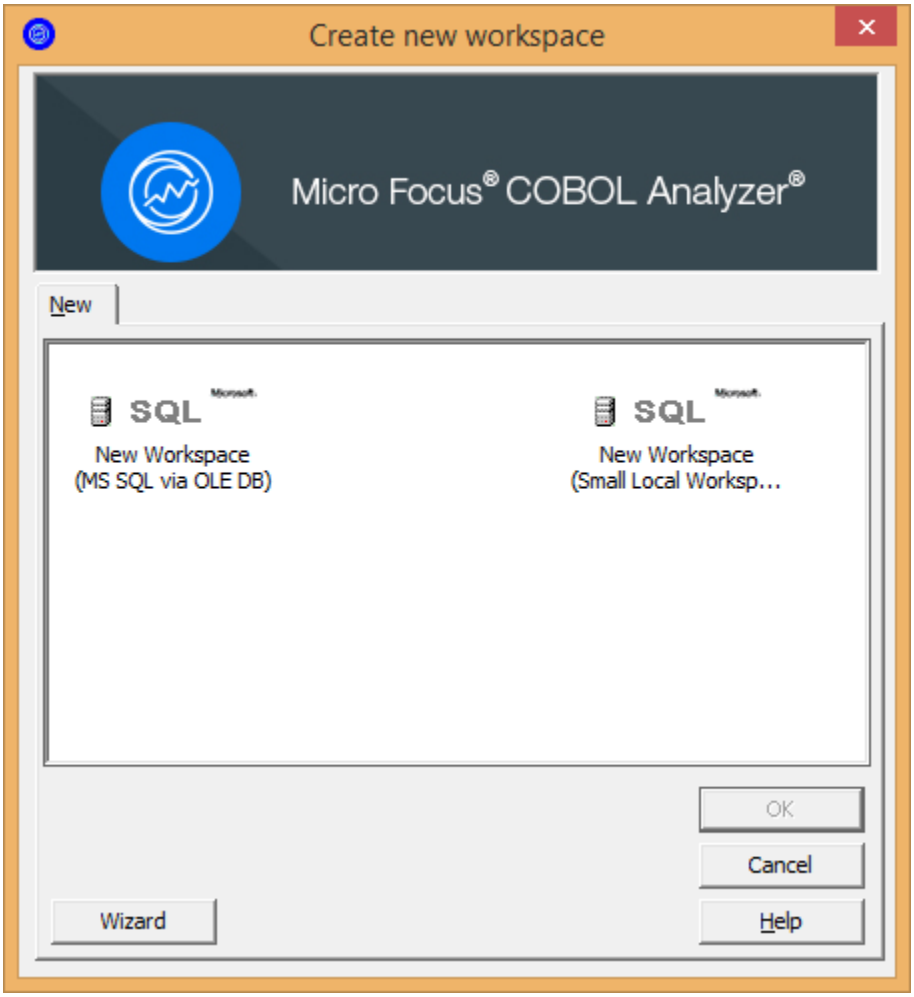
Exercise 0-2: Loading up the training material

1. Unzip CADEMO.zip
2. Open COBOL Analyzer
3. In the Open Existing Workspace dialog, click on Create New
4. **The tool will close, and a wizard will pop up, if this doesn't happen, re-open the tool as administrator and follow step 3. If this doesn't solve the problem, continue to the next exercise**
5. Follow the wizard instructions and click next
6. On the second wizard page, add **use(console.dir)** to the compiler directives box or copy and paste the content of console.dir that resides next to the source files.
7. On the next page add the MicroFocals root folder (Browse, drag and drop or copy paste should all work)
8. Click next until you get to the 5th wizard page.
9. Choose Analyze all and click build
10. You should have 1 file with errors (autosetup.cbl), this is expected and part of the training.
11. You can try solving the problem yourself using the getting started pane or follow the tutorial. In any case, one file with errors doesn't prohibit you from analyzing the rest of your project.

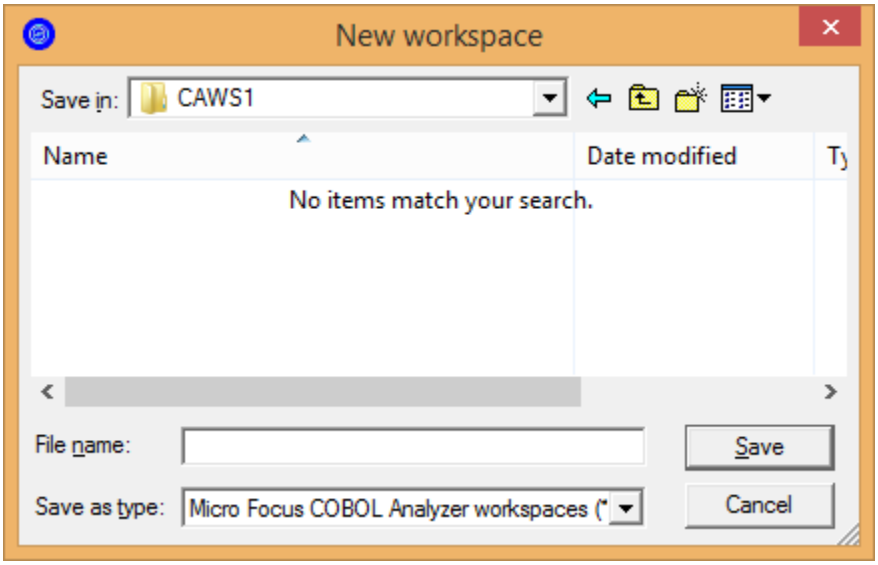
Exercise 0-3: Creating a workspace without the wizard

1. Open the COBOL Analyzer Administration
2. Click Administer→New workspace…

3. Choose the MS SQL Server Express option and click OK

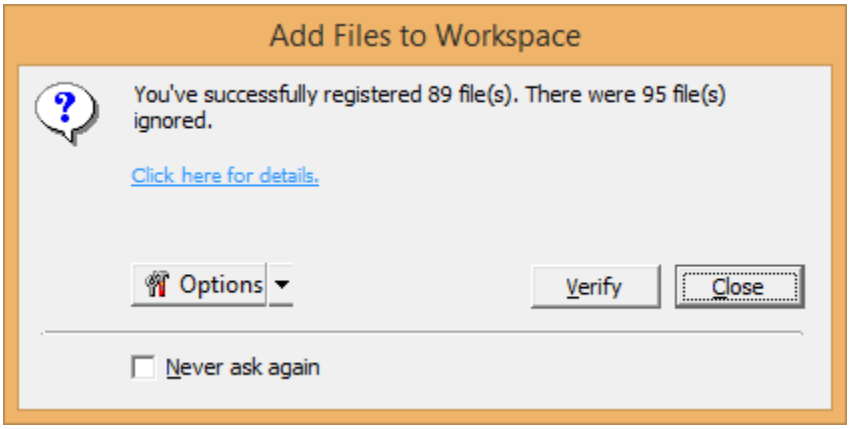


4. Assign a name and location to the workspace and click save

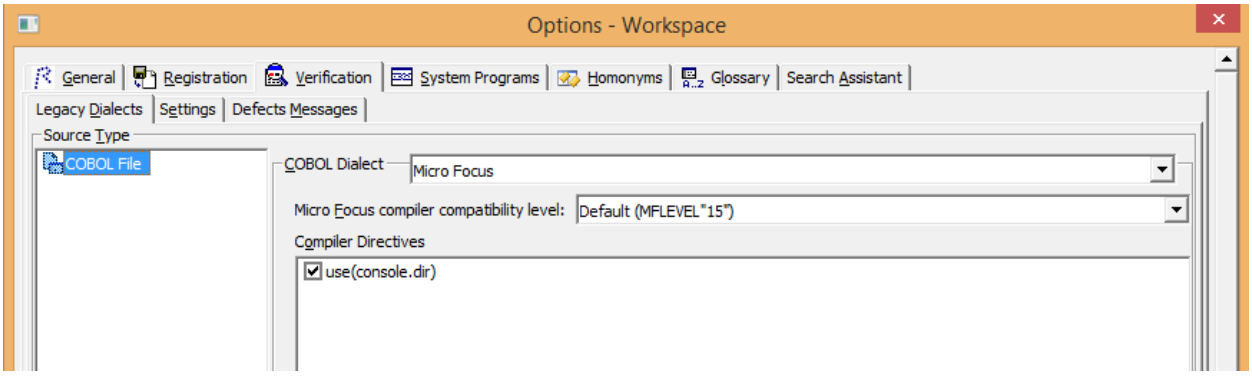


5. Open COBOL Analyzer and open the newly created workspace

6. Click Workspace→Add files to workspace and choose Add Folders
7. Browse to the MicroFocals folder that you extracted from CADEMO.zip
8. Click Continue on the Registration dialog
9. In the Add Files to Workspace dialog, DO NOT VERIFY, click close



10. Go to Options→Workspace options→Verification tab

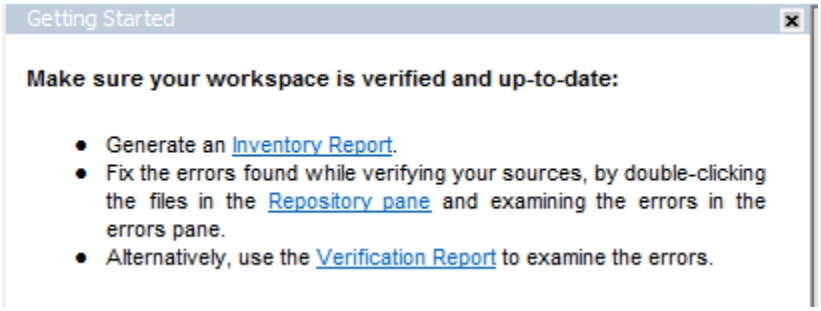


11. Right-click on the Compiler Directives box and choose Add
12. Add **use(console.dir)** or check one of the entries that already contain *console.dir*
13. Click OK
14. Right-click on the project in the repository pane and choose Verify
15. You should have 1 file with errors (autosetup.cbl), this is expected and part of the training.
16. You can try solving the problem yourself using the getting started pane or follow the tutorial. In any case, one file with errors doesn't prohibit you from analyzing the rest of your project.

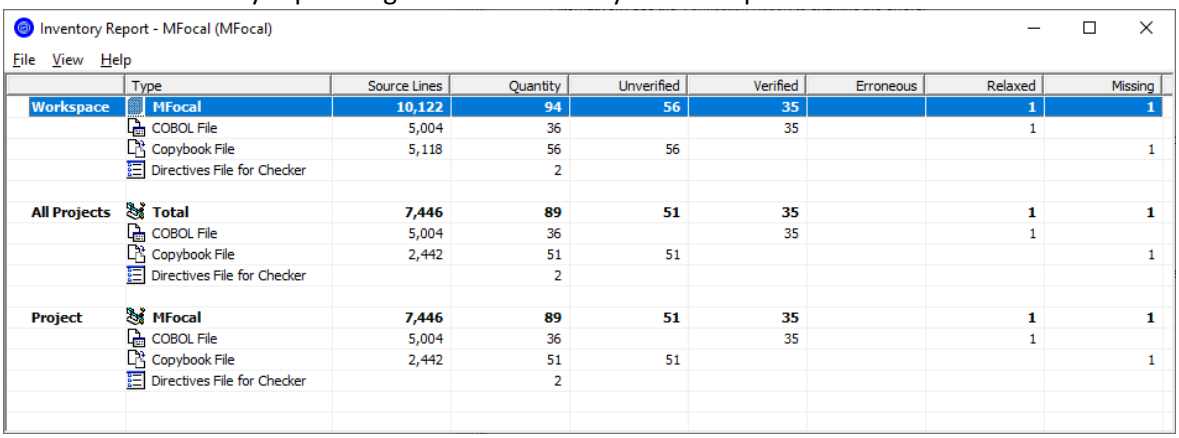
Exercise 0-4: Solving analysis errors and issues

In COBOL Analyzer and this tutorial errors in the analysis are sometimes referred to as verification errors.

1. Follow the getting started first category:



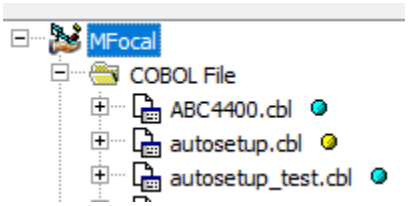
2. Click on the inventory report to get an overview of your workspace:



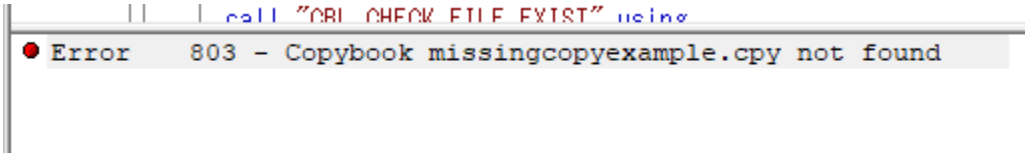
	Type	Source Lines	Quantity	Unverified	Verified	Erroneous	Relaxed	Missing
Workspace	**MFocal**	**10,122**	**94**	**56**	**35**		**1**	**1**
	COBOL File	5,004	36		35		1	
	Copybook File	5,118	56	56				1
	Directives File for Checker		2					
All Projects	**Total**	**7,446**	**89**	**51**	**35**		**1**	**1**
	COBOL File	5,004	36		35		1	
	Copybook File	2,442	51	51				1
	Directives File for Checker		2					
Project	**MFocal**	**7,446**	**89**	**51**	**35**		**1**	**1**
	COBOL File	5,004	36		35		1	
	Copybook File	2,442	51	51				1
	Directives File for Checker		2					

You can see that you have one file with minor errors (relaxed) and one missing file.

3. Double click on the autosetup.cbl file (notice the yellow icon that indicates a minor error)

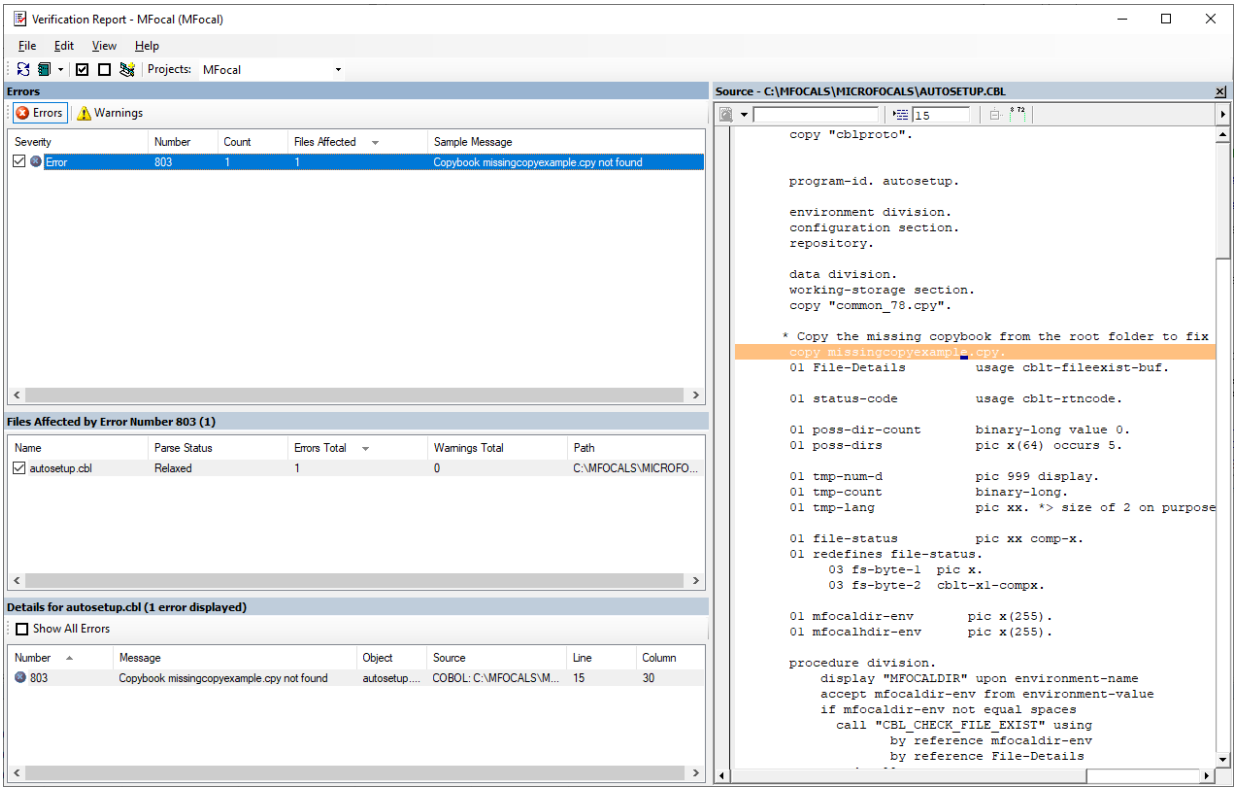


4. Look at the errors pane just below the source editor, there is a missing copybook “missingcopyexample”:



5. When you have a larger source repository with multiple errors the recommended way to resolve them is by going over them in the verification report.

