X-Analysis  
Tutorial Professional

Software version 13.3.01, 20230403

Contents

[Contents 2](#_Toc130913461)

[Publication Information 4](#_Toc130913462)

[Preface 5](#_Toc130913463)

[About this guide 5](#_Toc130913464)

[Version 5](#_Toc130913465)

[How to use this guide 5](#_Toc130913466)

[What is new in the latest release? 6](#_Toc130913467)

[Overview 7](#_Toc130913468)

[Benefits of X-Analysis 7](#_Toc130913469)

[Prerequisites 8](#_Toc130913470)

[Acronyms used in the manual 8](#_Toc130913471)

[Using X-Analysis Client 9](#_Toc130913472)

[Sign-On Dialog 10](#_Toc130913473)

[Session Information 14](#_Toc130913474)

[X-Analysis Menu 15](#_Toc130913475)

[X-Analysis Preferences 16](#_Toc130913476)

[Impact Analysis & Graphical Documentation 17](#_Toc130913477)

[Chronology of Features 18](#_Toc130913478)

[Work With Objects 19](#_Toc130913479)

[Object Where Used 22](#_Toc130913480)

[Export a List to PDF, MS Word or MS Excel 27](#_Toc130913481)

[Data Flow Diagram 28](#_Toc130913482)

[Hide/Show Objects 30](#_Toc130913483)

[Expand/Collapse Referenced Objects 31](#_Toc130913484)

[View Annotation 33](#_Toc130913485)

[Attributes 34](#_Toc130913486)

[Detail DFD icon 36](#_Toc130913487)

[Legend icon 38](#_Toc130913488)

[DFD Restricted to an Application Area 40](#_Toc130913489)

[Variable Where Used 42](#_Toc130913490)

[Legend 44](#_Toc130913491)

[View Levels 45](#_Toc130913492)

[File/Field Where Used (FFWU) 52](#_Toc130913493)

[Structure Chart Diagram 53](#_Toc130913494)

[Legend 54](#_Toc130913495)

[Annotate 55](#_Toc130913496)

[Function Type 56](#_Toc130913497)

[Hierarchy Exclusions in SCD 57](#_Toc130913498)

[Repeated nodes in SCD 58](#_Toc130913499)

[Structure Chart Diagram with Files 59](#_Toc130913500)

[Hierarchical Structure Chart Diagram 60](#_Toc130913501)

[Narration 62](#_Toc130913502)

[Referred Files 64](#_Toc130913503)

[Show Root Node 66](#_Toc130913504)

[Legend icon 67](#_Toc130913505)

[Document Manager 69](#_Toc130913506)

[Source Browser 80](#_Toc130913507)

[Source Browser Toolbar 83](#_Toc130913508)

[Source Options 83](#_Toc130913509)

[Indented Source Code view 84](#_Toc130913510)

[Source Outline 84](#_Toc130913511)

[Pseudo Code 85](#_Toc130913512)

[Source Level 85](#_Toc130913513)

[Flowchart 86](#_Toc130913514)

[Other Options on the Source Browser 90](#_Toc130913515)

[Member X-Reference 90](#_Toc130913516)

[Jump to Subroutine 91](#_Toc130913517)

[Program Structure Chart 92](#_Toc130913518)

[Zoom Source to Files defined in File Specification 94](#_Toc130913519)

[Screen/Report Design 95](#_Toc130913520)

[Zoom in SEU/Lpex Editor 96](#_Toc130913521)

[SEU 97](#_Toc130913522)

[Relational Data Model Extraction for IBM i 99](#_Toc130913523)

[Generating Data Model Diagram 100](#_Toc130913524)

[LFs/Access Paths 106](#_Toc130913525)

[File Field Details/FFD 108](#_Toc130913526)

[Access Path Diagram/APD 110](#_Toc130913527)

[View Data 112](#_Toc130913528)

[Record Display 113](#_Toc130913529)

[Data Model Diagram - Application Area Specific 115](#_Toc130913530)

[What’s Next 118](#_Toc130913531)

[Appendix A – Enabling the SEU Interface 119](#_Toc130913532)

[Create User Profile XAN4SEU 119](#_Toc130913533)

[Creating Macro for User Profile XAN4SEU 123](#_Toc130913534)

[Index 126](#_Toc130913535)

Publication Information

© 2023 Fresche Solutions Inc.

**Published by:**

Fresche Solutions Inc.  
995 Wellington Suite 200  
Montreal, QC  
Canada, H3C 1V3

Telephone numbers:

(514) 747.7007  
(toll-free in US and Canada): 1.800.361.6782  
(toll-free in Belgium, France, Germany, UK): 00 800 361 67 82 0  
(toll-free in Australia): 0011 800 361 6782 0

E-mail for support: [support@freschesolutions.com](mailto:support@freschesolutions.com?subject=Support)

E-mail for inquiries: [info@freschesolutions.com](mailto:inquiry@freschesolutions.com?subject=inquiry)

Web: [www.freschesolutions.com](http://www.freschesolutions.com/)

Title: **X-Analysis Tutorial Professional**, software version 13.3.01

Publication Date: April 2023

**Trademarks**

X-Analysis and X-Analysis Professional are trademarks or registered trademarks of Fresche Solutions Inc. <iSeries, Power Systems, Power8> are registered trademarks of IBM Corporation. Microsoft and Microsoft Windows are registered trademarks of Microsoft Corporation. All other brand and product names are trademarks or registered trademarks of their respective companies.

No part of this document may be reproduced or transmitted in any form or by any means, without prior permission in writing from Fresche Solutions.

The information in this manual is believed to be correct at the time of publication. However, Fresche Solutions Inc. makes no warranty, express or implied, about the accuracy of this information and reserves the right to revise this document or make changes to the products described herein at any time without notice and without obligation. Fresche Solutions Inc. is not liable for any loss of data, damage to databases or other software, or any other losses arising from the use of this manual.

Preface

About this guide

This guide, X-Analysis Tutorial Professional is a handy, step-by-step, illustrative document that has been specially designed to highlight the practical features of X-Analysis. It discusses the following topics:

* Impact Analysis
* Graphical Documentation
* Data Model Extraction

Version

This is version 13.3.01 of the X-Analysis Tutorial Professional.

How to use this guide

Each chapter in this guide focuses on the advanced analytical aspects of X-Analysis. The guide begins with a few words on the overall benefits of X-Analysis and mentions certain software pre-requisites to be met. Thereafter, the guide discusses the main topics comprising Impact Analysis, Graphical Documentation, and Data Model Extraction for IBM i.

The X-Analysis suite of products contains a total of eight modules. For information about the other modules, please contact your Fresche Solutions representative, or visit us at: <https://freschesolutions.com/products/x-analysis-suite/>

|  |  |
| --- | --- |
|  | X-Analysis client release 10.x onwards is digitally signed. |

|  |  |
| --- | --- |
|  | Windows 8 (or above) users – In case any problem occurs in running X-Analysis:  - Right-click on the ‘Clean start Eclipse\*’ or the ‘X-Analysis for Eclipse’ shortcuts.  - Then, select ‘Run as administrator’. |

|  |  |
| --- | --- |
|  | \*The requirement to take the "Clean start Eclipse" option during installation is no longer necessary with V12.1.0 and onwards. For details, refer to the document “X-Analysis\_Installation\_And\_Upgrade\_Guide” for this release. |

What is new in the latest release?

|  |  |
| --- | --- |
|  | For this release, we updated this document. Here is the change that were made to this document. |

* [Object Where Used](#_Ref-2001550459): Update the definition for the option OWU>Update References.
* [X-Analysis Menu](#_Ref-79986385): Replaced the screen X-Analysis Menu.

Overview

X-Analysis is an advanced and easy-to-use analysis tool. The tool’s instant and automated features like visual impact analysis, cross-referencing, and graphical documentation of RPG and COBOL applications endow this software with a remarkable edge. This section briefly explains the benefits of X-Analysis and the hardware prerequisites needed for successful installation of the software.

Benefits of X-Analysis

The X-Analysis product suite is an essential tool for IBM i application documentation, design recovery, and re-engineering. The X-Analysis software works as a plugin under IBM's RDi and Eclipse, enabling the user to browse and visualize applications from a high-level model down to individual lines of code. This globally renowned tool is packed with features like RPG & COBOL as pseudo code, variable level impact analysis, graphical flow/structure charts, graphical data flowcharts, PDF/MS Word, and MS Visio documentation wizards. X-Analysis allows technical analysts to navigate graphically through an IBM i application from a Windows PC. Working in a true client-server mode, X-Analysis automatically cross-references databases and ensures that the display is always up-to-date and a true representation of the application on the IBM i.

Prerequisites

Before starting this tutorial, the following are required:

* The X-Analysis server and the Client components should be installed to an IBM i server and a Windows PC, respectively.
* The demo library – XAN4CDEM, and its cross-reference library – XAN4CDXA, should be successfully restored on the IBM i.

Acronyms used in the manual

|  |  |
| --- | --- |
| Acronym | Full form |
| APD | Access Path Diagram |
| AAD | Application Area Diagram |
| DFD | Data Flow Diagram |
| DMD | Data Model Diagram |
| FFD | File Field Details |
| SCD | Structure Chart Diagram |
| HSC | Hierarchical Structure Chart |
| PSC | Program Structure Chart |
| OSC | Overview Structure Chart |
| VWU | Variable Where Used |
| FFWU | File/Field Where Used |
| UML | Unified Modeling Language |
| SSL | Secure Sockets Layer |

Using X-Analysis Client

The following topics are important to start using X-Analysis:

* Sign-on dialog
* Session Information
* X-Analysis Menu
* X-Analysis Preferences

Sign-On Dialog

Start IBM's Rational products 9.5 and above or Eclipse (see <https://www.myfreschesolutions.com/nexus/ui/docs?root=0&docId=590&kbnum=15954> for supported Eclipse version) .

**Window > Open Perspective > Other > X-Analysis**

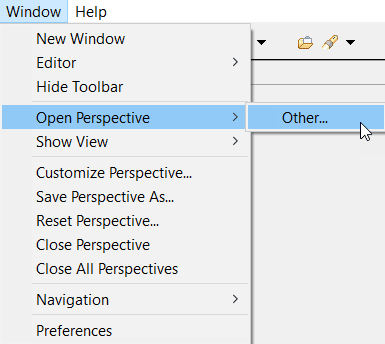


Fig. 1.1.1 – Open Perspective > Other

Select the X-Analysis Perspective.

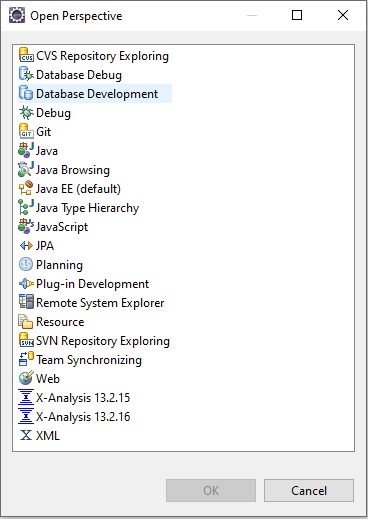


Fig. 1.1.2 Open Perspective

Click OK to start the X-Analysis Perspective.

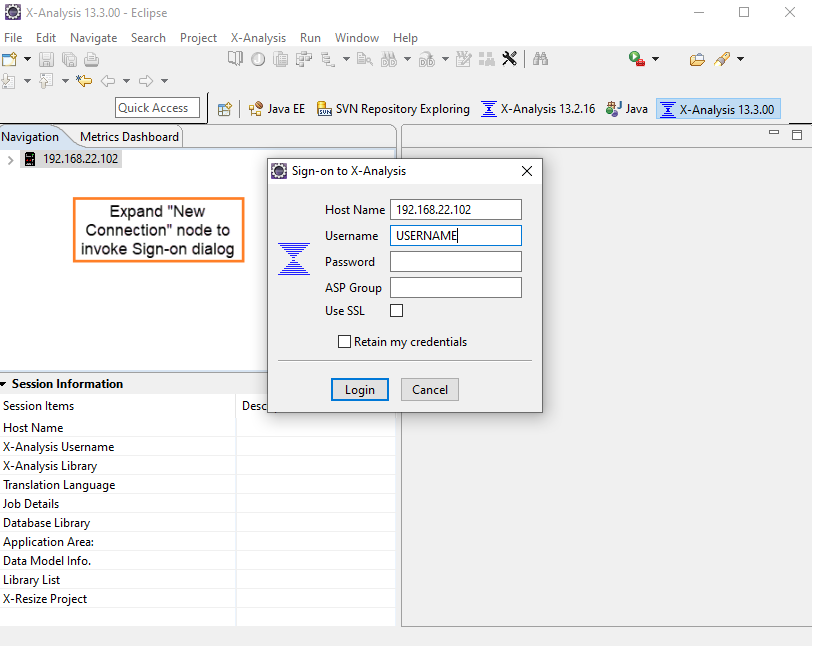


Fig. 1.1.3 – X-Analysis Perspective

Expand the New Connection node to invoke the Sign-on dialog box.

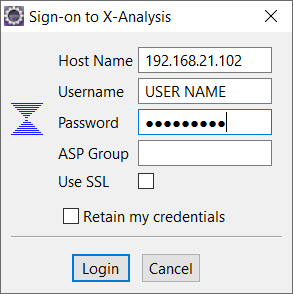


Fig. 1.1.4 X-Analysis Sign-on dialog box

Enter the following information to the Sign-on dialog box:

1. Enter the TCP/IP address/Computer Name of the IBM i to be accessed.
2. Enter the username and the password of a valid IBM i profile.
3. Specify the ASP Group If X-Analysis is installed on an ASP group.
4. Check the box for ‘Use SSL’ feature for additional security. Refer to the section Appendix E - Use SSL feature in the X-Analysis User Manual.
5. Click Login.

After successful sign-on, X-Analysis lists the application libraries which were initialized from the X4WRKAPP screen on the IBM i.

Session Information

The X-Analysis Client displays detailed session information about the connection in the associated Session Information view.

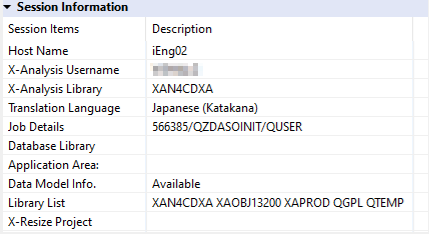


Fig. 1.2.1 Session Information

The Session Information details are as follows:

* Host Name: Displays the IP or the web link of the connected IBM i.
* X-Analysis Username: Displays the user name connected to the IBM i.
* X-Analysis Library: Name of the cross-reference library, the user has currently selected.
* Translation Language: Displays the information about the Database Language Translation, and if a value is set in Preferences> Advanced> Database Language Translation to No Translation Required, a blank entry will be visible.
* Job Details: Displays job details in format – Job Number/Job Name/Job User.
* Database Library: The Subset Library which is being used to get the data from where the Data View options are selected. This displays a value only when a subset library has been selected.
* Application Area: Displays the currently selected Application Area.
* Data Model Info: Informs the user whether Data Modeling is available or not.
* Library List: Displays the library list for the current job.
* X-Resize Project: Displays the name of the currently selected X-Resize project.

X-Analysis Menu

X-Analysis provides the X-Analysis toolbar menu on the Eclipse toolbar. Below screen displays the options available on the X-Analysis menu.

|  |
| --- |
| Fig. 1.3.1 X-Analysis Menu |

X-Analysis Preferences

The X-Analysis Preferences provides the facility to modify product preferences.

The X-Analysis Plugin comes with default preferences settings. You can change default preferences settings as required. To change preferences settings, open IBM's Rational product 9.5 and above or Eclipse 4.4 and above.

From the menu bar, select Window > Preferences to invoke the Preferences dialog.

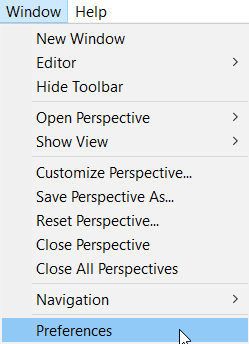


Fig. 1.4.1 Preferences option

|  |  |
| --- | --- |
|  | For more details, refer to the section “X-Analysis Preferences” in the document “X-Analysis\_User\_ Manual”. |

Impact Analysis & Graphical Documentation

Continual technical enhancements, syntax, variations in programming styles, financial and time constraints, absence of original application designers, and/or monolithic application codes are some reasons that necessitate changes in the existing application designs. System analysts or software professionals must be equipped with the technological know-how for redesigning business applications.

The X-Analysis toolset meets the requirements mentioned above. It is patterned to convert all IBM i applications into the latest format. This software suite has precise tools for conducting advanced application and documentation tasks.

This chapter covers Impact Analysis and Graphical Documentation. The features and options discussed here provide prompt and automated documenting and computing solutions.

Chronology of Features

The Analysis & Documentation solution set is a sophisticated tool for instant and automated cross-referencing, impact analysis and graphical documentation of Synon/2E, RPG, and COBOL applications. All the functionality above (and more) is contained in the following separate features, explained in this chapter:

|  |  |
| --- | --- |
| **Feature** | **Brief Description** |
| Work with Objects | A list of all the objects in a chosen application or application area. |
| Object Where Used | A list of all usage references for an object. |
| Export List to PDF/MS Word or MS Excel | The option to export data to MS Excel or a PDF/MS Word. |
| Data Flow Diagram | An interactive, color-coded bus routing block diagram for displaying an object relationship within an application. |
| Variable Where Used | A list of all instances of the specified variable in the application. |
| Structure Chart Diagram | A graphical display that shows program-to-program relationships. |
| Detailed Structure Chart Diagram | A graphic display that shows program-to-program relationships with files used by each program. It also explains each program’s generated program text. |
| Hierarchical Structure Chart Diagram | Color-coded bus routing block diagram, which enhances the visual of program-to-program interrelationships. Through this, the user gets an exact idea of the hierarchical set-up of the programs. |
| Document Manager | A utility that automatically generates structured system document in PDF / MS Word format. |
| Source Browser | The interface used for system analysis, reviewing, re-factoring, maintaining, and redeveloping in source code. |
| Source Browser Toolbar | The toolbar contains a list of different options for viewing code within the Source Browser, such as Pseudo Code and Source View Level options, etc. |

Work With Objects

Work with Objects displays a list of all the objects for a chosen application or application area.

To generate the Object List for all files, double-click the Files node available under the application library.

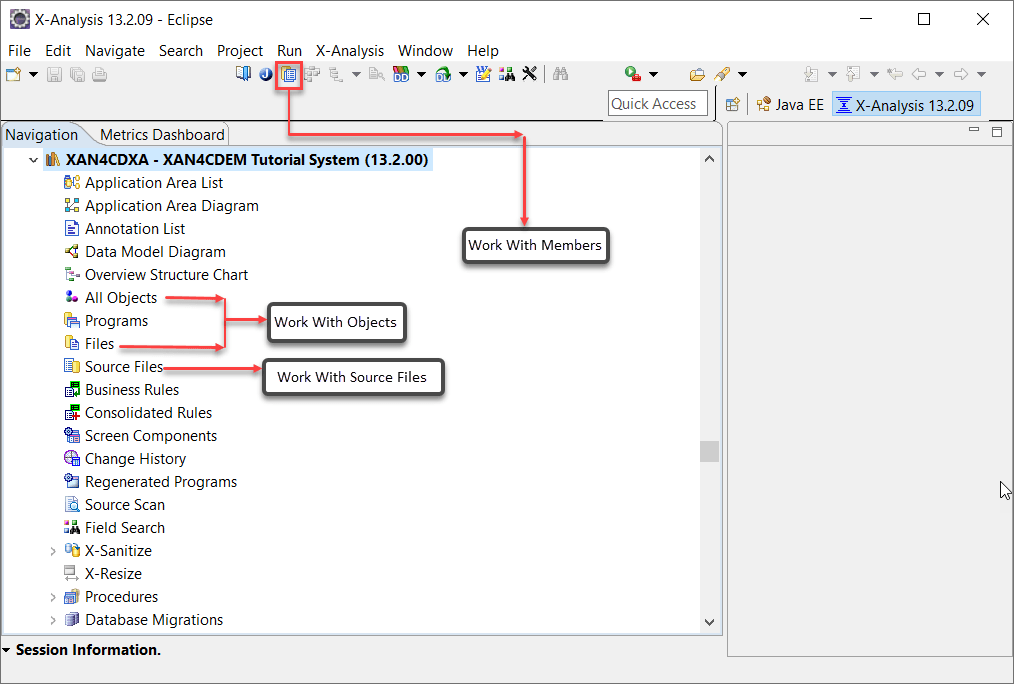


Fig. 2.1.1 – X-Analysis displaying the Files node

The double-click action invokes the Work with Objects dialog box with ‘Type’ set as \*FILE and ‘Attribute’ as PF. The dialog is displayed on the following page:

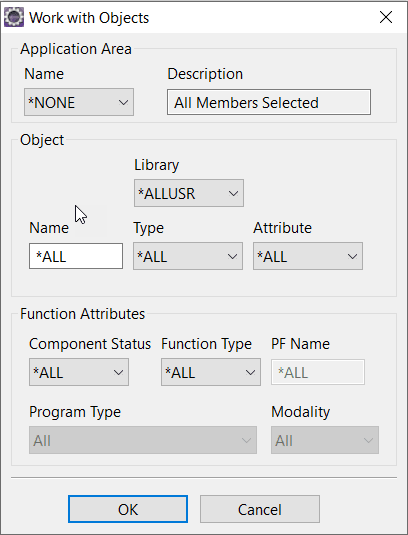


Fig. 2.1.2 – Work with Objects dialog box – Files

Click OK to invoke the Object List window showing all the files.

|  |  |
| --- | --- |
|  | Object Name - An individual name/a generic name/ ‘\*ALL’. Both individual and generic names can have wildcards specified. Enter & (Ampersand) to denote that any one character and \* (Asterisk) denotes any number of characters. Accordingly, C&&H\* means that all the object names where the first character is C and fourth character is H, will be selected. More examples: AA&&2, OE\*. |

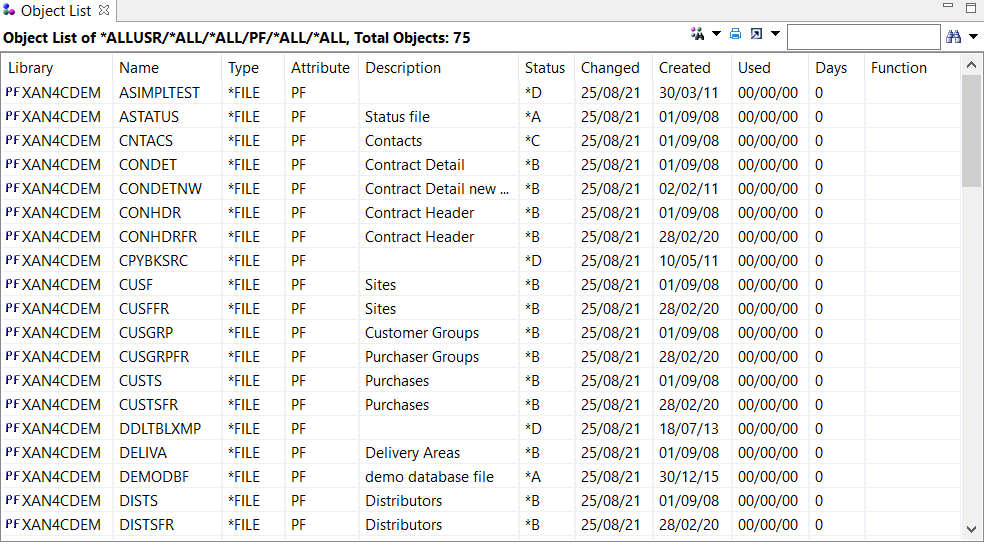


Fig. 2.1.3 – Object List window displaying all Files (PF)

The Object List window displays multiple columns. In this window, the user can sort a column’s data by clicking a column heading.

When the user double-clicks the All Objects/Files/Programs node of an application area, the Object List displays the column called ‘Rule’ which indicates the Application Area Rules that have been applied.

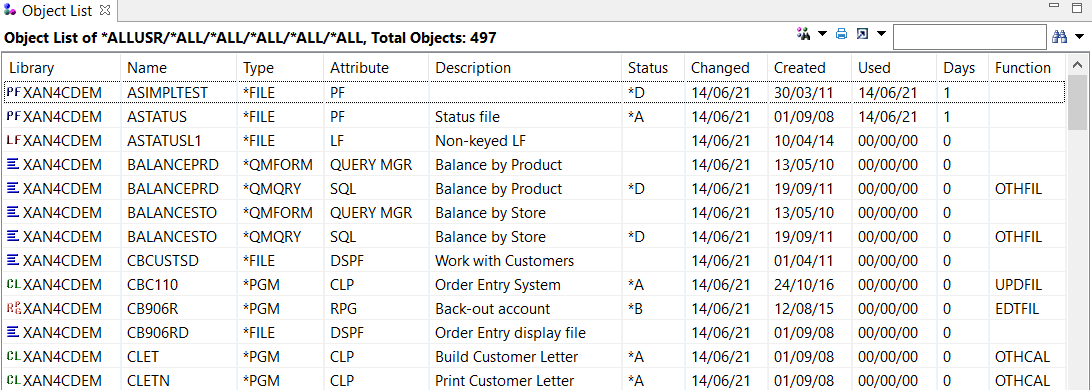


Fig. 2.1.4 – All Objects list on an application area showing the Rule column

Object Where Used

The Object Where Used option lists all the instances in the application where a specified object (\*FILE type) has been used/referenced. The following are the details of the sub-options of Object Where Used.

|  |
| --- |
| * Usage References: List of objects which are using this object. * All References: List of objects where this object is being used. Besides this, source references for which the objects do not exist are also displayed. * Update References: This option will be taken on a File which will list all programs, that use the file in an ‘Update’ mode. It does not include any file used in the ‘Output’ or ‘Input/Output’ modes. * Delete References: List of objects that are deleting the records from the specified object. * Entry Level References: Entry-level references in 'Object Where Used' list all entry-level programs (\*A), which directly or indirectly access the object in question. * Real Time References: It is the most flexible option because it facilitates live impact analysis and displays the Real-Time references or the object dependencies at runtime. It does not use any data existing in the X-Analysis tables populated during the last initialization/refresh of the X-Ref library. |

Select CUSF from the Member/Object List and choose the Object Where Used option from the context menu.

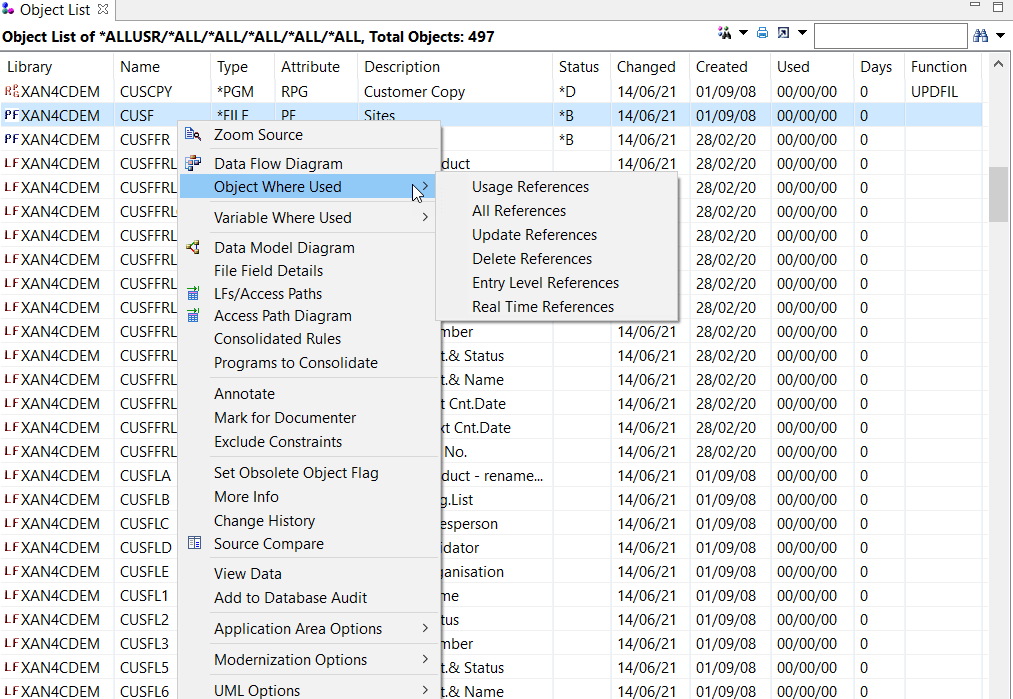


Fig. 2.2.1 – Object Where Used option

This produces an Object Where Used listing for all objects referring to CUSF.

If you select the Object Where Used on a PF (CUSF, in this case), then besides listing the objects using CUSF, the objects using the LFs built on CUSF (for example, CUSFL3) are also reported.

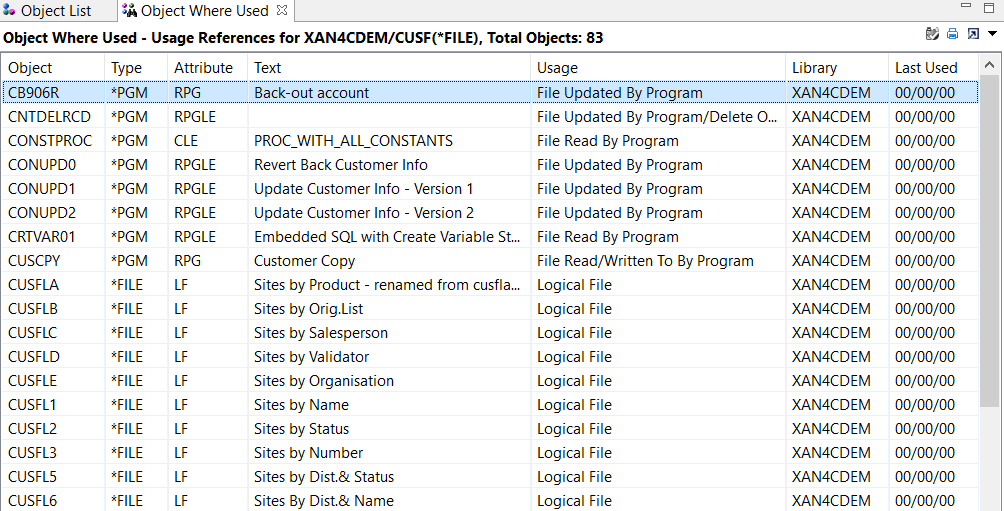


Fig. 2.2.2 – Object Where Used – Usage References screen

On the other hand, if you select the Object Where Used on a LF (CUSFL3, in this case), then besides listing the objects using CUSF, the Physical Files on which the Logical File created are also reported.

If you select the Object Where Used on a variable in a \*PGM type object, then all objects where the variable has been used are listed.

Select an object and click ENTER to access its Source List. Alternatively, double-click on that object row. Double-click on an object from the Object Where Used list to zoom on to the source line where the first reference has been made.

