helpsystems

User Guide
Abstract Plugin Clients

201703295746

Table of Contents

Table of Contents ..... **iii**

Introduction **1**

About ABSTRACT 1

About iSeries Navigator 1

About WebSphere Development Studio Client 4

Summary of ABSTRACT Features 5

Frequently Asked Questions **6**

Requirements **9**

iSeries Host 9

Workstation 9

Get Started – Fast Path **11**

Step 1 - Load iSeries Software 12

Step 2 - Load Client Software 12

Step 3 – Verify Installation 12

Step 4 - Install the WDSc Plugin 13

Using the WDSc plugin 13

Step 5 - Set Defaults 13

Accessing Defaults from iSeries Navigator 13

Accessing Defaults from WDSc 14

Step 6 - Run the Object Exception Report 15

Step 7 - Load the Cross-Reference 18

ABSTRACT Defaults **21**

Job Description 22

Job Queue 22

Hold on job queue 22

Object date 23

Object name 23

Data set 23

Select a Data Set 25

Add a New Data Set 25

Change the Data Set....26
Clear or Delete the Data Set....26

Object Selection Window....27

Look In:....27

(Main window)....27

Name, Type, Attribute....27

Exception Reports....28

About Exception Reports....28

Submit the Object Exception Reports....30
Name....31
Type....31
Attribute....32
Orphaned Objects....32
No Source....32
Source Changed....32
Not Used Since....33
Not Loaded Since....33

Report Information....34
Source Conflicts....35
Orphaned Objects....35
Unused Objects....36
Undocumented Objects....36

Loading the Cross-Reference Database....37

Overview....37

Object Authority Requirements....37

Cross-Reference Initialization Steps....38

Information in the Cross-reference....40

Library Selection....41

Work with Cross-Reference Data....44

Select Library(s) or Objects from AS/400 Explorer....45
Load Entire Library(s)....45
Load Specific Object Types....46
Load Specific Objects....47
Remove Libraries from the Cross-reference....47
Reload Existing Cross-reference Information....47

Object Relations....48

Overview....48

How to View Object Relations....49
Elements of the Object Relation Display....51
Parent objects....51
Indented objects....51
Type....52
Attribute....52
Usage....52
Text....53
Extended Description....53
Object Resolution....54
Object Relation Options....55
Types of Object Relation Displays....56
Object Where Used....56
Field Where Used....57
File Group Usage....57
Field Group Usage....58
Object References....59
Expand or Collapse the List Outline....60
Level Buttons....61
Object Relation Subset....62
Show Children of Type....63
Show children only once....63
Usage....64
Explosion Level....66
Print the Object Relation List....68
Save the Object Relation List....68
Flowcharting....69
Flowchart the Object Relation List....69
Edit the Flowchart....70
Select an Object....71
Draw a new object....71
Draw a link....71
Stretch a link....72
Draw a reflexive link....72

File Analysis....73

Overview....73
View File Analysis Information....74
External Layout....76
Database Relations....78
Member Information....79

Internal Layout 79
File Usage 80
File Group Usage 81
Analyze a Different File 81
Print File Analysis Information 81

Recreate Database Relations 82

About Recreate Database Relations 82

Recreate Options 84
Recompile Physical File 85
Allow Rename 85
Retain Data in Physical File 85
Number of Records 86
Compile Related Objects 86
Target Library 87
Next>> Button 87
Source File 88
Text 88
Audit Report 88

Find String 89

Wild Card 90

String(s) to scan for 90

The String Scanning Report 91

Source Flowchart 93

Edit Source 94

CODE Edit Window 95

Index 96

Introduction

About ABSTRACT

ABSTRACT is a powerful development environment available as a plugin to both WebSphere Development Studio Client and iSeries Navigator. Through its cross-referencing capabilities, ABSTRACT provides a complete, centralized information source for all of your iSeries software. The relationships among the various objects in your applications are kept in a cross-reference database. This cross-reference can give you high level views of the control and data flows within your applications. Coupled with the cross-reference is a set of development tools that will save a significant amount of time while maintaining existing applications or developing new ones.