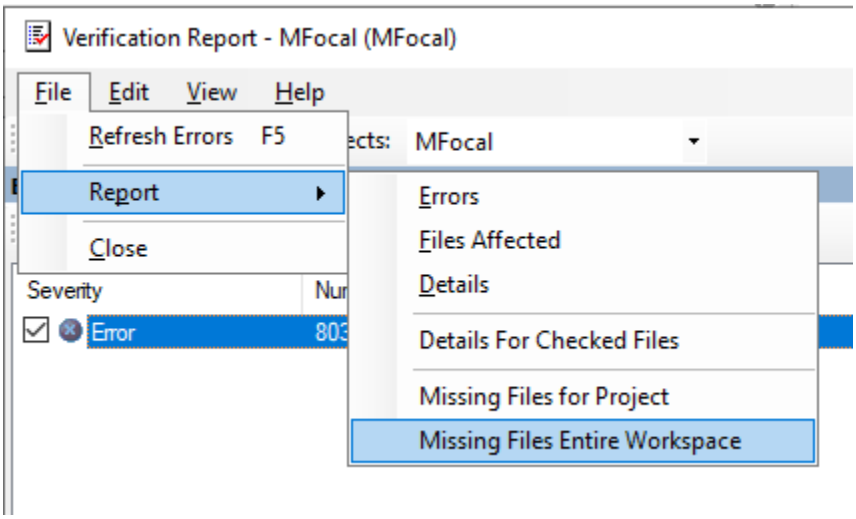
6. Click the Verification report hyperlink in the getting started pane or under Reports menu item in the main menu:



7. You can review the different errors, sort them by severity and see the number of files each error affects.

Fix the error by severity and re-verify to make sure they are fixed.

8. Click on File→Reports→Missing files Entire Worlspace:



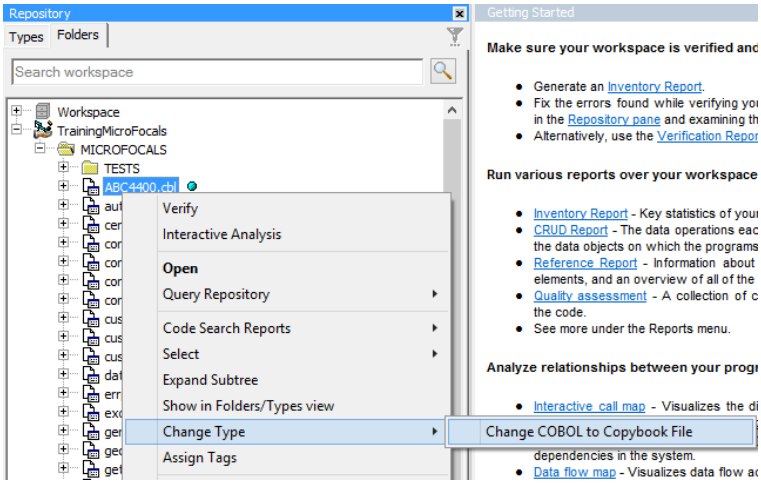
9. The report shows the missingcopyexample.cpy is missing and that it's a copybook.
10. Close the verification report.
11. Click on Workspace→Add files to workspace.
12. Choose add files.
13. Navigate to missingcopyexample.cpy

a. You can find it in the same level as the MicroFocals folder where you unzipped the training materials.

14. Right-click on autosetup.cbl and choose Verify.
15. Open the inventory report again and verify that everything verified correctly.

Common problems when loading new COBOL sources into COBOL Analyzer

- Missing files, especially copybooks:
  - Review the error
  - Search for the files
  - Add to workspace and re-verify
- Missing files due to mismatch in extensions or relative location
  - Make sure files are associated correctly in the wizard
  - If the files are already in the workspace but associated to the wrong type, delete them from the workspace and re-add with the right association.
  - If you have copybooks that was wrongly classified as COBOL files, you can right-click on them in the repository browser and select Change Type→ Change COBOL to Copybook File



- Special/custom preprocessors are used to compile the code so it's not parsed correctly by COBOL Analyzer.
  - If handling the code using the checker preprocessor directives (compiler directives box) doesn't help, preprocess the code before loading to COBOL Analyzer.
    You can create a batch file that preprocesses the code and run it on every update of the workspace (see source synchronization in Module 5 – Working with projects)
- Missing of compiler directives – As mentioned before, COBOL Analyzer uses the Micro Focus COBOL compiler to parse the COBOL code. If there are specific directives that are needed to be set in order to have the code passing compilation, make sure they are set in the appropriate place in the wizard or the workspace options.
- Wrong compiler compatibility level – If the code is originally compiler with one of the older products (Net Express or Server Express for example), the right compiler compatibility should be set in the wizard or workspace settings.

Module 1: INTRODUCTION

Purpose

This module of COBOL Analyzer training shows you how to use COBOL Analyzer to provide comprehensive analysis on your legacy system, ranging from data flow of a data item to visually exploring the relationships between legacy objects. COBOL Analyzer aids in your understanding, increasing not only knowledge of your system, but also assisting in any redevelopment or enhancement efforts.

The aims of this module are to enable an COBOL ANALYZER (CA) user to:

- Understand the relationships between repositories, workspaces, and projects
- Manage projects
- Diagrammatically and / or in report format, show relationships between legacy objects to facilitate application understanding.
- Trace the flow of data to and from a data element across a legacy system.
- Establish the potential impact on a system resulting from a change in a data item's definition or usage.

This training module is aimed at users of the COBOL Analyzer Developer Edition, containing all CA features. A separate training module with a subset of these features is aimed at users of the COBOL Analyzer Standard Edition.

Organization

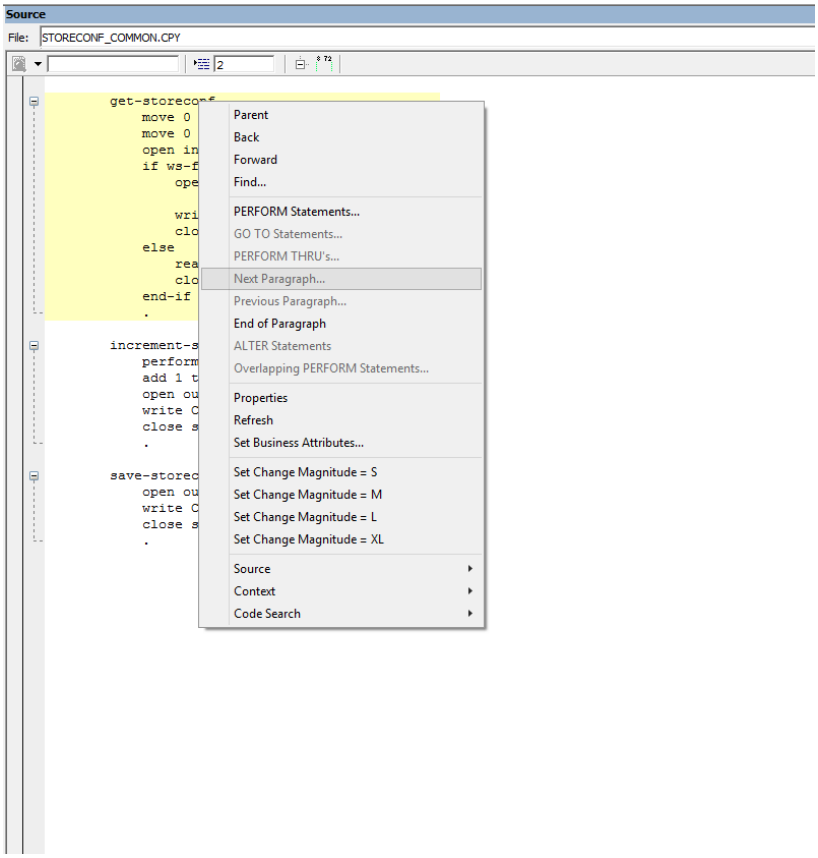
- Workspace – folder containing sources on the app server. Objects are stored at the workspace level, and there is only ever one version of the workspace sources.
- Repository – schema on the database server containing information and relationships created as a result of parsing the sources in the workspace.
- Project – a default project with the same name as the initial workspace is created automatically. Additional projects are created by the user. Projects contain "pointers" to the source; so, an object can belong to many projects. Dividing the entire workspace to project is very useful especially when using Visual COBOL or planning to upgrade to Visual COBOL. This will be covered in module 5 (working with projects)
- Folders - contain objects of a specific entity type (Cobolsource files, copybooks etc.) based on the source file extension. A Project can contain one or more folders. Some folders are automatically created as a result of parsing (Program, Screen, Transaction, etc.)
  In another view, folders can relate to actual folders on disk.

Navigation

There are multiple ways to access tools and functions within COBOL Analyzer. Many features can be accessed by one or more of the following methods:

- Menu drop-down
- Icon
- Right click

Options in CA are “context sensitive”, meaning that the options that display will apply to the object or entity you currently have highlighted or selected with the cursor. For example, you can perform the “Query Repository” function at the Project, folder or object level. You can even perform this function for a “list” in the View pane, if you have the entire list high-lighted. Options that are valid for the entity type selected, but NOT valid for the current object will be light grey (not selectable). The get-storeconf paragraph has been selected below in Interactive Analysis. There are PERFORM statements in the program for this paragraph, but no PERFORM THRU’s or GO TO statements for this paragraph. Therefore, those options are not selectable.



Each window within COBOL Analyzer Interactive Analysis consists of multiple panes which can be opened using the View option from the menu bar. Panes are activated by toggling on/off from the Menu drop-down. Some views can also be activated via function keys; for example, panes on the Repository Screen. You can close any pane in CA by clicking the ‘X’ in the upper right corner of the pane.

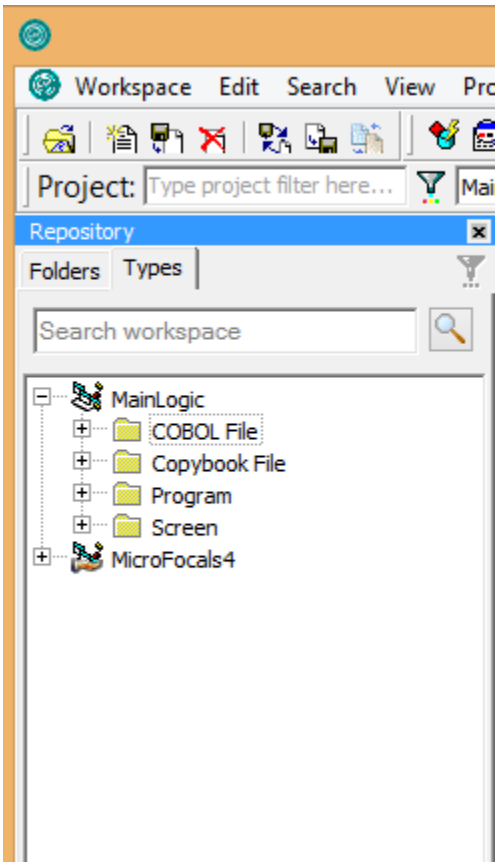
The Repository Browser pane can contain up to 3 tabs; Folders, Types and Legacy Search. In addition, “Repository Search Assistant” (Search workspace) is available as an entry box on both the Folders tab and the Types tab. The third tab, Legacy Search, becomes available if you have selected “Search” from the Menu option within the current session.

Folders

Displays the contents of the repository in the same folder structure as that in which the source was loaded. This can have the same structure as the host environment and allows for elements with the same name (homonyms) to be loaded and displayed.

Types

Displays the contents of the repository by project and entity type.



Legacy/Text Search

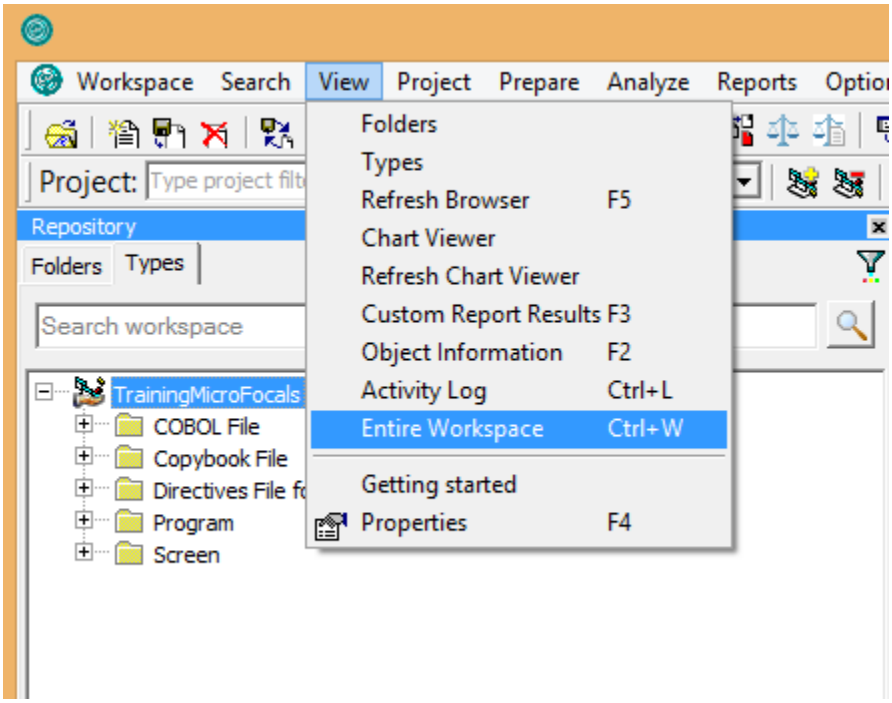
Displays the list of objects that met the Search Criteria of the last executed search. Each of the “Folder” and “Types” tabs consist of the following elements:

- Browser -- Shows the objects in the repository of the current workspace in a tree structure. Depending on the tab you have selected in the Repository Browser pane, the contents are displayed either by type or by folder.
- Search (Search Assistant) -- Lets you execute various searches in the repository of the current workspace.
- Project Filter -- Allows you to filter the projects that are displayed in the Browser pane. Project filter makes use of “wildcards” and regular expressions (except #). Wildcard * is assumed at the end of the filtered name.