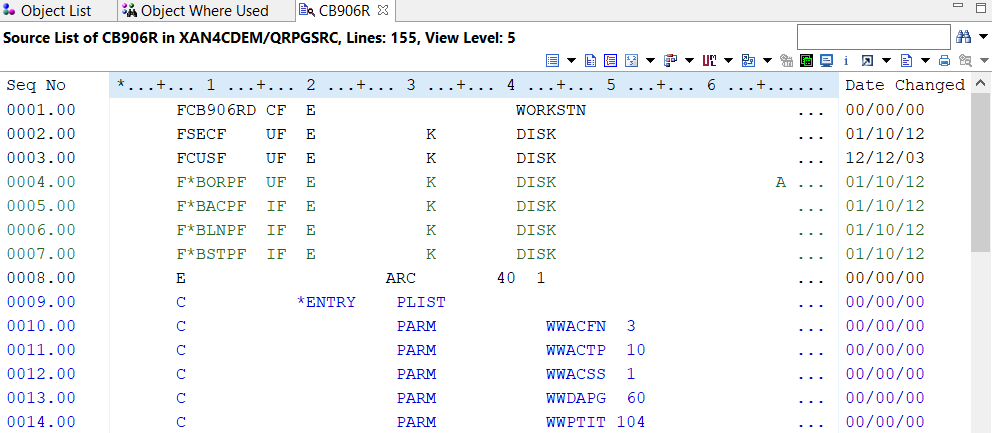


Fig. 2.2.3 – Object Where Used – Usage References screen

For Entry Level References, the Fetch Next References option allows the user to view the references for a particular object at different levels. The maximum attainable level is 11 and the Fetch Next References toolbar button will be disabled if, in case, there is no references available for the next level. Click the toolbar button successively in order to get the next references. Refer to the screen below:

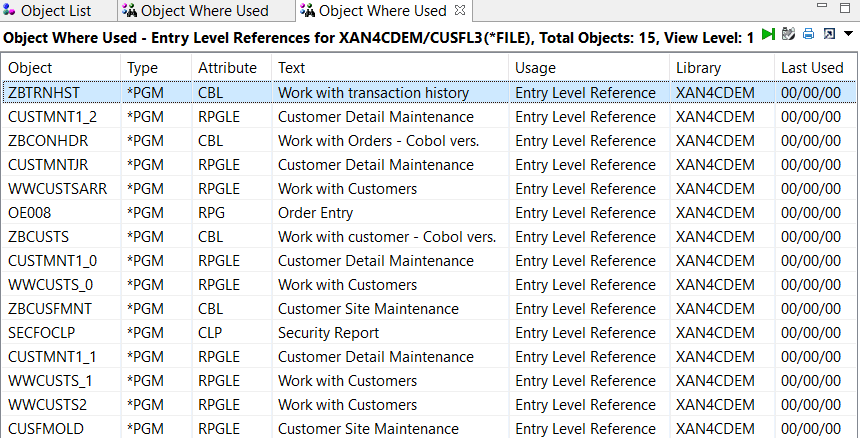


Fig. 2.2.4 – Entry Level References List

Select the Show Call Path option to display the path from the Entry level program to the referred object which may be a file/program, and on which the option for Object Where Used > Entry Level References are shown. Opt for the context menu on an object from the Entry Level references, and then select the Show Call Path option. Refer to the screen below:

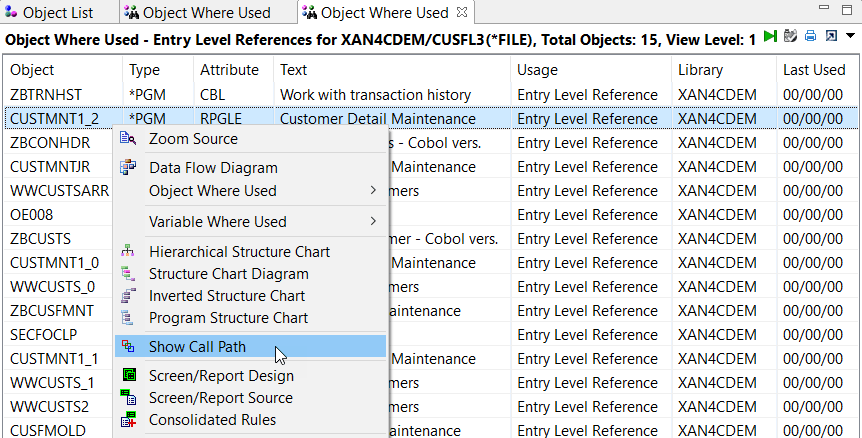


Fig. 2.2.5 – Object Where Used - Entry Level References

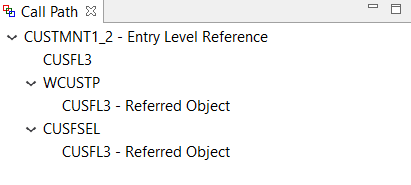


Fig. 2.2.6 – Entry Level References List

|  |  |
| --- | --- |
|  | The Object Where Used references will always show usage for DSPF as Input/Output and for PRTF as Output.  When the Object Where Used is selected on \*MODULE, the program to which this module is an entry module and the references to this program will be included.  The ‘Text’ Column will be renamed to ‘Text/Long Name’ column when at least one file with a long name is present and the Windows>Preferences>X-Analysis XX.X.X>Show Long Names for Files option is selected. |

Export a List to PDF, MS Word or MS Excel

X-Analysis provides a facility to export data from any displayed window. The window toolbar on the Object Where Used view has an Export Options drop-down button, with Export to PDF, Export to MS Word, and Export to MS Excel options.

Select the required option to export the list into PDF/MS Word or Excel.

|  |  |
| --- | --- |
|  | To export the data to DOCX or XLSX format, refer to the Appendix H “Export to Google Drive” in the document “X-Analysis\_User\_Manual” for this release. |

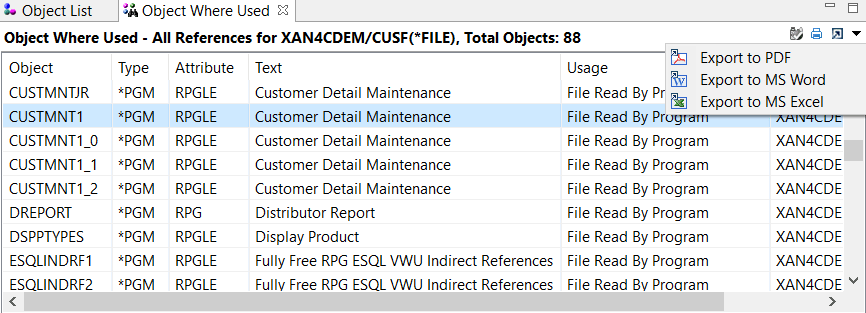


Fig. 2.3.1 – Export Options icon on Object Where Used

Data Flow Diagram

A Data Flow Diagram (DFD) is a diagrammatic representation of the data flow. A particular application contains many files and programs, and a DFD depicts all possible interrelationships among these multiple files and programs contained within that application. A color-coded DFD simultaneously plays the dual role of presenting the data flow at the high object level, in addition to providing contextual details regarding specific variables and parameters passed between objects.

The Bus Routing DFD offers two benefits. First, it gives a precise idea of where and how the inputted data affects the desired output; second, it helps to visualize how an object within an application correlates with the other objects. The diagrams showing the data flow in neat blocks let even non-technical users get a clear picture of the object(s) interrelationships.

Select the Data Flow Diagram option from the context menu over an object, as shown below:

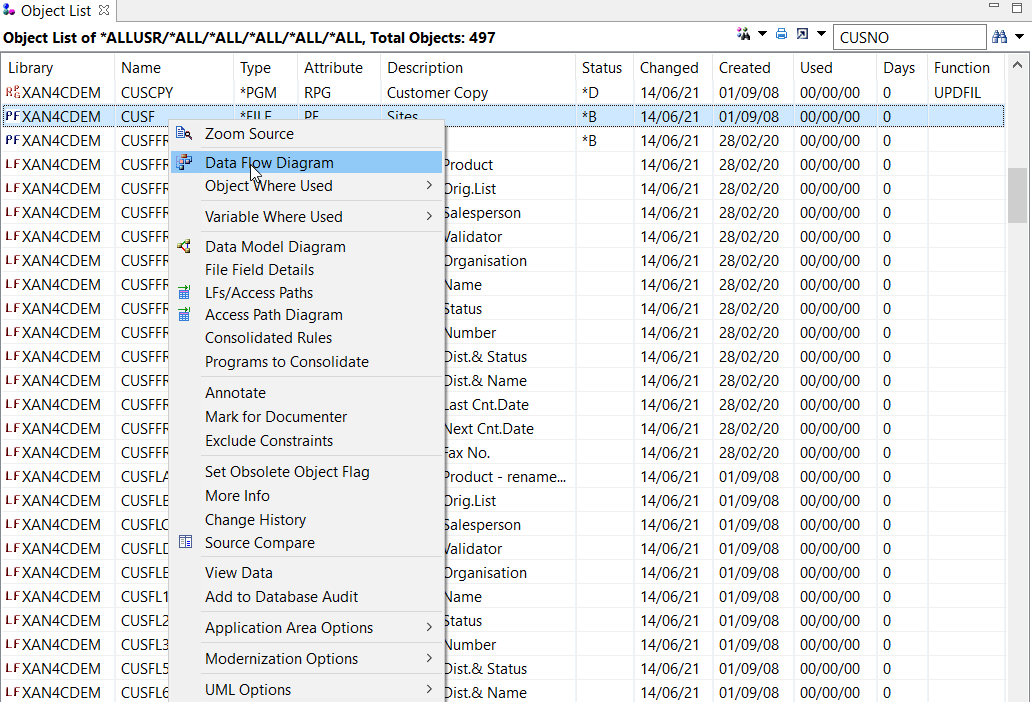


Fig. 2.4.1 – Data Flow Diagram option

This brings up the DFD for the object, CUSF.

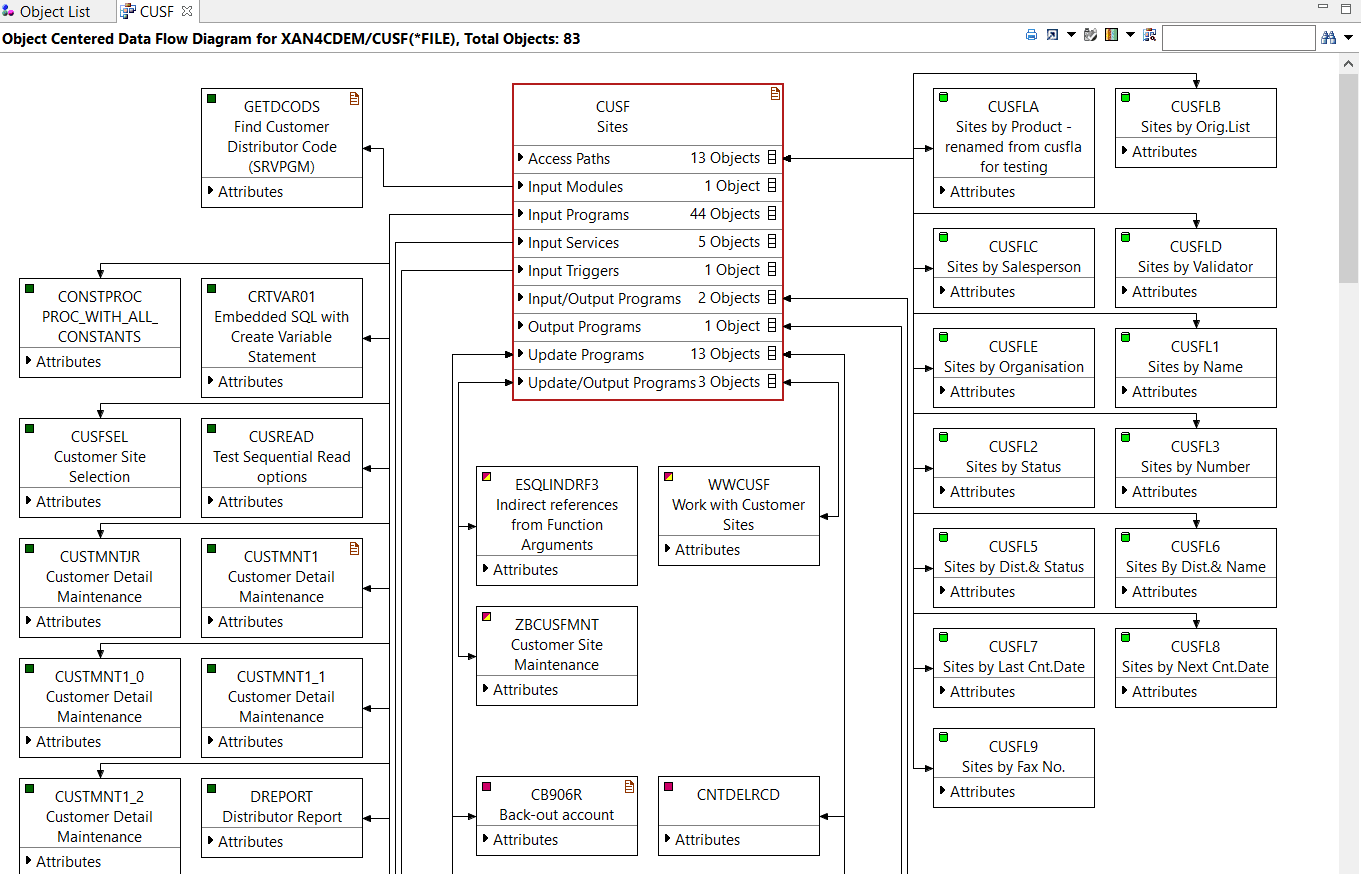


Fig. 2.4.2 – Data Flow Diagram

The DFD is also a graphic equivalent of the Object Where Used data. Besides the Logical Views/Access Paths for CUSF, you can see how CUSF fits into the application (that is, the programs that update CUSF, programs that use CUSF as an input file, and so on).

You will see how the dynamic and interactive diagram facilitates access to other features contained within an object. There are varied and flexible features and options like Hide/Show Objects, Expand/Collapse Referenced Objects, View Annotation, Attributes – all of these and more are explained ahead.

|  |  |
| --- | --- |
|  | In program-centered DFDs, the display files or DSPFs will be shown as Input/Output Files. In an object-centered DFD on a DSPF, programs will be shown as Input/Output Programs.  In an object-centered DFD on a PRTF, the programs will be shown in the Output mode whereas in the program-centered DFD, the PRTF will be shown in the Output mode. |

Hide/Show Objects

To have a better understanding of object interrelationships, the user can hide/show all objects pertaining to programs. A single-click action on the icon alongside the objects enables you to hide/show objects.

In the following screen, the icon against Access Paths is clicked to hide objects.

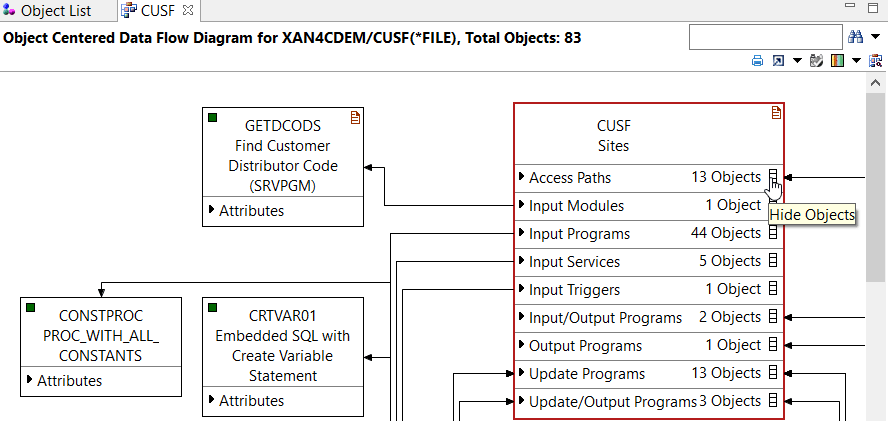


Fig. 2.4.3 – Hide Objects icon

The following screen shows how another click displays all the objects within.

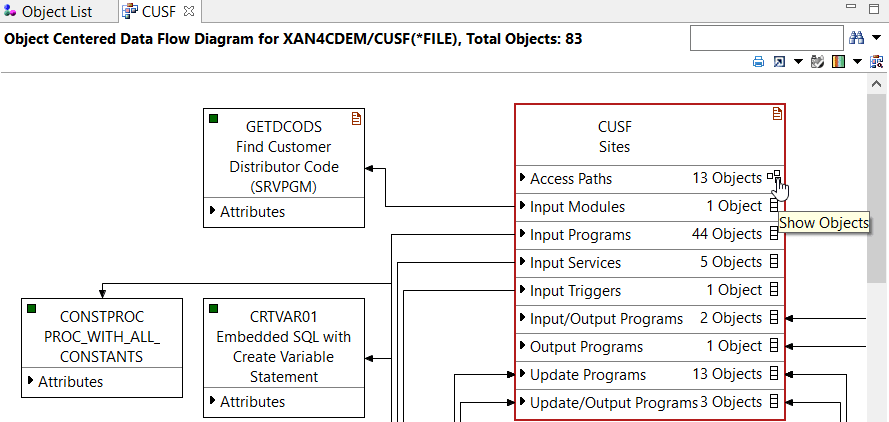


Fig. 2.4.4 – Show Objects icon

Expand/Collapse Referenced Objects

The DFD takes another dynamic form when you expand the referenced object(s) on selecting a particular category from the main object’s box. The following screen displays how the user can perform the action:

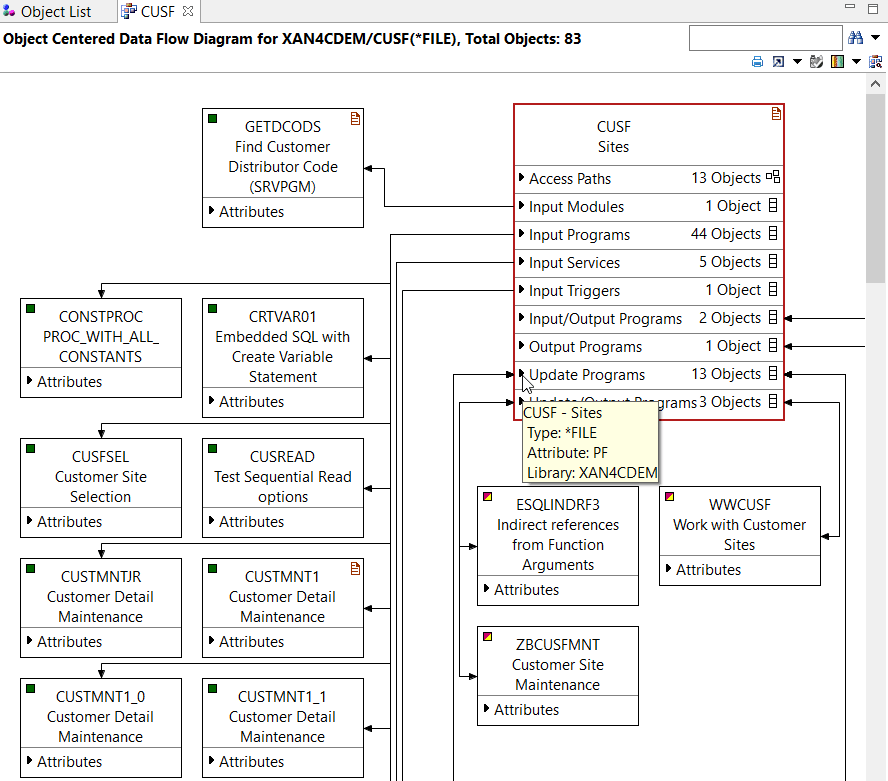


Fig. 2.4.5 – Expand Referenced Objects

A single click on the arrow icon adjacent to Update Programs shows the objects updating CUSF. The following screen displays the expanded view of the referenced objects’ list:

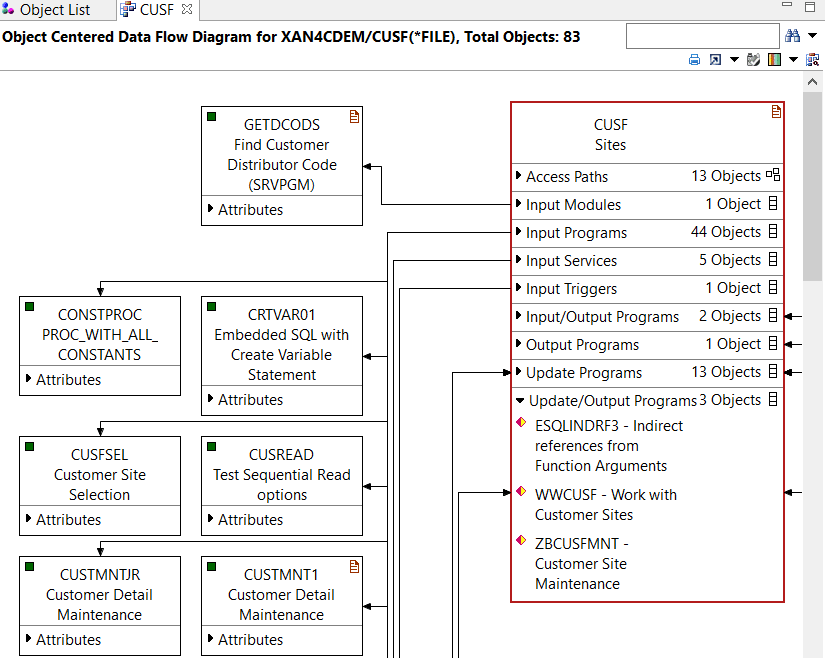


Fig. 2.4.6 – Expanded view of Update Programs

View Annotation

The View Annotation option allows user to see annotation for a specific object. There is a red icon denoting Annotated text on the right side of the Object. Refer to the below screen.

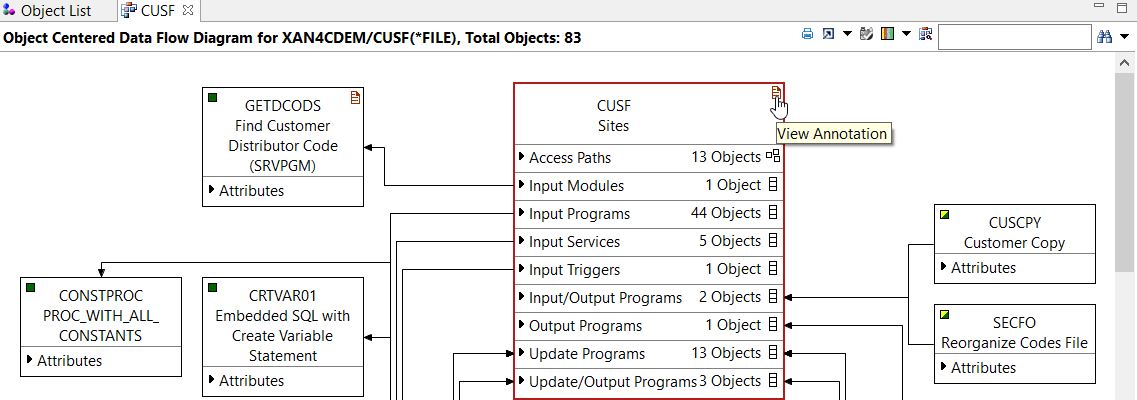


Fig. 2.4.7 – View Annotation option

Double-click on the View Annotation icon to invoke the Annotation dialog box, displaying the notes that had been written earlier.

You can further add/modify notes in the Annotation dialog box and click Save. The user can view these notes later, using the same option.

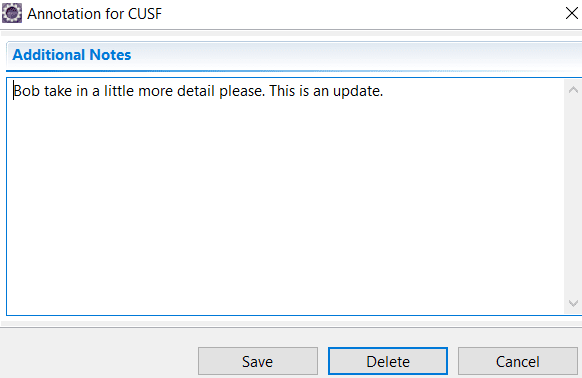


Fig. 2.4.8 – Annotation dialog box

Attributes

The Attributes feature comprises the impacted fields of a particular object. Through these fields, the main object is referenced.

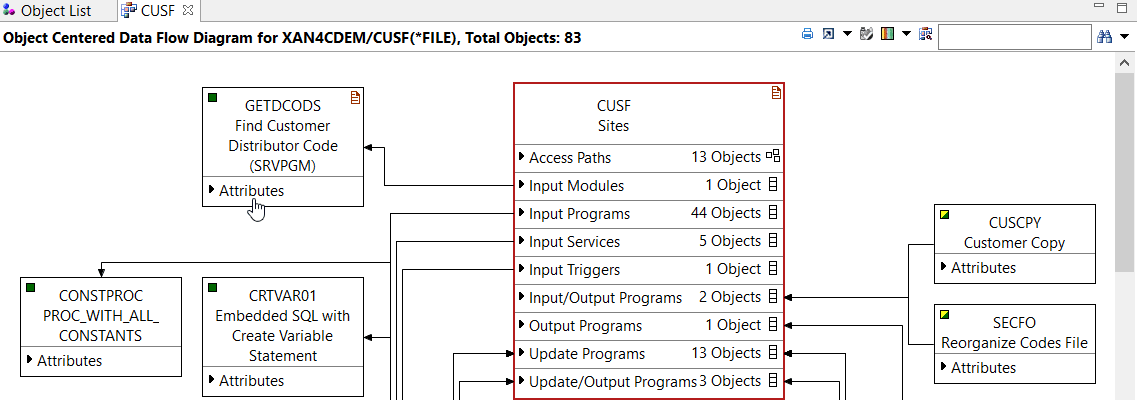


Fig. 2.4.9 – Attributes icon

The different objects corresponding to the main object have fields within them, implicit as ‘Attributes’. These are easily accessible by a single-click on the Attributes icon. The term ‘Attributes’ is changed to related attribute – input, update, output, etc., on expanding the Attributes button, as shown below:

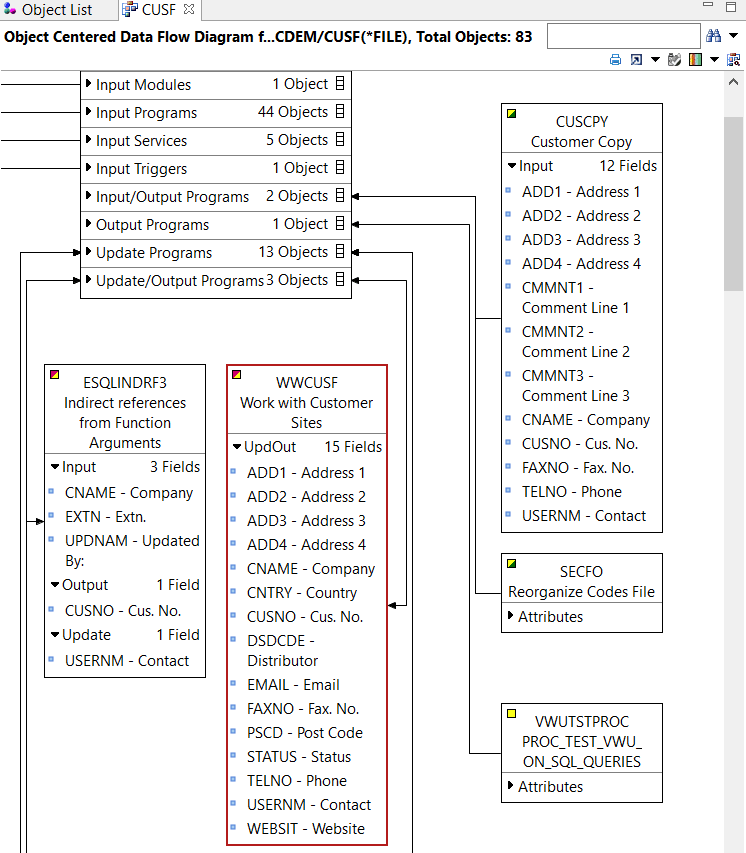


Fig. 2.4.10 – Expanded Attributes

Detail DFD icon

The Detail DFD presents the field usage of all the objects.

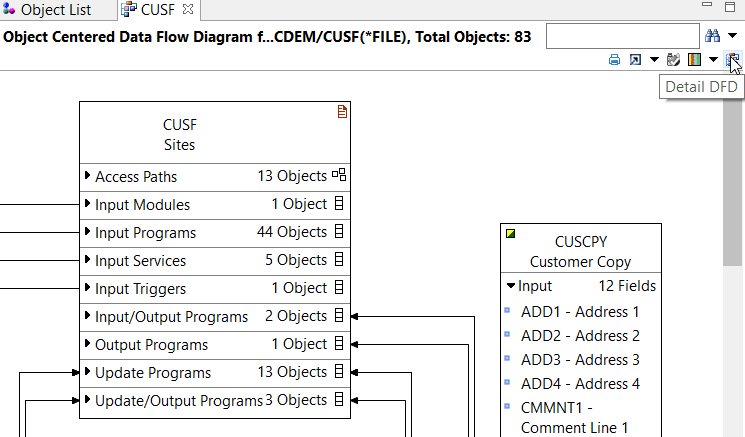


Fig. 2.4.11 – Detail DFD icon

When you click on the Detail DFD icon on the toolbar, the Attributes section of all the objects gets expanded, displaying referred fields from all objects.

The following screen displays the detailed DFD screen:

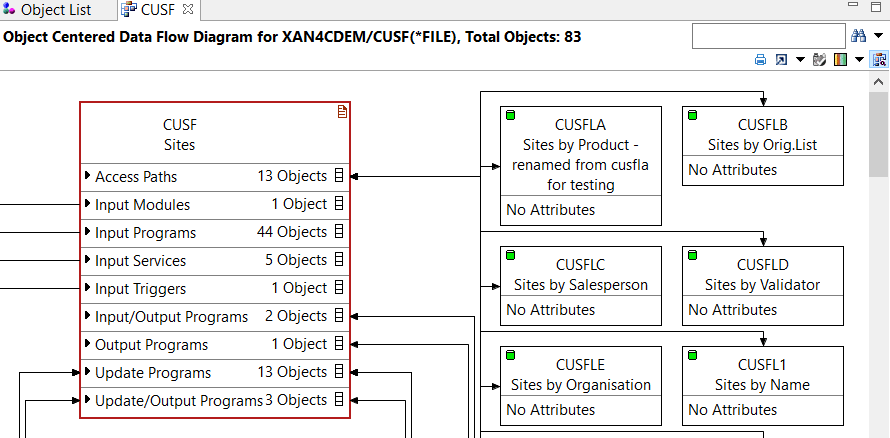


Fig. 2.4.12 – Detail DFD view

|  |  |
| --- | --- |
|  | The Detailed Data Flow Diagram or Field Usage on Data Flow Diagram will now show the long name for the files when the long names exist and the Windows>Preferences>X-Analysis XX.X.X>Show Long Names for Files option is selected. |

Legend icon

The Legend icon on the toolbar displays details of the color scheme used by the DFD. Click the icon shown below:

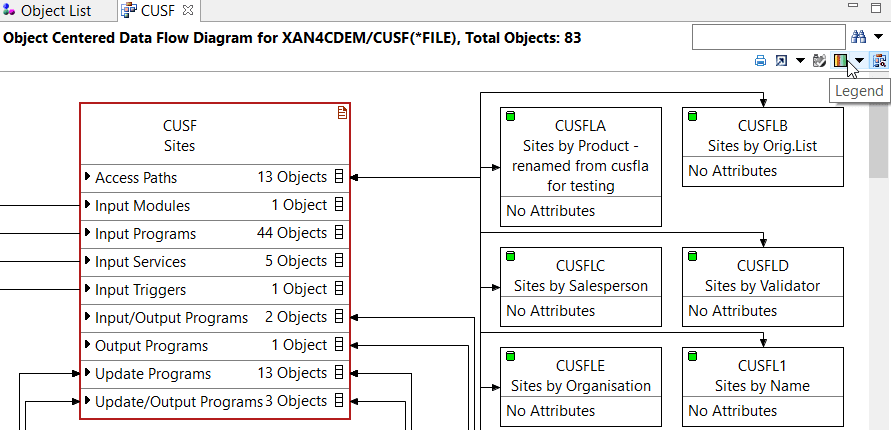


Fig. 2.4.13 – Legend icon

The color scheme helps in understand the reference and association of specific objects, like how each object is referred to or used by the main object on which the DFD was opted. The expanded view of the Legend is given below.

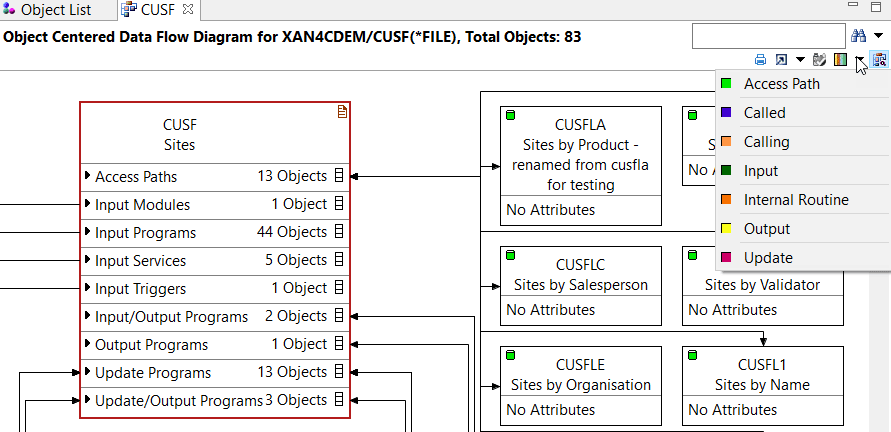


Fig. 2.4.14 – Legend Expanded

Each object box has a colored square or disk on the top left corner. A disk denotes that the object is a file, whereas a square denotes a program. The description of the DFD Legend is as follows:

* Access Path – This is the LF for the File on which the Data Flow Diagram has been opted.
* Called – This is to represent programs called by the main program.
* Calling – This denotes the programs calling the main program.
* Input – For a program-centered DFD, this denotes an input file. For a file-centered DFD, this denotes a program taking input from the file.
* Internal Routines – These are Synon-specific routines being called from a 2E program.
* Output – For a program-centered DFD, this denotes an output file. For a file-centered DFD, this denotes a program giving output to the file.
* Update – For a program-centered DFD, this denotes an update file. For a file-centered DFD, this denotes a program updating the file.

DFD Restricted to an Application Area

X-Analysis provides an additional feature related to DFDs. You can restrict the DFDs to the selected application area by clicking on the Restrict To Application Area icon. Clicking the icon will display only those child objects which belong to that particular selected application area.