ABSTRACT has many extraordinary ease-of-use features as well. ABSTRACT is designed to allow easy navigation of iSeries objects using an intuitive 'Explorer-like' navigation interface. This makes it easy to learn and remember the product functions.

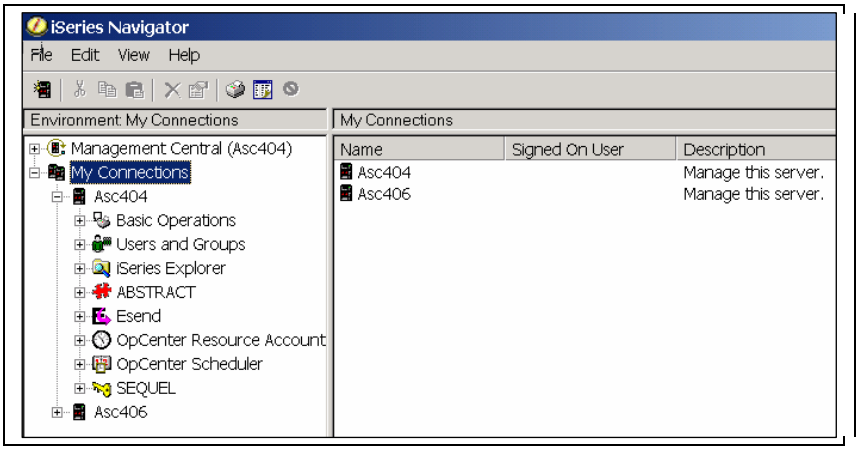
Either ABSTRACT plugin may be installed and used along with the "green screen" version of ABSTRACT. Both versions of the product use the same cross-reference database resident on the iSeries.

In short, everything that you need for a complete programming environment is integrated into one consistent, easy to use interface. Once you install ABSTRACT and it analyzes your applications, you will be able to use it as your primary development environment - working from ABSTRACT displays from the time you begin your work day until you leave for the day.

About iSeries Navigator

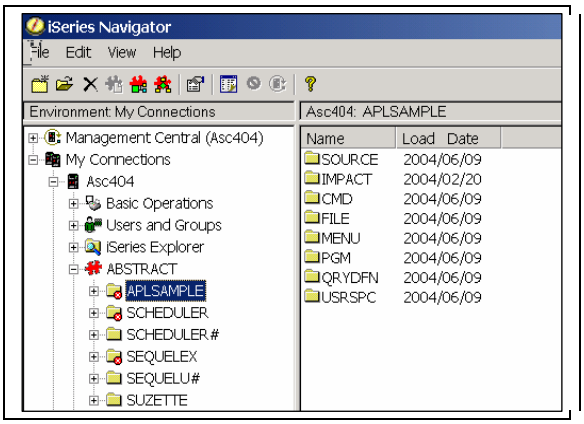
IBM's iSeries Navigator is a Windows-like graphical user interface for managing the iSeries. iSeries Navigator makes operation and administration of the iSeries easier, faster and more productive. It allows you to point-and-click your way through administration tasks instead of using the traditional command line interface. For instance, you can copy an iSeries user profile onto another iSeries system by dragging the user profile from one iSeries to the other iSeries. Wizards guide you through setting up security, TCP/IP, and more. iSeries Navigator also includes Management Central, which allows management of multiple iSeries from a central iSeries system, including real-time graphical performance monitoring.

IBM is committed to iSeries Navigator as the new iSeries user interface. It has been included with each version of OS/400 since V3R7 and has seen significant enhancement with each new release. IBM encourages the integration of products like ABSTRACT using iSeries Navigator's Plug-In Support. ABSTRACT is a perfect fit for the iSeries Navigator hierarchy, allowing you to track and manage objects on your iSeries more easily than any other means.