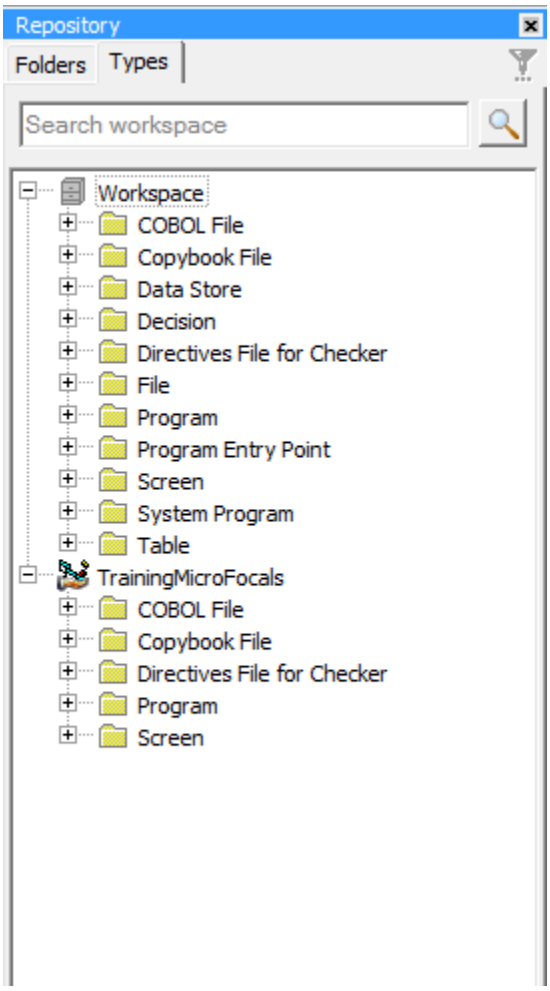
Exercise 1-1: View Entire Workspace

Objective: Open and View the Workspace

1. From the **View** menu, select **Entire Workspace**. We will use the Workspace View later on in the training. For now, be aware that the workspace contains additional information that may not be available at the project level.



2. Expand the folders under the workspace by clicking the + sign. Notice that the workspace is represented by a filing cabinet icon. Any functions that are available from the Analyze drop-down of the menu are ONLY valid for 'Projects'. So, if you have the workspace folder active (highlighted), you will not be able to choose any Analyzer functions. For that reason, we will close the Workspace view when not in use.



3. You can toggle View Entire Workspace ON and OFF by pressing **CTRL+W**.

Module 2: COBOL Analyzer DOCUMENTATION

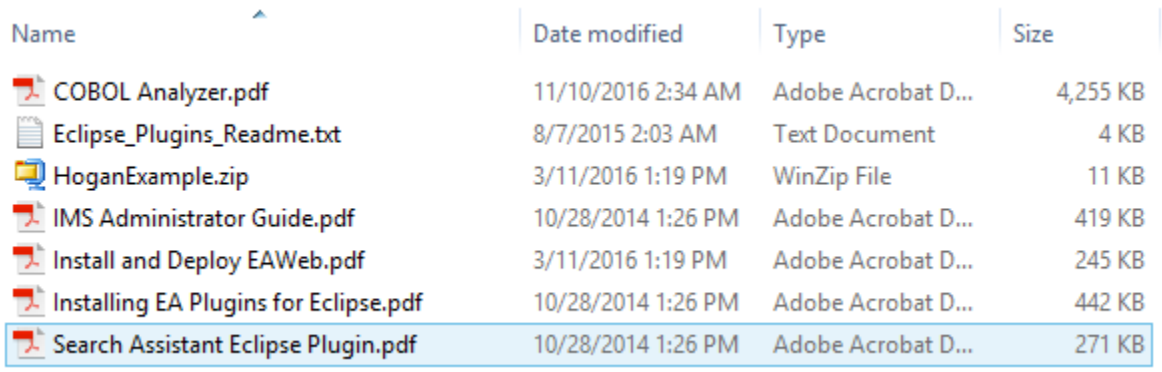
Introduction

A complete set of COBOL Analyzer manuals are installed in C:\Program Files (x86)\Micro Focus\COBOL Analyzer\Docs when you install the workbench using the default location.

Exercise 2-1: Reviewing the Documentation

Objective: To successfully open and review the COBOL Analyzer online documentation.

1. Navigate COBOL Analyzer installation folder (usually C:\Program Files (x86)\Micro Focus\COBOL Analyzer)
2. Navigate to the docs folder.
3. Review the available pdf files and select the appropriate file for viewing. COBOL Analyzer.pdf has a Getting Started section which is a good introduction to COBOL Analyzer. It also contains a description of the other manuals.

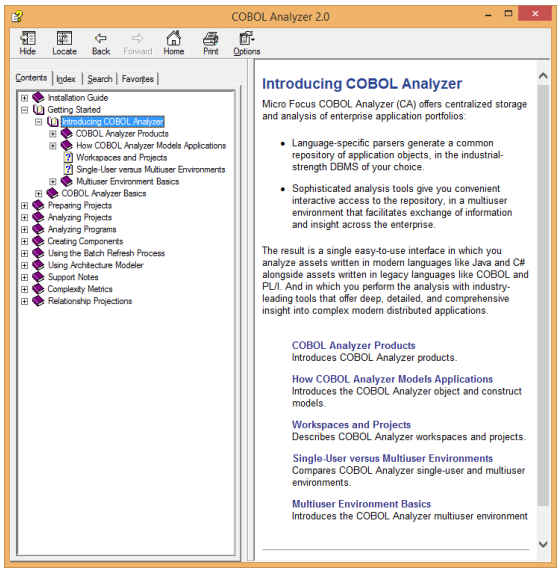


NOTE: In addition to the training manuals and User Guides provided in the Documentation folder, the application comes with a complete set of Online HELP.

Exercise 2-2: Accessing Online Help File

Objective: To successfully open and review the COBOL Analyzer online help files.

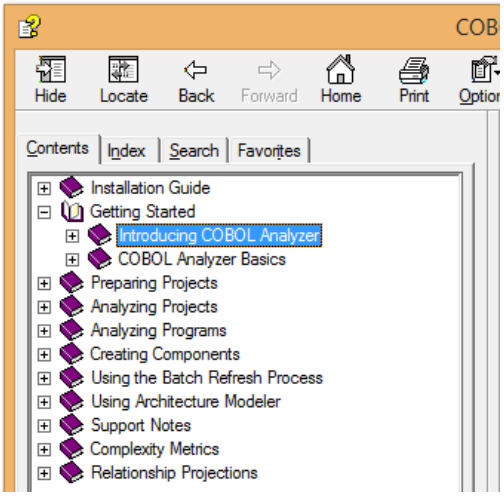
1. Press the **Help** menu item and select **Help Topics**.
2. Press the **F1** key to access context sensitive help.



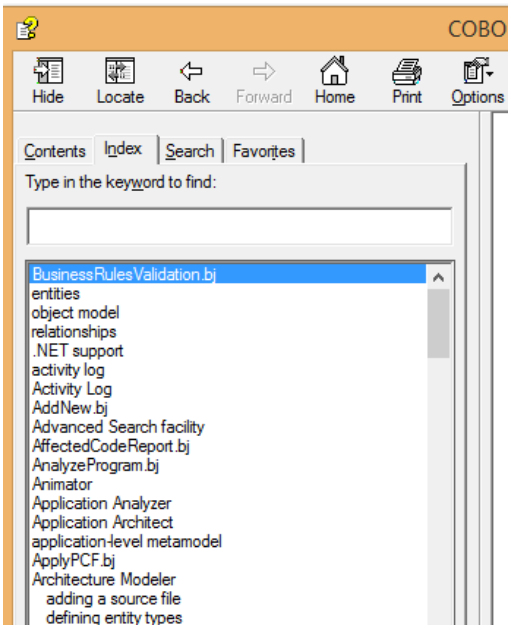
The information in the right pane is hyperlinked to enable you to navigate to the documentation easily.

The four tabs in the left pane, Context, Index, Search, and Favorites offer different interfaces for finding the information that you are looking for.

Context: The Table of Contents is organized like the external documentation. Expand the different sections of the document and click on the topic that you want displayed in the right pane.

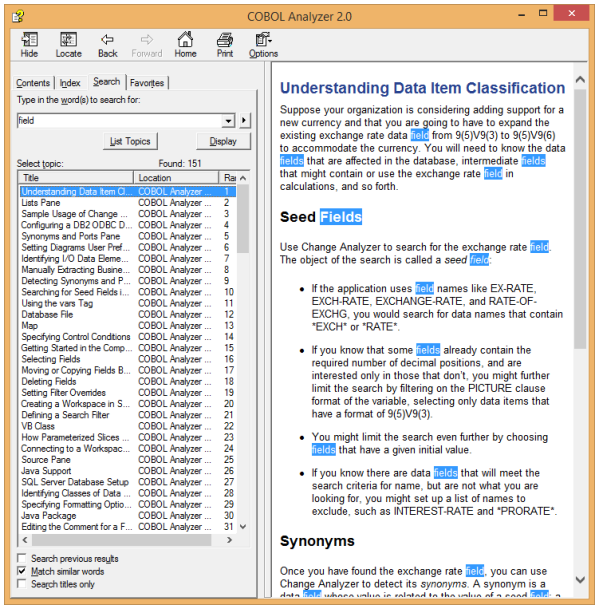


3. Click on the **Index** tab.

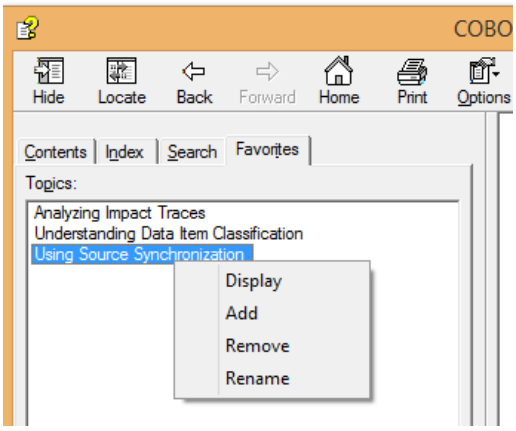


a. Type in a word in the top of the left pane to show the index for that area.
b. Click on a word or phrase in the left pane and the topic will be displayed in the right pane.
c. When there are multiple choices for the phrase, a list box pops up for you to choose which topic you are interested in.

4. Click on the **Search** tab. This tab allows you to find documentation easily.
   d. Type in a word or phrase and the matching words will display in the middle pane on the left of the window.
   e. Click on List Topics button and Topics containing that word display in the bottom left pane.
   f. Double-Click on a topic and it is shown in the right pane.



5. Click on the **Favorites** tab. This tab allows you store commonly used Help windows for easier access in the future.

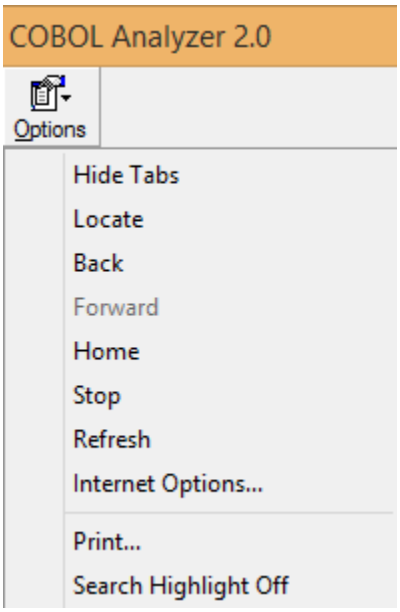


- **Add** stores the current page as a favorite
- **Display** shows the selected page in the main Help window
- **Remove** removes the currently selected page from the Favorite list
- **Rename** allows you to change the name (defaulted from the help text) to something more meaningful to you

6. Use the following buttons to navigate through the Help documentation:

- **Hide:** removes the list on the left side of the screen so that only the help topic is displayed. Show redisplays the list.
- **Locate:** Identifies and highlights the document where the help is defined
- **Back**: returns to the previous displayed page
- **Forward**: advances to the next page
- **Home**: returns to the help home page
- **Print**: prints the page

Under the Options drop-down, are all of the above choices plus these options:



- **Stop**
- **Refresh:** refreshes the screen display
- **Internet Options:** takes you to the Internet Options menu for your browser
- **Search Highlight Off:** toggles OFF or ON the highlighting of words that you are searching for in the Help text

7. When you are finished, press the **X (Close)** button to close Help.

Module 3: Configuration Options

Introduction

This section describes how to set various options within COBOL Analyzer. Options within COBOL Analyzer can be set in two ways — either from the Tools menu or from the  button on the toolbar.

Objectives

- You will explore the Options –User Preferences dialog box
- You will set values as necessary for your project needs

User Preference Options

There are three different types of options in COBOL Analyzer: User Preferences, Workspace Options, and Project Options. User Preferences apply only to the current user. Workspace Options apply only to the currently open workspace. Project Options apply only to the project currently selected in a workspace. Project options and Workspace options will be covered in detail in the COBOL Analyzer Administrator training manual but those important to the regular user are described during the relevant exercises in this course.

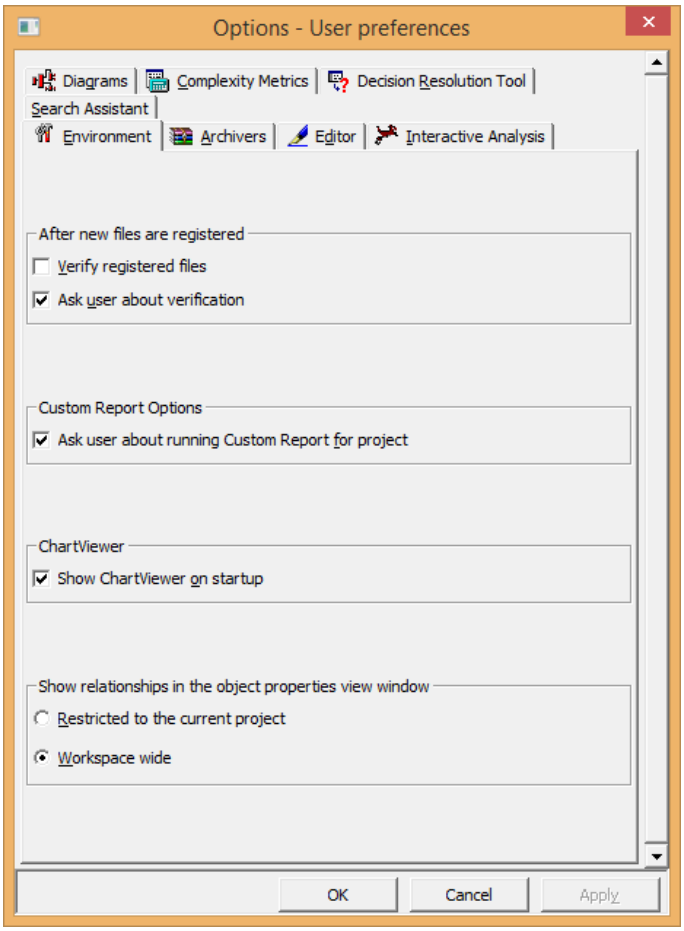
Note: In an Enterprise wide installation, unless you make these changes the default in Options Manager, changes you make to the Project or Workspace Options will apply only for your user id. The Master user is the ONLY one who should use Options Manager.

Environment

Exercise 3-1: Setting User Preferences

Objective: You will set user preferences in the User Preferences dialog box

4. From the **Options** menu, select **User Preferences.** Alternatively, you can press the Tools button  on the toolbar.