The following screen displays the DFD on CUSF. Notice that the Restrict To Application Area icon is disabled. This is so because this option only gets enabled when any application area is selected.

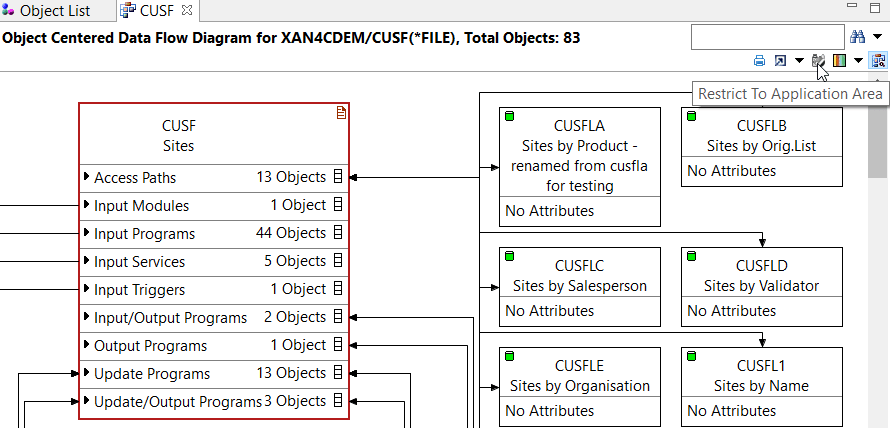


Fig. 2.4.15 – DFD on CUSF with disabled Restrict To Application Area icon

Now select the application area, MVCPROCESS.

Select the Data Flow Diagram option on CUSF; notice that the Restrict To Application Area icon is enabled (see the following screen). This is so because you have selected an application area.

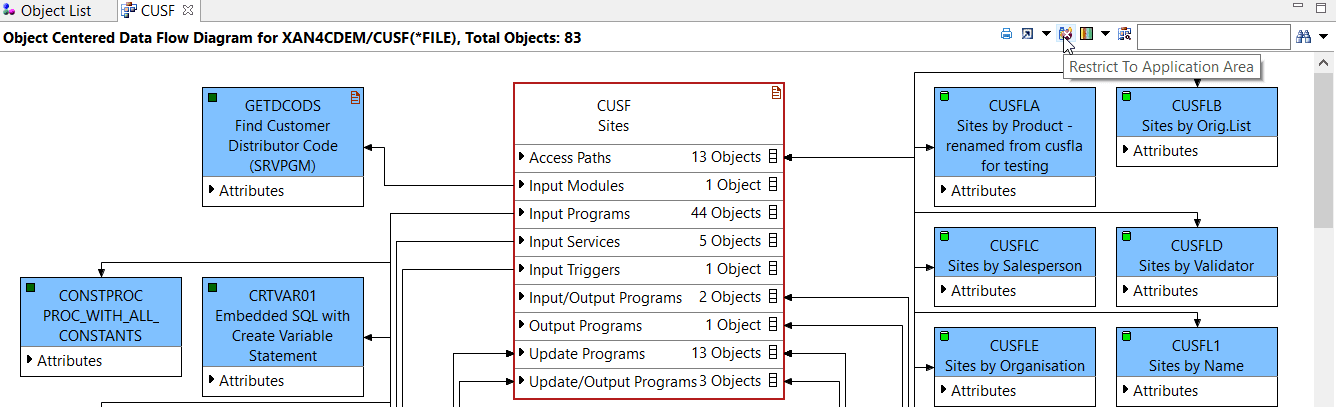


Fig. 2.4.16 – Data Flow Diagram on CUSF with enabled Restrict To Application Area icon

Now, click Restrict To Application Area icon as shown below:

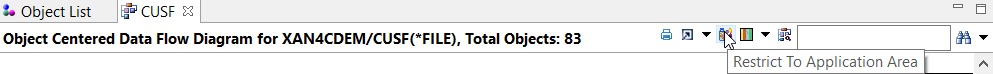


Fig. 2.4.17 – Clicking Restrict To Application Area icon

The following screen is displayed:

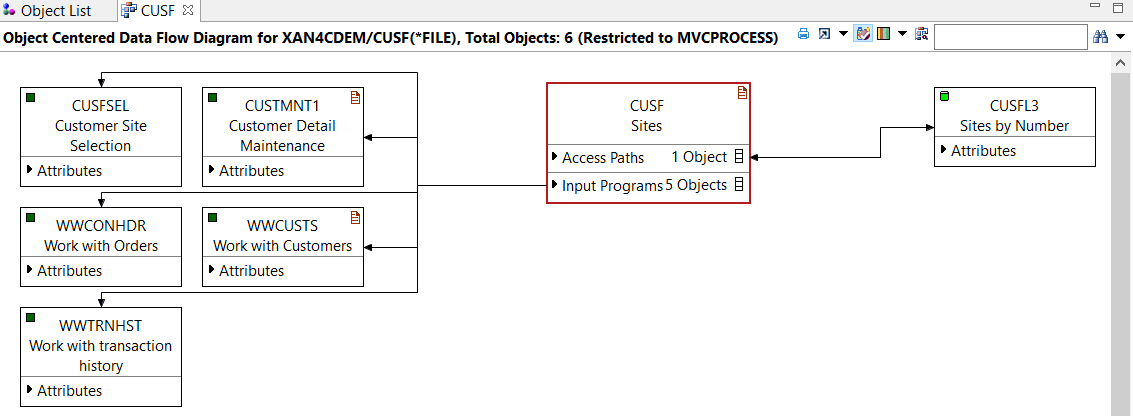


Fig. 2.4.18 – Restricted DFD on CUSF for application area, MVCPROCESS

This shows that the child objects belong to the application area, MVCPROCESS.

|  |  |
| --- | --- |
|  | While using the Restrict To Application Area feature, ensure that the originally-selected application area is not changed. When the Data Flow Diagram is selected on \*MODULE, the program to which this module is an entry module and the references of this program will be included. |

Variable Where Used

The Variable Where Used option lists all source lines from the application, where the field/variable of a file/program is used/referenced in any of the source members and its associated device files and copybooks.

A wide variety of options can be specified including:

Files, Array Definitions, Data Structures, Sub-Fields Data Structures, Indicators, Key Lists, Data Fields, File Formats, Subroutines, Program Variable, Array Elements, Parameter Lists, Parameters, Key Fields, Message Ids, EXCPT Names, etc.

The Variable Where Used submenu provides the following options:

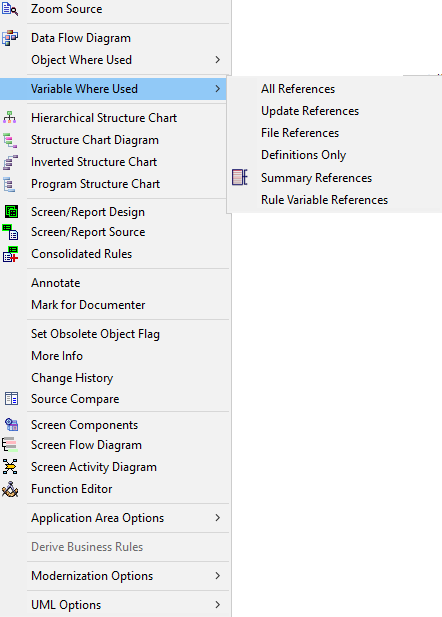


Fig. 2.5.1 – Options in Variable Where Used

* All References: All references of this variable.
* Update References: References where this variable is being updated.
* File References: All references where the object using this variable is a file.
* Definitions Only: Lists references where this variable was defined.
* Summary References\*: This option on a variable or File/Field, lists all members that use the selected variable or File/Field and also the usage for the same. The option is explained in more detail at the end of this section.
* Rule Variable References: Lists all Business Rules based on the selected object from across the entire application.

|  |  |
| --- | --- |
|  | Out of the different options in Variable Where Used submenu, a new icon is added for the option "Summary References". |

Select the Variable Where Used option for the CUSNO field using the JumpTo dialog.

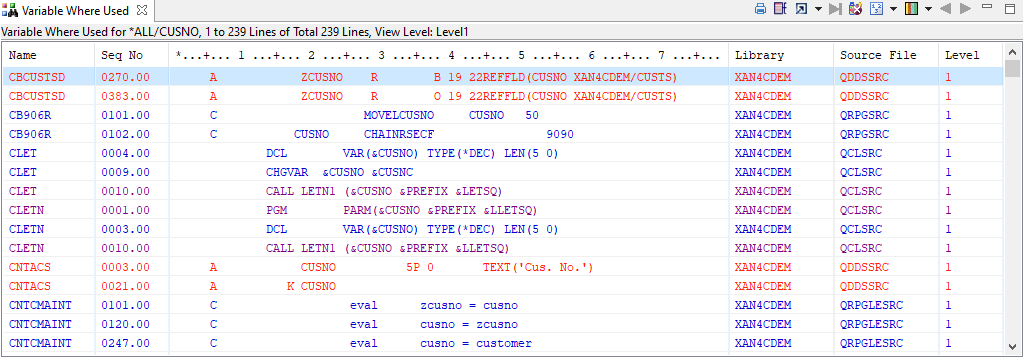


Fig. 2.5.2 – Variable Where Used view for CUSNO

Legend

* Dark Red depicts the Field Definition Statements in DDS sources only.
* Black depicts the File Operations.
* Magenta depicts the Parameters.
* Dark Brown depicts the Program Structure Operation.
* Dark Blue depicts the Field Operation.

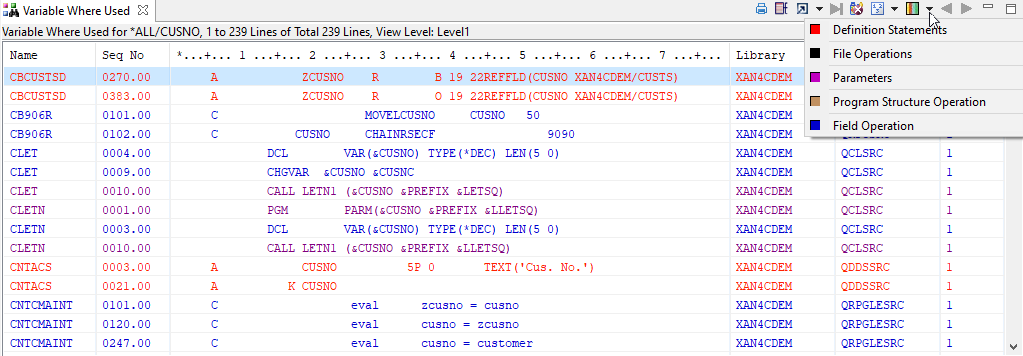


Fig. 2.5.3 – Legend for Variable Where Used

View Levels

The Variable Where Used view is available up to Level 6 view and the Default View Level icon, located on the toolbar allows access to these levels.

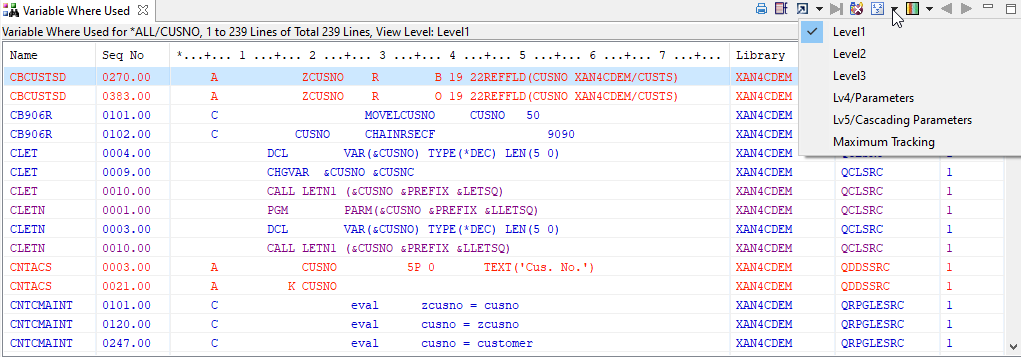


Fig. 2.5.4 – Expanded View Level Menu

|  |  |
| --- | --- |
|  | Default View Level for the Variable Where Used option can be set in the X-Analysis Preferences (General Preferences) window. |

The Level selected is displayed in a separate column as shown in the screen below:

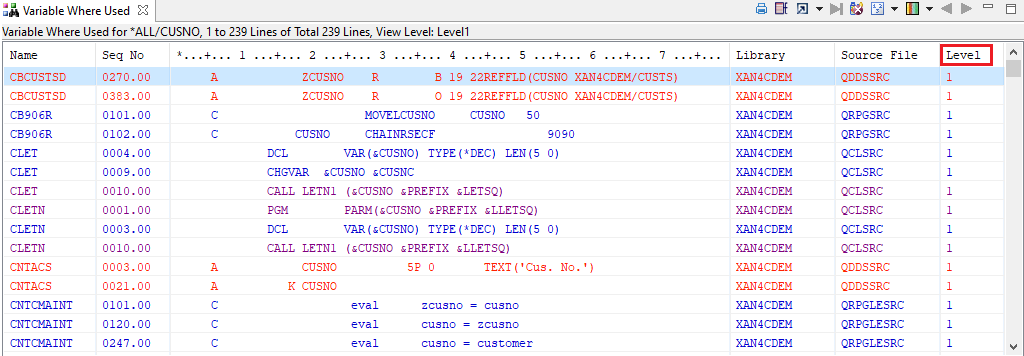


Fig. 2.5.5 – Variable Where Used window showing the Level column

Higher Variable Where Used levels can be requested using VWU Levels. The Variable Where Used levels display the following information:

* Level 1: At this Level, the VWU process displays the direct occurrences of the variable. The same is followed when you select the Member X-Ref option. However, VWU at Level 1 also tracks non-direct references such as program key field mapping, embedded SQL field-to-variable mapping (includes field-to-field and variable-to-variable mapping), etc. Note that the same level In File/Field Where Used is only allowed on file fields, tracking the presence of that field with the combination of file in the programs.

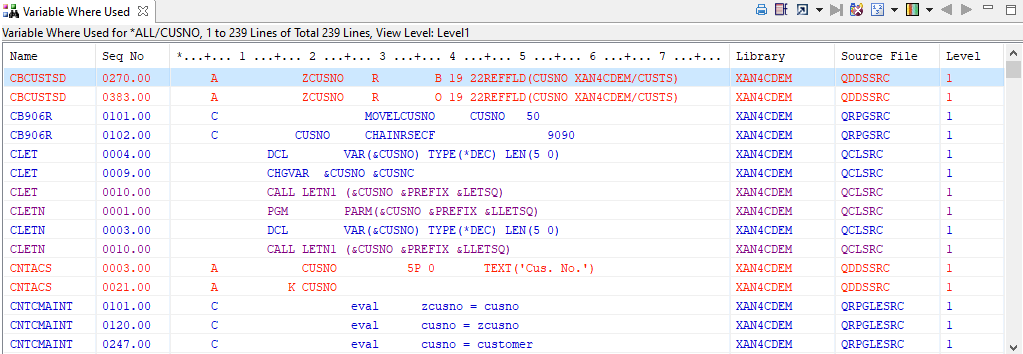


Fig. 2.5.6 – Variable Where Used Level 1 View

* Level 2: This level considers the dependent variables found at level 1 (that is, the variables, which are modified or being modified with the level 1 variable) and tracks them in addition to the entries loaded for level 1.

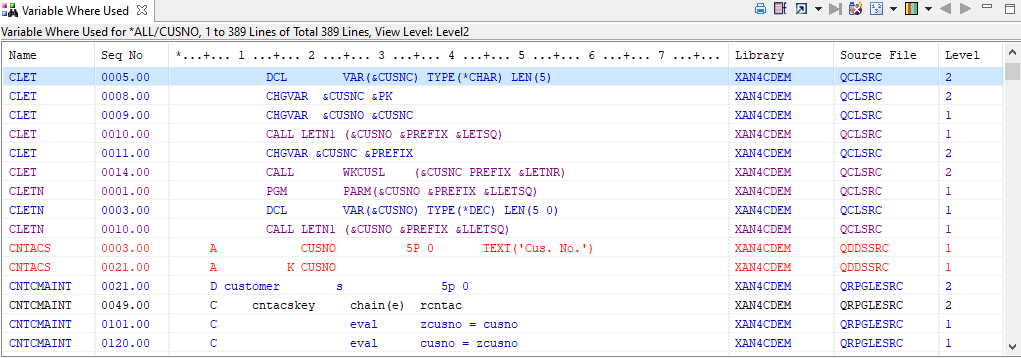


Fig. 2.5.7 – Variable Where Used Level 2 View

* Level 3: This level considers the dependent variables found at level 2 (that is, the variables, which are modified or being modified with the level 2 variable) and tracks them in addition to the entries loaded for level 2.

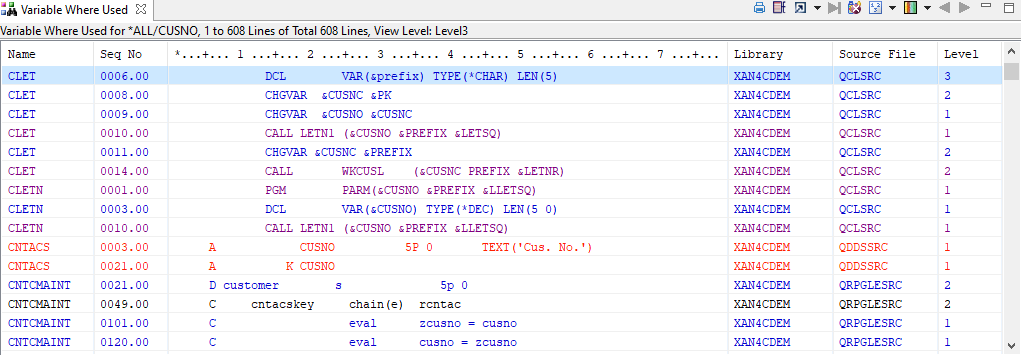


Fig. 2.5.8 – Variable Where Used Level 3 View

* Level 4/Parameters: This level considers the dependent variables found at level 3 (that is, the variables, which are modified or being modified with the level 3 variable) and tracks them in addition to the entries loaded for level 3. It also considers the dependencies of up to level 4 within the specific program for the parameters tracking in calling and called programs.

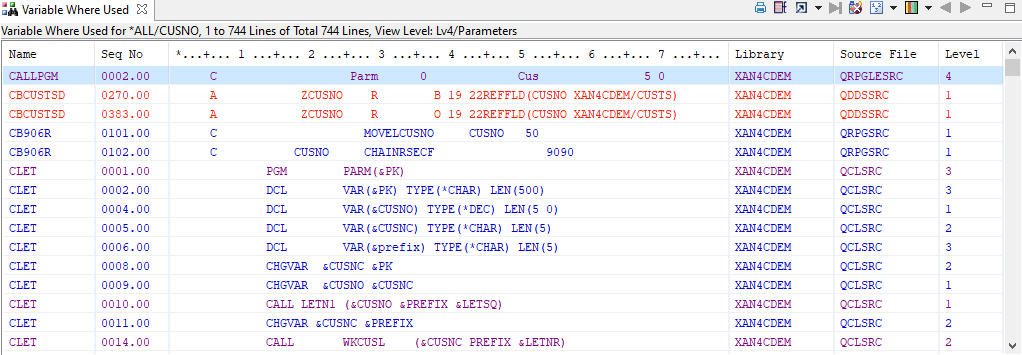


Fig. 2.5.9 – Variable Where Used Level 4 / Parameters View

* Level 5/Cascading Parameters: This level considers the dependent variables found at level 4 (that is, the variables, which are modified or being modified with the level 4 variable) and tracks them in addition to the entries loaded for level 4. Extending the information in Level 4, Level 5 includes references where CUSNO fields are parameters in a function, called by another function.

In addition, parameters tracking (of the variables from level 1 to 4) is done but differently than level 4. In the case of level 4, the mapped parameter is tracked only for level 1. However, in this case, level 5 call gets submitted recursively to track the mapped parameter in that program (that is, calling and/or called program) at level 5. As the recursive call gets submitted at level 5, this will go till the top or bottom of the call stack to trace everything and would keep submitting level 5 dependency for everything.

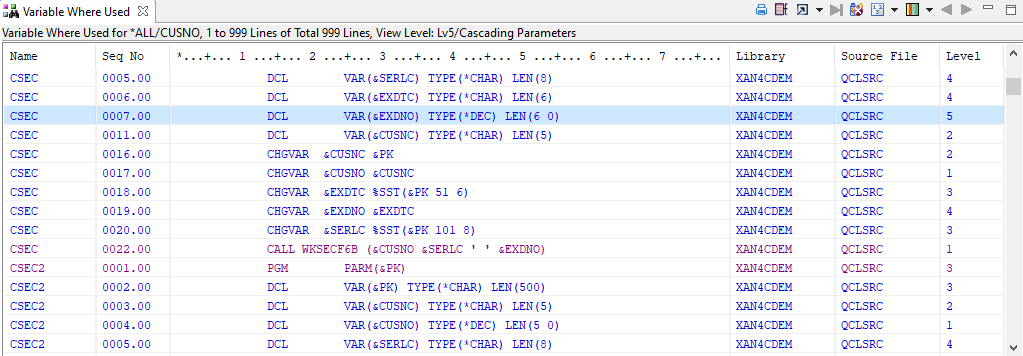


Fig. 2.5.10 – Variable Where Used Level 5 / Cascading Parameters View

* Maximum Tracking: Tracking at this level like that for level 5 in addition to identifying level 6 and level 7 dependencies (only in the program where the request is being made). The tracking outside the program boundary (that is, for calling and called program for cascaded parameter tracking) has no impact.

This way, we can say that Level 1, 2, 3, 6, and 7 have lightweight processing. Level 4 goes for parameter tracking of the entries found up to Level 3 and Level 5 goes for cascaded parameter tracking of the entries found up to Level 3.

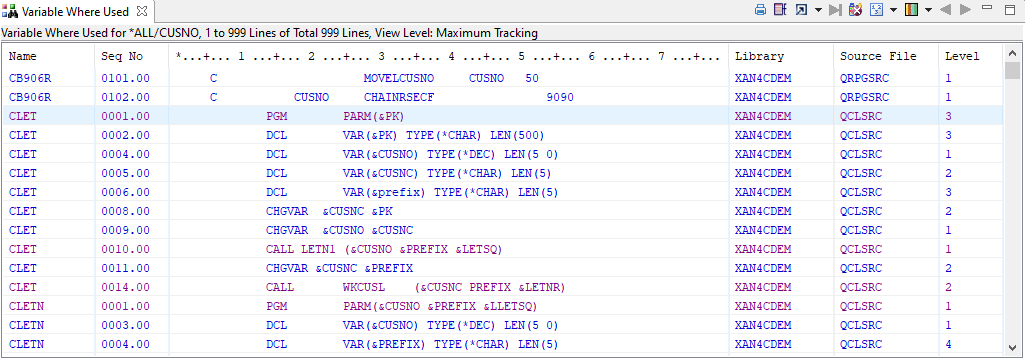


Fig. 2.5.11 – Variable Where Used – Maximum Tracking View

This is the highest view level, also referred to as ‘Maximum Tracking’. It displays Variable Where Used information for all the levels mentioned above, as well as any other possible references of the CUSNO field.

* Backtrack: The Backtrack feature allows a line on 'Variable Where Used' to be traced back when shown at level 2 or higher. The selected line highlights the Variable Where Used option of the variable line which was previously selected. For example, at level 4, the corresponding level 3, level 2, and level 1 lines will display. This allows users to understand why the current line was being shown as level 4 in the VWU result.

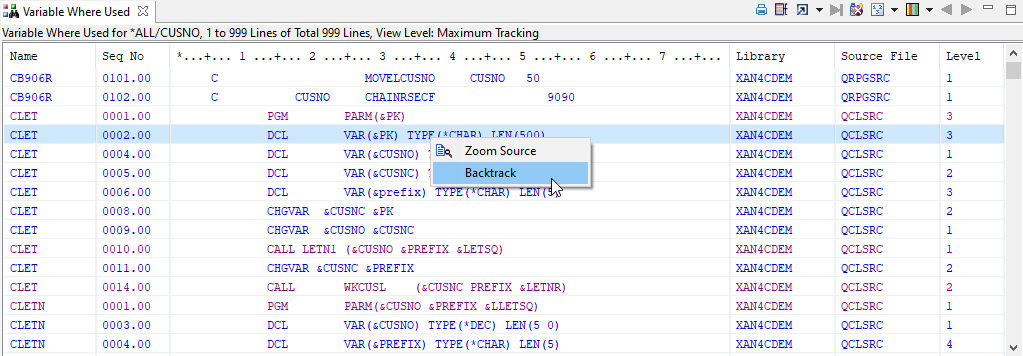


Fig. 2.5.12 – Backtrack – Level 3 Variable

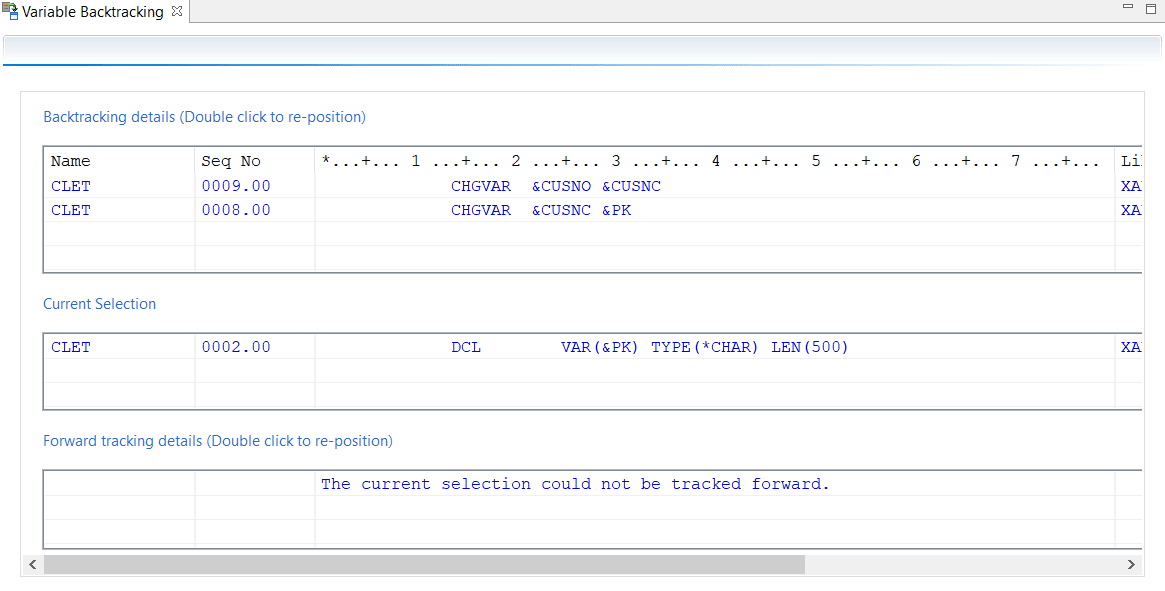


Fig. 2.5.13 – Backtrack – Level 3 Variable

Please note that the Implicit Field Used In Program indicate fields which are being used as a key field but not directly used in the program.

|  |  |
| --- | --- |
|  | Implicit Field Used In Program indicate fields which are being used as a key field but not directly used in the program. |

|  |  |
| --- | --- |
|  | Important note: Under the new VWU Process, tracking of constant literals is restricted to display only those lines where that constant is used. This action disables the tracking of the constant dependencies at further levels. To expand its usage, take the VWU option on the appropriate tracked variable.  The Paging Mechanism allows the user to see 20000 records per page at a time. Navigation is allowed through the pages by using the Next and Previous buttons present on the toolbar. Any sorting done on the columns is limited to the records on that page only. Navigation to the next page would result in the loss of any sorting.  The Export All Records option, to an Excel sheet, is allowed on Variable Where Used records when the option is enabled, and the record count exceeds 20000. The other Export options allow exporting of records on that page only. |

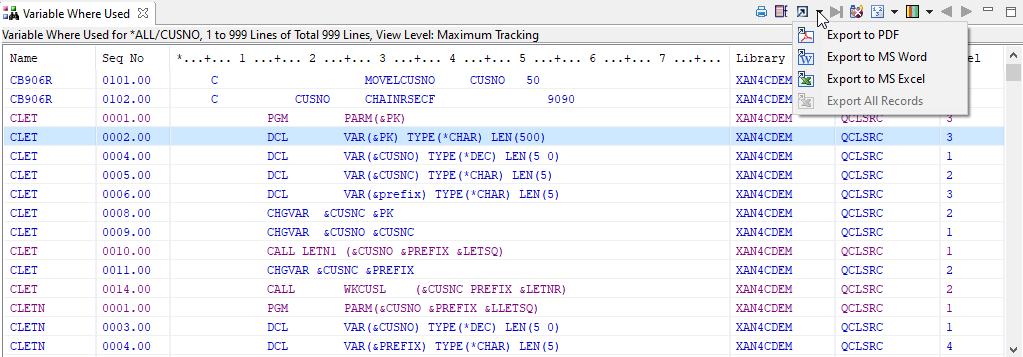


Fig. 2.5.14 – Export All Records

\*Summary References – The Variable Where Used > Summary References option opens a new editor showing the distinct summary of members, using the variable. The Usage Count column shows the number to times the variable is used.

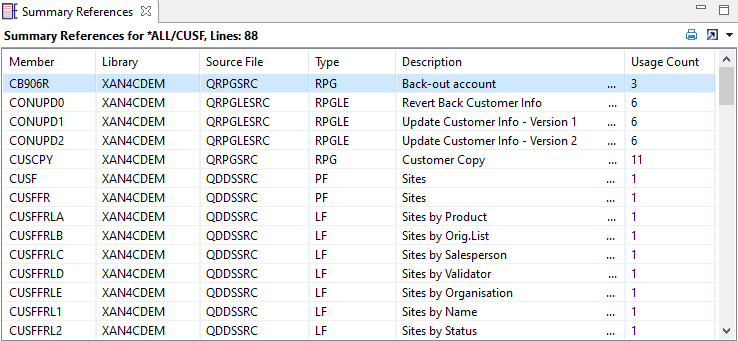


Fig. 2.5.15 – Window showing Summary References for CUSF

Qualified Field and File references in Embedded SQL

SQL statements can have qualified field and file references.

Variable Where Used can be performed on each element of the qualified field or file reference. Additionally, combinations from the qualified reference, that make sense, can also be used for Variable Where Used.

For example,

SELECT FNAME INTO: NAME FROM SALES

WHERE SALES.FNAME = 'Scott'

For the above SALES.FNAME, following can be searched via Variable Where Used feature:

* FNAME
* SALES
* SALES.FNAME

If the above qualified field were being modified in SQL statement like UPDATE, then following will return results when Update References will be opted.

* FNAME
* SALES.FNAME

|  |  |
| --- | --- |
|  | At times, the desired combination may not be specified from the source right click menu. In such cases, and for any possible combination, the JumpTo dialog can be used to specify Variable Where Used, Variable Where Updated. |

File/Field Where Used (FFWU)

The File/Field Where Used (FFWU) functionality works like the Variable Where Used – with the only difference being that the tracking of the references is limited to the fields within the specified file; the VWU process does tracking at the global level.

The FFWU for LFs works like FFWU for the PF. If multiple DB files (including the LF selected in FFWU) are used in any program then lines from all the referred files (that is, database file, printer file, device file, etc.) having same field name will be traced in FFWU. Besides this, if the same name field is also present in any PF and its LF is used in the program, then lines from both PF and LF will also be traced.

Structure Chart Diagram

Before going to documenting program logic, it is imperative to understand how control transfers from one program to the next.

A Structure Chart Diagram (or SCD) addresses this issue and represents the control through graphical, color-coded block diagrams. SCD is a nested tree diagram that shows the complete call hierarchy of the ‘programs called’. These diagrams contain relevant information as per control flow and call structure. Also, the user can view data input objects and avail of a summarized description for each of the objects. Important functional aspects like updates, prints, and displays are color-coded to help the user to instantly focus on these commonly preferred details.

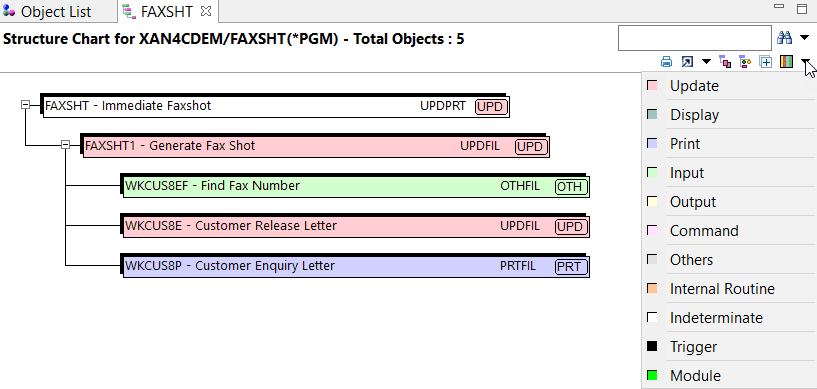


Fig. 2.6.1 – Structure Chart Diagram

Legend

The SCD Legend describes the type of object displayed.