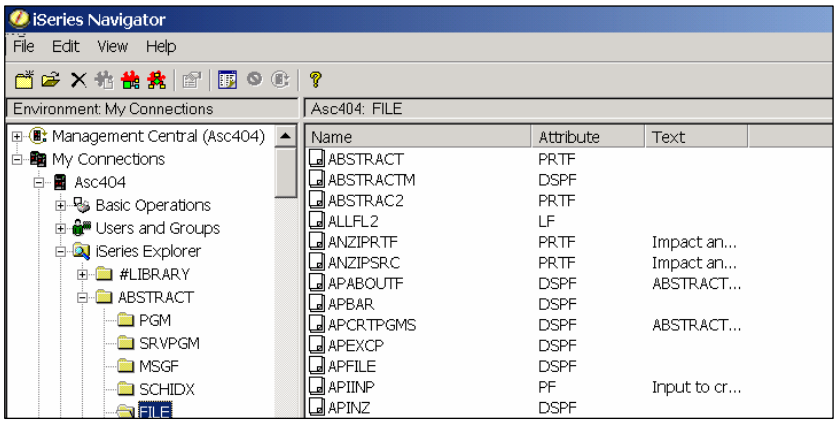


iSeries Navigator appears as a graphical hierarchy tree comprised of a number of nodes that may be expanded for each function. When ABSTRACT is installed, two additional nodes are available: **ABSTRACT** and **iSeries Explorer** (see next page).

ABSTRACT is the main node that shows the libraries and objects that are loaded into the cross-reference database. The functions in the ABSTRACT node extend to the iSeries Explorer node as well as the Database and File Systems nodes within iSeries Navigator.



iSeries Explorer provides a fast method of browsing system libraries and objects rather than going through QSYS.LIB in the File Systems node. It is similar in appearance and functionality as MS Windows Explorer. The benefit of using iSeries Explorer is that you can use ABSTRACT's real-time functions (like File Analysis) without having to load the object(s) into the cross-reference.



About WebSphere Development Studio Client

IBM intends for all RPG programmers to use to WebSphere Development Studio Client (WDSc) as the primary GUI for development. It encapsulates all of their development tools and editors in a single, graphical palette. The entire client is actually a plugin to Eclipse, the open source, Java-based framework. WDSc lets you seamlessly code, edit, and debug, while using their Remote System Explorer to replicate the kind of development environment afforded by PDM from a green screen.

Summary of ABSTRACT Features

Exception Reports – Show potential problems with your applications or ABSTRACT cross-reference data (page 28).

Cross-Reference Database - Contains object relationship information about your applications. The cross-reference database is easy to maintain and update using ABSTRACT'S intuitive menus and options (page 37).

Object Relations - Different object reference and 'where used' scenarios are easy to display and interpret. This information is based both on real-time information and on data loaded into the cross-reference files (page 48).

File Analysis - Provides information about database, display, and printer files. Information includes external and internal file layouts, database relations and member definitions (page 73).

Recreate Database Relations – Recreates all affected objects after changes are made to a physical file record layout and/or programs are modified (page 82).

Find String – Searches source and message files for occurrences of one or more specified strings (page 89).

Edit Objects – Interfaces with the IBM editor (CODE).

Frequently Asked Questions

1. **How is iSeries Navigator related to ABSTRACT?**

   The IBM iSeries Navigator development team works closely with HELP/SYSTEMS in ensuring a tightly integrated solution via their plug-in support. ABSTRACT is a perfect fit for the iSeries Navigator hierarchy, allowing you to track and manage objects on your iSeries more easily than any other means.

2. **Do I need Client Access to use iSeries Navigator, and therefore ABSTRACT?**

   If you are a current user of Client Access Express or iSeries Access for Windows, iSeries Navigator is included with the software. Beginning with V4R2, iSeries Navigator comes free with OS/400 regardless if you purchased Client Access or not. It can be installed on your PC through the NetServer path for your iSeries. Even if you want to use the WDSc client, the Navigator client is still required for installation.