Here you can see and change your current User Preferences settings. Review the other tabs and the options described on them. Use F1 for help on what each option does.

We recommend setting the options as you see here. This should ensure that what follows is not affected by any option changes.

"Show Chart Viewer on startup" is typically an option that can be turned off, especially if you do not have the correct software to display the diagrams.

Depending on the Language add-ons that may be installed, this pane may vary. For example, if Java parser is installed there is an additional option to provide the path to the required JRE folder.

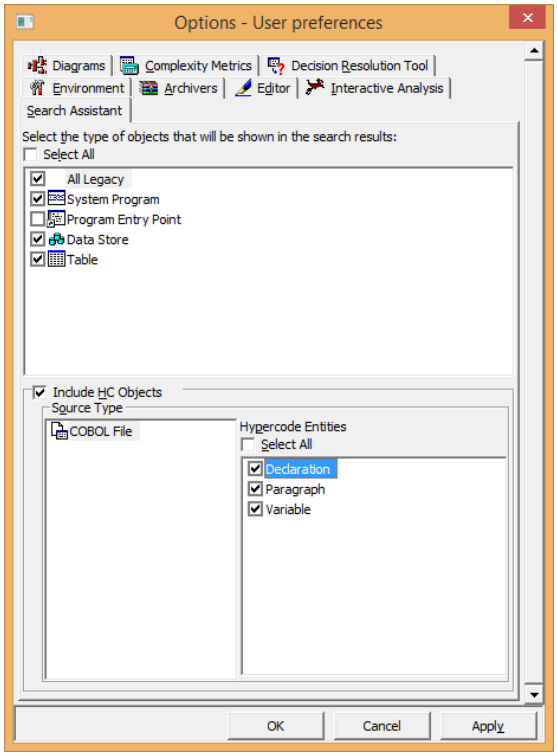
5. Click on the appropriate tab across the top of the dialog box and set the values as necessary. You can press the F1 key for information regarding each option.
6. When you are finished setting User Preferences, press **OK**.

Search Assistant

Exercise 3-1: Setting the Search Assistant options

1. Click the **Search Assistant** tab.

2. Set the Search Assistant options as seen on the left. This will ensure that future exercises will have consistent results with the training material.



3. Click **Apply** and/or **OK**.

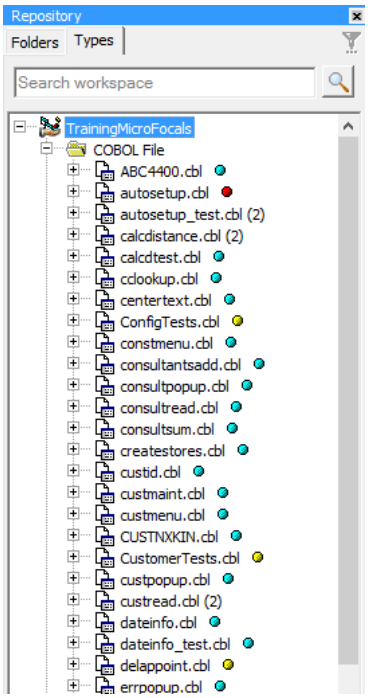
Module 4: Repository Features

Introduction

On the Repository Browser pane, COBOL Analyzer allows you to filter and select objects by name with or without using wildcards. Filtering is done by object type and can be used to select specific objects by full or wild carded name.

Exercise 4-1: Advanced Type Ahead

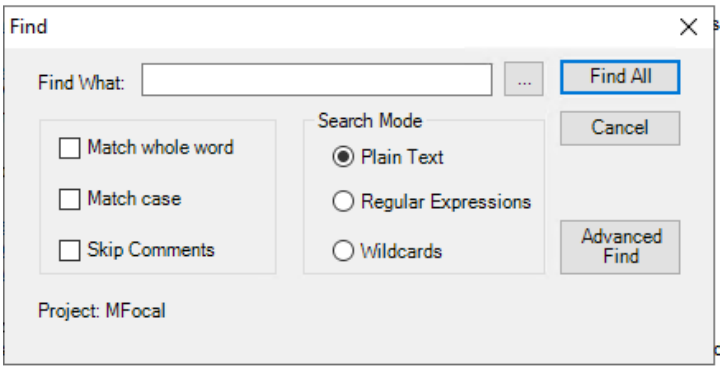
7. Switch to the Types tab in the **Repository** pane.
8. In the **Repository** pane, click the **+** icon next to the **Cobol File** folder to expand it.
9. Click anywhere within the list and start typing dat. *Depending on how fast you type,* your cursor will be positioned at the first object within the list that matched what you typed.



Exercise 4-2: Text Search of the Repository

COBOL Analyzer allows you to search the objects in the workspace repository using various search criteria as filters. You can filter on object name, verification status, and matching text. You can also replace text, if you have CA Administrator access. A more advanced Search facility is available in Interactive Analysis that we will look at later.

1. Select the project.
2. From the **Search** menu, select **Find** or press CTRL+F. You will see the **Find** dialog box.

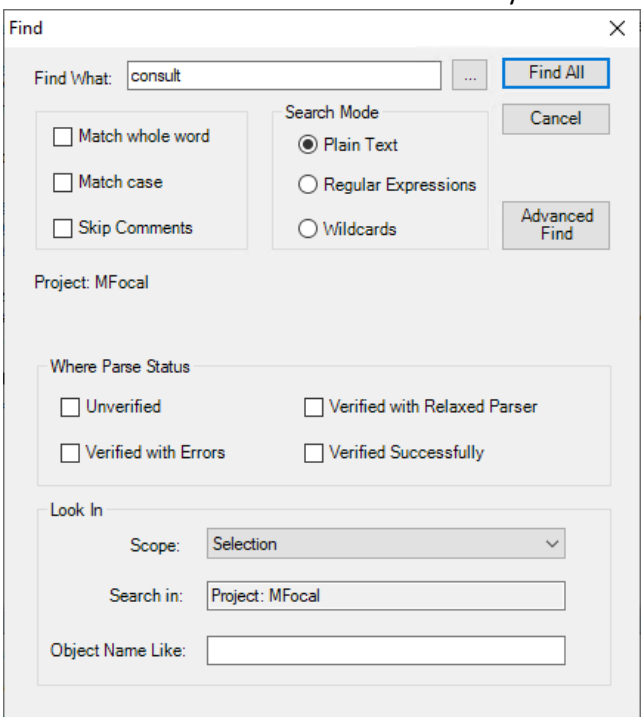


3. Click the Advanced Find button to see additional options
4. Enter **consult** as the search string. You can enter plain character strings, regular expressions or wildcarded search strings.

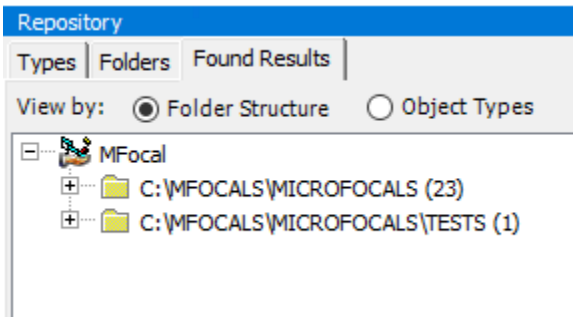
   For regular expressions or wildcards, the right arrow button next to the drop-down will display a list of elements you can insert in the pattern.

   You can use POSIX regular expressions and wildcard patterns allowed in LIKE statements by .NET Applications.

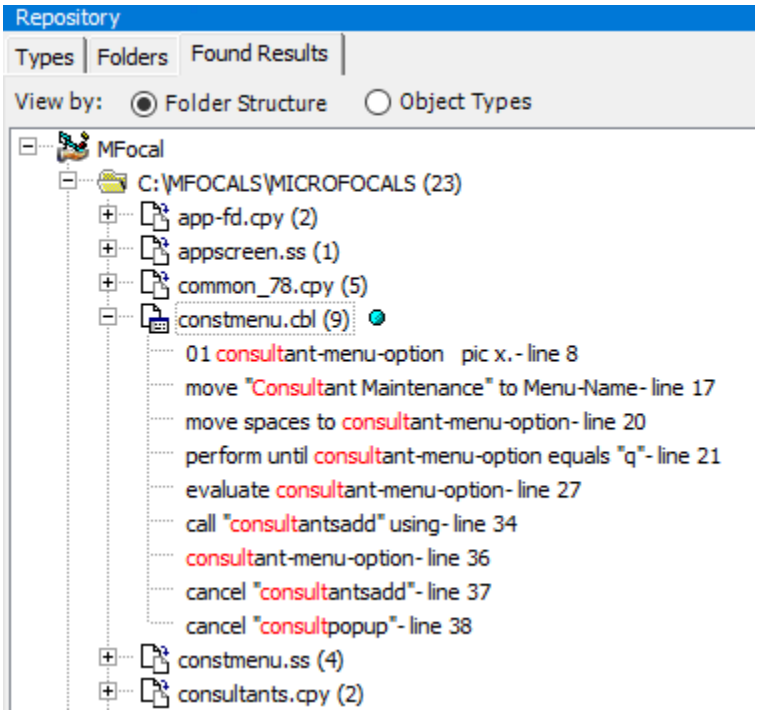
5. Check the **Skip Comments** and the **Verified Successfully** check boxes. This will return results from actual uncommented code in files that were analyzed without any errors or warnings only:



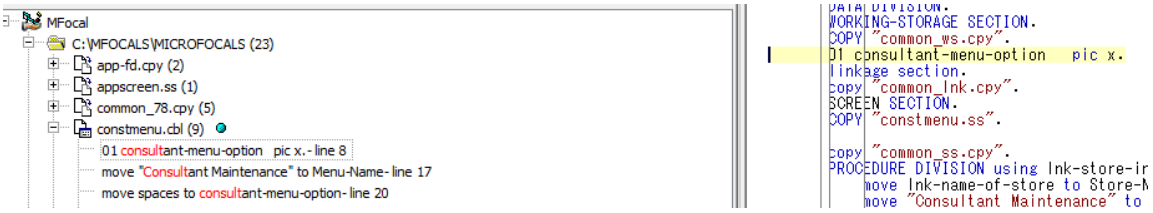
6. Press the **Find All** button. The Found Results tab of the Repository pane will be populated with the list of folders containing files with the string "consult".



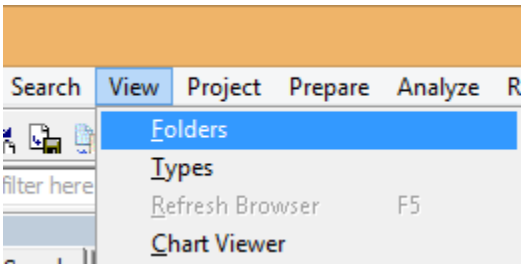
7. Expand the nodes to view the files and results



8. Double clicking on a result will navigate to the relevant line of code and highlight it



9. Click **View > Folders** or **Types**. This takes you back to the main Repository Browser window.

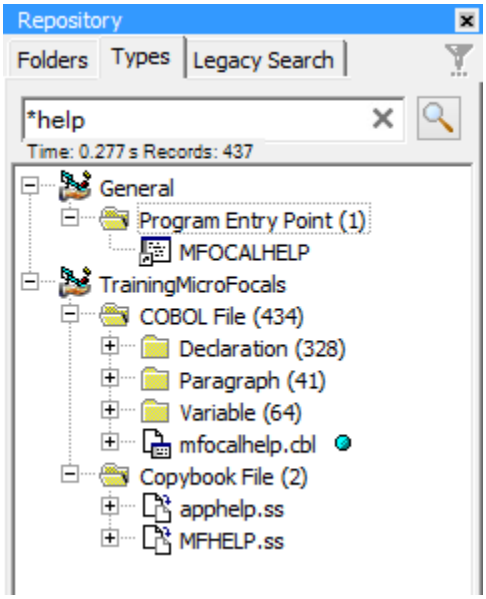


You may also click the **Folders** or **Types** tab at the top left of the Repository Browser pane.

Exercise 4-3: Using the Repository Assistant

CA has a facility to allow for a quick search over the repository. With this ability, the user can type in any name in the system; Program, Variable etc. and get a result list of all objects that match the search pattern (within Entity type). For each object in the list, the user can "right click" and see the possible actions available for that object. User Preferences contains the default options selections which can be modified.

1. From the Repository Browser, type a value into the **Search** field and click the Find icon or press Enter.
2. Enter ***help*** in the item to search for: input area.

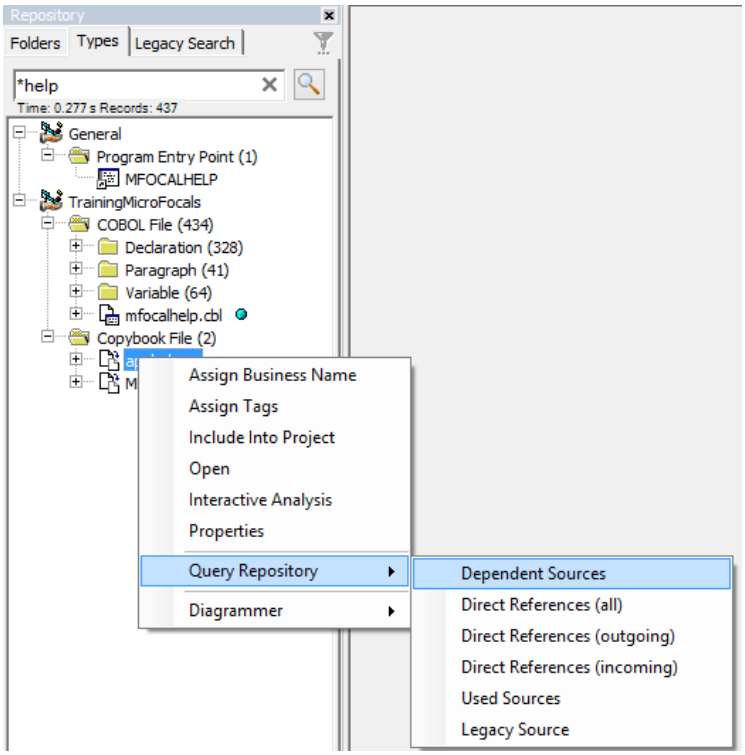


Please note that prior to accessing this search you will need to turn on ALL options under User Preferences so that your results will be the same as the ones on the left.

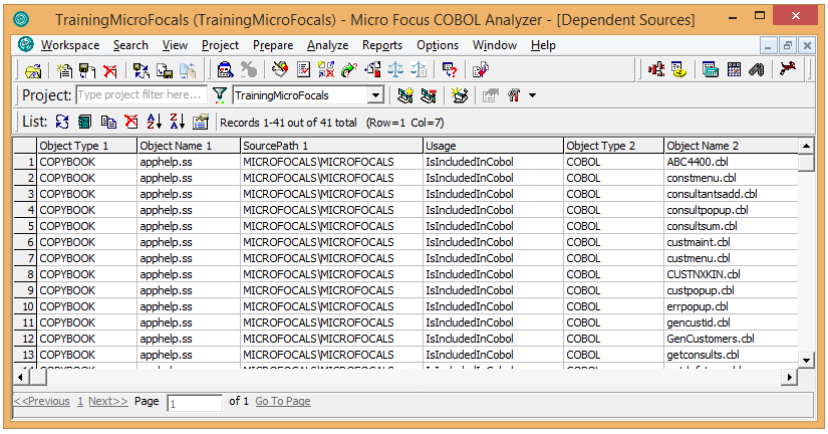
Right click on any object to see the available functions for that object. Options depend on entity type.