* Update – This depicts the program that updates a file.
* Display – This depicts the program that uses a Display file.
* Print – This depicts the program that uses a Print file.
* Input – This depicts the program that uses an Input file.
* Output – This depicts the program that writes to a file.
* Command – This simply depicts a Command.
* Others – This refers to the Programs where it is referring to a file which is not Update/Display/Print/Input/Output.
* Internal Routine – This refers to the Synon-specific routines.
* Indeterminate – This depicts the programs where the usage cannot be programmatically determined.
* Trigger – This depicts the program which is a Trigger.
* Module – This depicts the program which is a Module.



Fig. 2.6.2 – Rollup Group

The Rollup group describes the cumulative component function of the program and its dependents.

The following types of Rollup groups are available:

* UPD – At least one program updates a file.
* PRT – Program and dependent programs create a printed report.
* DSP – Program and dependent programs use input files and display files.
* OTH – No cumulative component function can be determined.

Annotate

X-Analysis also allows annotation on the Structure Chart.

Select an object displayed in the SCD and opt for the Annotate option from the context menu.

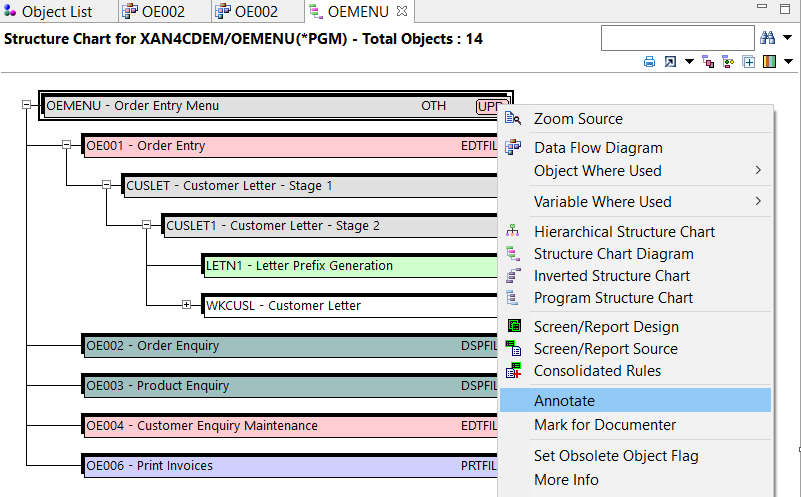


Fig. 2.6.3 – Annotate option

This brings up the Annotation dialog box. The user must write their note and click Save. The annotation is stored in a file available in the cross-reference library.

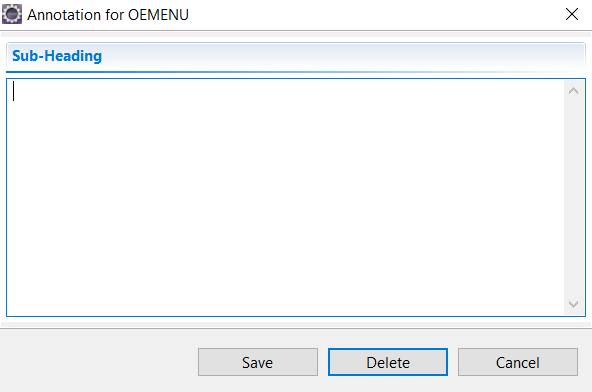


Fig. 2.6.4 – Annotation dialog box for OEMENU

Regenerate the SCD to check for any object annotations.

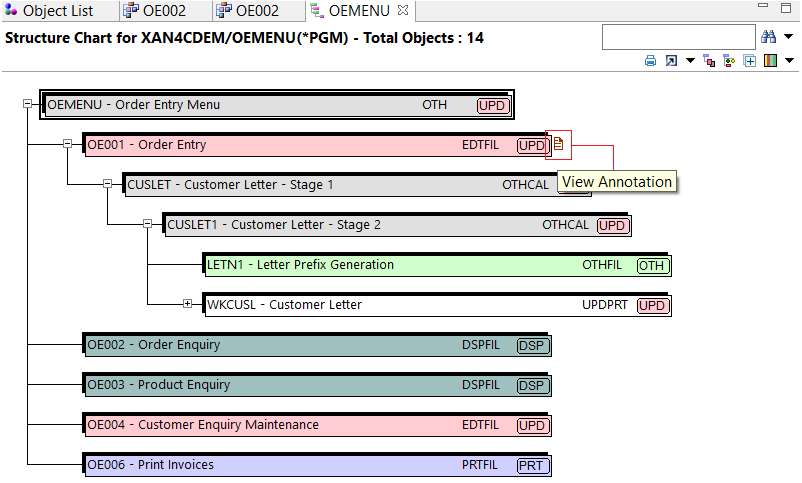


Fig. 2.6.5 – Annotation icon for OEMENU in a re-generated SCD

Double-click the Annotation icon to bring up the Annotation dialog box.

Function Type

Function Type describes the function of the object and is based on COOL: 2E definitions.

Hierarchy Exclusions in SCD

Work with Exclusions is Option 16 on the Work with X-Analysis/4 Applications screen. With the help of this feature, the user can add a program’s name for Hierarchy Exclusions. This affects the SCD and the OSC. This feature removes the child nodes of the excluded object, and a green arrow is displayed to indicate the hierarchy exclusion. The below sample screen shows this feature.

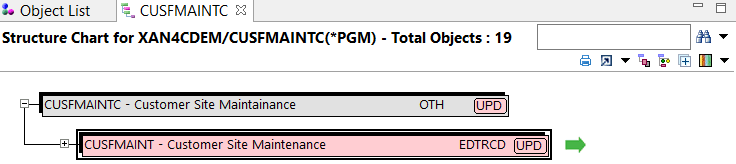


Fig. 2.6.6 – Hierarchy Exclusions in SCD

Repeated nodes in SCD

The SCD displays the repeated objects (having child nodes) with a green arrow beside them. The following sample screen demonstrates the repeated nodes in an SCD.

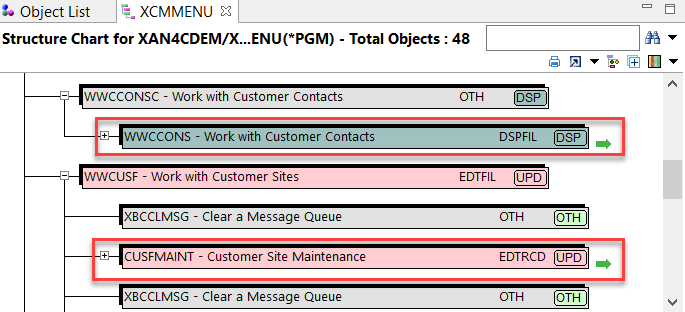


Fig. 2.6.7 – Repeated nodes in SCD

Structure Chart Diagram with Files

The SCD provides a graphic display of the program-to-program relationships. The SCD with files also displays associated files along with programs. Note that the associated files, data areas along with programs get displayed, but the user-defined printer files and the display files are not displayed.

To see the SCD with files, click Show Files icon available on the SCD toolbar.

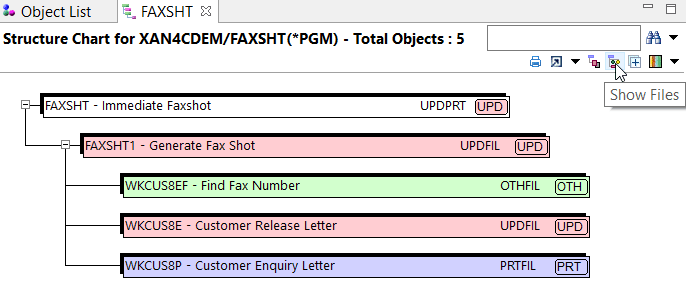


Fig. 2.6.8 – Show Files icon

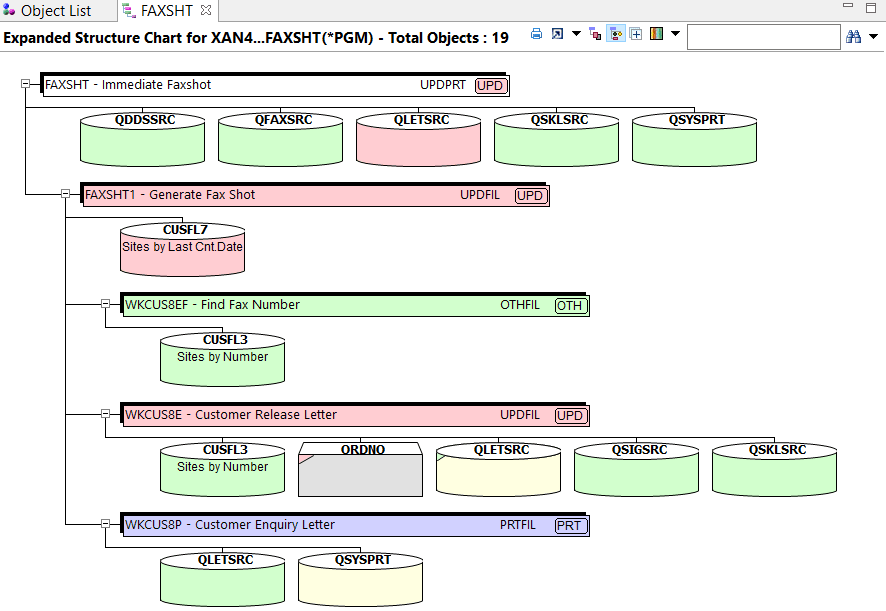


Fig. 2.6.9 – Structure Chart Diagram with I/O Files

Hierarchical Structure Chart Diagram

The Hierarchical Structure Chart Diagram offers a different layout for viewing the SCD. This illustration gives a well-defined view of all the programs by representing their control flow and call structure. Neat, color-coded bus routing block diagrams depict the movement of control/programs.

Like the SCD, the Hierarchical Structure Chart Diagram displays the repeated objects (having child nodes) as expanded and shaded in green.

Select the Hierarchical Structure Chart option for any object from the context menu. In the following screen, OEMENU is selected.

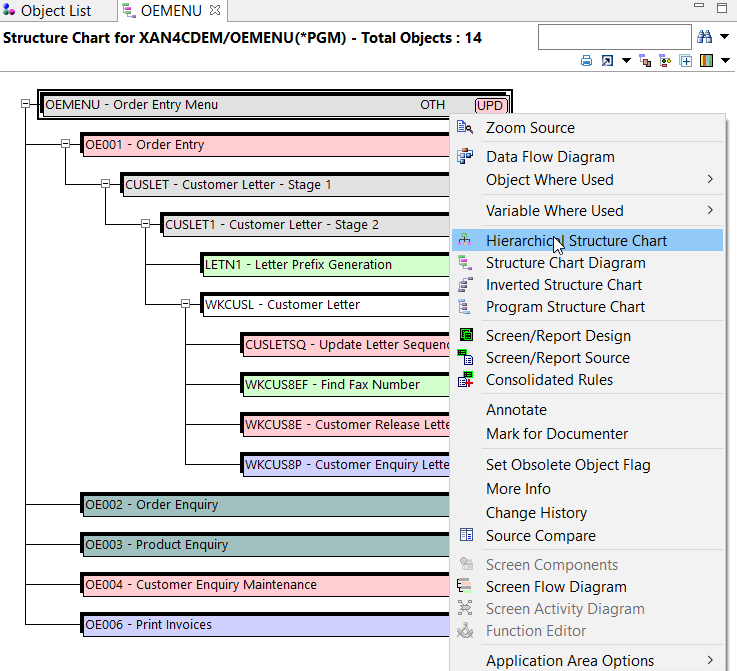


Fig. 2.7.1 – Hierarchical Structure Chart option

The screen below shows the delineation of the main object OEMENU into several blocks of related objects in the order of calling. The color codes signify the identity of objects as command-based or input-based or print-based, and so on.

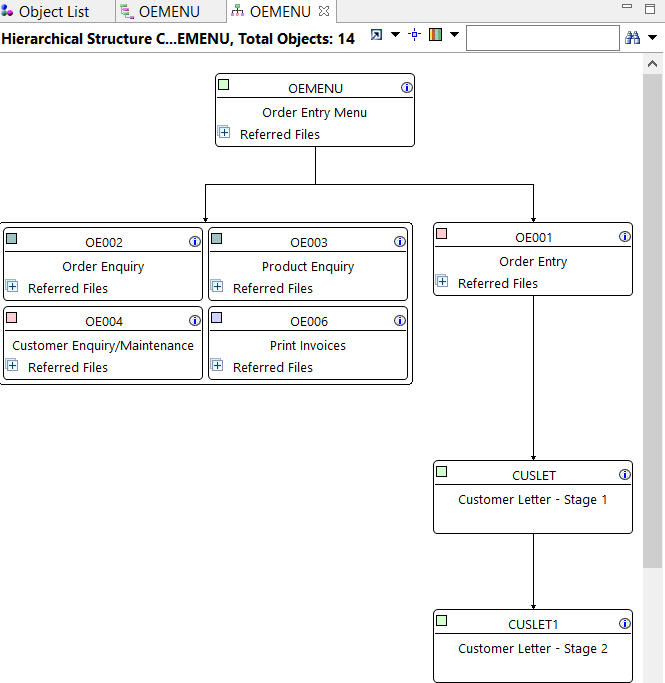


Fig. 2.7.2 – Hierarchical Structure Chart

Narration

Access the additional details related to an object through the icon in the screen below.

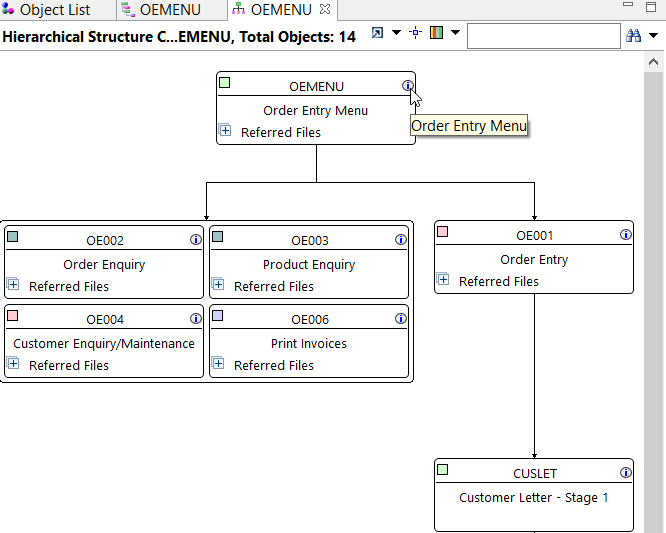


Fig. 2.7.3 – Narratives icon

Click on the icon displayed above to invoke a pop-up window which provides the auto-generated narration for the program, as shown below:

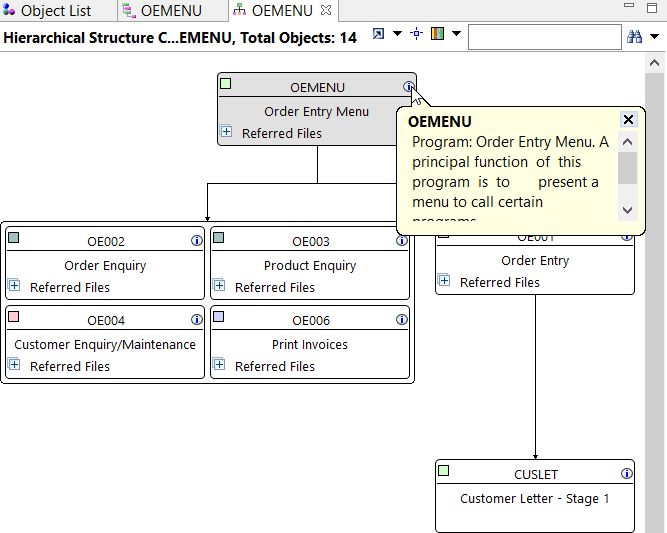


Fig. 2.7.4 – Narratives pop-up box

Referred Files

The Hierarchical Structure Chart has a feature that displays the referred files inline. An expandable icon called Referred Files is available in each box. Click on the icon preceding Referred Files in the screen below.

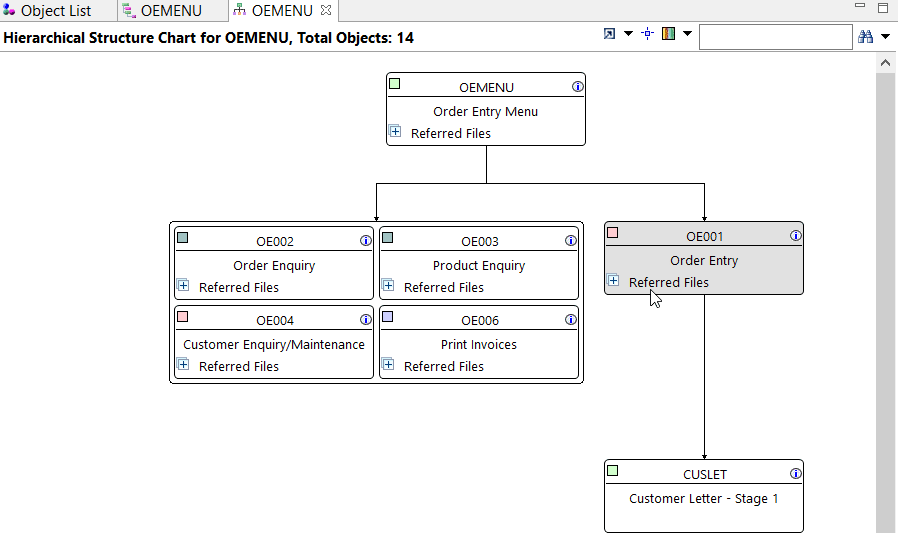


Fig. 2.7.5 – Referred Files

The box will expand to display the files referred to by OE001. The color-coded geometric shape before the file name indicates the file type.

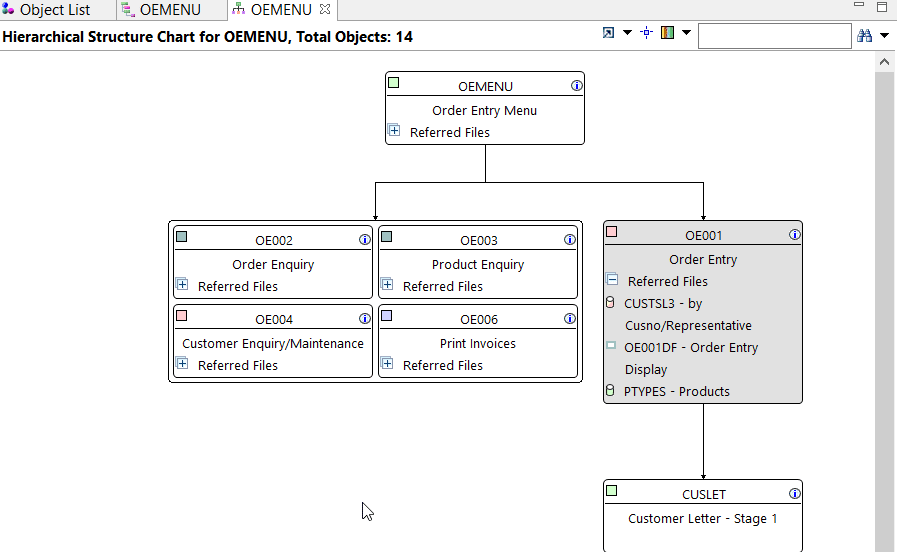


Fig. 2.7.6 – Referred Files – Expanded

Show Root Node

Click the Show Root Node icon to display the root node:

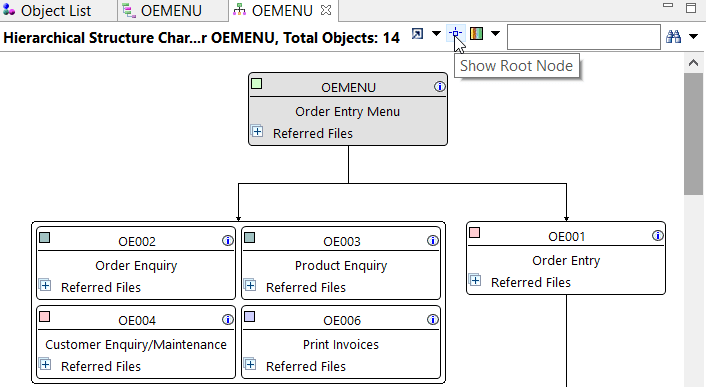


Fig. 2.7.7 – Show Root Node icon

Legend icon

The Legend icon on the toolbar displays details of the color scheme used by the Hierarchical SCD. The colors help in establishing the reference and association of specific objects. Click the Legend icon, as shown below:

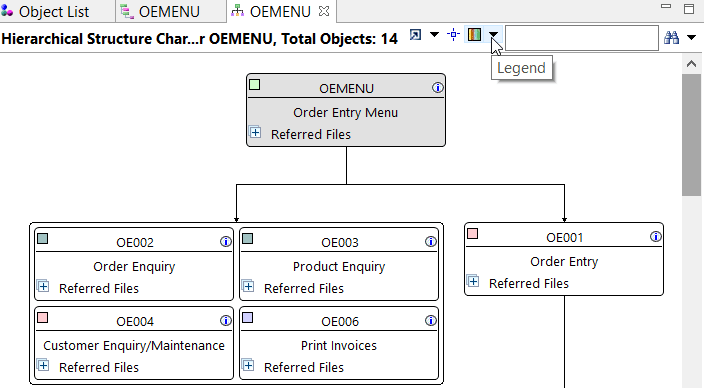


Fig. 2.7.8 – Legend icon

The following screen shows the expanded view of the Legend:

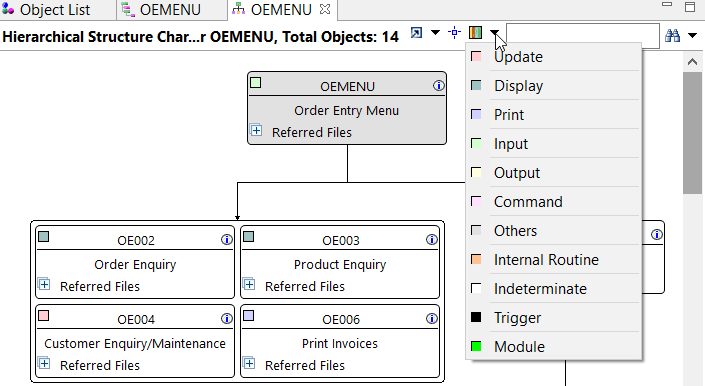


Fig. 2.7.9 – Legend expanded

The description of the HSC Legend is as follows:

* Update – This depicts the program that updates a file.
* Display – This depicts the program that uses a Display file.
* Print – This depicts the program that uses a Print file.
* Input – This depicts the program that uses an Input file.
* Output – This depicts the program that writes to a file.
* Command – This simply depicts a Command.
* Others – This refers to the Programs where they are referring to a file which is not Update/Display/Print/Input/Output.
* Internal Routine – This refers to the Synon-specific routines.
* Indeterminate – This depicts the programs where the usage cannot be programmatically determined.
* Trigger – This depicts the program which is a Trigger.
* Module – This depicts the program which is a Module.

Document Manager

One of the most powerful functionalities of X-Analysis is its ability to automatically generate structured documentation. The built-in Document Manager facilitates the generation of PDF/MS Word documents containing the system design information of selected objects.

First, mark the objects for documentation using the Mark for Documenter option from the context menu over OEMENU.

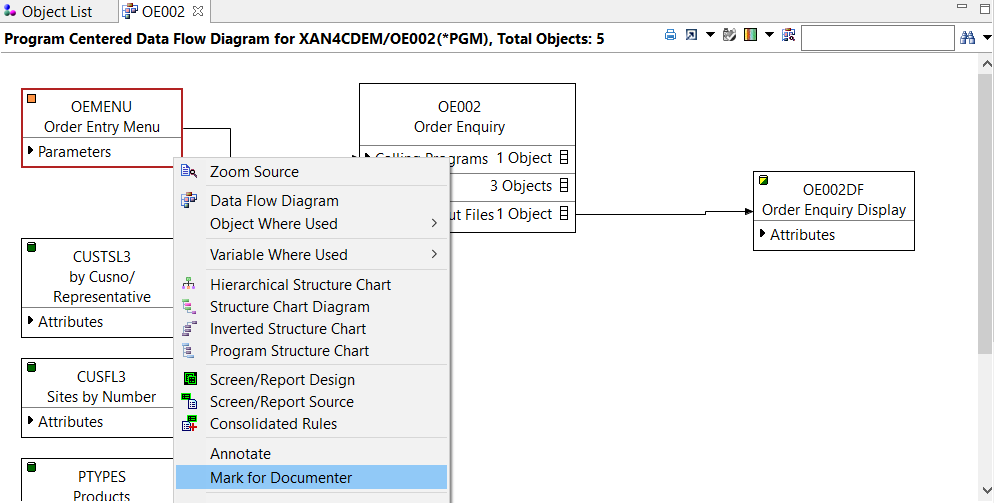


Fig. 2.8.1 – Mark for Documenter option

The Mark all for Documenter option is also available under the X-Analysis menu. Using this option will flag all objects displayed in the diagram.

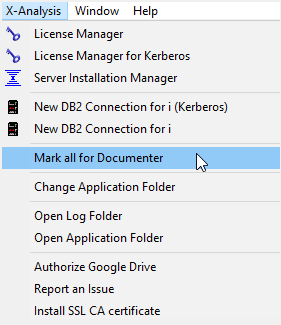


Fig. 2.8.2 – X-Analysis Menu

|  |  |
| --- | --- |
|  | The ‘Mark all for Documenter’ option is available for all diagrams and lists. |

Next, navigate back to the Object List window using the back arrow on the Eclipse toolbar and select CUSF.

Then, select the Mark for Documenter option from the context menu on CUSF.

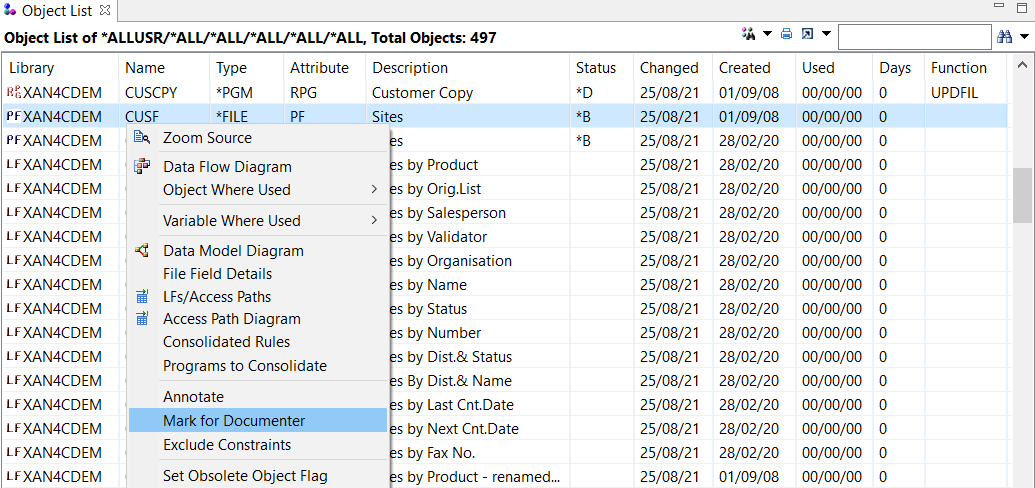


Fig. 2.8.3 – Mark for Documenter option on CUSF

Click the Documenter icon on the toolbar to begin the documentation process.

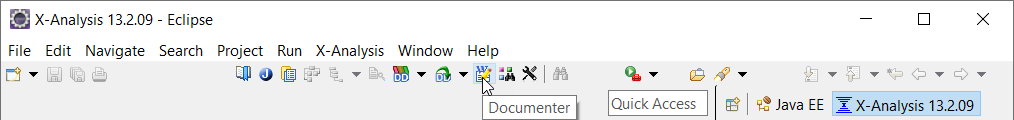


Fig. 2.8.4 – Documenter icon

This invokes the Document Manager dialog box.

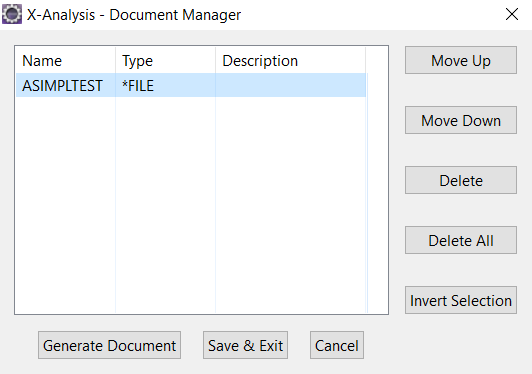


Fig. 2.8.5 – X-Analysis – Document Manager dialog box

The Document Manager dialog box lists all the marked objects for documentation. Use the Move Up / Move Down / Delete / Delete All buttons to organize Objects/Members on the documenter list.

The Invert Selection button on the Document Manager dialog box unselects the selected objects and selects all the un-selected objects.

After organizing the listed objects, click Generate Document. This invokes the Documenter Wizard as displayed on the following page.

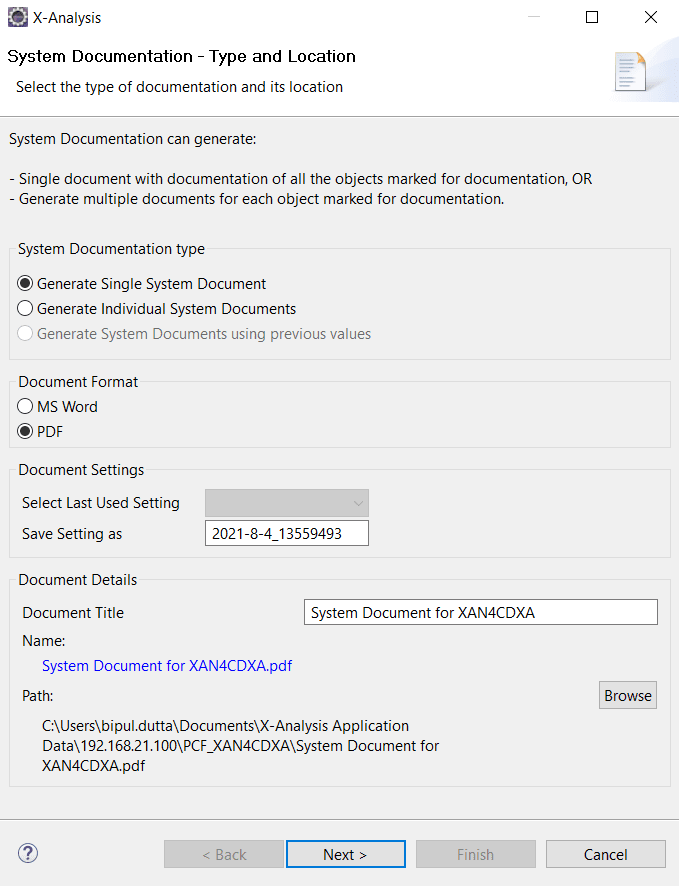


Fig. 2.8.6 – System Documentation Wizard – Type and Location

The Type and Location dialog consists of two sections:

* System Documentation type
* Document Details

The System Documentation type section provides options to generate the following types of documents:

* Generate Single System Document: Generates a single PDF/MS Word document.
* Generate Individual System Documents: Generates a PDF/MS Word document for each object.
* Generate System Documents using previous values: Generates system documentation using the options selected in previous instance of system documentation. (This option is only available for generated system documents.)

Select the Generate Single System Document option.

**Document Format**

* MS Word – Select this if it is required to generate the report in MS Word.
* PDF – Select this if it is required to generate the report in PDF.

**Document Settings**

* Select Last Used Setting – This field displays the last used setting and appears as disabled when the system documentation is generated for the first time.
* Save Setting as – This field is editable. The user can edit and save the setting with the required name and the same name can be used when the system document is generated subsequently.

The Document Details section specifies the Document Title, generated system document Name and its location. The user can choose a different file name for the generated PDF/MS Word document and a different location to save the generated document.

Click Next.