   (e.g. /QYOURSYSTEMNAME/QIBM) or from the Client Access Express CD that was shipped with your iSeries.

3. **Will a host-based version of ABSTRACT exist?**

   A "green screen" version of ABSTRACT is still available on the iSeries for those who prefer a host-based interface.

4. **Does that mean I need to load separate cross-reference data for both the plugin and "green screen" versions of ABSTRACT?**

   No. All of the data still resides on the iSeries and both products share the same cross-reference database.

5. **Do the various options take any longer in either plugin?**
   Your response time should be comparable. Most of the response time delay in the product is related to the performance characteristics of your PC. A faster PC with more memory will bring up tree-lists more quickly.

6. **How do I view whatever errors occurred during the load of the cross-reference?**

   There is an error log you can access by right-clicking on any library that appears beneath the ABSTRACT node.

7. **What is "service data" and how does it affect this tool?**

Every source-based object has information attached to it that tells you where the source is located for the compiled object. You can view service data with the OS/400 command DSPOBJD with DETAIL(*SERVICE).

Since ABSTRACT is an automated product, it needs an automated method of matching up a program object with its source. While ABSTRACT will look at the service data on the object and immediately be able to find the corresponding source code, sometimes service data becomes incorrect if the source code gets moved after the object is compiled. This is a fairly common occurrence in iSeries shops and can be problematic, often with source code getting lost.

It is important to analyze your service data situation before proceeding with any cross-reference loading in ABSTRACT. One of the best ways to do this is to run the ABSTRACT Exceptions report over your program object libraries you intend to load.

When source cannot be located (the service data is pointing to the incorrect source file), the source file and member names will be underlined on the report. See the ABSTRACT manual for complete documentation on this subject.

8. How do you submit a cross-reference build if the service data is (A) Correct or (B) Incorrect?

A) - There are several ways to load multiple libraries into the cross-reference. Probably the easiest way is to go into the iSeries Explorer node and use your control key to select mulitple libraries on the right hand side of the navigator display. Once you have selected them all you can right click on any of the selected libraries and choose the Load Cross Reference option. This will load all of the selected libraries via a submitted job on the iSeries. The job's status can be monitored through the Job Management node in iSeries Navigator.

B) - If you suspect that service data is incorrect, consider your environment before proceeding. In most scenarios, the missing source for a given program is simply in another library. If you load this "other" library in the cross-reference, ABSTRACT will match the program with the source. There are some less frequent cases where you may have multiple source members for a given object, making it impossible for ABSTRACT to determine the correct version to reference. In this case, you must either use the ABSTRACT CHGSVCDTA command on the iSeries to point the object(s) to the correct source, or you can also recompile the object over the correct source.

9. I notice there is a node that gets installed called "iSeries Explorer". What is its purpose?

We included the iSeries Explorer because there was no efficient way in iSeries Navigator to obtain a real-time list of system libraries and objects. The only alternative was to go into the IFS and get the list by opening QSYS.LIB, which unfortunately is very slow. With the iSeries Explorer, you can bring up libraries that are not loaded in the cross-reference, but still interact with them. A good example would be the File Analysis function, which can be used with any file regardless if it is loaded into the cross-reference or not. It's also a handy way of loading libraries into the cross-reference. The Remote Systems Explorer eliminates the need for this node in WDSc.

102. Does ABSTRACT work with RPGII?

Yes. Its cross-reference abilities are restricted to program-to-file-to-field relationships. Since ABSTRACT does not analyze certain source code unique to S/36 applications (e.g. procedures, OCL), the list excludes program-to-program relations.

11. Does ABSTRACT work with ILE RPG?

Yes.

12. Does ABSTRACT handle program-described files (program-described input specs)?

ABSTRACT analyzes program-described files and fields.

13. When I'm running a library through ABSTRACT that consists of programs, data, and source, how do I make the product examine just that library? It seems to want to pull in references to objects outside of that library.