3. Right click on copybook **apphelp.ss**. You will see the options that can be selected for Copybooks.

4. Right click on Query Repository and you can see the queries that are available for Copybooks.



5. Select Dependent Sources (sources that are dependent on this copybook). The results are displayed in the Edit / View pane on the right side of the screen.



Syntax Table for POSIX Regular Expressions

Syntax	Expression Description

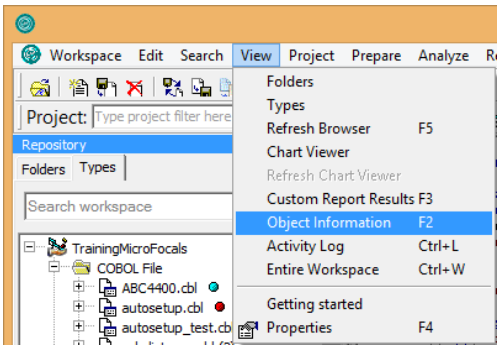
\	Marks the next character as a special character. "n" matches the character "n". "\n" matches a linefeed or newline character.
^	Matches/anchors the beginning of line.
$	Matches/anchors the end of line.
*	Matches the preceding character zero or more times.
+	Matches the preceding character one or more times.
.	Matches any single character except a newline character.
(expression)	Brackets or tags an expression to use in the replace command. A regular expression may have up to 9 tagged expressions, numbered according to their order in the regular expression. The corresponding replacement expression is \x, for x in the range 1-9. Example: If (h.*o) (f.*s) matches "hello folks", \2 \1 would replace it with "folks hello".
[xyz]	A character set. Matches any characters between brackets.
[^xyz]	A negative character set. Matches any characters NOT between brackets.
\d	Matches a digit character. Equivalent to [0-9].
\D	Matches a non-digit character. Equivalent to [^0-9].
\f	Matches a form-feed character.
\n	Matches a linefeed character.
\r	Matches a carriage return character.
\s	Matches any white space including space, tab, form-feed, etc. but not newline.
\S	Matches any nonwhite space character but not newline.
\t	Matches a tab character.
\v	Matches a vertical tab character.
\w	Matches any word character including underscore.
\W	Matches any non-word character.
\p	Matches CR/LF (same as \r\n) to match a DOS line terminator.

NOTE: You can search text that contains special characters ("\", "^", "$", "+", "." and brackets) by entering preceding back slash before each special character. For example, expressions "A + B is ^C" and "C + D at ^@" will match to the pattern "\D \+ \D .. \^\D".

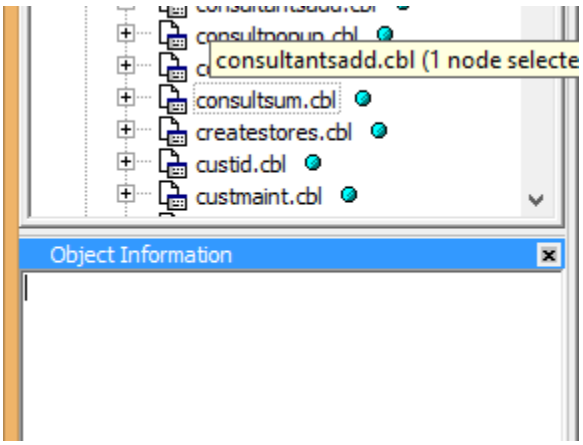
Exercise 4-4: Adding Object Information

Within your repository, you can add annotation to an object for clarification or informational purposes. This can be very useful to share information among multiple users of COBOL Analyzer.

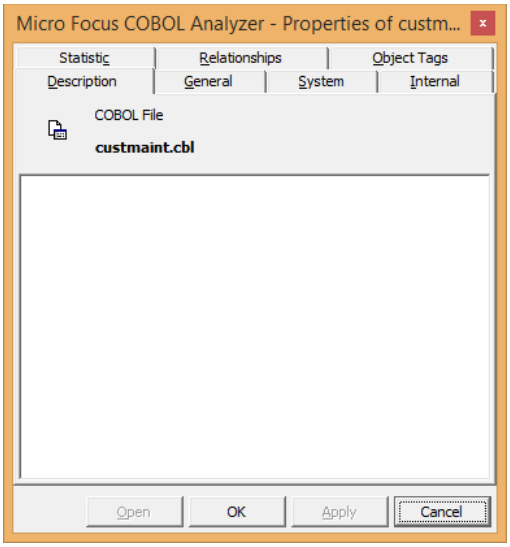
1. From the **View** menu, select **Object Information**. You will see the Object Information pane.



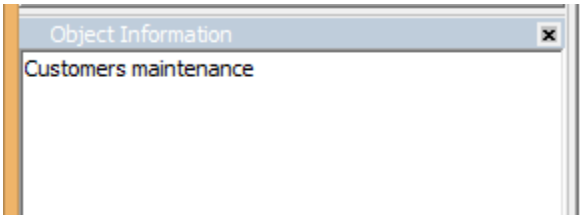
Notice the new pane that appears. This new pane will display any information that we wish to use to describe this object.



2. Right-click on **custmaint.cbl** and select **Properties**. You will see the Properties dialog box into which you can make annotations or add information about the selected object.
3. Click the **Description** tab.



4. Inside the description field, type **Customers maintenance**. This information is internal to the tool, but when provided can be a valuable asset to entry level programmers or staff who are unfamiliar with this program.
5. When you are finished, press **OK**. Notice the new information that appears in the Object Information pane (you may need to reselect **custmaint.cbl** or press F5 to refresh this screen). You can use this to add Comments or any other vital information necessary for your Project.

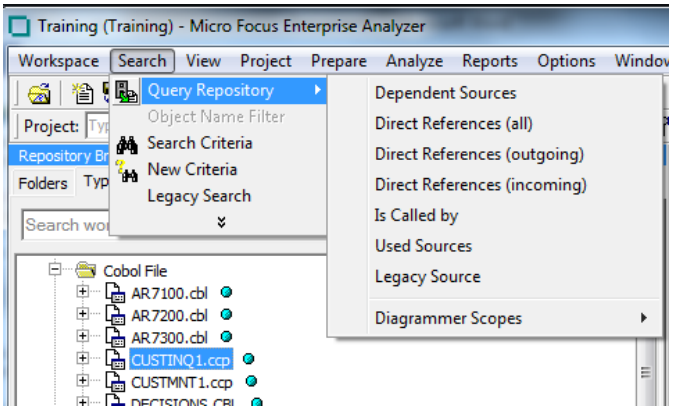


Exercise 4-5: Querying the Repository

Within your repository, you can quickly perform queries to return information about relationships between objects in the repository. This can be very useful to provide high level impact analysis on your legacy objects or answer basic usage questions.

Querying the Repository can quickly identify high-level relationship information across legacy objects. It can be used to answer basic questions developers have about their system, especially regarding related components which can be affected by a change.

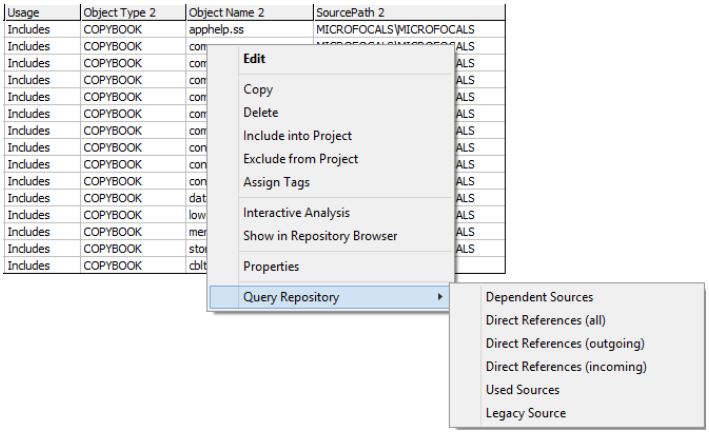
1. Return to the **Repository Browser** select **consultpopup.cbl**.
2. From the **Search** menu, select **Query Repository** and then **Used Sources**. Opens and runs the Repository Query tool and runs 'Used Sources' query. Notice the available queries for the Cobol File type. This menu is context-sensitive, meaning that the queries will differ based on the file type that you choose.



Notice the new report that appears on the right. This pane will display the results of the Used Sources query. In this case, it returned the copybooks used by our selected program.

	Object Type 1	Object Name 1	SourcePath 1	Usage	Object Type 2	Object Name 2
1	COBOL	consultpopup.cbl	MICROFOCALS\MICROFOCALS	Includes	COPYBOOK	apphelp.ss
2	COBOL	consultpopup.cbl	MICROFOCALS\MICROFOCALS	Includes	COPYBOOK	common.cpy
3	COBOL	consultpopup.cbl	MICROFOCALS\MICROFOCALS	Includes	COPYBOOK	common_78.cpy
4	COBOL	consultpopup.cbl	MICROFOCALS\MICROFOCALS	Includes	COPYBOOK	common_ex.cpy
5	COBOL	consultpopup.cbl	MICROFOCALS\MICROFOCALS	Includes	COPYBOOK	common_lnk.cpy
6	COBOL	consultpopup.cbl	MICROFOCALS\MICROFOCALS	Includes	COPYBOOK	common_ss.cpy
7	COBOL	consultpopup.cbl	MICROFOCALS\MICROFOCALS	Includes	COPYBOOK	common_ws.cpy
8	COBOL	consultpopup.cbl	MICROFOCALS\MICROFOCALS	Includes	COPYBOOK	consultants-fc.cpy
9	COBOL	consultpopup.cbl	MICROFOCALS\MICROFOCALS	Includes	COPYBOOK	consultants.cpy
10	COBOL	consultpopup.cbl	MICROFOCALS\MICROFOCALS	Includes	COPYBOOK	consultpopup.ss
11	COBOL	consultpopup.cbl	MICROFOCALS\MICROFOCALS	Includes	COPYBOOK	dataflow.cpy
12	COBOL	consultpopup.cbl	MICROFOCALS\MICROFOCALS	Includes	COPYBOOK	lowerpopup.ss
13	COBOL	consultpopup.cbl	MICROFOCALS\MICROFOCALS	Includes	COPYBOOK	menuheader.ss
14	COBOL	consultpopup.cbl	MICROFOCALS\MICROFOCALS	Includes	COPYBOOK	stores.cpy
15	COBOL	consultpopup.cbl	MICROFOCALS\MICROFOCALS	Includes	COPYBOOK	cbltypes.cpy

3. Right-click on an object in the results pane. You will see different options to perform on your results. You can include these results in other projects and perform other queries on the resulting set in the Query Repository.



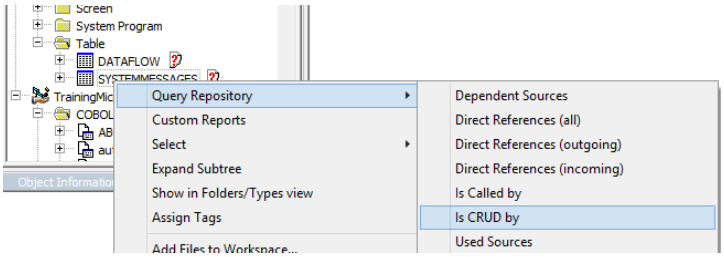
We can also perform Query Repository on multiple objects using the Ctrl / left click on each object. Additionally, you can select the folder or project level to perform queries on an entire folder (all COBOL files) or to query against the entire Project.

Query Repository can be used to quickly identify high-level relationship information across legacy objects. It can be used to answer basic questions developers have about their system, especially regarding related components which can be affected by a change.

Exercise 4-6: Querying Repository Workspace Objects

Some folders are only available at the workspace level. In the Training workspace, table relationships were created as a result of parsing the sources that reference those tables. The ? indicates that the Create Table DDL does not exist in the Workspace. View / Entire Workspace and expand the Table folder. We will use the existing table to find which program(s) access the table and how it is accessed.

1. With the SYSTEMMESSAGES table selected, right-click and select **Query Repository > Is CRUD BY**. This will open and run the CRUD (Create, Read, Update, and Delete) query for tables.



This is a very useful query to run in a production environment when you are interested in identifying all table access within your application. The report below can be saved by using the report icon above the List pane.

	Object Type 1	Object Name 1	Relationship	Object Type 2	Object Name 2	SourcePath
1	TABLE	SYSTEMMESSAGES	IsInsertedIn	COBOL	ABC4400.cbl	MICROFOCALS\MICROFOCALS
2	TABLE	SYSTEMMESSAGES	IsReadIn	COBOL	mfocal.cbl	MICROFOCALS\MICROFOCALS

In the View Pane on the right side of the screen, you will see that table SYSTEMMESSAGES is read by mfocal.cbl updated in ABC4400.cbl.

2. When you are finished, close the workspace view by pressing CTRL+W.

Module 5: Working with Projects

Introduction

It is useful to create additional projects in a workspace when you want to break large source deliveries down into subsystems, either to partition work into more manageable units, or to perform separate analysis exercises on partial subsets of sources.

When breaking a system down into separate workspaces and projects, it is important to first understand the relationships between the two containers. When a workspace has multiple projects, objects in one project can still recognize objects in another project. Therefore, some relationships and workspace-wide features will still apply across all projects in a workspace. However, relationships do not exist across different workspaces. One workspace is completely independent of another workspace. In addition, you can only work with one workspace at a time within COBOL Analyzer. Understanding these relationships can be crucial in proper management and partitioning of an application portfolio in COBOL Analyzer.

Also, only verified files have the object relationships defined that enable you to include referenced or referencing objects in a project to ensure a closed system. You can include all referencing objects or only "directly referencing" objects: If program A calls program B, and program B calls program C, A is said to directly reference B and indirectly reference C. You can also remove unused support objects.

Visual COBOL users and migration paths

While currently there is no automatic recognition of Visual COBOL projects in COBOL Analyzer, it is very useful to create the corresponding projects to match the Visual COBOL setup. In this way, the analysis information can help you assess potential impact across the entire workspace (cross Visual COBOL projects) and highlight missing dependencies in your different projects.