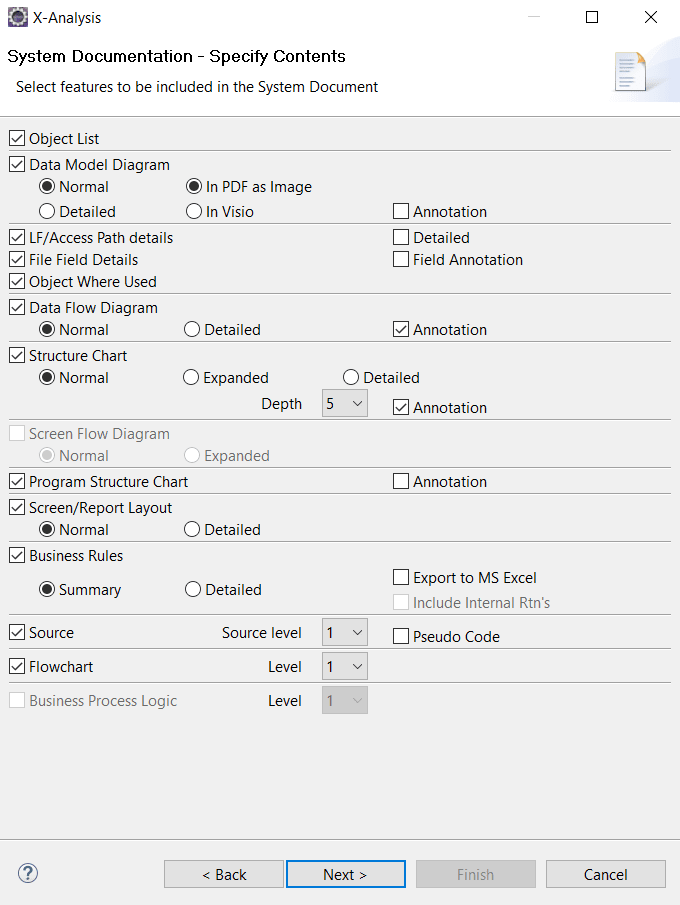


Fig. 2.8.7 – System Documentation Wizard – Specify Contents

Choose the features or attributes to be included in the System Document as displayed in the above screen.

Click Next.

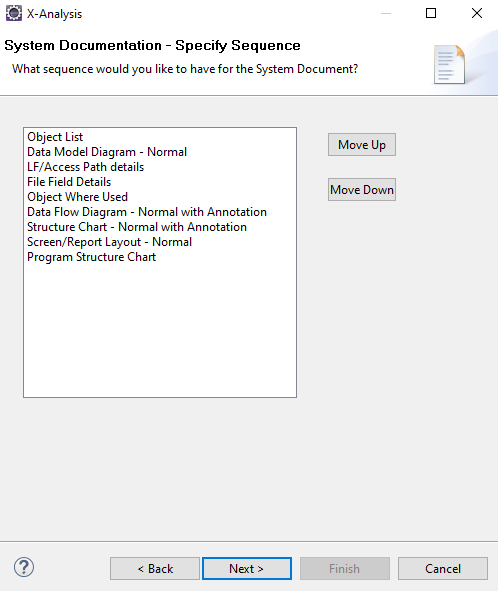


Fig. 2.8.8 – System Documentation Wizard – Specify Sequence

Confirm the sequence of the selected features and click Next.

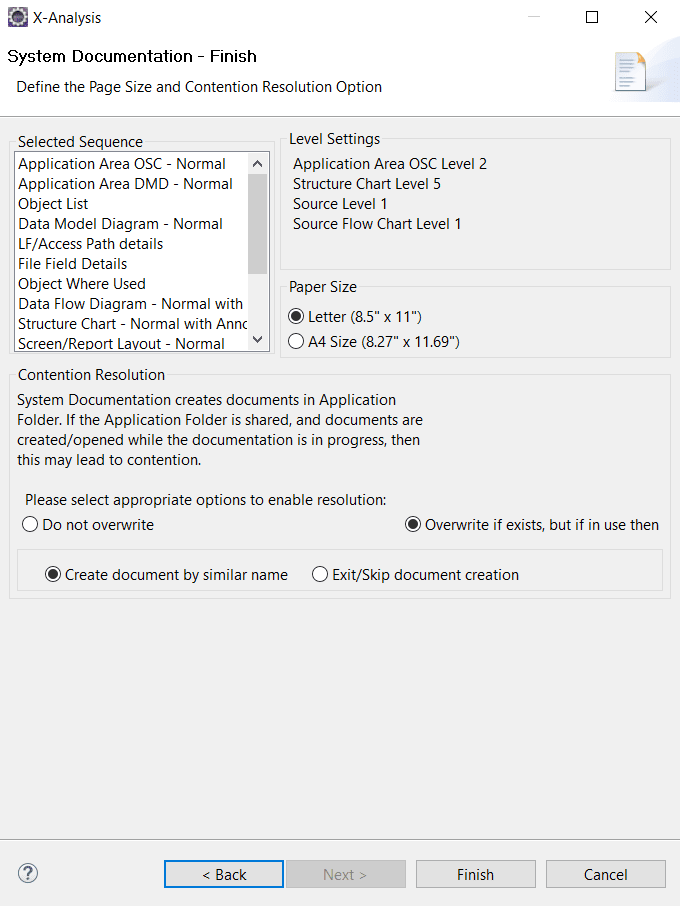


Fig. 2.8.9 – System Documentation Wizard – Finish

Choose the appropriate paper size and contention resolution options. Click Finish. This will end the configuration process and start generating the document.

After the process of generating the document gets completed, the following dialog box appears:

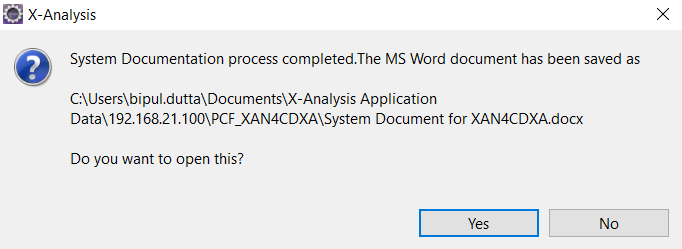


Fig. 2.8.10 – System Documentation Completion dialog box

Click Yes to open the generated document.

To view a previously generated document, select the Open Application Folder option from the X-Analysis menu.

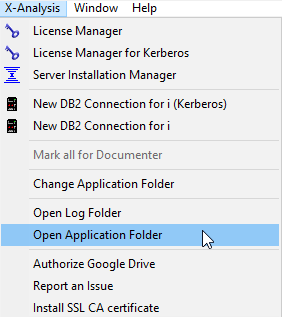


Fig. 2.8.11 – Open Application Folder option

This will open the Application Folder containing the previously generated documents. Double-click on a generated document to open it.

|  |  |
| --- | --- |
|  | X-Analysis also creates a log file in the Application Folder which logs what the ‘Document Generator’ does in the process of documenting the selected objects. |

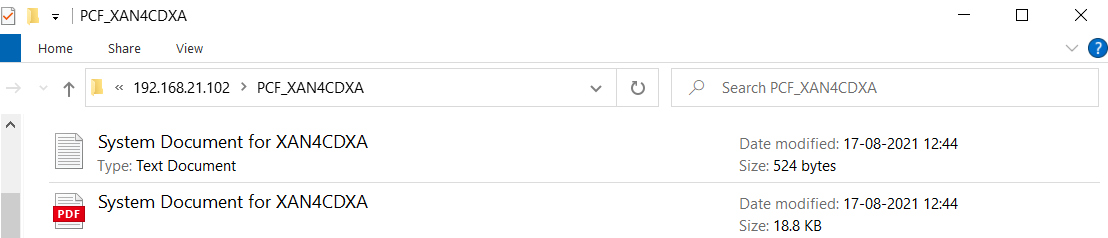


Fig. 2.8.12 – Application Folder containing the generated documents

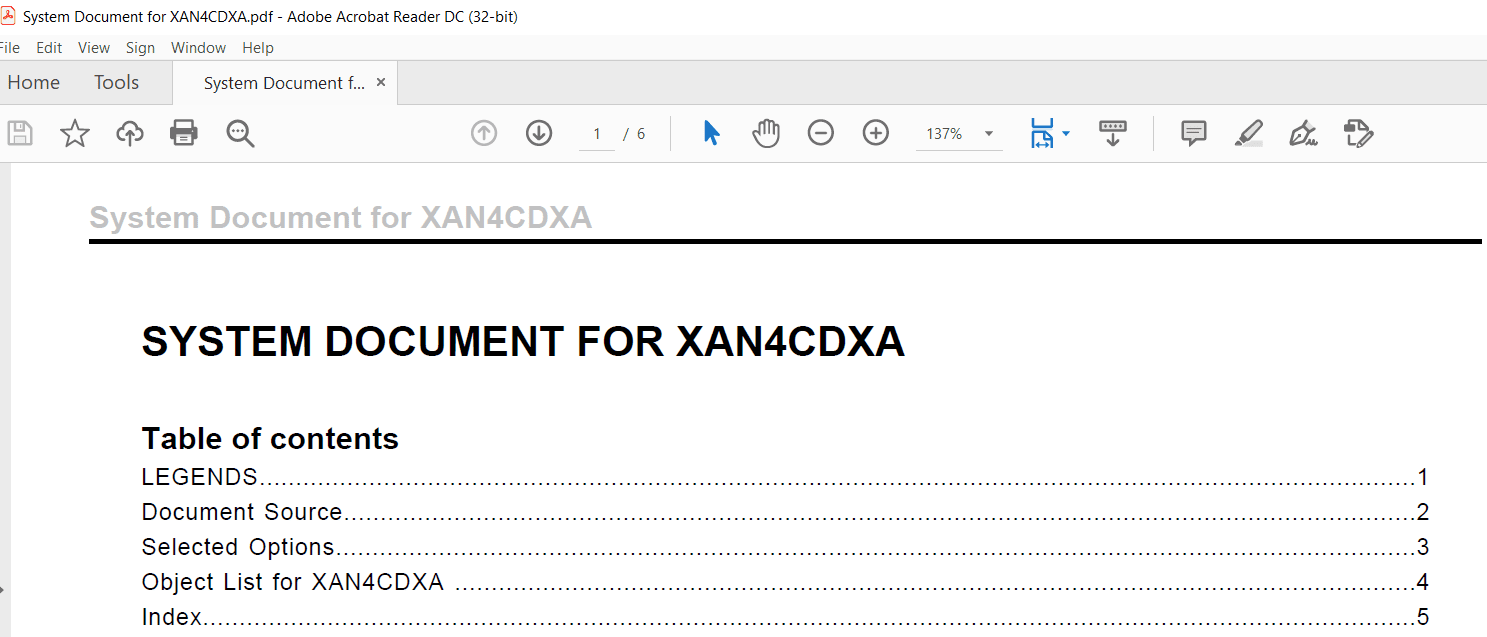


Fig. 2.8.13 – System document – TOC

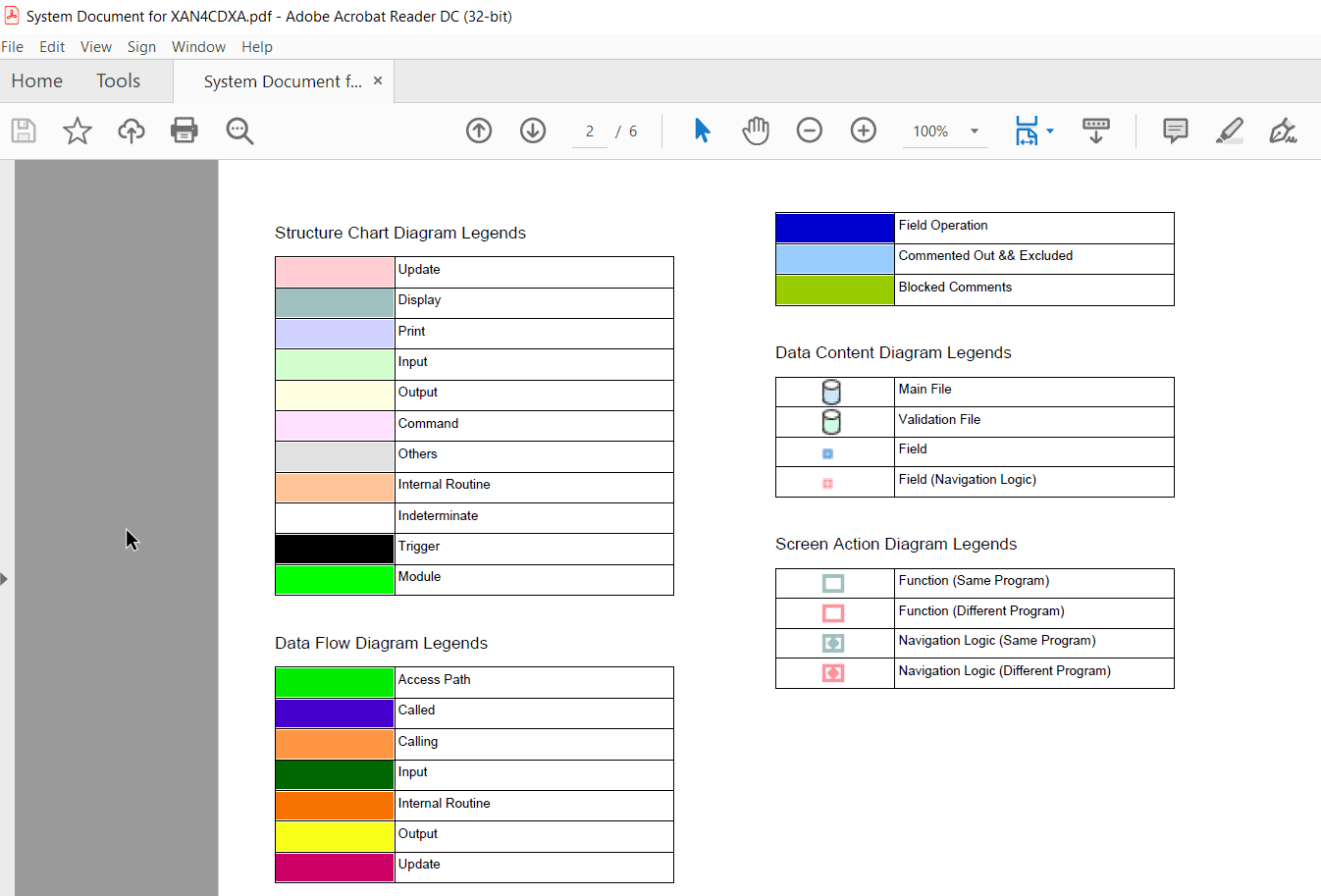


Fig. 2.8.14 – System Document – Legends

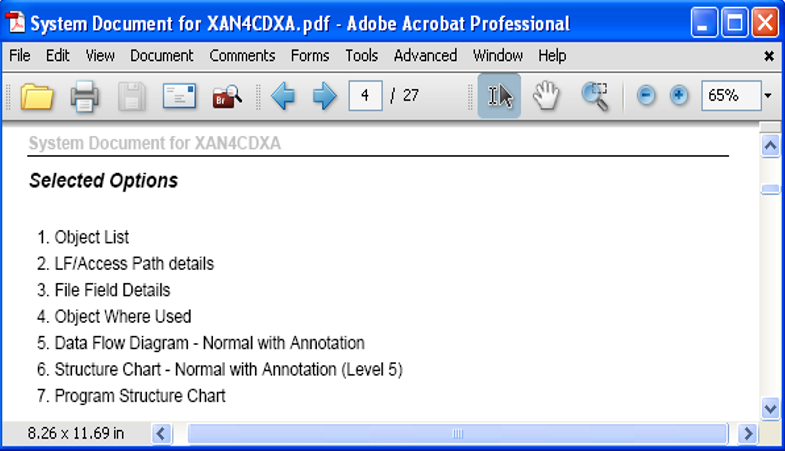


Fig. 2.8.15 – System Document – Selected Options

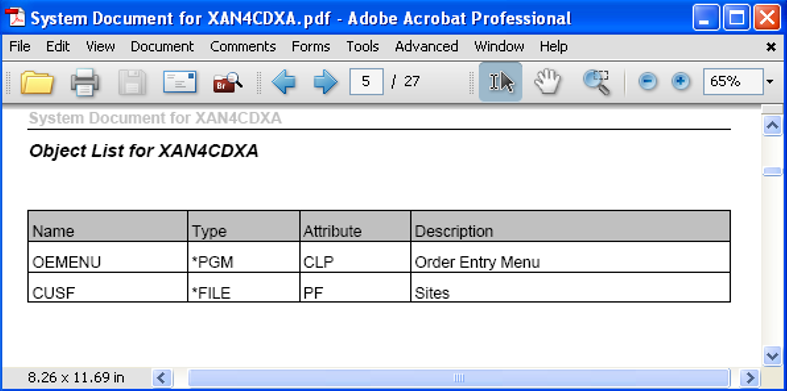


Fig. 2.8.16 – System Document – System Document – Object List

Source Browser

The Source Code of the application is the key to the process of analyzing, reviewing, refactoring, maintaining, or re-developing a system. X-Analysis supports some very advanced features in its Source Browser.

Double-click the Programs node to get the Object List for all programs available in the application library.

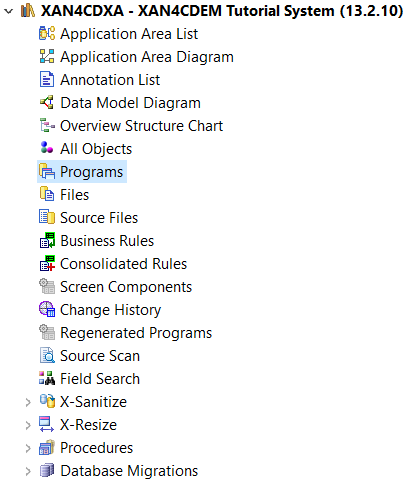


Fig. 2.9.1 – Programs node

The Work with Objects dialog box for \*PGM object type will be invoked.

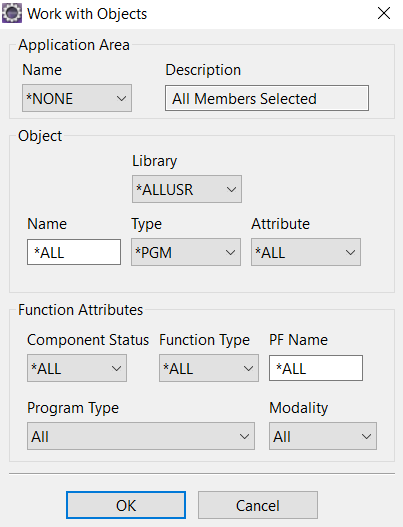


Fig. 2.9.2 – Work with Objects dialog box – Programs

Click OK to generate the list of program objects.

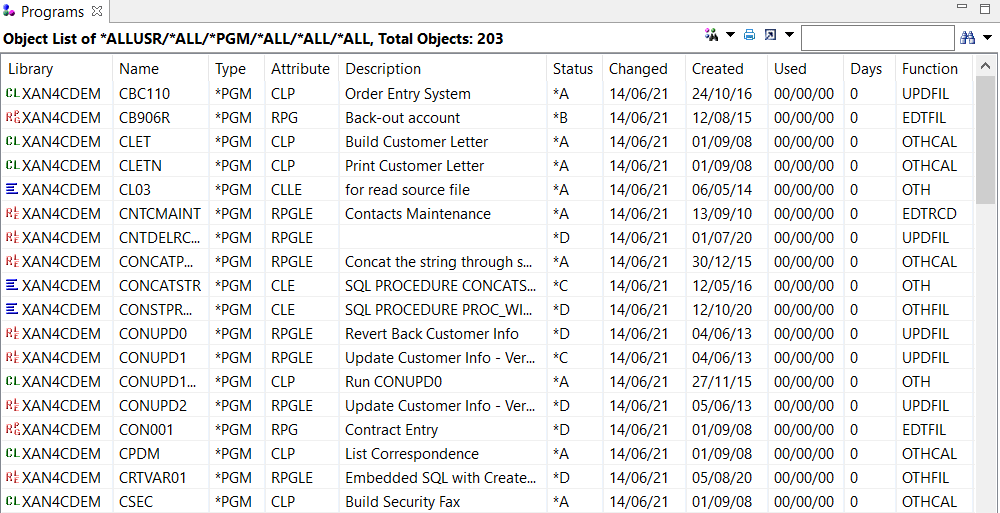


Fig. 2.9.3 – Object List – Programs

Double-click the CB906R program. This will call its Source Browser view.

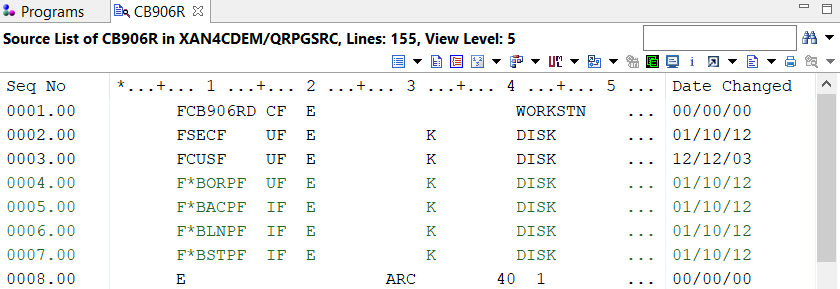


Fig. 2.9.4 – Source Browser view displaying the CB906R source

Source Browser Toolbar

The Source Browser toolbar is displayed below:

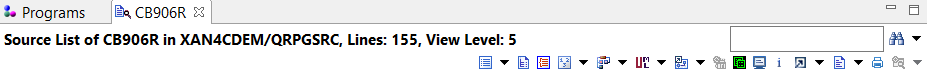


Fig. 2.10.1 – Source Browser toolbar

The toolbar consists of the following options:

* Source Options
* Source Outline
* Show Split Panel
* Pseudo Code
* Source Level
* Diagram Options
* Generate UML
* Flowchart
* Screen Components
* Screen/Report Design
* Zoom in Lpex (available on WDSc/RDi only)/SEU option (Set XA Preferences to select the appropriate option)
* More Info
* Export Options
* Download Source Code
* Print
* View Code

Source Options

Use the Source Options menu to view various source modes. The Normal Source Code is the default source view.

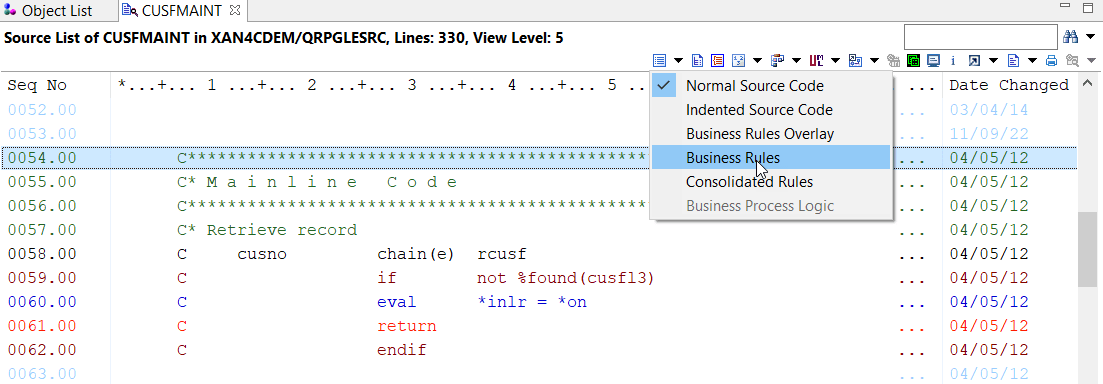


Fig. 2.10.2 – Source Options

|  |  |
| --- | --- |
|  | Source Options related to Business Rules is only available to customers who have purchased the X-Rules module. |

Indented Source Code view

Select the Indented Source Code option from the Source Options menu. This displays the Indented Source Code view.

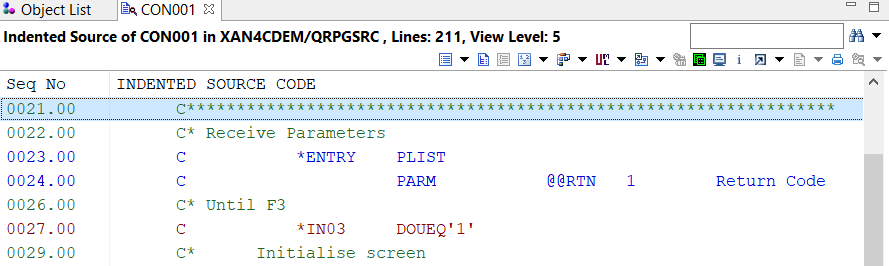


Fig. 2.10.3 – Indented Source Code

Select the Normal Source Code option to return to the normal source code view.

Source Outline

The Source Outline option displays an outline of the source member. This view lists all the subroutines available in the code.

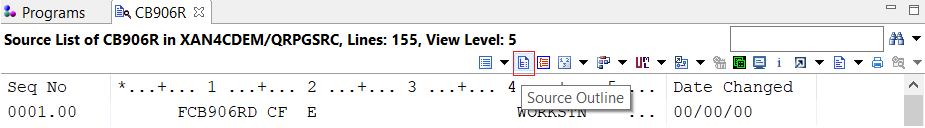


Fig. 2.10.4 – Source Outline option

Double-click any of the listed subroutines. This positions the cursor on the subroutine inside the source browser.

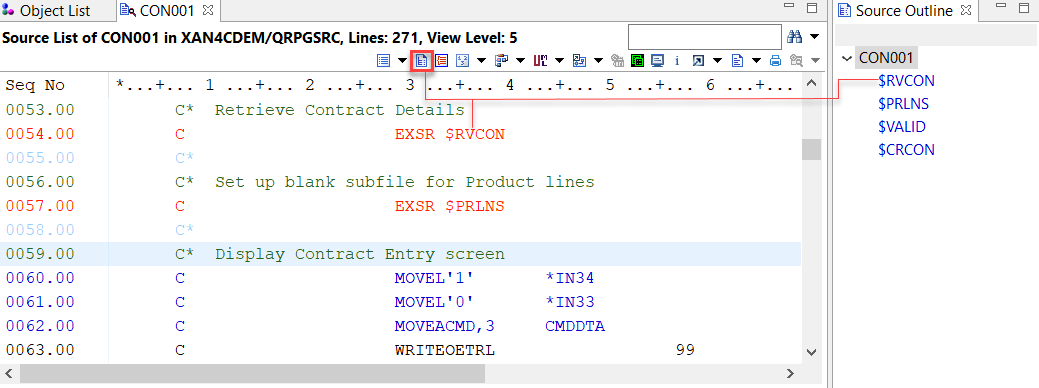


Fig. 2.10.5 – Source Outline view of CON001

Pseudo Code

The Pseudo Code (or the structured English view) is meant to help a non-technical user understand the code structure better. Click the Pseudo Code option on the toolbar.

This invokes the Pseudo Code view.

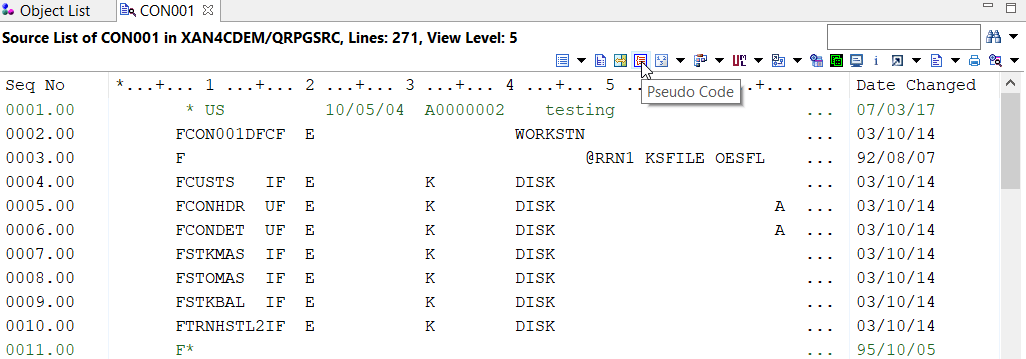


Fig. 2.10.6 – Pseudo Code option

The Pseudo Code icon has a toggle behavior. Click it again to return to the Normal Source Code. Use the X-Analysis ‘General Preferences’ option to set the Pseudo Code Indentation Depth level (0-4).

Source Level

The Source Level icon is used to access different view levels to view a specific type of source lines. It enhances the understanding of the code. The default source level is ‘5’.

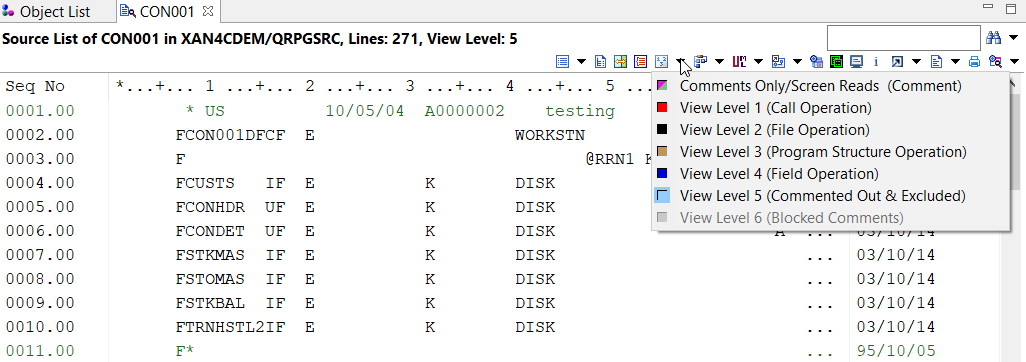


Fig. 2.10.7 – Source Levels

For instance, switching the source level to View Level 1 restricts the Source List from displaying comments and only allows for call operations.

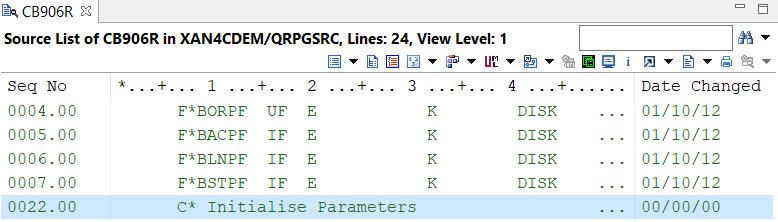


Fig. 2.10.8 – Source List at View Level 1

* View Level 2 includes File Operations.
* View Level 3 includes Program Structure Operations.
* View Level 4 includes Field Operation.
* View Level 5 includes the Commented Out & Excluded codes.

Flowchart

One way to view a program source code is to check the execution model of that program through the Data Flowchart. Select the Flowchart option from the source browser.

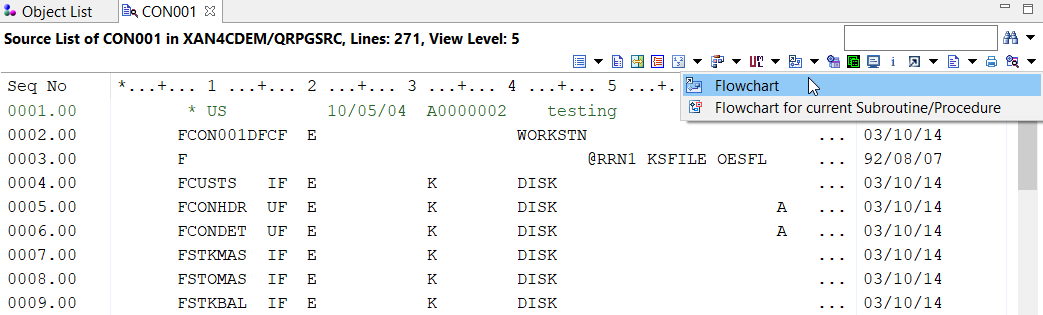


Fig. 2.10.9 – Flowchart option

This generates the program flowchart in MS Visio, as shown in the following page.

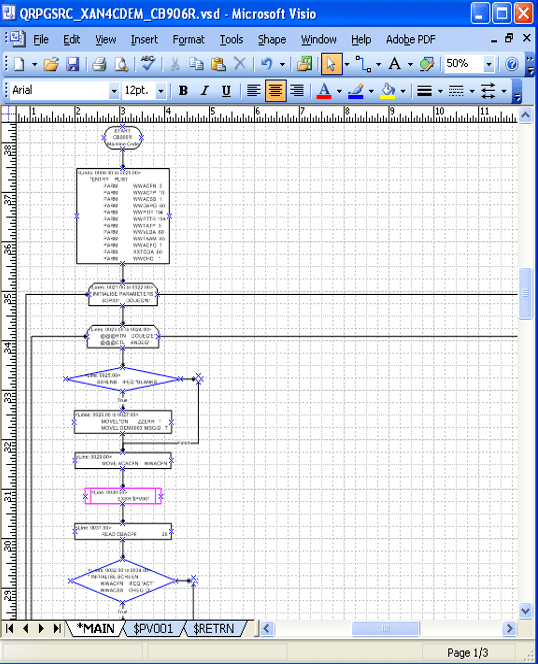


Fig. 2.10.10 – Flowchart in MS Visio

Note that the Flowchart option creates an individual flowchart for each subroutine. The first page of the flowchart displays the processing for the Main Procedure. From the Main Procedure, the user can jump to any of the subroutines available in the program.