You can use TYPE(*SRCF). Extra care should be taken when using *SRCF, since you are essentially ignoring any external relationships with objects in other libraries.

14. How do I recover space used by deleted records when I remove libraries from the cross-reference.

Since ABSTRACT cross reference files are created with REUSEDLT(*YES), deleted records are not normally a concern. If you remove a large amount of cross reference data, you can use the RGZXREF command to free up space taken by deleted records.

Requirements

iSeries Host

- OS/400 V4R5 and higher.

Workstation

- Pentium IV or higher processor.
- 256 MB of RAM if using iSeries Navigator. More if using WDSc.
- V4R4 or higher of **Client Access Express** (with a current Service Pack installed). V5R2 or higher of iSeries Access for Windows is also acceptable.
- NT Workstation users need Service Pack 5 or greater installed.
- WDSc version 5.1 (with a current Service Pack installed) or higher - this is only required if you want to use the WDSc plugin.

Get Started – Fast Path

This section describes how to install ABSTRACT and load the cross-reference database. It contains excerpts from later sections, arranged to give you a “fast path” of the basic steps required to get ABSTRACT up and going. Complete the following steps to get started:

Load System i Software – page 12.

Load Client Software- page 12.

Verify Installation – page 12.

Install the WDSc Plugin – page 13.

Set Defaults - page 13.

Run the Object Exception Report – page 15.

Load the Cross-Reference – page 18.

Once these steps are completed you’ll be ready to take advantage of ABSTRACT’s powerful capabilities. Refer to the other sections in this user guide for more information on getting the most out of this exciting development environment.

Step 1 - Load iSeries Software

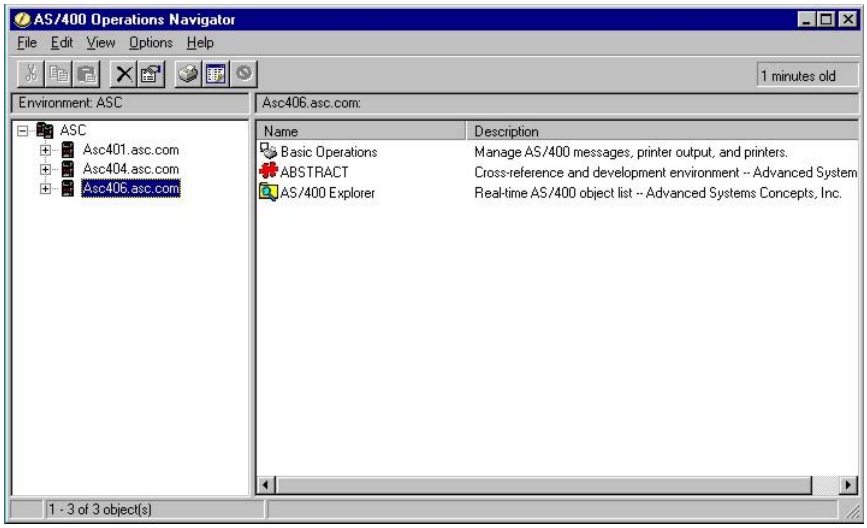
Refer to the document titled Instructions for Installing or Updating to ABSTRACT 10. This document (available in the download section of the SEQUEL Software Web site) will step you through the installation process.

Step 2 - Load Client Software

Refer to the document titled Instructions for Installing or Updating to ABSTRACT 10. This document (available in the download section of the SEQUEL Software Web site) will step you through the installation process.

Step 3 – Verify Installation

Start iSeries Navigator and select the system where ABSTRACT was installed. If the installation procedure completed successfully, the **ABSTRACT** and **iSeries Explorer** nodes will appear along with the other iSeries Navigator nodes.



Step 4 - Install the WDSc Plugin

Refer to the document titled Instructions for Installing or Updating to ABSTRACT 10. This document (available in the download section of the SEQUEL Software Web site) will step you through the installation process.