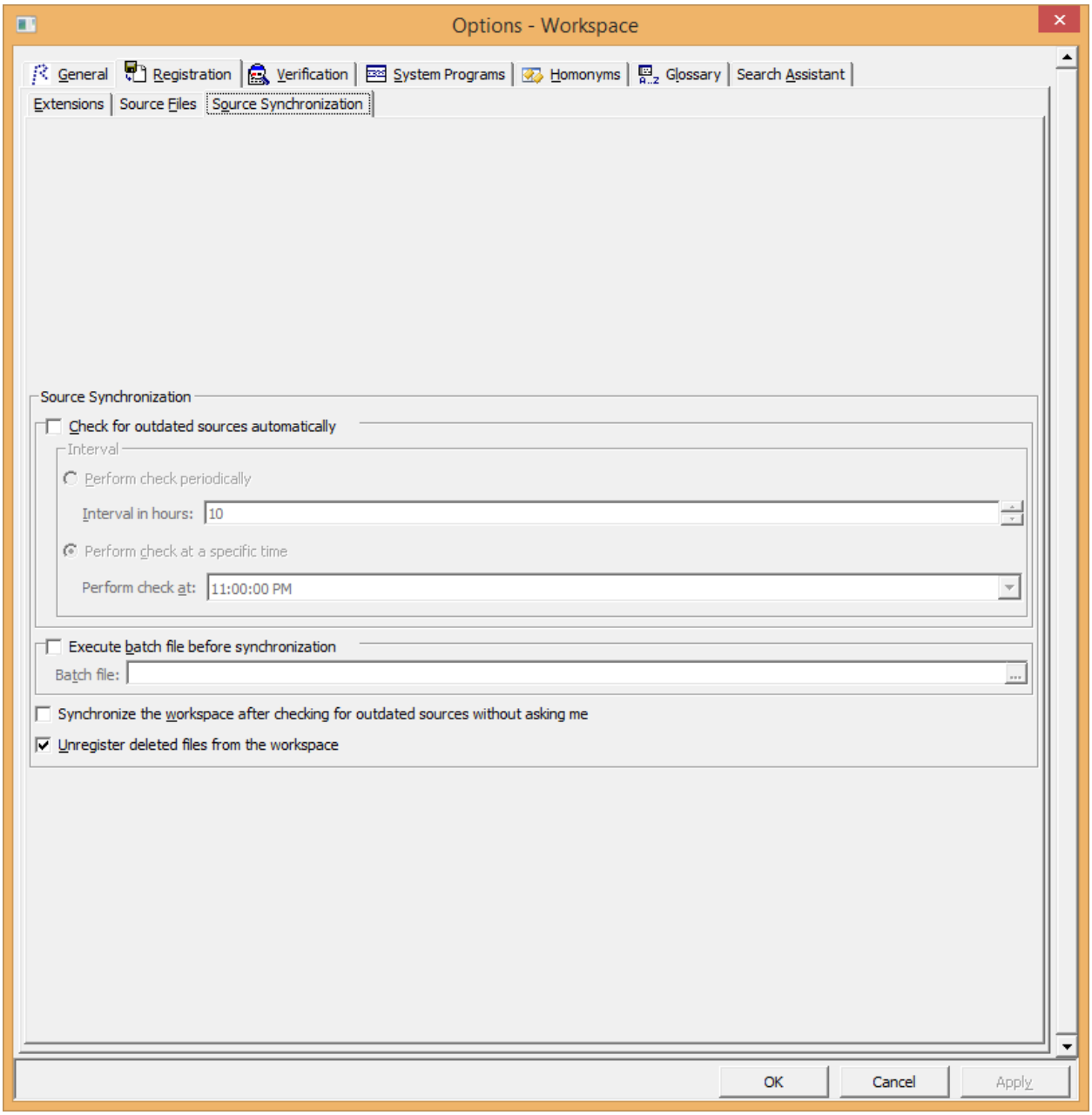
For customers who are exploring migrating to Visual COBOL, it is highly recommended to first break the source deliveries down into project as mentioned in the introduction and then create the matching projects in Visual COBOL (for Visual Studio or Eclipse).
It is also recommended to run the **Migration Report** that will include useful information for planning and executing such a migration (See module 6 – Working with reports for more details).

Analyzing the working copy of your code while developing

A key feature of COBOL Analyzer is the ability to analyze the code you are currently working on (working copy). This allows a developer to analyze the code before and after changes and fixes and assessing the impact of those changes before committing the code.

While you can choose your favorite method of code editing, being it a simple text editor or an advanced Integrated Development Environment (IDE) such as Visual COBOL, the source synchronization feature will allow you to keep your analysis up to date with you working copy.

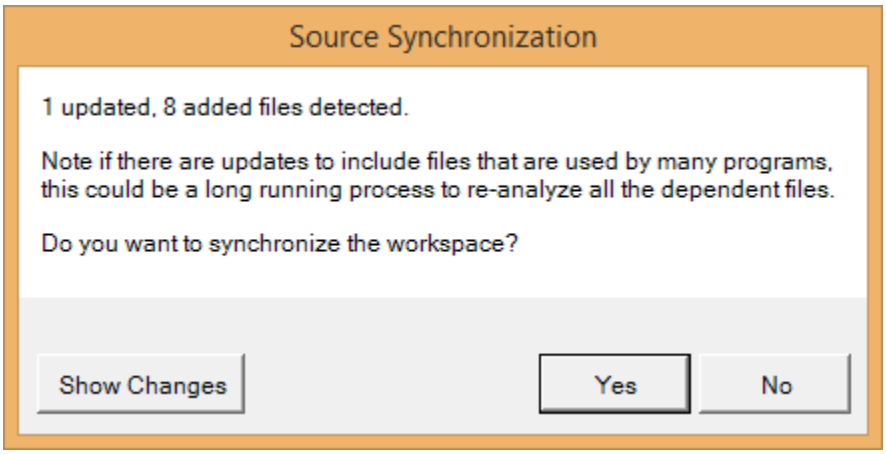
Source synchronization is achieved by several methods depending on your settings (Options→Workspace options→Registration tab→Source Synchronization tab:

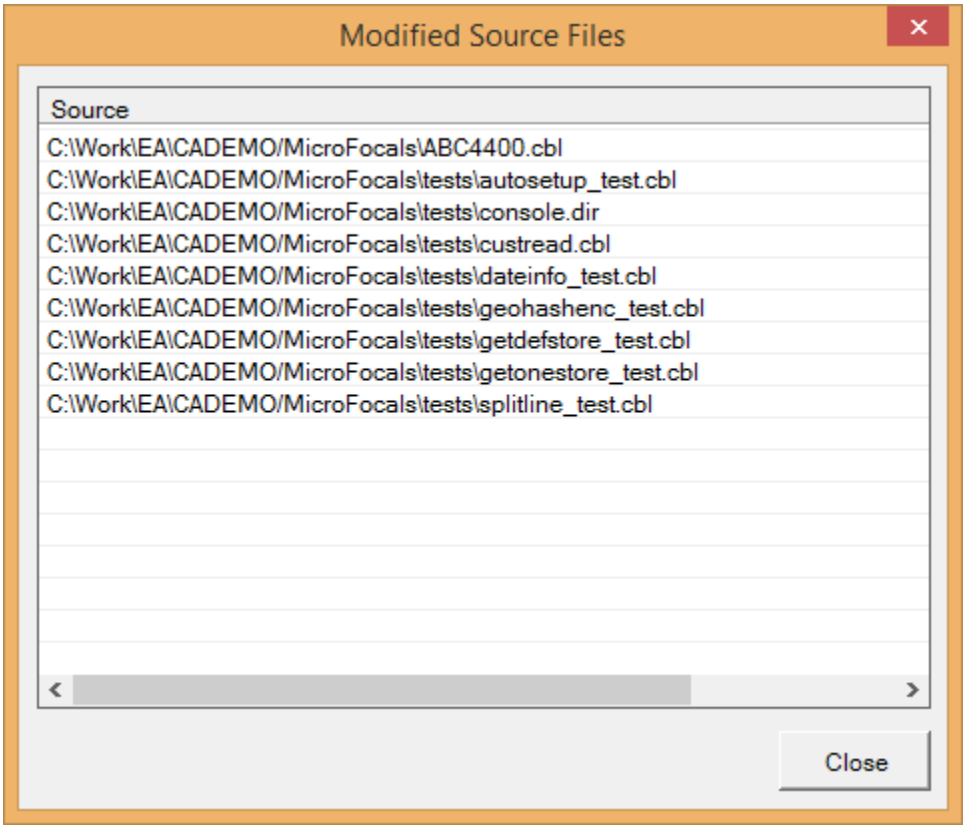


The Execute batch file before synchronization can be used for copying code, updating from source control, preprocessing etc.

Once you have returned to COBOL Analyzer after making code changes in another tool, you can run the source synchronization from Workspace→Source Synchronization or using the toolbar button ()

You'll get a message box with information regarding the changed files:





Importing analysis data into Visual COBOL for Eclipse

In Visual COBOL for Eclipse, you can import code searches results from COBOL Analyzer into the IDE.
In order to add this capability, you'll need to install the COBOL Analyzer Eclipse plugin by following the instructions in the COBOL Analyzer installation folder under docs.

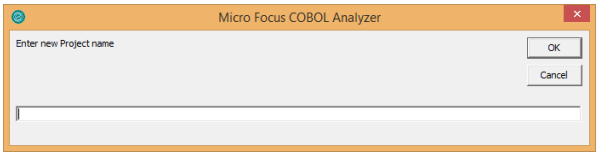
When you perform code searches in COBOL Analyzer, you create a list of Point of Interest (POI). These lists can be imported to Eclipse allowing you to continue the development work inside the IDE by quickly navigating the results and make the necessary changes.

More information about code searches and importing them is available in module 7 – Working with Analysis Tools.

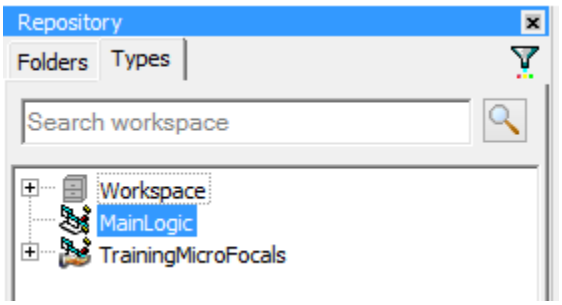
Exercise 5-1: Create New Projects in a Workspace

Objective: You will create a new project in a workspace

10. Click **Project** on the menu and select **New Project**. You will see a screen similar to the following:



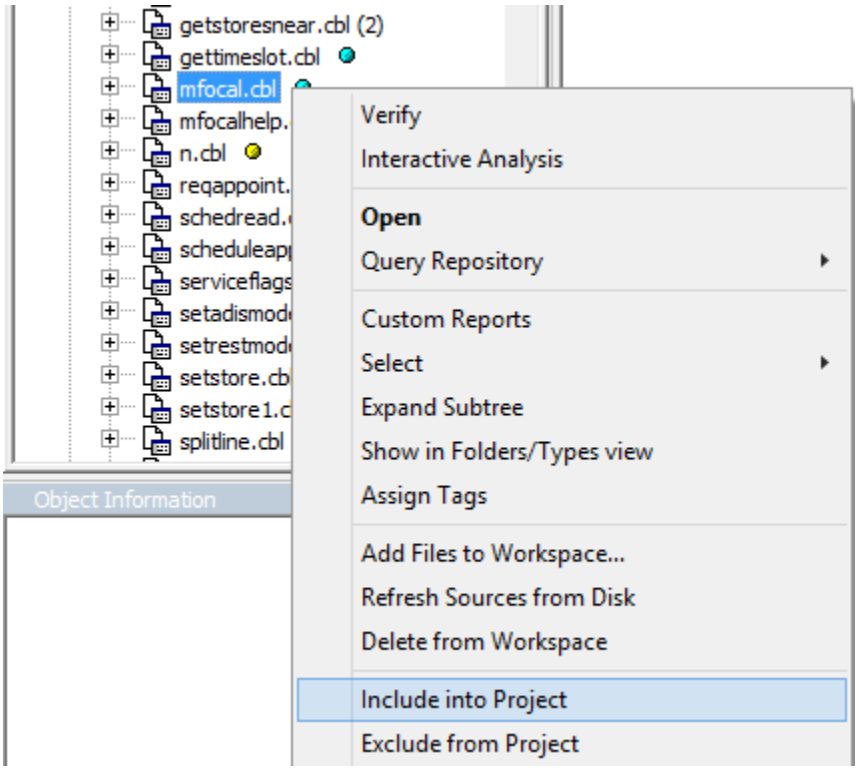
2. In the **Enter New Project Name** text field, type **MainLogic**. The project name must be unique within the workspace, so, in a multiuser installation, it is preferable to include a user id or change request number as part of the Project name.
3. Press **OK**. You will see the new project appear in the Repository pane.

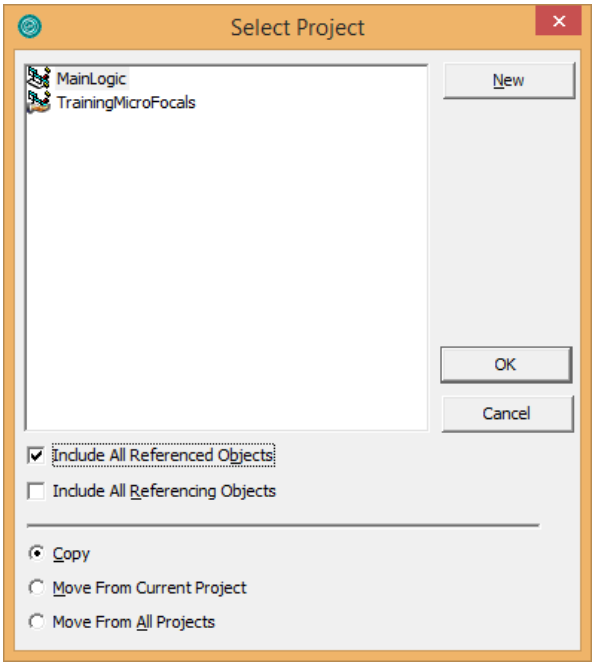


Exercise 5-2: Copying Sources between Projects

When you want to analyze subsystems separately, or partition a large system into smaller, more manageable units, it is helpful to copy sources between two projects in the same workspace.

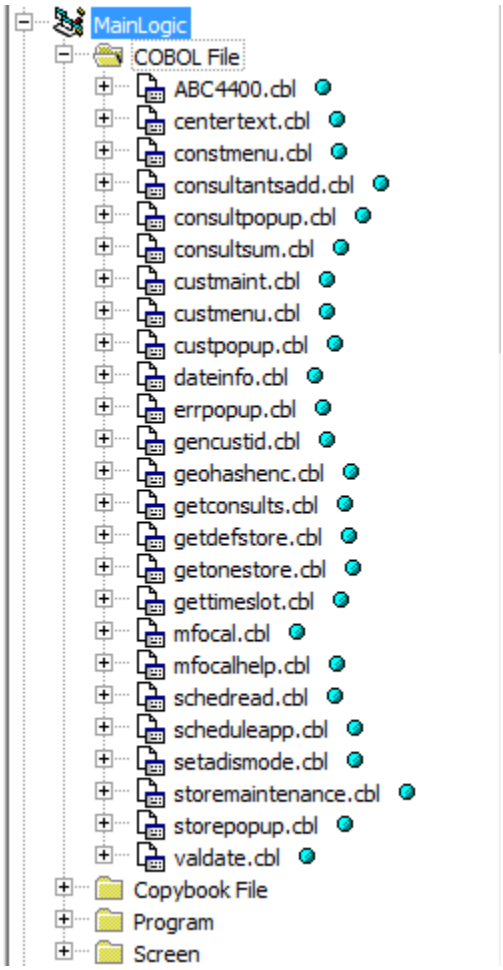
1. In the repository pane, right-click the COBOL file **mfocal.cbl** from the Training project and select **Include Into Project**.



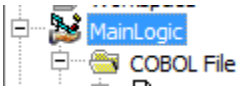


NOTE: Copy *does not create a separate instance of the object but rather a "link" to the object. Changes made to a file in one project will also be made in the other project.* Move *will remove the object from the 1st project.* Include All Referenced Objects *will also include all "child" objects related to the selected object.*

2. In the **Select Project** dialog box, check the **Include All Referenced Objects** checkbox and ensure that the **Copy** option is selected. This ensures that all legacy files associated with **mfocal.cbl** are imported along with the COBOL file itself. In this example, **Copybooks** and referenced programs have also been copied over into the new project.



3. Click **Project > Toggle Sharing**.



4. Press **F5** to refresh the Repository Browser. This option enables newly created Project to be viewable by other users accessing the Workspace. You will see a hand icon to denote that the project is now shared.
5. To disable project sharing, select **Project > Toggle Sharing**.