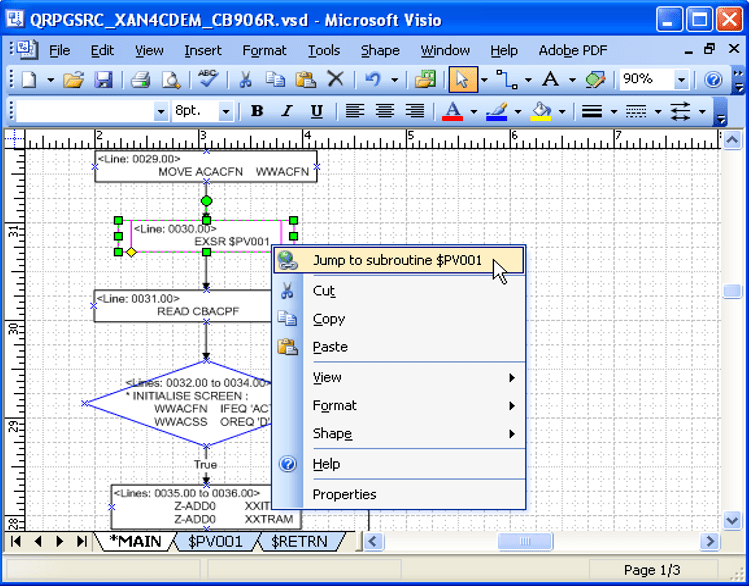


Fig. 2.10.11 – Jump option on the Flowchart

As shown above, click the Jump to subroutine $PV001 option to switch to the flowchart for the subroutine $PV001.

The related screen is shown on the following page.

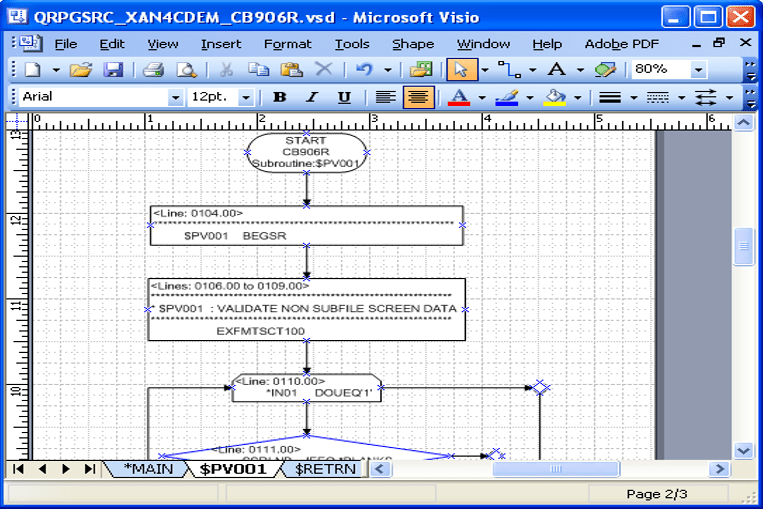


Fig. 2.10.12 – Subroutine Flowchart

The RPG/CBL flowchart has special handling for the GOTO/CAB opcode. Logically GOTO/CAB forces the logic to jump to another part of the code, with no return. Hence, the flowchart does not show a connection coming out of GOTO/CAB and shows the flow as abruptly terminated.

|  |  |
| --- | --- |
|  | GOTO/CAB as well as the TAG lines are represented by a circle in the flowchart. |

Other Options on the Source Browser

There are several other options available on the source browser toolbar. To understand these, open the source browser for the program, OE008.

Locate and select OE008 from the Object List. Right-click, to bring up the context menu and then select the option, Zoom Source from it.

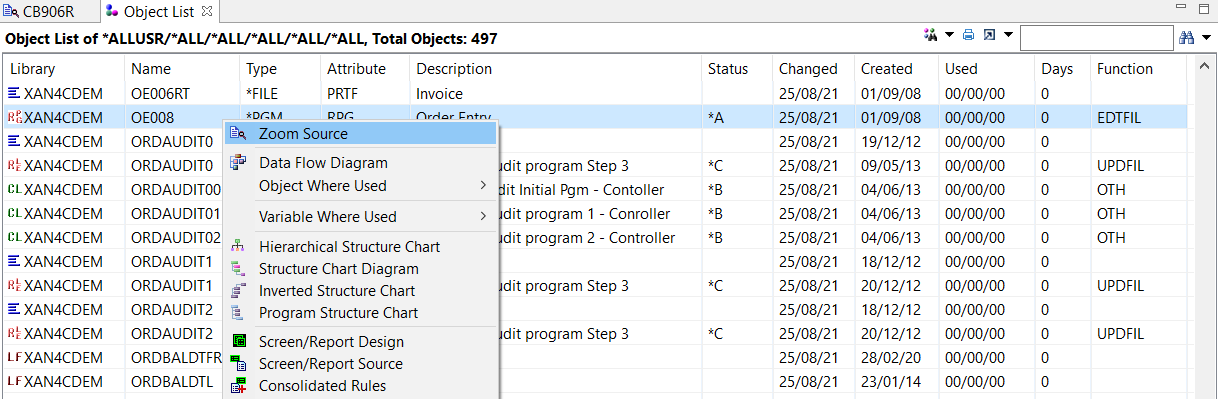


Fig. 2.11.1 – Context menu on OE008 – Zoom Source option

This brings up the Source Browser.

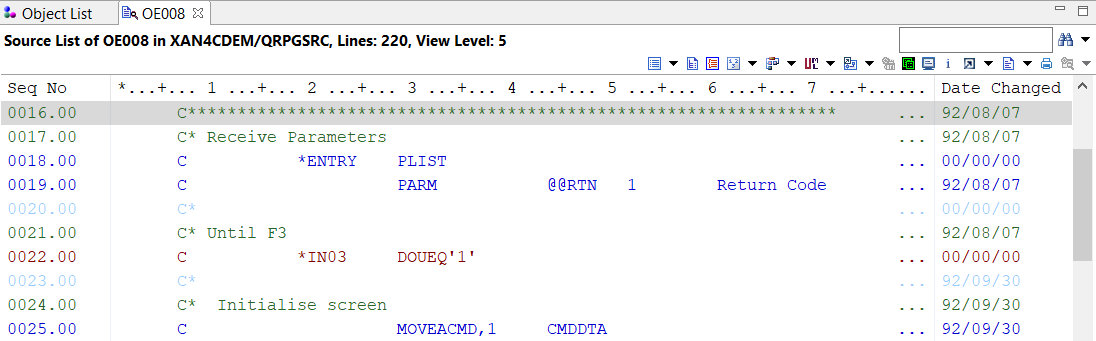


Fig. 2.11.2 – Source Browser for OE008

Other important options available on the source browser are:

Member X-Reference

On the source browser for OE008, select the line number 0022.00 with indicator \*IN03. Right-click for the context menu on \*IN03 and select the Member X-Ref option.

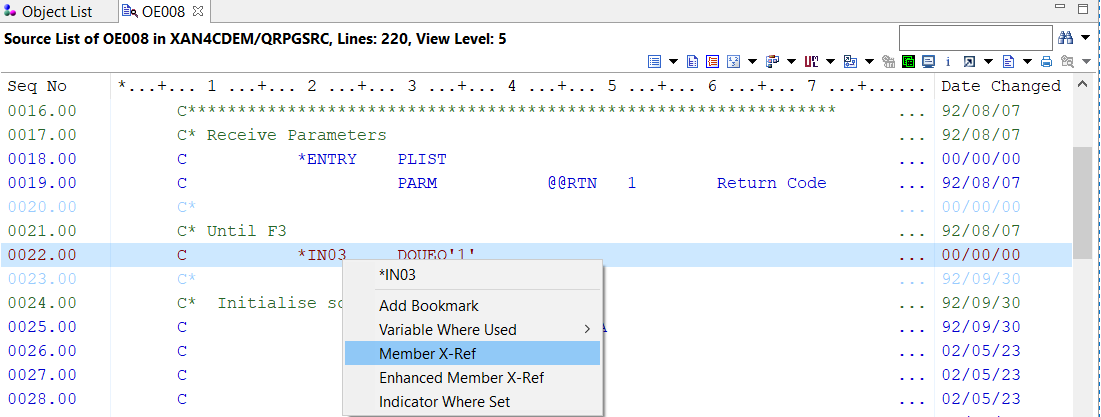


Fig. 2.11.3 – Member X-Ref option

This displays the Member X-Ref view for the selected variable, \*IN03. The Member X-Ref view lists all source lines where the selected Field/Variable has been used/referenced.

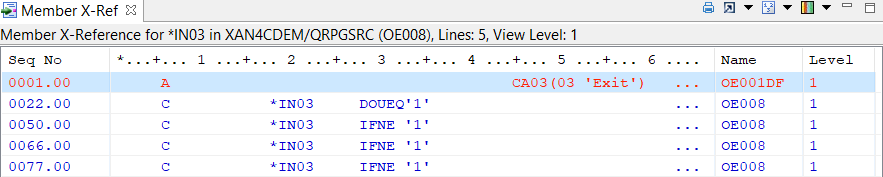


Fig. 2.11.4 – Member X-Ref window

|  |  |
| --- | --- |
|  | User may copy the variable name to the clipboard by simply clicking the option depicting variable name in the context menu as shown below: |

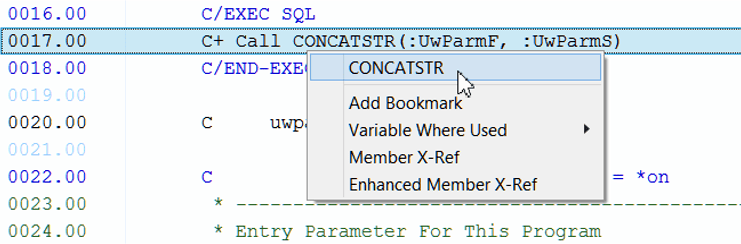


Fig. 2.11.5 – Copy Variable Name

Jump to Subroutine

Jump to a subroutine by using the source browser. Double-click one of the subroutines in the source browser for OE008.

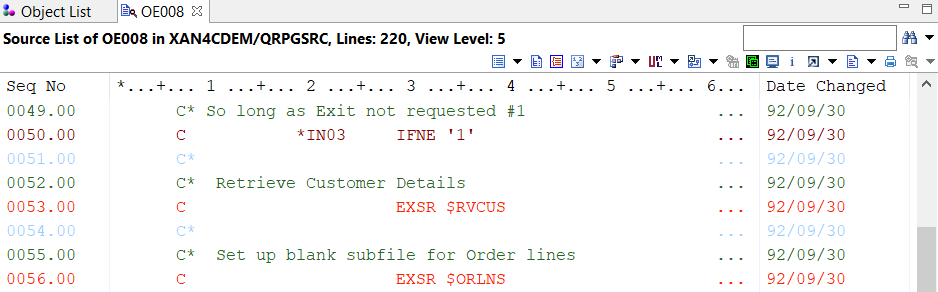


Fig. 2.11.6 – Select subroutine for Jump action

Double-click $RVCUS to position the cursor at the beginning of the subroutine, $RVCUS.

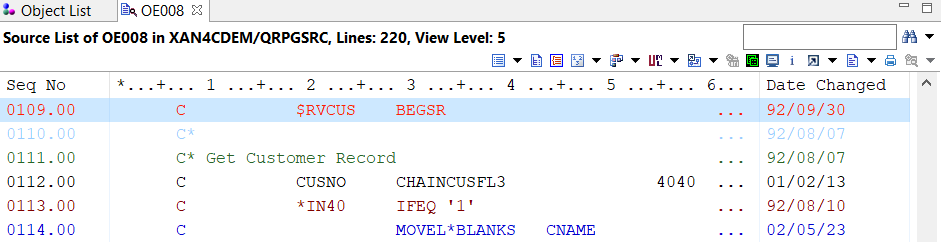


Fig. 2.11.7 – Highlighted first line of the subroutine

Click the back navigation arrow to return to the line where the $RVCUS subroutine is called.

Program Structure Chart

The Program Structure Chart option provides a different way of looking at the program structure.

A PSC graphically displays the sequence of calls in the program. Click the option from the Diagram Options dropdown menu on the Source Browser toolbar.

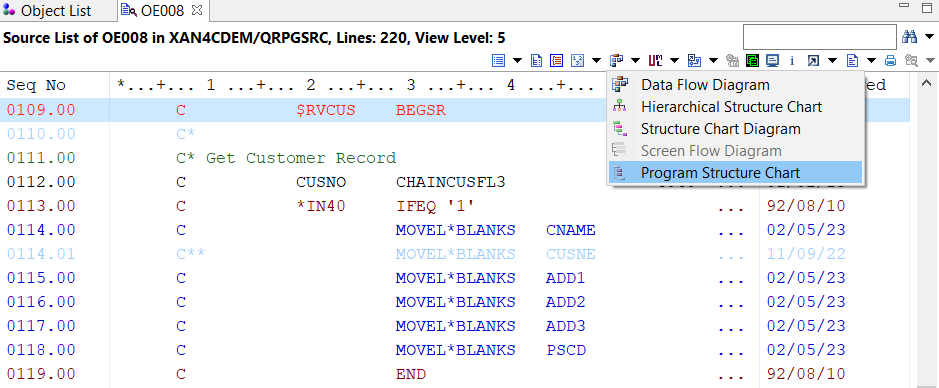


Fig. 2.11.8 – Program Structure Chart option

Selecting this option will display a diagram that displays the PSC, for that program.

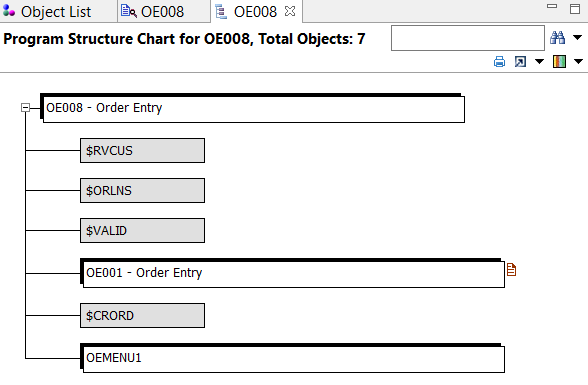


Fig. 2.11.9 – Program Structure Chart for OE008

The PSC presents each subroutine as a small grey rectangle displaying the name of each subroutine. The object types that are, the Function Type, which determines the coloring of all other shapes, Programs, Modules, and Service Programs.

|  |  |
| --- | --- |
|  | (Other than the subroutines, boxes are the same size of a program element as used in the Structure Chart Diagram). |

On the PSC, double-click a subroutine. This initiates the Source Browser for the OE008 program with the subroutine selected.

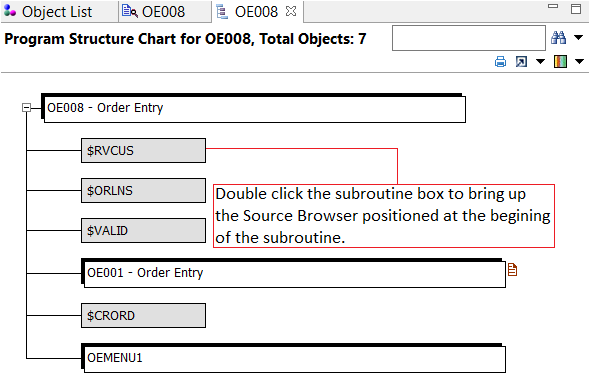


Fig. 2.11.10 – PSC with the selected subroutine

The source browser for OE008 with the selected $RVCUS subroutine is shown below:

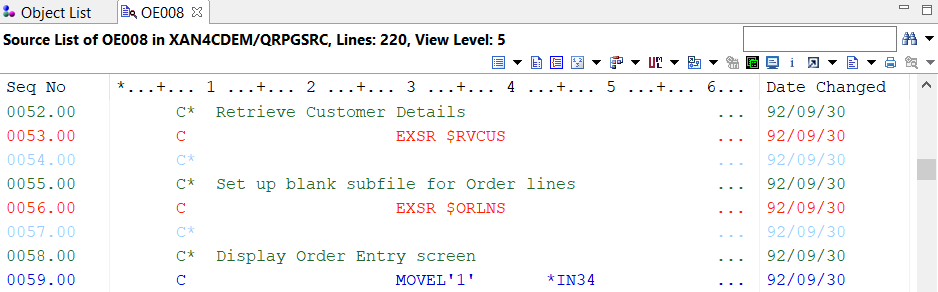


Fig. 2.11.11 – Source Browser for OE008 with the selected subroutine

Zoom Source to Files defined in File Specification

The selected program OE008 has files defined in the File Specification.

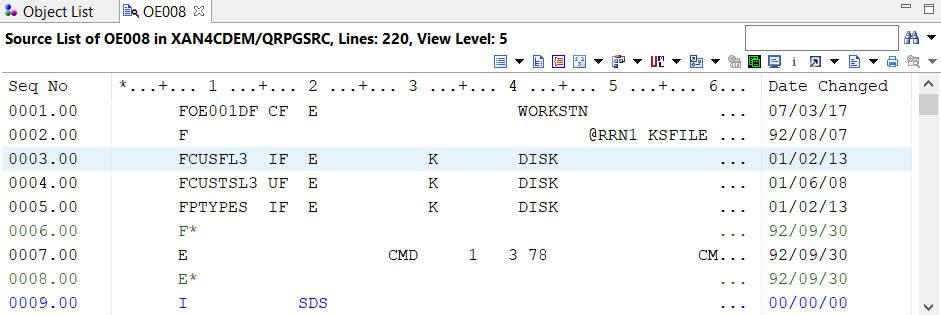


Fig. 2.11.12 – Files listed in OE008 source

Browse the source codes of the associated files by double-clicking them. Alternatively, select any file from the source browser and opt for the Zoom Source option from the context menu. In this case, CUSFL3 is selected.

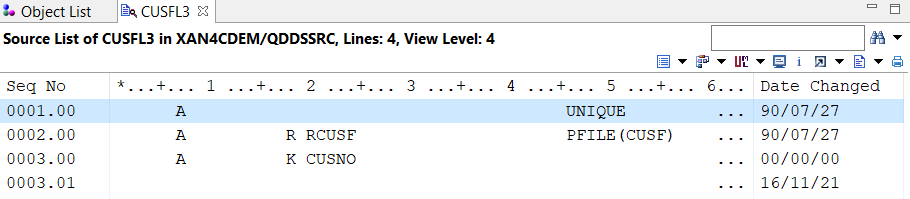


Fig. 2.11.13 – Source Browser for CUSFL3

The following steps demonstrate the drill down to the source code of the associated PFILE (CUSF) of the LF and CUSFL3.

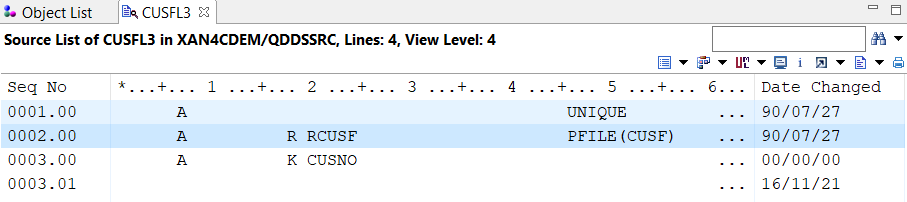


Fig. 2.11.14 – PFILE selected on the Source Browser for CUSFL3

Double-click CUSF to invoke the Source Browser for it.

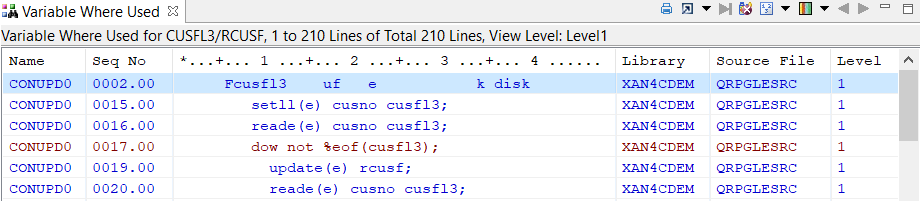


Fig. 2.11.15 – Source Browser for CUSF

The Source Browser display helps to navigate and drill down to various related features, thus, facilitating the overall browsing and understanding of the application.

Bring up the OE008 source browser to become familiar with some more features.

Screen/Report Design

Most of the application programs interact either with the display files (DSPFs) or the printer files (PRTFs). The Source Browser provides a feature to view the layout of the associated display/printer files.

This feature does not require the execution of an actual program.

Click the Screen/Report Design icon available on the Source Browser.

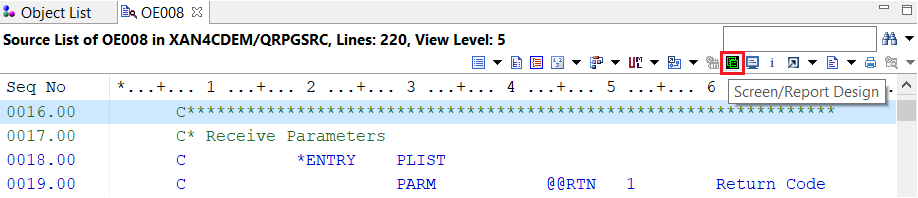


Fig. 2.11.16 – Screen/Report Design option on the Source Browser

This invokes the Screen/Report Design window for the OE008 program.

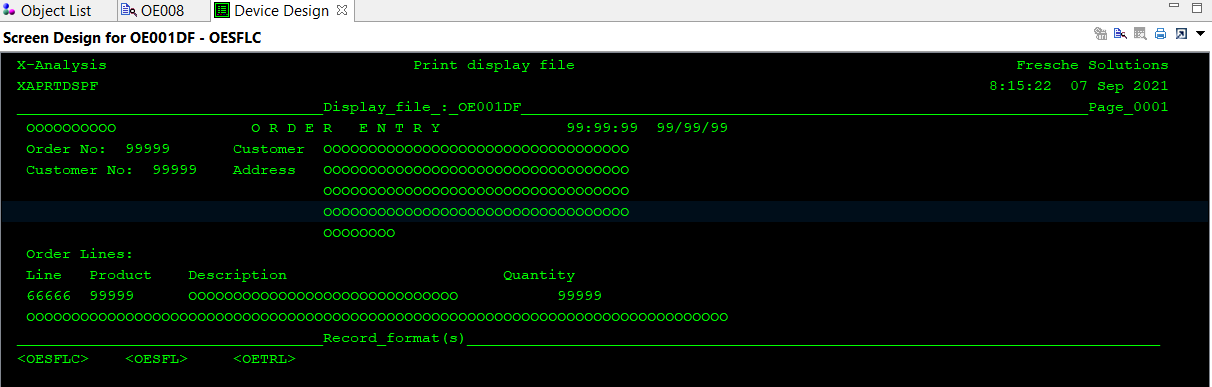


Fig. 2.11.17 – Screen/Report Design window for OE008

Navigate back to the Source Browser for OE008 using the back navigation arrow.

Zoom in SEU/Lpex Editor

**Lpex Editor**

Use the Zoom in Lpex option to make changes to the source code. Click the Zoom in Lpex icon available on the Source Browser toolbar.

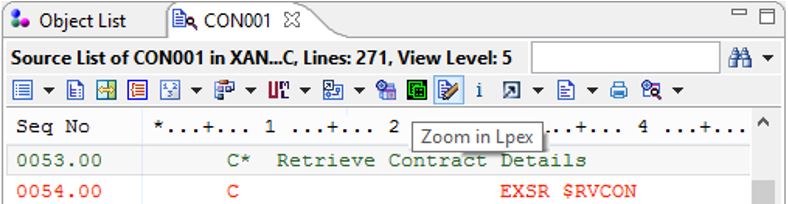


Fig. 2.11.18 – Zoom in Lpex option on the Source Browser

This invokes the Lpex editor from within the X-Analysis Perspective.

SEU

Make changes to the source code using the SEU option. Click the SEU icon available on the Source Browser toolbar.

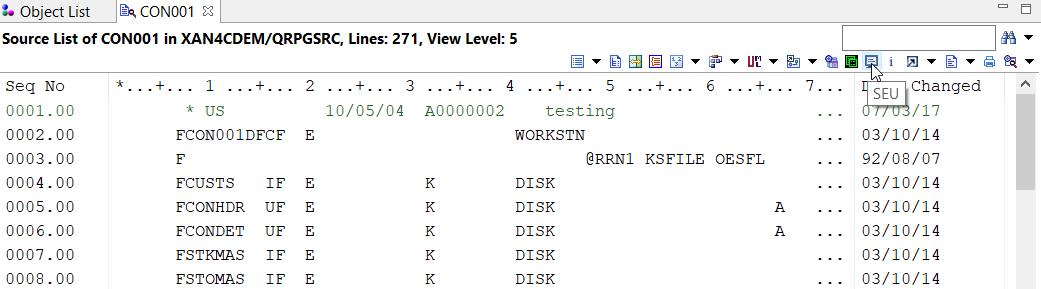


Fig. 2.11.19 – SEU option on the Source Browser

|  |  |
| --- | --- |
|  | ‘Zoom in Lpex’ is a default option for all IBM Rational products having an RSE plugin. For all other Eclipse products, SEU (5250 Emulator) is the default option. The user can change the default settings using the X-Analysis ‘General Preferences’ option. Creating an SEU (5250 Emulator) session requires the XAN4SEU IBM i user. Refer to the Appendix for creating the XAN4SEU user. |

The X-Analysis Plugin comes with default preference settings. To change the preference settings, start IBM's RDi 9.0 and above or Eclipse (see <https://www.myfreschesolutions.com/nexus/ui/docs?root=0&docId=590&kbnum=15954> for supported Eclipse version).

Select Window > Preferences to display the Preferences dialog.

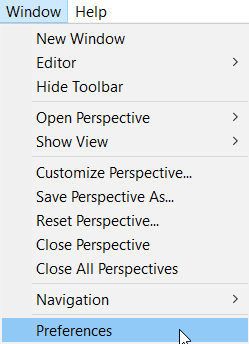


Fig. 2.11.20 – Preferences option

To view / modify various X-Analysis Preferences, select the X-Analysis node.

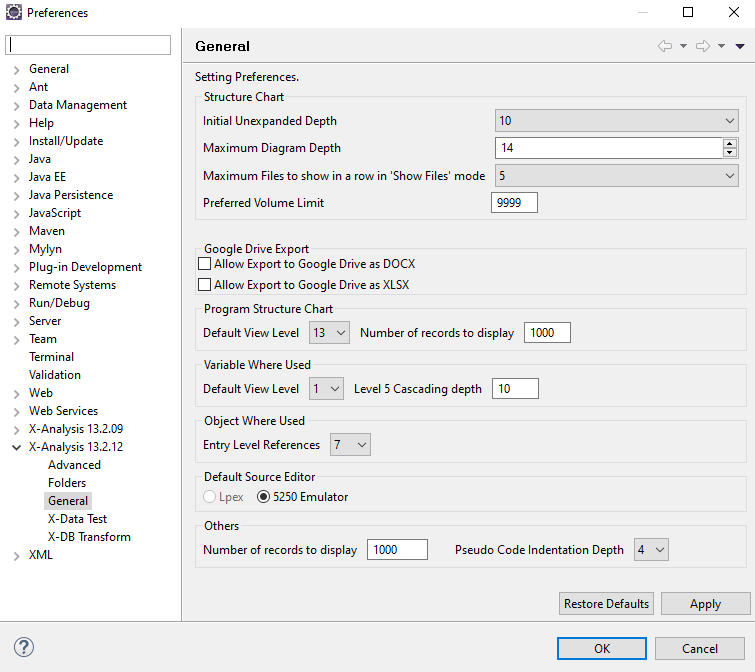


Fig. 2.11.21 – General X-Analysis Preferences Settings

Relational Data Model Extraction for IBM i

Configuring the design of an application depends on an accurate data model.

X-Analysis features the advanced capability of automatically extracting data model from an RPG/COBOL-based application. The data model derived through X-Analysis is viewable as an interactive diagram. The software extends the benefit of perusing the file-centric or application area-specific filtering of the diagram.

This section highlights the importance of understanding an application design and describes how the Data Modelling solution set helps in deciphering complicated codes of databases. Thereby, making data maintenance and development work a less complicated process.

|  |  |
| --- | --- |
| **Feature** | **Brief Description** |
| Data Model Diagram | Diagram which displays file relationships for an application or an application area. |
| LFs/Access Paths | Displays all access paths related to a specific file. |
| File Field Details | Displays the field details of a specific file. |
| Access Path Diagram | Diagram which displays all access paths belonging to a specific file. |
| View Data | Displays actual records held in the PFs. |

The Data Modelling solution set is an add-on module dedicated to data model extraction. Before analyzing a legacy system, it is important that the user must understand the application design. An analyst looks for the Data Model information of the legacy system, which forms the foundation of the application design.

In the context of X-Analysis, the term Data Model refers not only to the physical model of a database design but also to the foreign key or the relational model.

X-Analysis automatically derives an explicit physical and logical data model by analyzing both the actual data and the programs that use that data, and thus verifies the existence of any potential cross-file relationships. It performs an integrity check to ensure that all data from the dependent file validly refer to the data records from the owning file.

A complete data model that accurately describes all possible relationships among the files is essential for productive maintenance and development work. Such a model also provides a foundation to automate critical data administration tasks such as referential integrity testing and test data extraction.

The graphical representation of automatically generated Data Model information in X-Analysis is a Data Model Diagram. The Data Model Diagram displays file relationships for an application or an application area.

Generating Data Model Diagram

Double-click the Data Model Diagram node under the cross-reference library, XAN4CDXA.

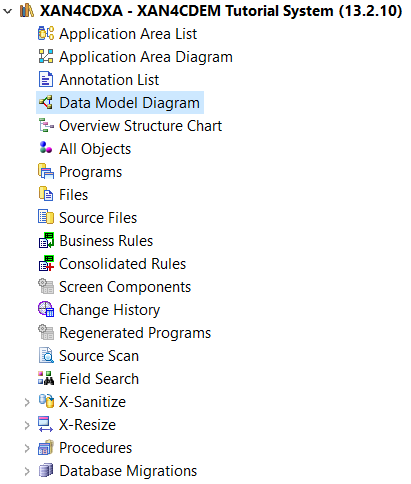


Fig. 3.1.1 – Data Model Diagram option

This invokes the Data Model Diagram dialog box as shown below:

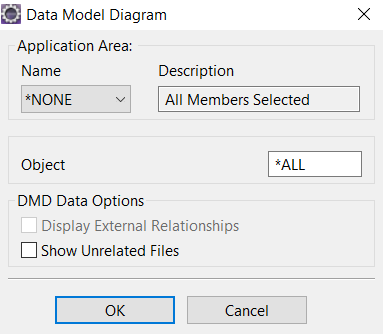


Fig. 3.1.2 – Data Model Diagram dialog box

It prompts the user to enter the following information:

1. Application Area Name (leave it as \*NONE)
2. Object Name (leave it as \*ALL)
3. DMD Data Options to Display External Relationships (available only when the DMD is opted on an Application Area) or Show unrelated files.

Click OK to generate the Data Model Diagram for the entire application.

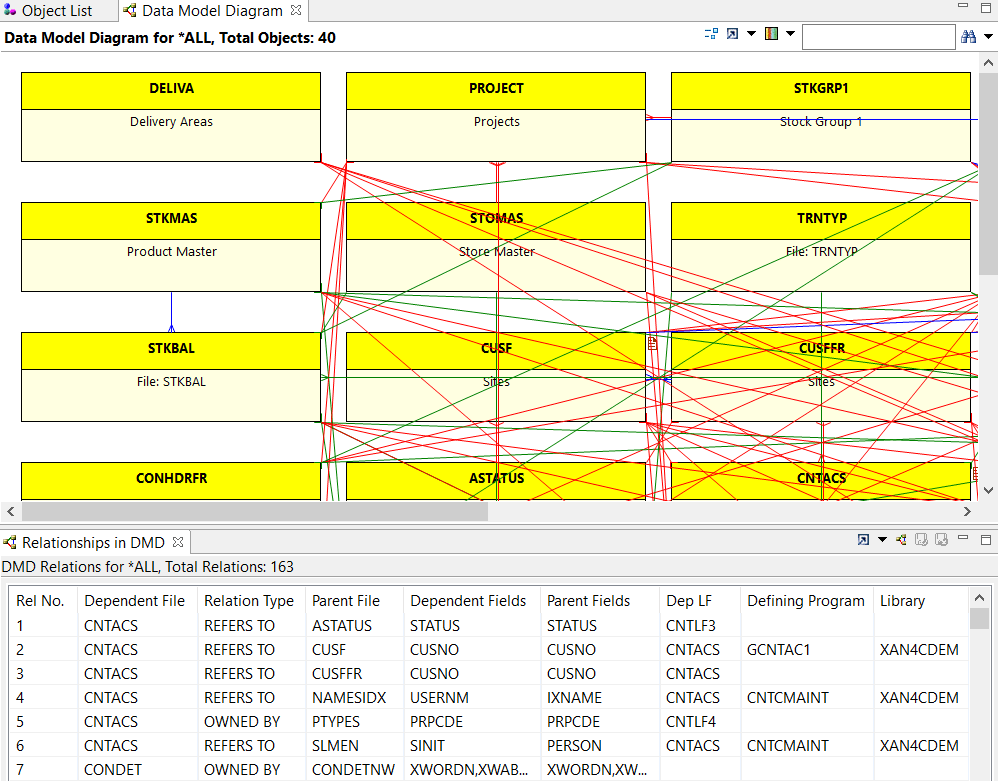


Fig. 3.1.3 – Data Model Diagram for XAN4CDXA

The Data Model Diagram or DMD displayed above consists of two halves. The upper half is the DMD and the lower half is the Relationships in DMD view.