Using the WDSc plugin

To navigate to the WDSc plugin, start the WDSC client, choose 'Window:Show View:Other…' and then choose 'HELP/SYSTEMS Abstract Tree View' beneath the 'HELP/SYSTEMS.Abstract' node.

Step 5 - Set Defaults

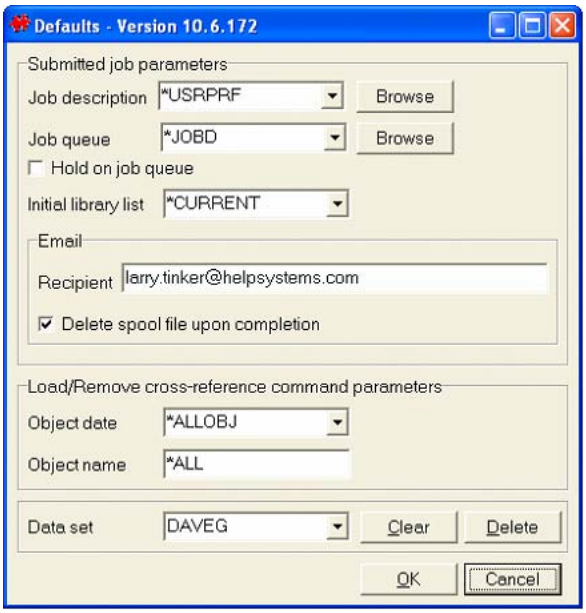
For more information on ABSTRACT Defaults turn to page 21.

Accessing Defaults from iSeries Navigator

Start iSeries Navigator and select the system where ABSTRACT was installed. Expand the branch for this system by clicking on the .

Right-click on the ABSTRACT branch to display the pop-up menu.

Select Properties from the pop-up menu to display the ABSTRACT properties.



Change any of the Job settings to meet your requirements.

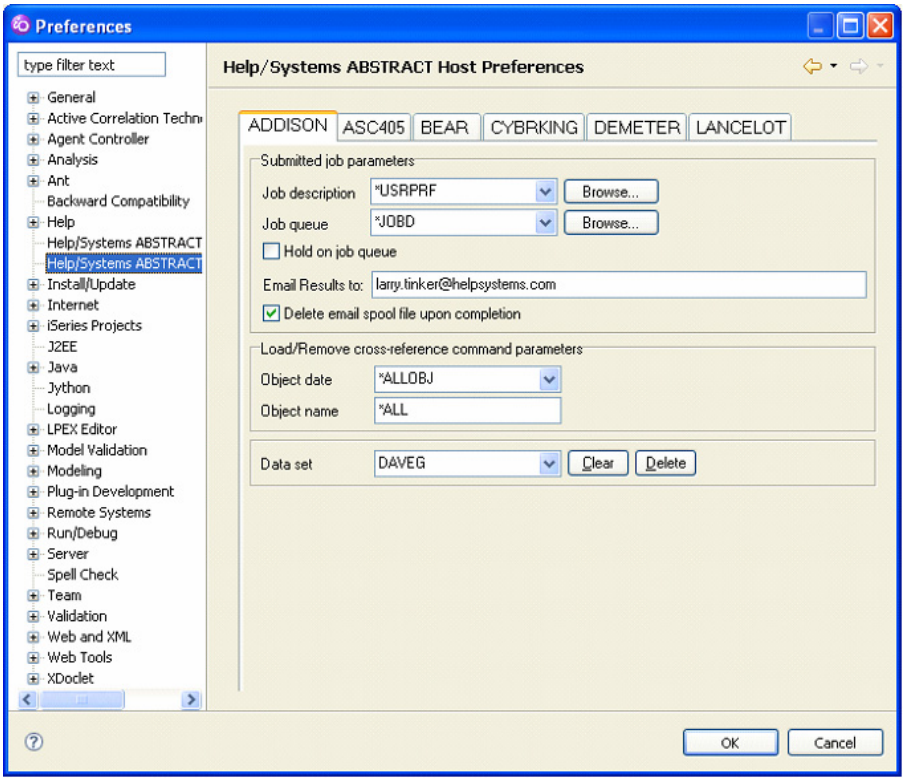
Specify an Email Recipient to receive reports generated by the find string feature, recreate database relations, and exception reports.