NOTE: If you select Toggle Protection, *it prevents anyone (including yourself) from changing the contents of a project (including or excluding objects). It also prevents you from accidently deleting your project (since you must first unprotect it).*

Exercise 5-4: Deleting a Project from a Workspace

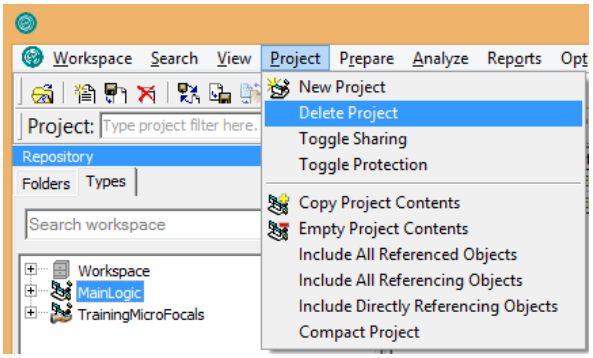
A project can be removed from the workspace while preserving the legacy objects in the repository. If two projects contain the same legacy source file, and one of the projects is deleted, the source file is not removed from the workspace but rather "unlinked" from the deleted project. It may be desirable to remove a project to reduce clutter in the workspace.

Objective: You will remove a project from a workspace

1. Perform the previous exercise but give a different name to the new project.
2. Select the new project.

NOTE: The Delete Project operation will be applied to the active project, so make sure the correct project has been selected.

3. On the menu bar, click **Project > Delete Project**.



4. When you are prompted to delete the current project, press **Yes**. The objects in the deleted project will not be deleted from the workspace. They will be "unlinked." The objects will remain in the workspace and in any other projects in which they are located.

Summary

In this module you learned:

- How to create new project in a workspace

- How to copy sources between projects
- How to filter projects using the repository browser
- How to delete a project from the workspace
- How to keep your analysis data up to date with your working copy of the code

Module 6: Working with Reports

Introduction

COBOL Analyzer includes tools for creating a variety of reports with useful information about your projects and workspaces.

Objectives

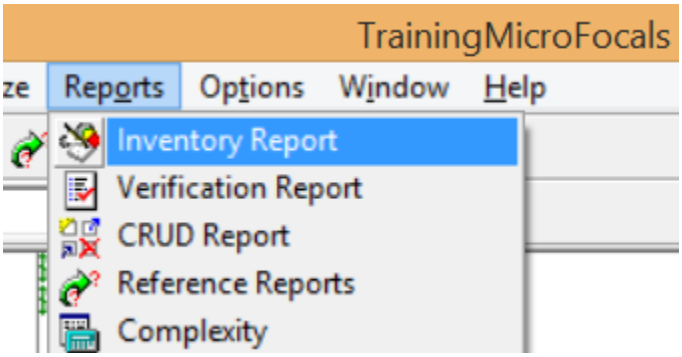
- You will learn how to create an inventory report
- You will learn how to create various kinds of reference reports
- You will learn how to create a portability assessment

Exercise 6-1: Create an Inventory Report

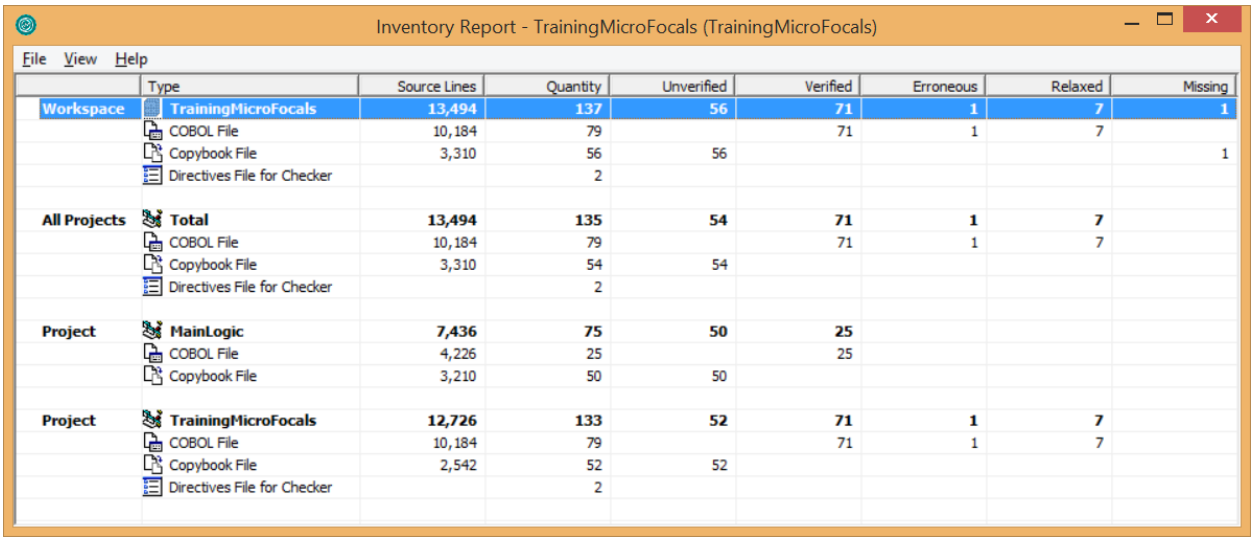
The Inventory Report contains high-level statistics on the contents of the current work-space. This includes the Workspace folder, a sum of all projects and a section for each project defined to the workspace.

Objective: You will create a new inventory report

11. On the main menu bar, click **Reports > Inventory Report**.



You will see a screen similar to the following that lists a project for each user.



	Type	Source Lines	Quantity	Unverified	Verified	Erroneous	Relaxed	Missing
Workspace	TrainingMicroFocals	13,494	137	56	71	1	7	1
	COBOL File	10,184	79		71	1	7	
	Copybook File	3,310	56	56				1
	Directives File for Checker		2					
All Projects	Total	13,494	135	54	71	1	7	
	COBOL File	10,184	79		71	1	7	
	Copybook File	3,310	54	54				
	Directives File for Checker		2					
Project	MainLogic	7,436	75	50	25			
	COBOL File	4,226	25		25			
	Copybook File	3,210	50	50				
Project	TrainingMicroFocals	12,726	133	52	71	1	7	
	COBOL File	10,184	79		71	1	7	
	Copybook File	2,542	52	52				
	Directives File for Checker		2					

NOTE: Source lines, number of objects and the verification status and count is stored for each 'source' folder in each project, 'All' projects and for the workspace.

12. When you are finished, close the Inventory Report by clicking the **X** in the upper right-hand corner.

Reference Reports

In any analysis or modernization project, it is important to ensure that no critical source components are missing or unused. The cost/risk associated with a missing source component can be very high. This is a powerful way to validate the completeness of your production code, confirm that all code is the most recent, and increase the efficiency of your production code by removing any unused components.

There are four types of reference reports:

- **Unreferred Report** – Identifies unreferenced (unused) entities. This may or may not indicate that an object is obsolete, based on what we have loaded in the workspace.
- **Unresolved Report** – Identifies missing entities. Typically, at the onset of a deployment of CA, we find that we are missing copybooks.
- **Cross-Reference Report** – Identifies all object relationships the system.
- **External Reference Report** - Identifies references in object-oriented applications to external files that are not registered in the workspace, such as .java, Java Archive (JAR), or C++ include files (assuming you have identified the locations of these files in the Workspace Verification options window for the source files). These references are not reported as unresolved in the Unresolved Report.

All reports have the same options and functions. For the sake of brevity, we will only perform the Unreferred Report exercise. However, the following steps apply in the same manner to the Unresolved, Cross-Reference and External References Reports.

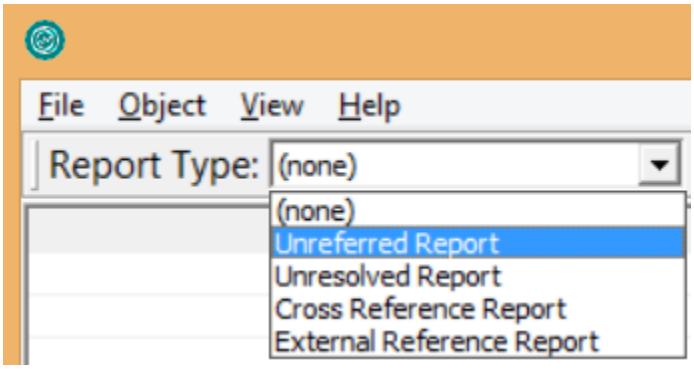
Exercise 6-2: Create an Unreferred Report

This report shows source components that are registered in the repository but that have no references elsewhere. For example, copybooks are not used in programs, program entry points are not called or executed in JCL or invoked by a transaction.

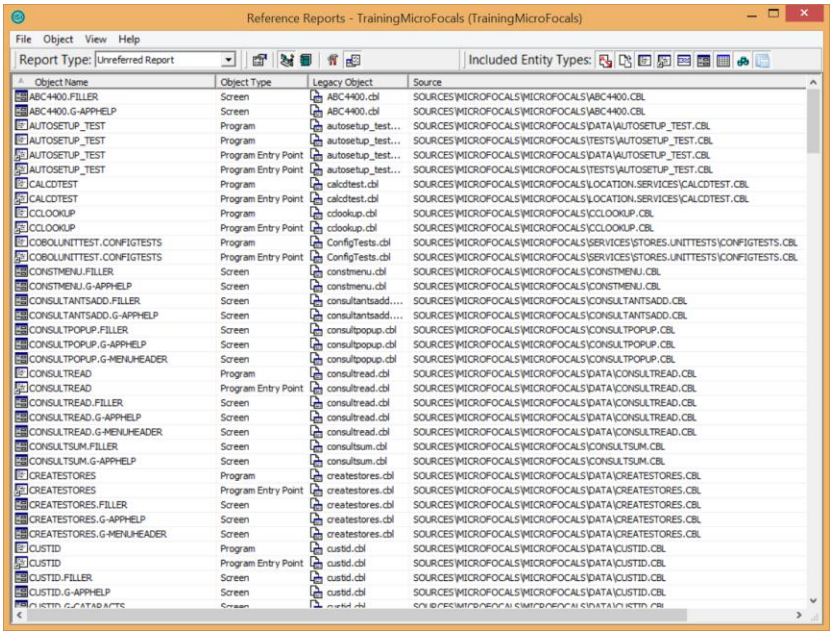
In general, all unreferred components should be investigated to ensure they have not been included by mistake, or that the component using them (e.g., COBOL program) is not missing.
Various preprocessing methods should be also taken into account when analyzing this report.

Objective: You will create a new unreferred report

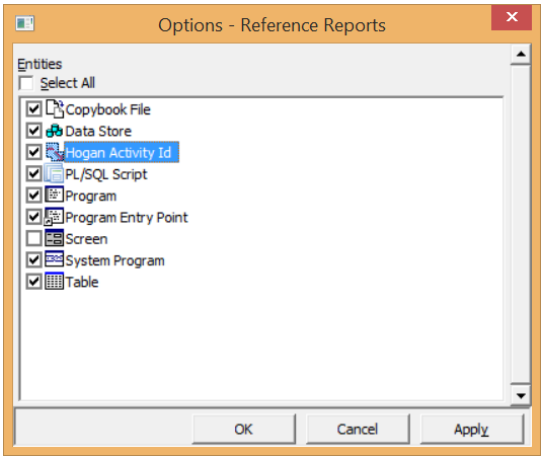
1. click **Reports > Reference Reports**.



2. From the **Report Type** drop-down box, select **Unreferred Report**.
3. Close the **Preview** pane by clicking the **X** in the upper right-hand corner. Make sure you click the **X** in the **Preview** pane and NOT on the **Reference Reports** window. You will see a reference report similar to the following:



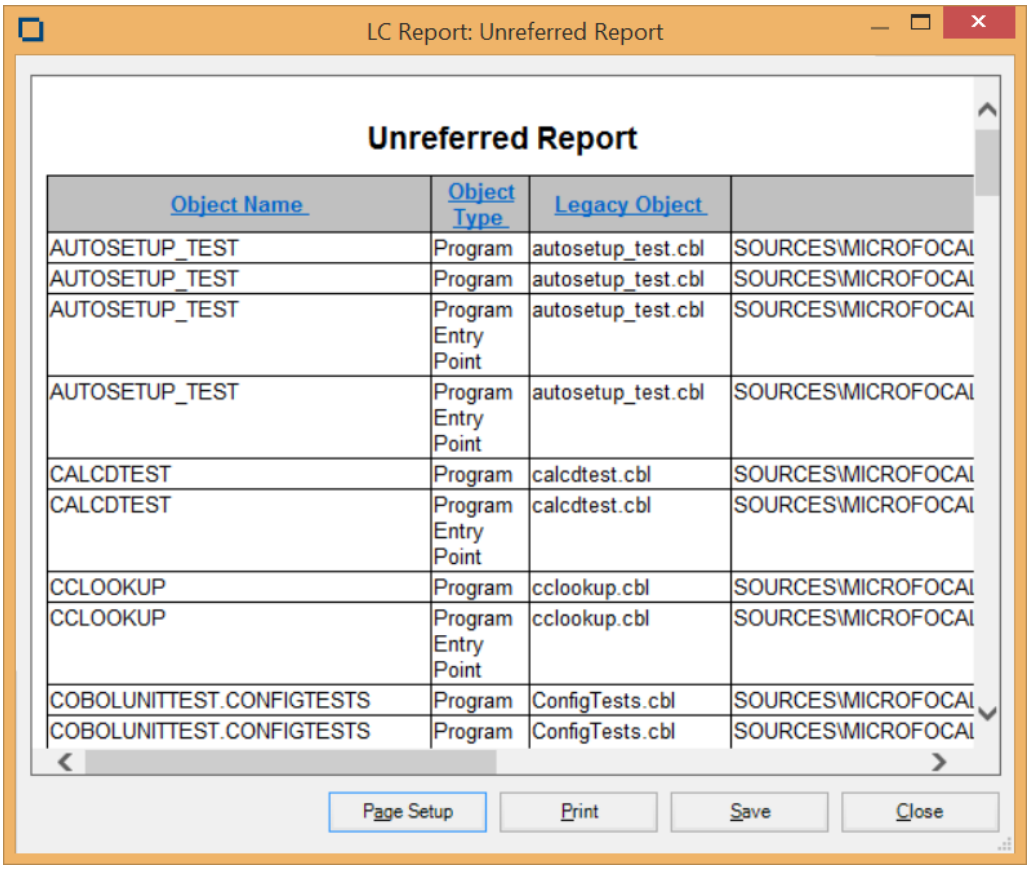
4. To sort the **Source** column, click on the **Source** column header.
5. From the menu bar, click **View > Options**. You will see the **Options – Reference Reports** dialog box. From here you can limit the file entities shown in the report, such as Copybook files and screens.



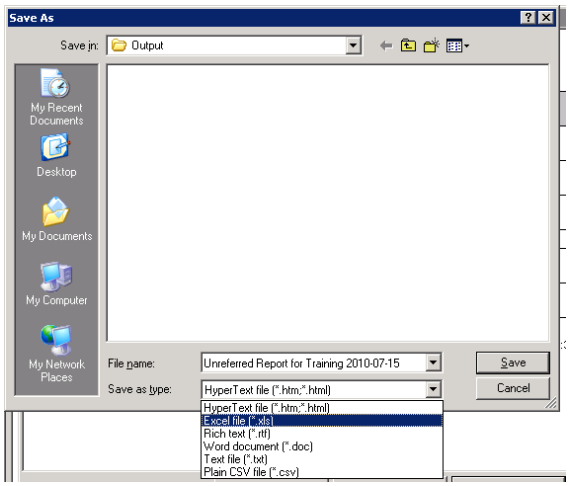
6. In the **Options – Reference Reports** dialog box, uncheck the **Screen** entity and press **OK**. In the reference report, you will see that **all the screens** are no longer displayed.
7. Click the **Copybook File** icon on the **INCLUDED ENTITY TYPES** toolbar next to the drop-down box. The **screens** should appear in the report again.