The color of the lines across the diagram means:

* Red line represents 'Refers to' relation
* Blue line represents 'Owned by' relation
* Black Line represents ‘Multiple’ relations

|  |  |
| --- | --- |
|  | The unrelated member is depicted in faded brown color which distinguishes it from the related members. |

When the number of files and relations are high, then two parallel lines may show as overlapped. When the lines are too close, it may seem that both connect to the same set of files, but in reality, maybe connecting to a different file in the same Row/Column.

The tail of the arrow depicts the dependent file whereas the head represents the Parent file. As displayed in the above DMD for XAN4CDXA, CNTACS is a Dependent file whereas ASTATUS is the Parent file.

The DMD toolbar has a Legend icon. The different colors on the Legend drop-down convey the type of association represented by the colored lines connecting various files.

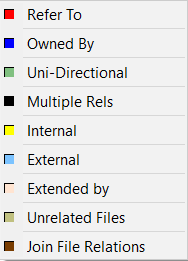


Fig. 3.1.4 – DMD Legend

With the help of the Legend, the user can easily understand the relations displayed on the DMD.

The Relationships in DMD view displays the Files Connection Details. The Relation Type column states the relationship between the files in the Dependent File column and the files in the Parent File column. The Dependent Fields and the Parent Fields columns list the fields which relate to the Dependent File and the Parent File, respectively.

The Relationships in DMD list displays actual names of the listed Files and Fields. The list has a toggle button, Show Description, to switch the DMD Relations list to display the description of the listed Files and Fields.

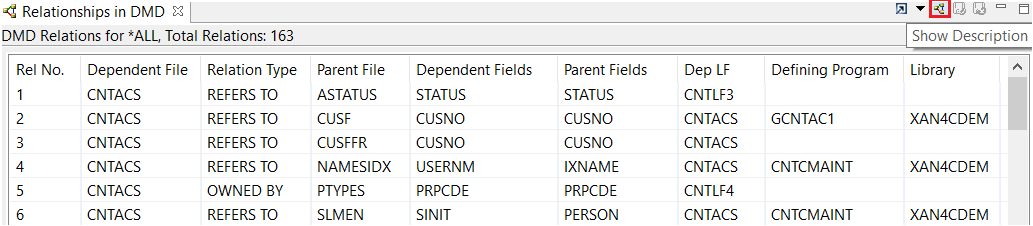


Fig. 3.1.5 – Relationships in DMD view with Show Description icon

Click the Show Description icon to see descriptive names for all the files and the fields.

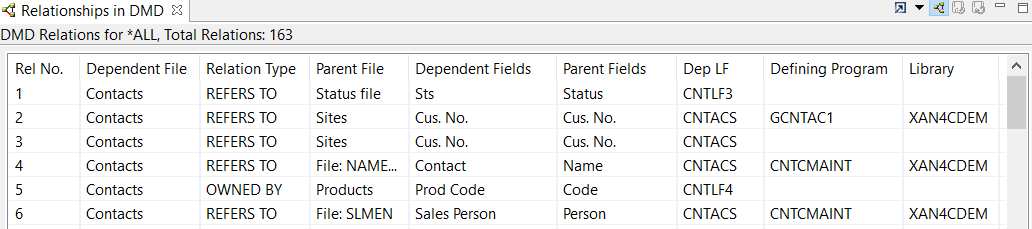


Fig. 3.1.6 – Relationships in DMD view with Description

Click the Hide Description icon to switch back the DMD Relations list to the actual names’ mode.

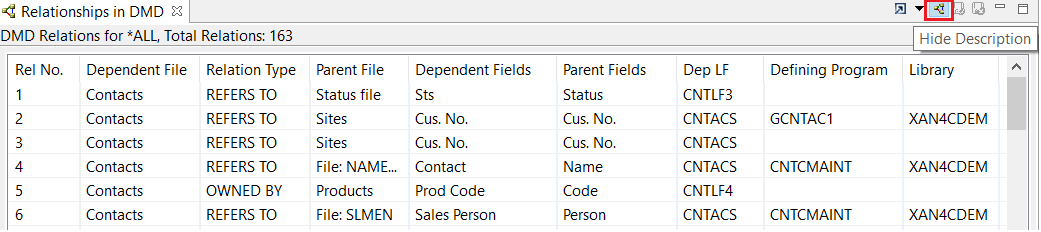


Fig. 3.1.7 – Relationships in DMD view with Hide Description icon

Let us consider another aspect of the DMD. A single-click on an object displays the references of that object with the other objects in the DMD.

Click STKBAL to view its references.

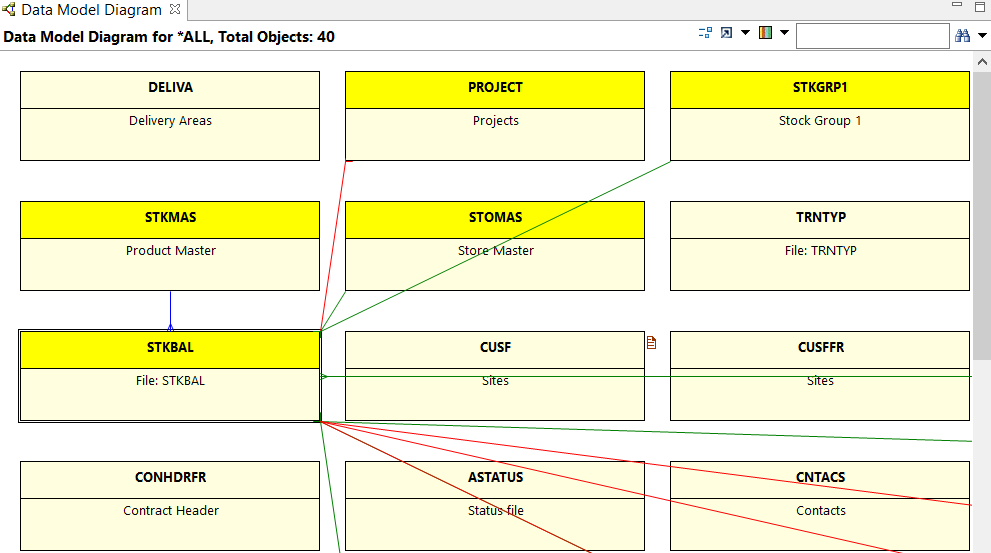


Fig. 3.1.8 – Data Model Diagram with the selected object, STKBAL

The screen above displays relations for STKBAL which helps the user to understand all relationships for the selected object.

Another way of understanding file relation for any specific file is to view a file-specific Data Model Diagram. A file-specific DMD displays the data model information for a given file only. Double-click STKBAL to get the file-specific DMD.

|  |  |
| --- | --- |
|  | The box size for all the files in the Data Model Diagram will increase to accommodate the long names when any of the following condition is true.   * The Display setting for the PC is set to more than 100% resolution. * At least one or more files are defined using a long name and Windows> Preferences>X-Analysis XX.X.X>Show Long Names for Files option is selected. * The Windows>Preferences>X-Analysis XX.X.X>Use Wider Boxes for DMD option is selected. |

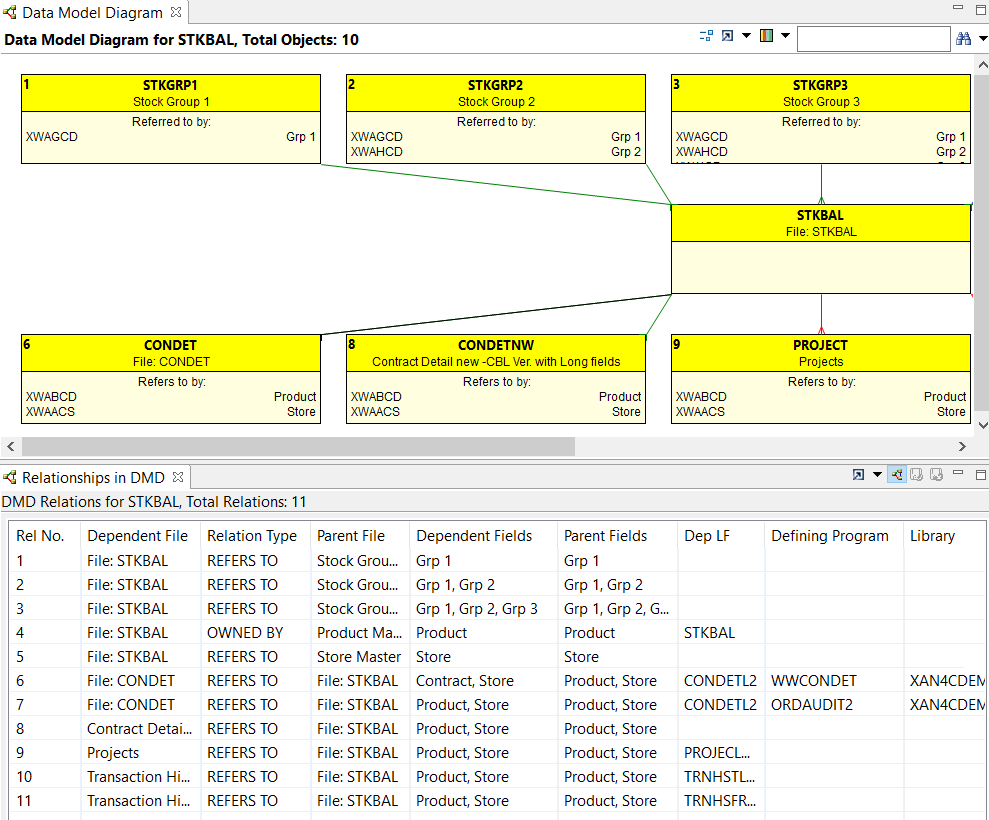


Fig. 3.1.9 – File-specific DMD for STKBAL

It is clear that the file-specific DMD only contains objects related to STKBAL. Furthermore, the Relationships in DMD list displays only those relationships related to STKBAL.

The numbers within the box indicate the sequence number, and this matches the number shown in the Relationships in DMD view. The matches are provided to assist the user to coordinate the diagram with its Relationships in DMD view.

The numbers shown in red color signify that the relationship is repeated, and that the two files are already related to another set of fields. On the DMD Relations list, a repeated Dependent file is suffixed with numbers \_2, \_3.

With the help of a file-specific DMD, the user can view relationships between any available files. These relationships can be understood properly after reading the View Data section.

LFs/Access Paths

Select the LFs/Access Paths option for a file from the DMD. The LFs/Access Paths option shows all access paths associated with a selected file.

Select CUSF, and then choose the LFs/Access Paths option from the context menu.

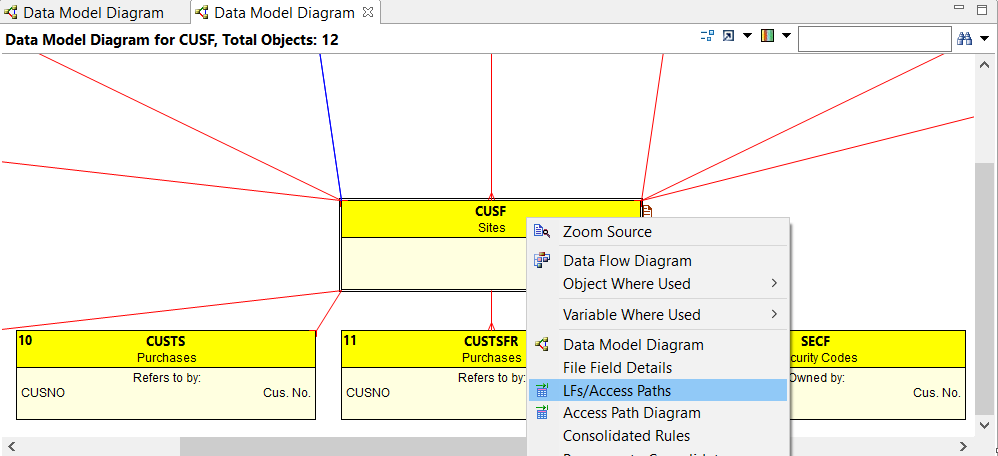


Fig. 3.2.1 – LFs/Access Paths option on DMD

This brings up the Access Paths list.

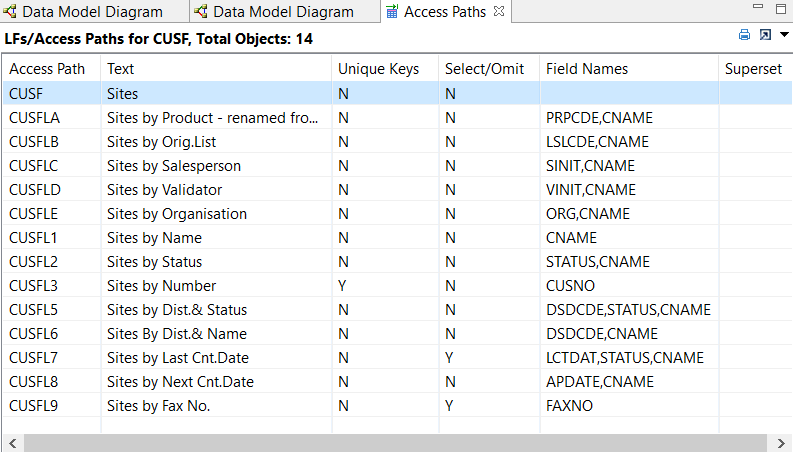


Fig. 3.2.2 – LFs/Access Paths for CUSF

To browse the source of any LF, double click on it or select the Zoom Source option from the context menu. Note that the information in the Superset column is related to the number of key fields in different access paths for the specific PF. In other words, superset against an access path in this panel means another file contains the key fields of this access path (in the same order) and some additional key fields.

|  |  |
| --- | --- |
|  | The LFs/Access Paths will now show the long name of Files and Fields on the LF/Access Paths list when the long names exist and the Windows>Preferences>X-Analysis XX.X.X>Show Long Names for Files option is selected. |

File Field Details/FFD

The user needs more details of a file to get a full understanding of the data model of an existing application. A file may consist of several fields. The File Field Details option helps the user here. Select the option from the context menu to get an easy access to field details of a selected file, CUSF.

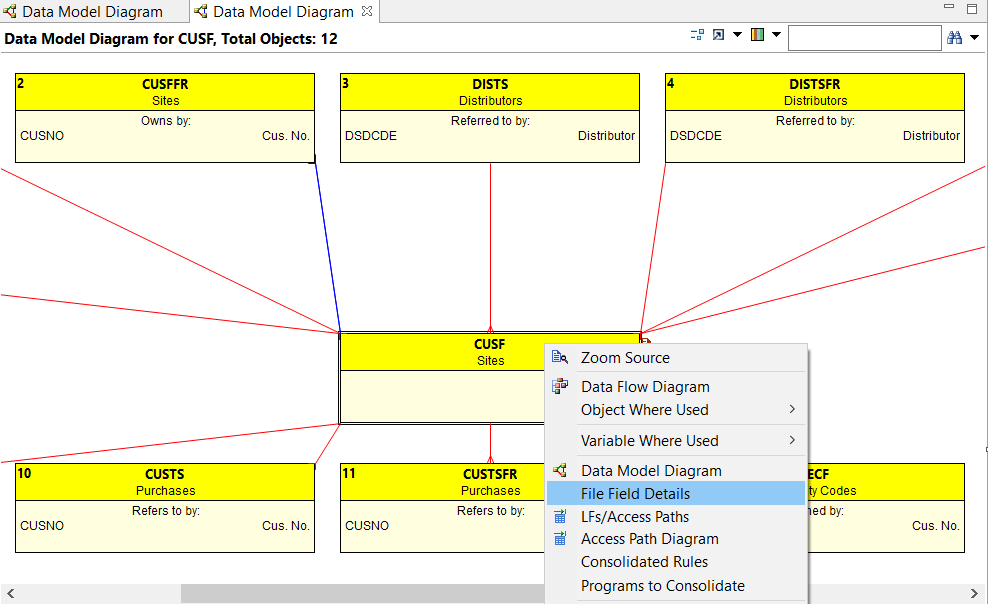


Fig. 3.3.1 – File Field Details option

A new window appears displaying File Field Details information.

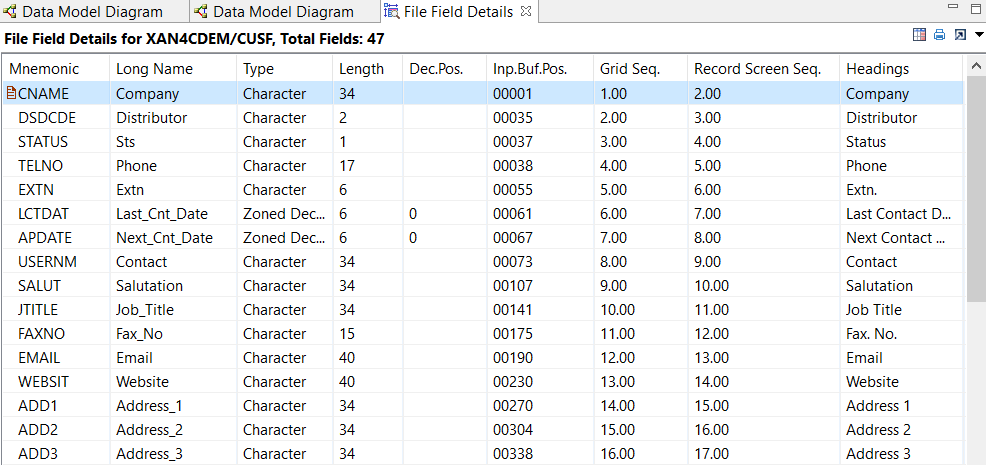


Fig. 3.3.2 – File Field Details for CUSF

|  |  |
| --- | --- |
|  | The ‘Display’ column is related to modernization features and will always show up as ‘Y’ for a normal user. |

Access Path Diagram/APD

The Access Path Diagram is a graphical presentation of all the access paths related to a specific file. Generate the diagram by selecting the Access Path Diagram option from the context menu over a selected file, CUSF.

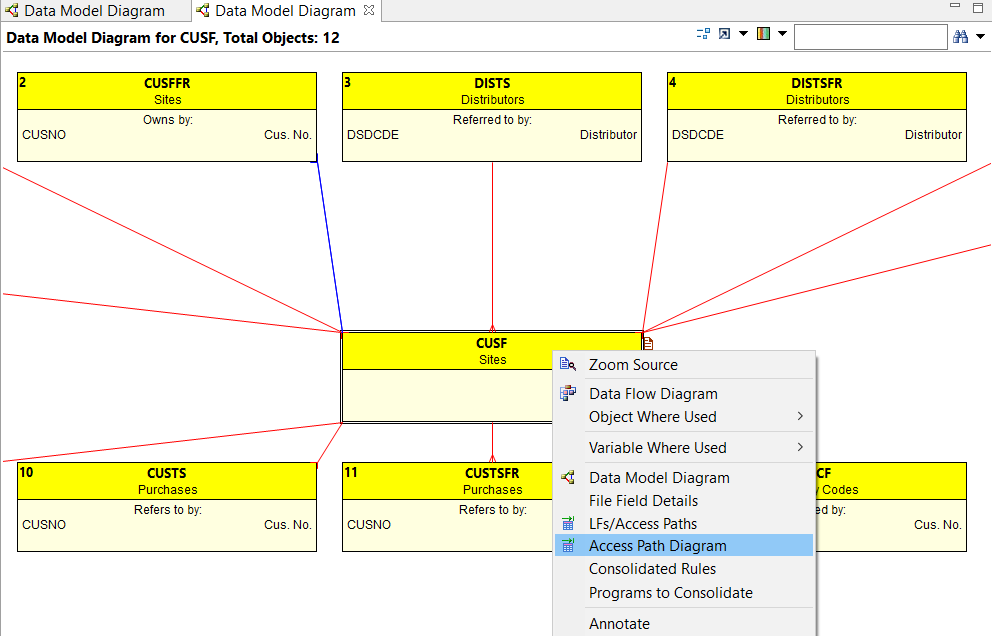


Fig. 3.4.1 – Access Path Diagram option on DMD

This brings up the APD for CUSF.

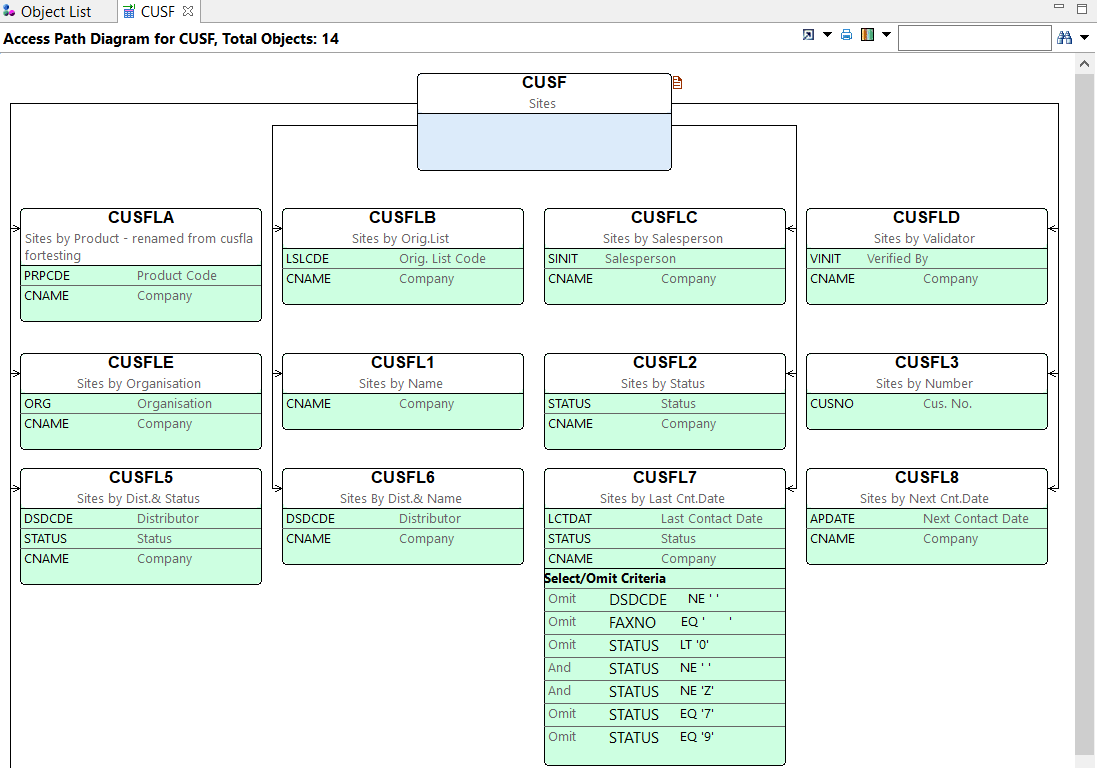


Fig. 3.4.2 – Access Path Diagram for CUSF

Go back to the DMD displaying the file-specific DMD for CUSF.

|  |  |
| --- | --- |
|  | The Access Path Diagram now shows the long name for the files when the long names exist and the Windows>Preferences>X-Analysis XX.X.X>Show Long Names for Files option is selected. |

View Data

The user can also view file records from the data model of an application.

Select the View Data option to view the actual records held in PFs. The option is available on all the objects displayed in the DMD.

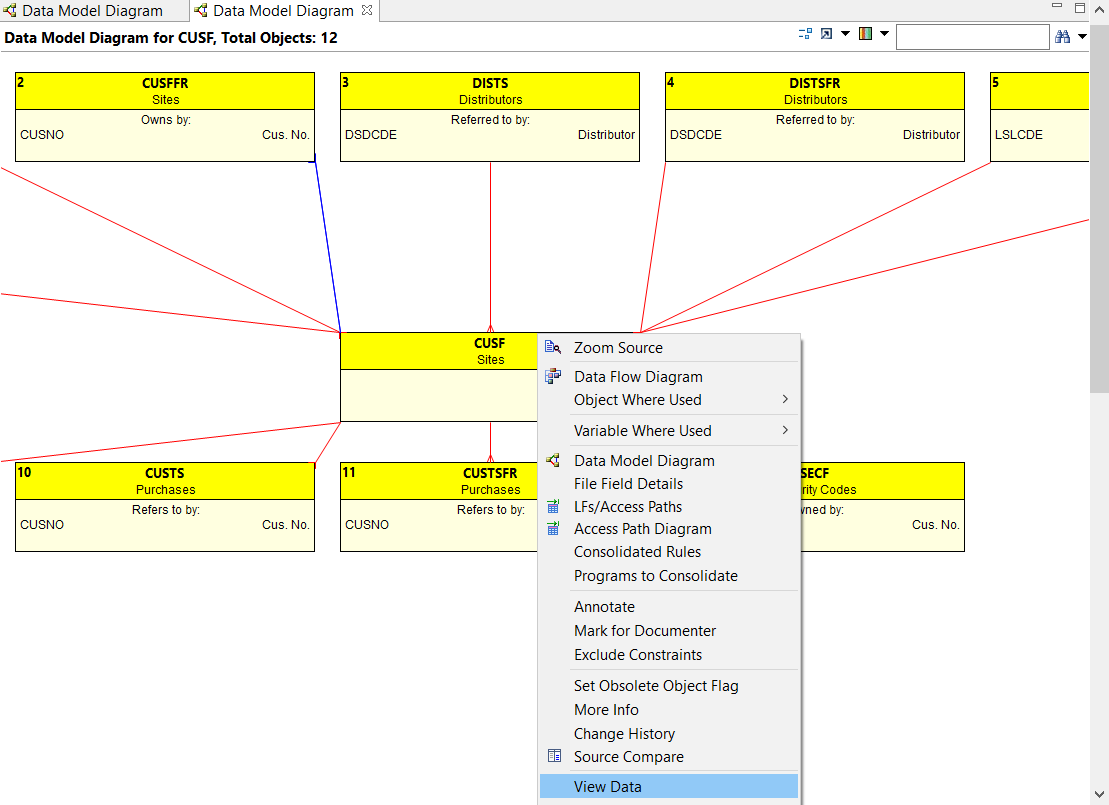


Fig. 3.5.1 – View Data option on the DMD object

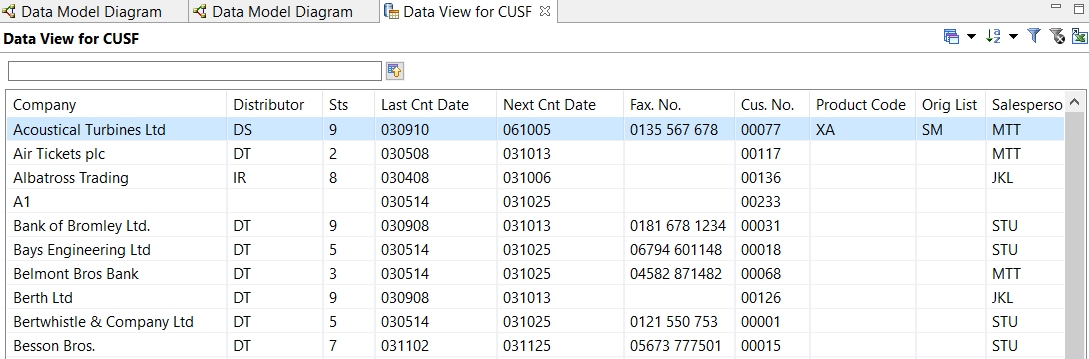


Fig. 3.5.2 – Data View for CUSF

Record Display

The View Data window lists records for a selected file. Double-click on a record to display the complete record.

If the user double-clicks Bertwhistle & Company Ltd., the record gets displayed along with its dependent files data.

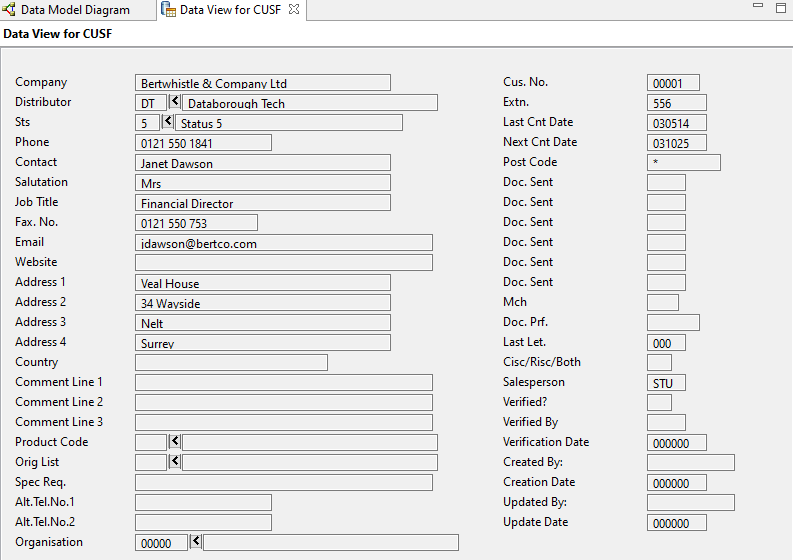


Fig. 3.5.3 – Record Display with Dependents

The tab at the bottom displays the dependent file data which includes any ACCESSES/OWNS/ REFERS TO relationships for CUSF in the DMD.

On the record display, the Descriptors button displays the ACCESSED BY/REFERS TO relationships. This is like a JOIN to a foreign file, to get the description for the code used from the Master/Owning file.

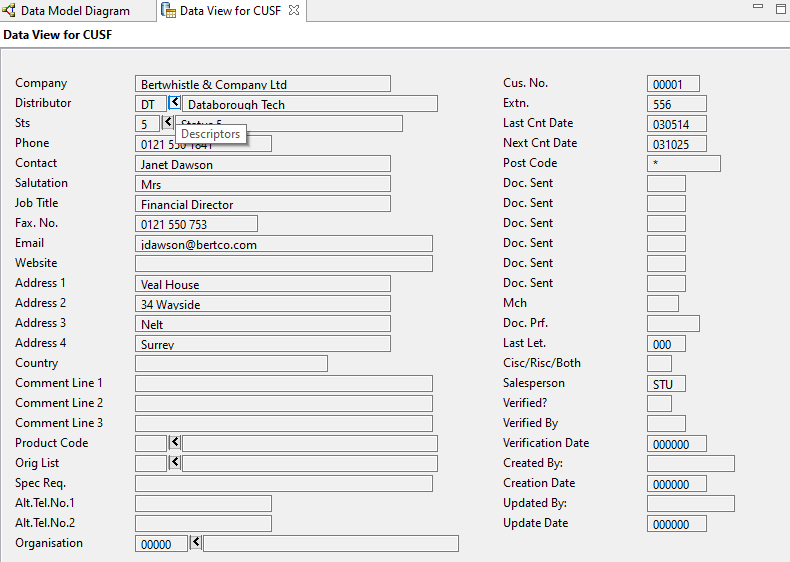


Fig. 3.5.4 – Descriptors button on Record Display

Click the Descriptors button to invoke the following window:

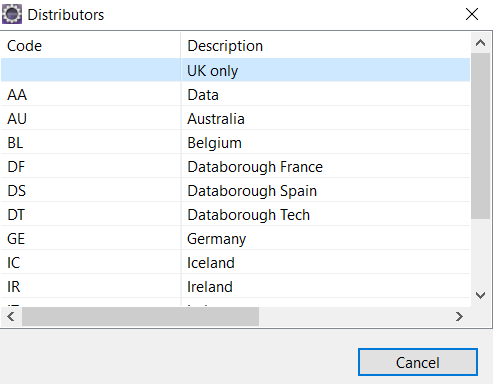


Fig. 3.5.5 – Window presenting Descriptors from Master/Owning file

Data Model Diagram - Application Area Specific

The DMD for an application area represents relationships among the related objects that belong to that application area only. Select the Data Model Diagram option for the application area – MVCPROCESS.