The Data set parameter may be left as *FIRST unless you want to create multiple data sets right away.

Accessing Defaults from WDSc

Start WSDc and click on Window>Preferences in the toolbar. Click on ASC ABSTRACT Host Preferences.

The Defaults window is listed below, common to each plugin.



Change any of the Job settings to meet your requirements.

Specify an Email Recipient to receive reports generated by the find string feature, recreate database relations, and exception reports.

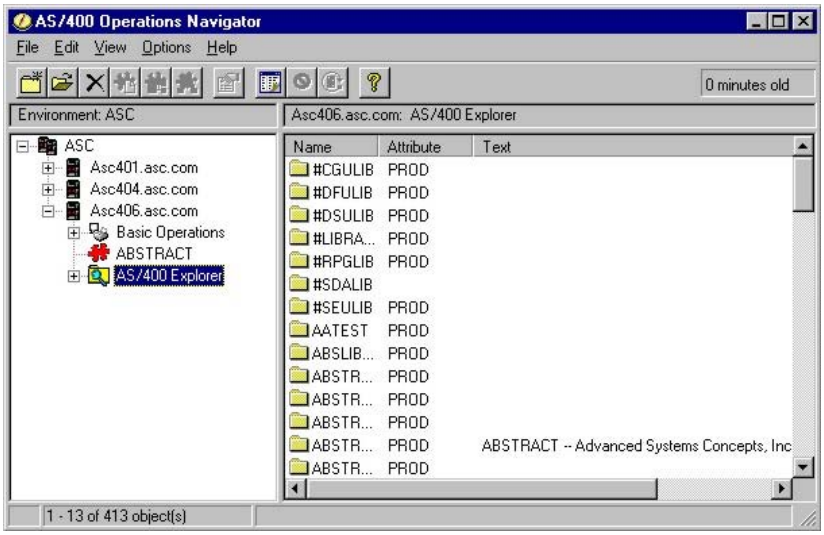
The Data set parameter may be left as *FIRST unless you want to create multiple data sets right away. Refer to page 23 for more information on data sets.

Step 6 - Run the Object Exception Report

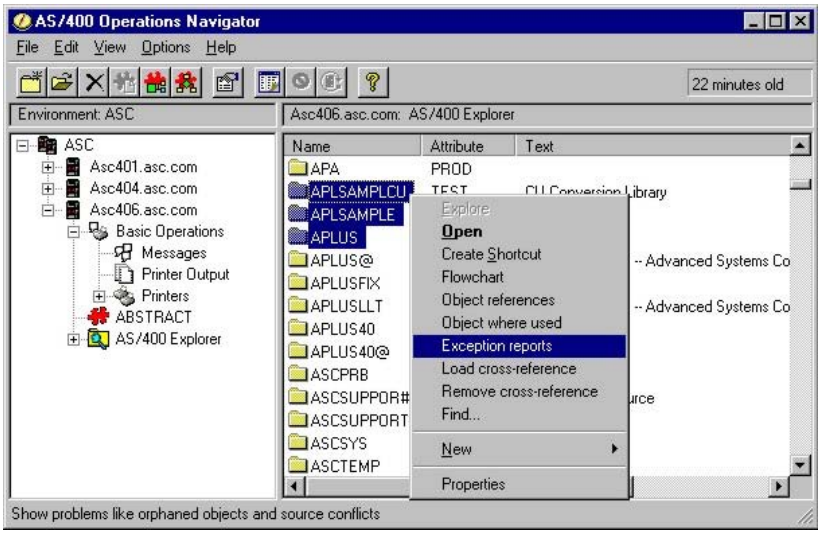
For more information on Exception Reports turn to page 28.