8. To create a reference report suitable for delivery to a customer, from the menu bar click **File > Report…** You can also click the **Report** icon shown below.

 You will see a screen similar to the following:



9. Either **Save** or **Print** the report using the appropriate buttons at the bottom of the dialog box. The default location for any report or diagram that you save is the Output folder under the Workspace folder on the server. You can override the Save in: to a network location or a local drive on your PC.



You can modify the report name to add your id and / or date so that the same report run multiple times will not be overplayed. You can change the report type to any valid extension provided you have the supporting software to open the file. The most useful format for most reports is Excel.

10. When you are finished, close the **Unreferred** and **Reference** report windows.

Determining Complexity

An object's *complexity* is based on weighted values calculated for selected file attributes. COBOL Analyzer produces a complexity report to help managers and users determine the depth and complexity of their legacy system. The Complexity Report displays metrics specific to the entity type selected. Better understanding of a legacy system's complexity will aid in predicting the effort and resources required to maintain, enhance, or transform the system.

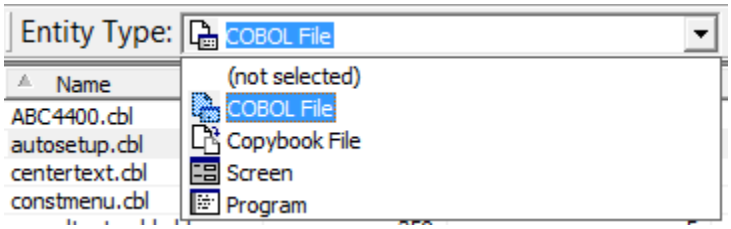
COBOL Analyzer provides a combination of Micro Focus generated and Industry standard complexity statistics. For COBOL "programs", the industry standard metrics used are based on Halstead and McCabe. The Halstead measure (Program Volume) is based on operators and operands while McCabe or Cyclomatic Complexity is based on the number of logical paths. Descriptions of these measures can be found in the Online Help.

Complexity Metrics show raw complexity values for source (i.e. COBOL) and generated (i.e. Program) files in your project. Initially the Complexity Metrics window is empty. You must select the entity type for which you want statistics.

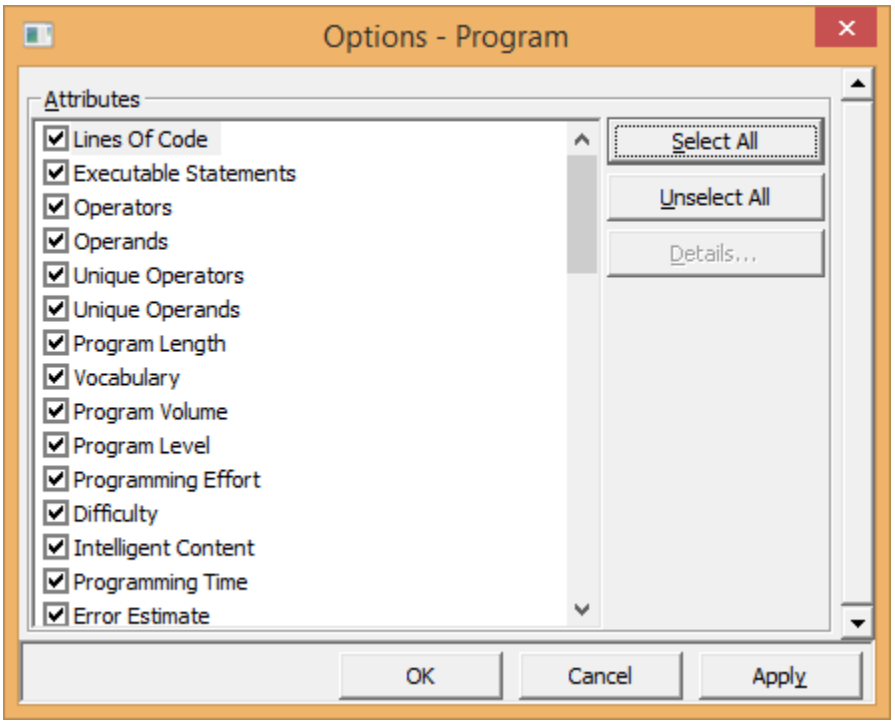
Exercise 6-3: Determining Complexity

Objective: You will create a complexity metrics report

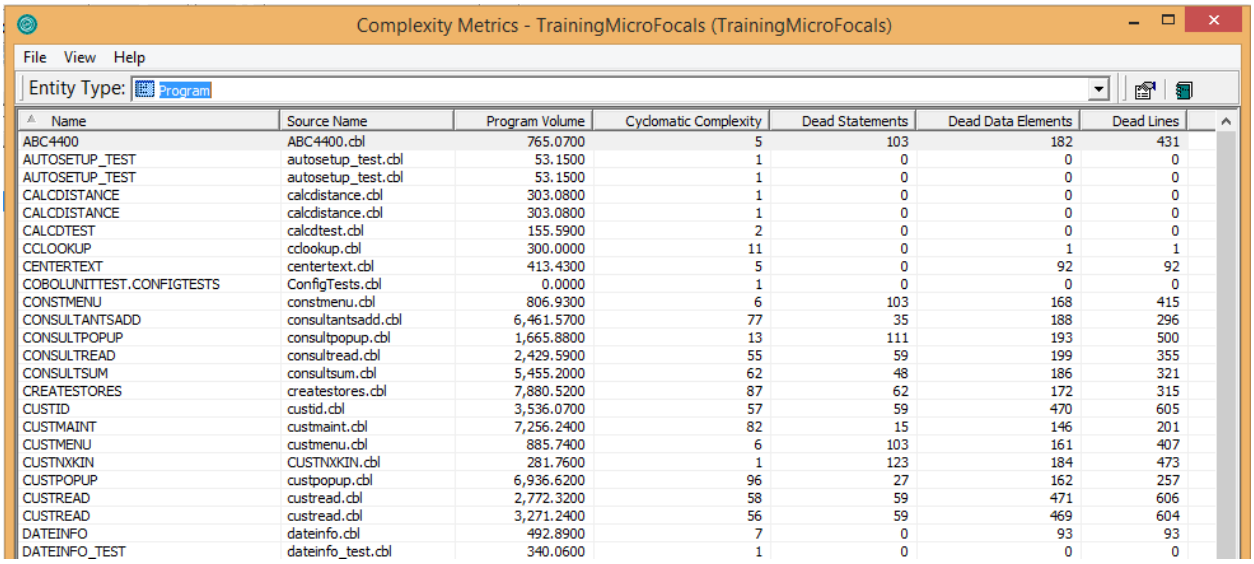
1. On the menu bar, click **Reports > Complexity.**
2. In the **Entity Type** drop-down box, select **COBOL File**.



3. Repeat Step 2 and select **Copybook File.** You will see the complexity statistics for your Copybooks.
4. Repeat Step 2 and select **Program**. You will see the complexity statistics for **ALL** of the Programs in the project are displayed. Programs are the parsed version of COBOL, Java sources etc. Programs have many metrics available.
5. On the menu bar, click **View > Columns…** You will see the **Options – Program** dialog box. The available metrics for the chosen entity are displayed, and you can select the metrics that you want included in the report and on the screen.

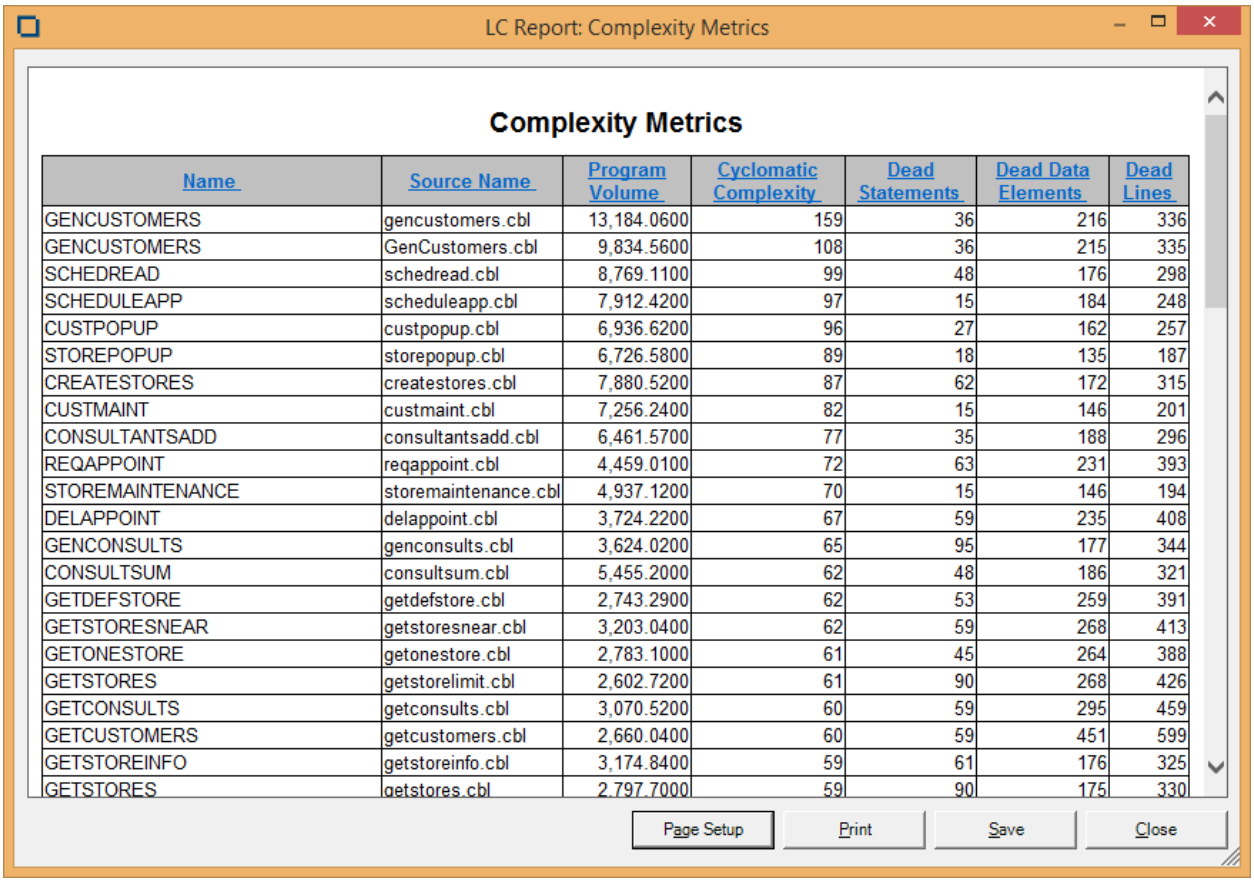


6. Select **Program Volume**, **Cyclomatic Complexity**, and any others of your choice, then click **OK.** You will see the selected statistics for the programs displayed.



Name	Source Name	Program Volume	Cyclomatic Complexity	Dead Statements	Dead Data Elements	Dead Lines
ABC4400	ABC4400.cbl	765.0700	5	103	182	431
AUTOSETUP_TEST	autosetup_test.cbl	53.1500	1	0	0	0
AUTOSETUP_TEST	autosetup_test.cbl	53.1500	1	0	0	0
CALCDISTANCE	calcdistance.cbl	303.0800	1	0	0	0
CALCDISTANCE	calcdistance.cbl	303.0800	1	0	0	0
CALCDTEST	calcdtest.cbl	155.5900	2	0	0	0
CCLOOKUP	cclookup.cbl	300.0000	11	0	1	1
CENTERTEXT	centertext.cbl	413.4300	5	0	92	92
COBOLUNITTEST.CONFIGTESTS	ConfigTests.cbl	0.0000	1	0	0	0
CONSTMENU	constmenu.cbl	806.9300	6	103	168	415
CONSULTANTSADD	consultantsadd.cbl	6,461.5700	77	35	188	296
CONSULTPOPUP	consultpopup.cbl	1,665.8800	13	111	193	500
CONSULTREAD	consultread.cbl	2,429.5900	55	59	199	355
CONSULTSUM	consultsum.cbl	5,455.2000	62	48	186	321
CREATESTORES	createstores.cbl	7,880.5200	87	62	172	315
CUSTID	custid.cbl	3,536.0700	57	59	470	605
CUSTMAINT	custmaint.cbl	7,256.2400	82	15	146	201
CUSTMENU	custmenu.cbl	885.7400	6	103	161	407
CUSTNXKIN	CUSTNXKIN.cbl	281.7600	1	123	184	473
CUSTPOPUP	custpopup.cbl	6,936.6200	96	27	162	257
CUSTREAD	custread.cbl	2,772.3200	58	59	471	606
CUSTREAD	custread.cbl	3,271.2400	56	59	469	604
DATEINFO	dateinfo.cbl	492.8900	7	0	93	93
DATEINFO_TEST	dateinfo_test.cbl	340.0600	1	0	0	0

7. In the Complexity Metrics pane, click on the **Cyclomatic Complexity** column header **twice** to sort the report by descending order of complexity.
8. Create a Complexity Report:
    - From the main menu click **File > Report**.



Name	Source Name	Program Volume	Cyclomatic Complexity	Dead Statements	Dead Data Elements	Dead Lines
GENCUSTOMERS	gencustomers.cbl	13,184.0600	159	36	216	336
GENCUSTOMERS	GenCustomers.cbl	9,834.5600	108	36	215	335
SCHEDREAD	schedread.cbl	8,769.1100	99	48	176	298
SCHEDULEAPP	scheduleapp.cbl	7,912.4200	97	15	184	248
CUSTPOPUP	custpopup.cbl	6,936.6200	96	27	162	257
STOREPOPUP	storepopup.cbl	6,726.5800	89	18	135	187
CREATESTORES	createstores.cbl	7,880.5200	87	62	172	315
CUSTMAINT	custmaint.cbl	7,256.2400	82	15	146	201
CONSULTANTSADD	consultantsadd.cbl	6,461.5700	77	35	188	296
REQAPPOINT	reqappoint.cbl	4,459.0100	72	63	231	393
STOREMAINTENANCE	storemaintenance.cbl	4,937.1200	70	15	146	194
DELAPPOINT	delappoint.cbl	3,724.2200	67	59	235	408
GENCONSULTS	genconsults.cbl	3,624.0200	65	95	177	344
CONSULTSUM	consultsum.cbl	5,455.2000	62	48	186	321
GETDEFSTORE	getdefstore.cbl	2,743.2900	62	53	259	391
GETSTORESNEAR	getstoresnear.cbl	3,203.0400	62	59	268	413
GETONESTORE	getonestore.cbl	2,783.1000	61	45	264	388
GETSTORES	getstorelimit.cbl	2,602.7200	61	90	268	426
GETCONSULTS	getconsults.cbl	3,070.5200	60	59	295	459
GETCUSTOMERS	getcustomers.cbl	2,660.0400	60	59	451	599
GETSTOREINFO	getstoreinfo.cbl	3,174.8400	59	61	176	325
GETSTORES	getstores.cbl	2,797.7000	59	90	175	330

- Click the **Save** button to save the report. Save in any format you wish.

9. Close the Report window by clicking the **Close** button**.**
10. Close the Complexity Tool by clicking the **X** in the upper right-hand corner.

Summary

In this module you learned:

- How to create an inventory report
- How to create an unreferred reference report
- How to create a complexity report

Next steps

You have concluded the getting started part and should have the basic understanding of COBOL Analyzer concepts. You can now start using the various analysis features using the getting started pane (View→Getting Started) or the different menus.

We encourage users to continue the tutorials found in the Micro Focus community site to explore the different analysis capabilities of COBOL Analyzer.