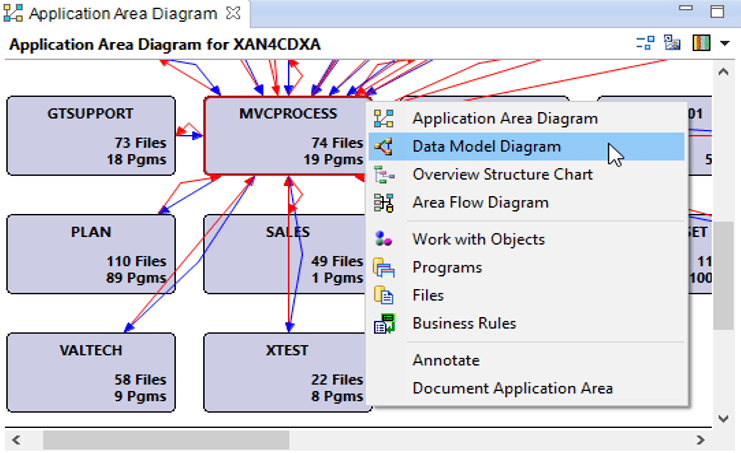


Fig. 3.6.1 – Data Model Diagram option from the Application Area Diagram

This invokes the Data Model Diagram dialog box for the application area – MVCPROCESS.

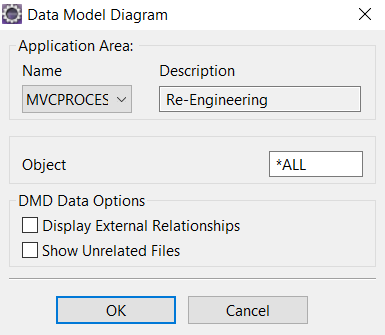


Fig. 3.6.2 – Data Model Diagram dialog box for an application area

Click OK to generate the DMD for the selected application area.

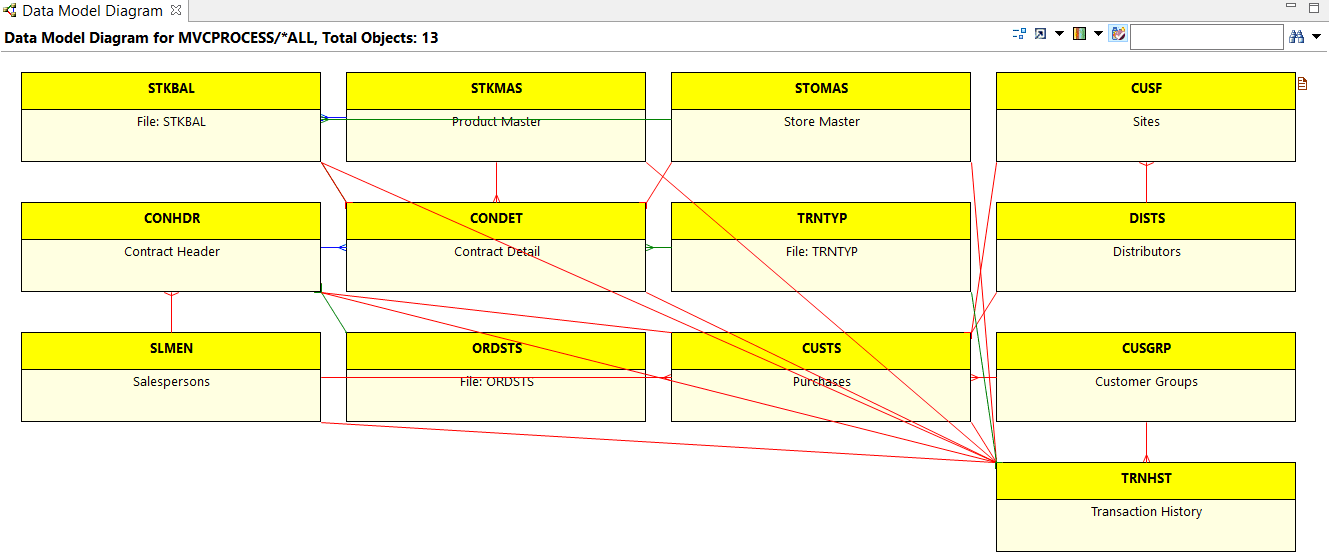


Fig. 3.6.3.A – Data Model Diagram for an application area (Restricted)

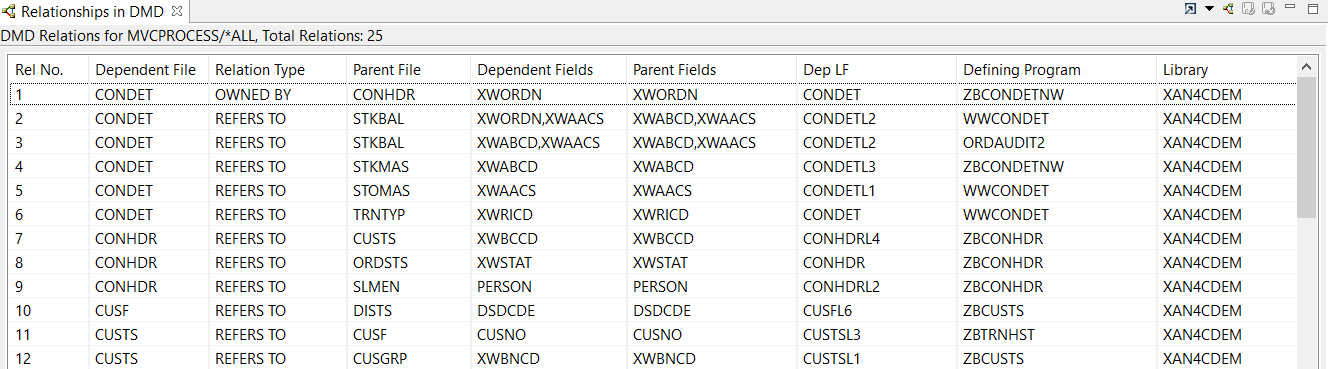


Fig. 3.6.3.B – Data Model Diagram for an application area (Restricted)

The DMD displayed above is a restricted Data Model and it only displays the objects which belong to the application area – MVCPROCESS.

The toolbar on the DMD has an option to generate the unrestricted DMD. Click the Unrestrict To Application Area icon.

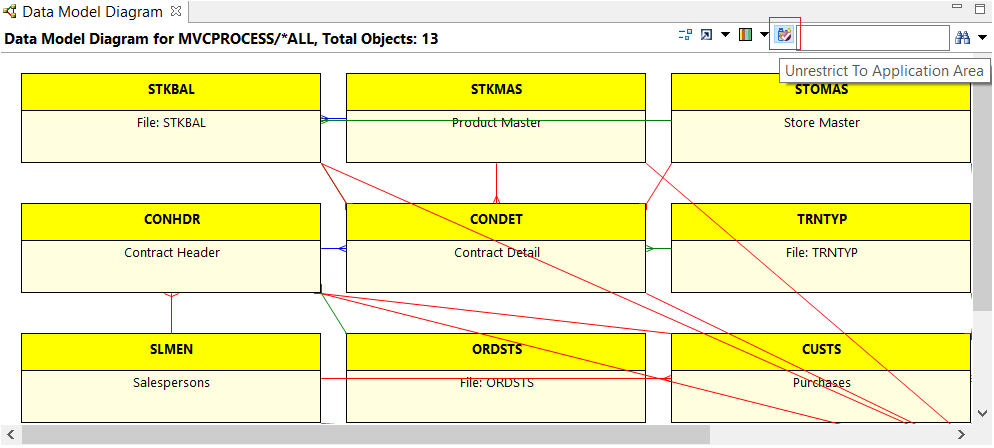


Fig. 3.6.4 – Unrestrict To Application Area icon

The following window will be displayed.

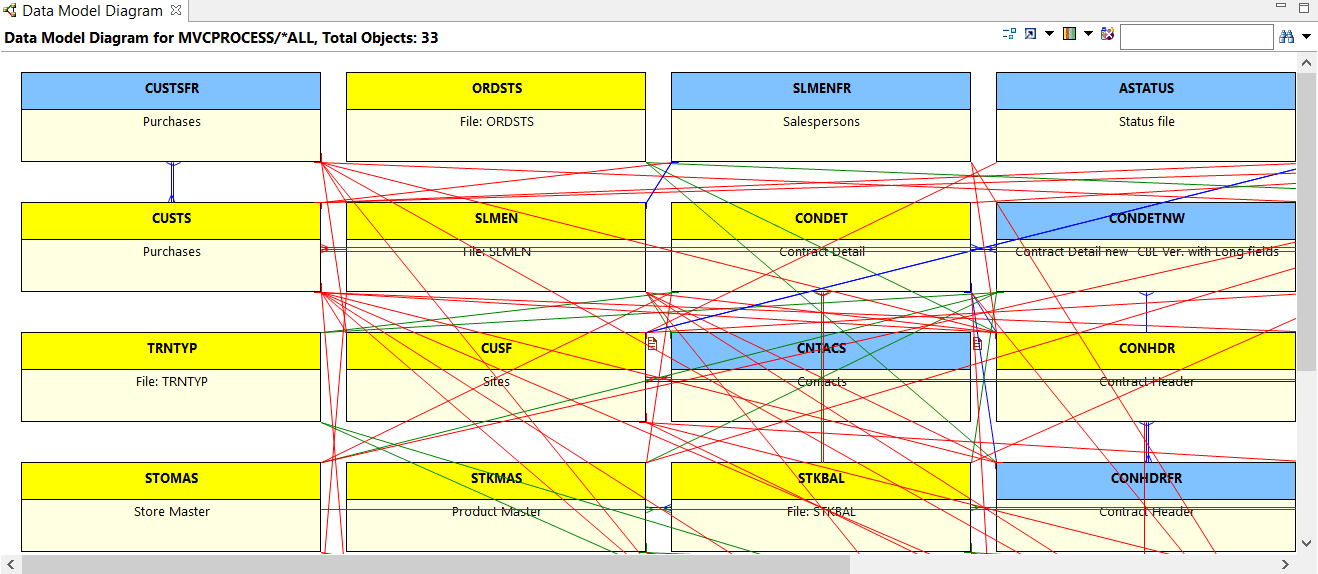


Fig. 3.6.5 – Data Model Diagram for Application Area (Unrestricted)

|  |  |
| --- | --- |
|  | The objects displayed in blue are the ones which do not belong to the application area, MVCPROCESS.  While using the ‘Restrict To Application Area’ feature, ensure that the originally-selected application area remains the same. |

What’s Next

Over three decades of dedicated research and development has rewarded us with a sophisticated software solution like X-Analysis. Built to meet the requirements of advanced computing and analysis, X-Analysis and its varied features make it easy to understand how a legacy application works. We also experienced how X-Analysis helps in analyzing one big application as smaller logical business areas. Its enhanced Data Modelling feature not only exposes the Business Rules implicit in the data and the process models of an application, but also digs out the vast amount of business logic buried inside the source code of the programs.

As we already know, X-Analysis dissects legacy application programs to decouple screens from embedded business logic. This componentization can be leveraged by the Transformation Tools, thereby assisting in re-architecting a legacy application in J2EE.

Appendix A – Enabling the SEU Interface

Create User Profile XAN4SEU

When browsing source code using the IBM i SEU, it is necessary to be logged in as user, XAN4SEU. The following gives instructions for creating the correct user profile.

Step 1 Logon as QSECOFR.

Step 2 Create XAN4SEU user profile.

Use the CRTUSRPRF command to create the XAN4SEU user profile. Make sure that the following parameters are set:

User Password : XAN4SEU  
User Class : \*PGMR  
Initial Program: XSEUCLP  
Library : XAPROD

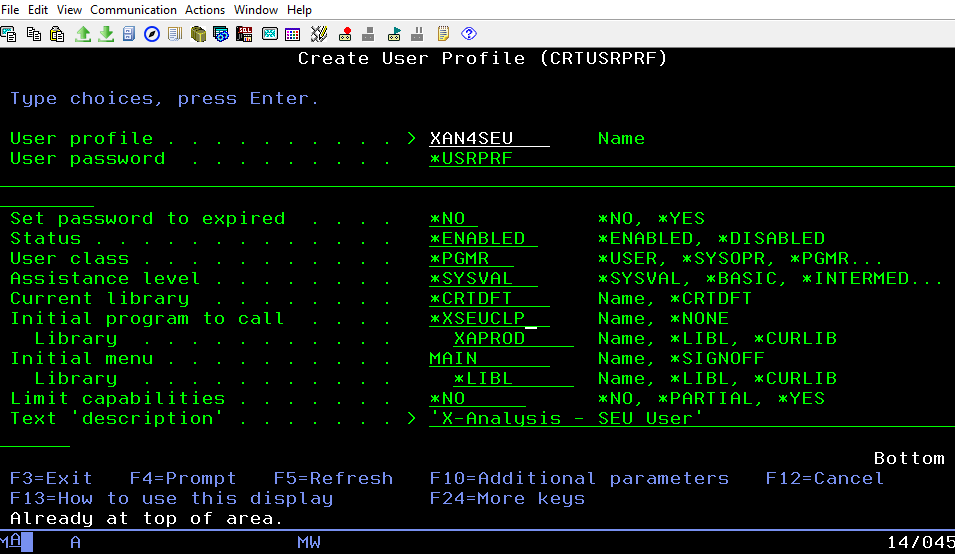


Fig. 4.1.1 – Create User Profile (CRTUSRPRF)

Press F10 and Page Down

Set Special Authorities:

* **\*ALLOBJ**
* **\*JOBCTL**
* **\*SAVSYS**

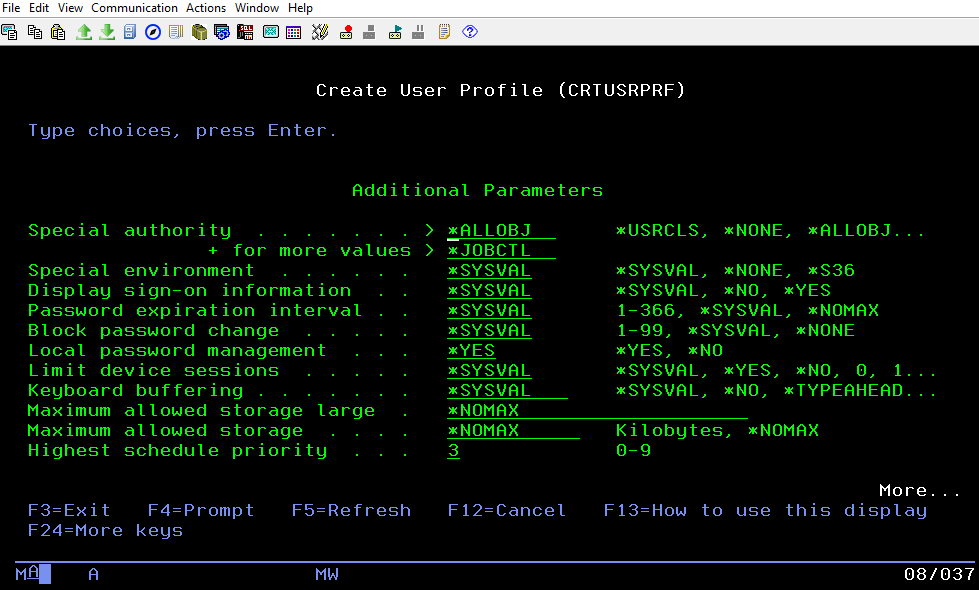


Fig. 4.1.2 – Change User Profile (CHGUSRPRF)

Step 3 Logoff.

|  |  |
| --- | --- |
|  | If there is an instance where a source member is already open for editing, then that specific source member cannot be opened using SEU. |

The user must follow the below steps, for the option SEU to work independently with IBM i ACS:

1. Update the ACS version to 1.1.9.0.
2. Open the Source code and click on the SEU button. Now, it will redirect to the login page in ACS.
3. Enter the Signon information with XAN4SEU as a User name and Password. The details reflected in client side screen will be same as displayed on ACS. Refer to the below screens.

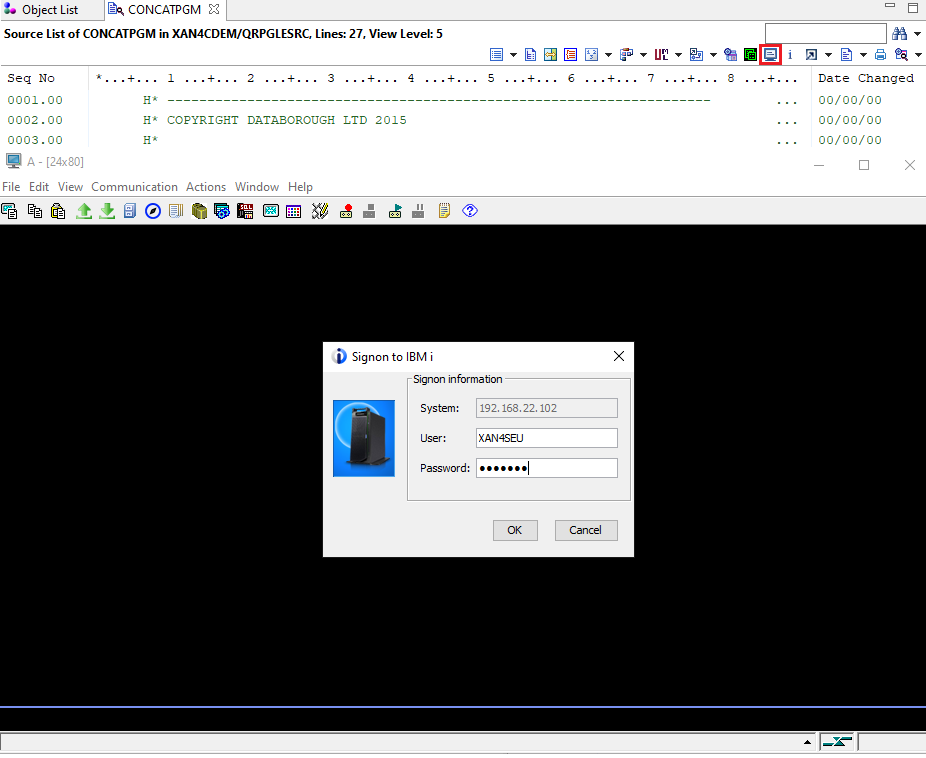


Fig. 4.1.3 – Signon to IBM i

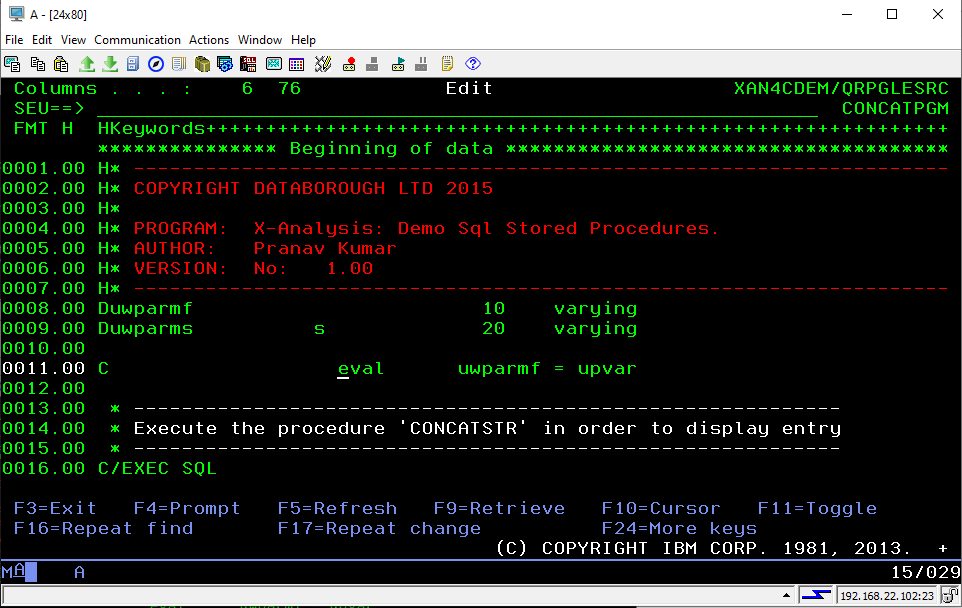


Fig. 4.1.4 – ACS screen

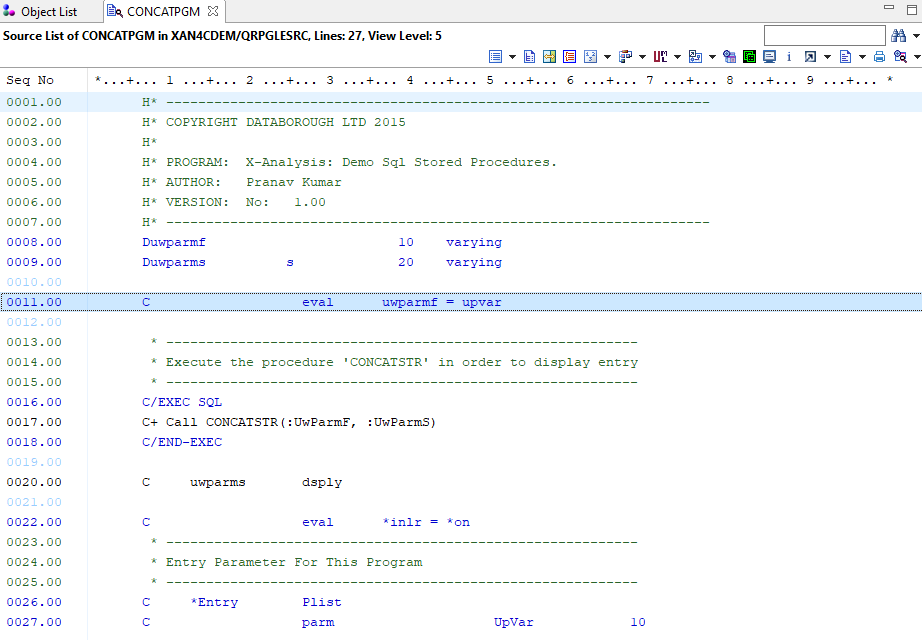


Fig. 4.1.5 – Client screen

|  |  |
| --- | --- |
|  | Any changes made in ACS will now get reflected on the Client side. |

Creating Macro for User Profile XAN4SEU

Macros allows the user to automatically sign on for XAN4SEU., Same time it also allows the user to work with the SEU option. So, it is necessary to verify whether the XAN4SEU macro has been created or not. If not, then the user must record this macro using the following steps:

Step 1: Press the Start Recording button. The Record Macro/Script as dialog box will be displayed. Sign on with the XAN4SEU user profile.

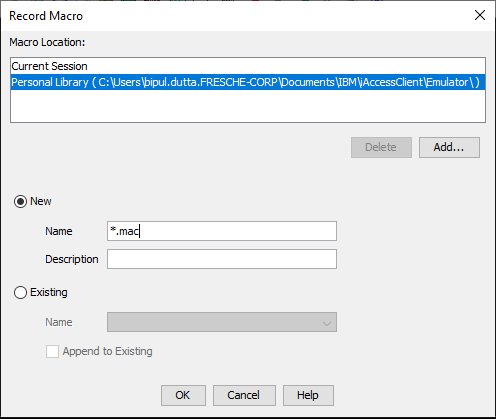


Fig. 4.2.1 Record Macro/Script

Step 2: On the Sign On screen, enter the User Profile – XAN4SEU. Press Tab and enter the Password also as XAN4SEU.

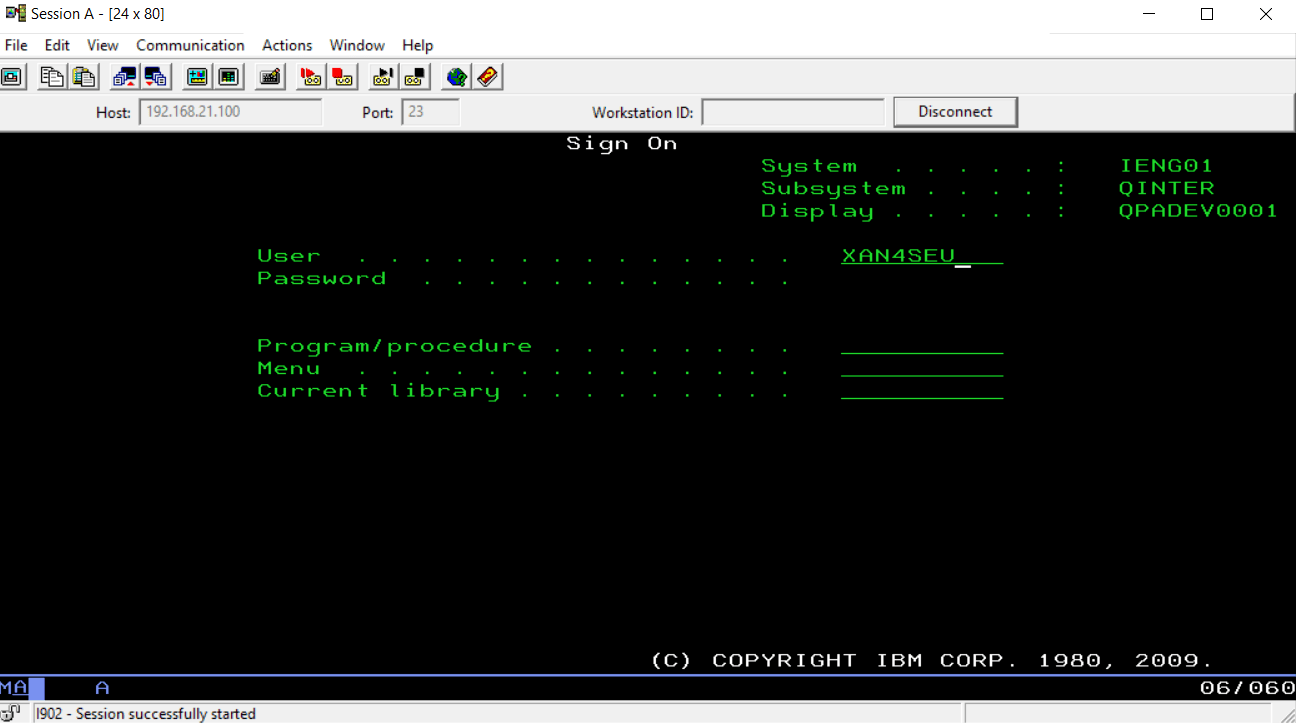


Fig. 4.2.2 – Sign On

Press ENTER. The IBM i main menu screen will be displayed:

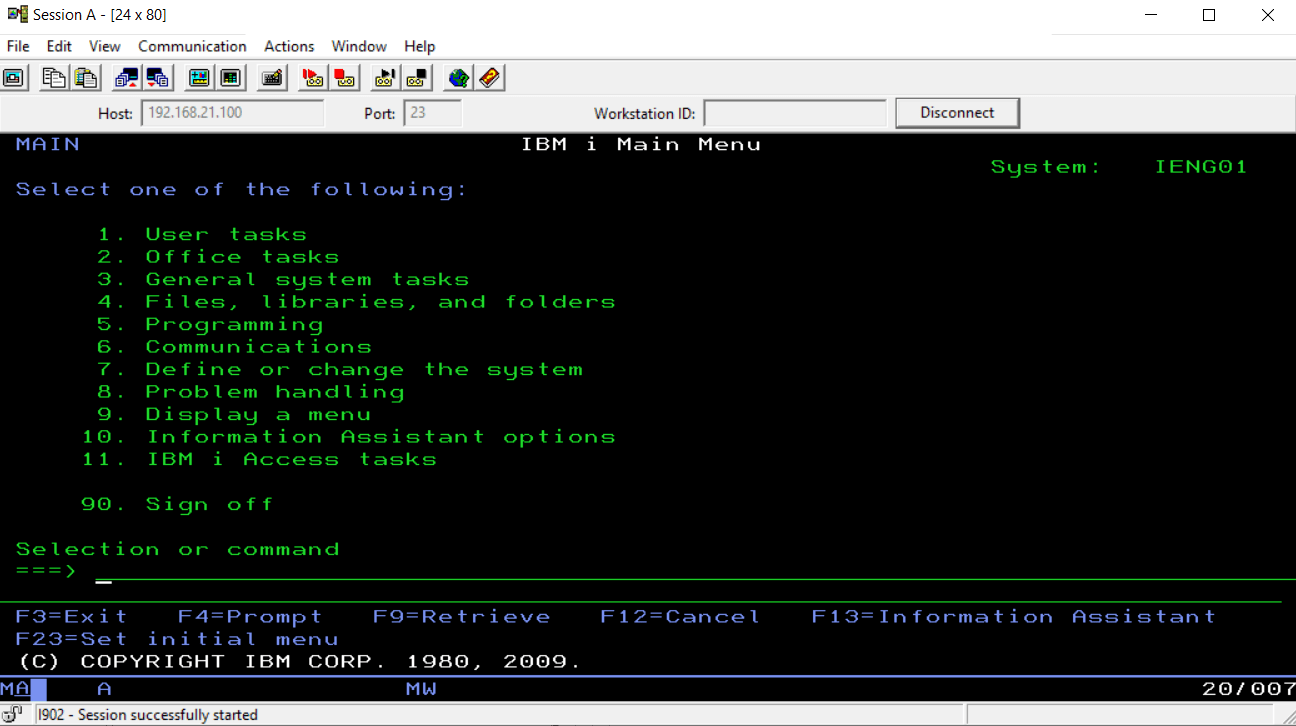


Fig. 4.2.3 – IBM i Main Menu

Step 3: Stop the recording by selecting the Stop Recording Macro option from the Actions submenu or clicking the related button directly from the toolbar.

|  |
| --- |
| Fig. 4.2.4 – Stop Recording Macro |

Index

2E, 18, 39, 56

5250 Emulator, 97

Access Path Diagram, 8, 99, 110

Access Paths, 29, 99, 106, 110

All References, 22, 43

Annotation, 29, 55

APD, 8, 110

application area, 8, 14, 18, 19, 40, 99, 101, 115

Application Area Rules, 21

Application Folder, 77

Attributes, 29, 74

block diagrams, 53, 60

Business Rules, 43, 84, 118

Called Programs, 47

Client, 5, 7, 9, 14, 120

Client Components, 8

COBOL, 7, 18, 99

configuration, 76

cross-reference library, 8, 14, 55, 100

Data Flow Diagram, 8, 18, 28

data model, 5, 8, 14, 99, 100, 108, 112, 115

Data Model Diagram, 8, 99, 100, 115

Data Modelling, 99, 118

Database Library, 14

Default View Level, 45

Dependent Fields, 102

Dependent File, 99, 102

Detail DFD, 36

Device Files, 42

DFD, 8, 28

Diagram Options, 83, 92

Diagrams, 28, 53, 60, 70

DMD, 8, 101, 106, 110, 112, 115

Document Manager, 18, 69

documentation, 5, 7, 17, 18, 69

Documenter, 69

Eclipse, 5, 7, 10, 15, 16, 70, 97

Expand/Collapse Referenced Objects, 29

Export Options, 27, 49, 83

Export to MS Excel, 27

Export to PDF, 27

Fetch Next References, 24

FFD, 8, 108

Fields, 34, 42, 52, 102, 106, 108

File Field Details, 8, 99, 108

file-specific DMD, 104, 111

Flowchart, 83

Function Type, 56, 93

Generate UML, 83

Google Drive, 27

Hide/Show Objects, 29

Hierarchical Structure Chart, 8, 18, 60

Hierarchy Exclusions, 57

IBM, 4, 5, 7, 10, 14, 16, 17, 97, 99, 119, 124

IBM i, 5, 7, 12, 14, 17, 97, 99, 119, 124

Indented Source Code, 84

Invert Selection, 71

Jump to Subroutine, 88, 91

legacy application, 118

Legend, 38, 44, 54, 67, 102

Level, 7, 18, 22, 28, 52, 83

LFs, 23, 52, 99, 106

LFs/Access Paths, 99, 106

libraries, 13

Library List, 14

MS Visio, 7, 87

Narratives, 62

Object List, 19, 22, 70, 80, 90

Object Where Used, 6, 18, 22, 27, 29

Overview Structure Chart, 8

OWU, 6

parameters, 28, 42, 119

Parent Fields, 102

Parent File, 102

Preferences, 9, 14, 16, 26, 37, 45, 83, 97, 104, 107, 111

Print, 60, 83

program logic, 53

Program Structure Chart, 8, 92

PSC, 8, 92

pseudo code, 7, 18, 83

RDi, 7, 83, 97

Referred Files, 52, 64

Relation Type, 102

Relational Data Model, 99

Relationships in DMD, 101

RPG, 7, 18, 89, 99

SCD, 8, 53, 60

Screen Components, 83

Screen/Report Design, 83, 95

Server, 7

Signon, 120

SOURCE, 18, 22, 42, 80, 83, 90, 106, 118, 119

Source Browser, 18, 80, 83, 90

Source Lines, 42, 85, 91

Source Options, 83

Structure Chart Diagram, 8, 18, 53, 60, 93

subroutines, 42, 84, 91

Suite of products, 5

Synon, 18, 39, 54, 68

Use SSL, 13

Variable Where Used, 8, 18, 42, 52

VWU, 8, 45, 52

X-Analysis, 4, 5, 6, 7, 9, 10, 14, 15, 16, 17, 19, 22, 27, 37, 45, 55, 69, 80, 85, 96, 99, 104, 107, 111, 118

X-Analysis Client, 5, 9, 14

X-Analysis Perspective, 10, 96

XAPROD, 119

X-Rules, 84

Zoom Source, 90, 106