Before a library is loaded into the ABSTRACT cross-reference it's highly recommended that you run the Object Exception Report option to identify missing source or source conflicts within the library. Problems with source may corrupt the accuracy of the cross-reference information and should be resolved before the cross-reference is generated.

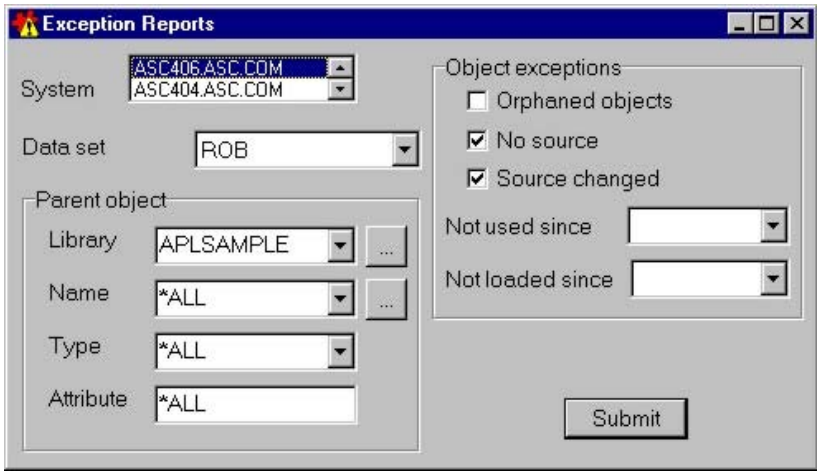
Start iSeries Navigator and select the **iSeries Explorer** node in the left-hand column. A list of libraries on the system will then be displayed in the right half of the window.



Scroll down the list of libraries and highlight the one(s) you want to include in the report. You may use the standard **Shift+click** or **Ctrl+click** conventions to select multiple libraries. After you make your selection, right click to display the pop-up menu shown on the next page.



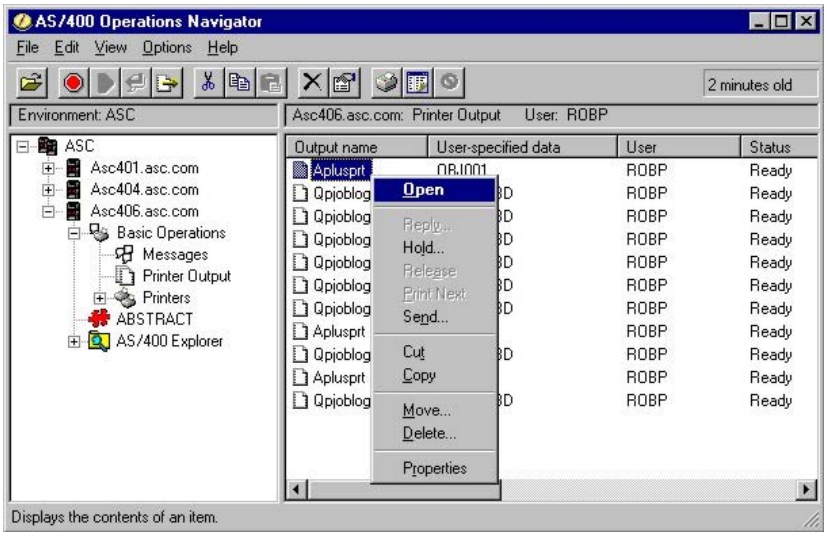
Click on the **Exception reports** option to display the window of the same name, below.



Make sure the **No Source** and **Source Changed** boxes are checked (they are, by default).

Click on the **Submit** button to submit the report. If you selected multiple libraries, the Exception Reports window will be redisplayed with the next library name in the selection displayed in the **Library** text box. Continue to click the **Submit** button until reports for all selected libraries have been submitted.

Click the **Basic iSeries** node on the left column of the **iSeries iSeries Navigator** window, then click on the **Printer Output** branch. This will display the list of spooled files on the system (next page).



Double-click on the exception report (named Aplusrpt) with your mouse to display the **Object Exception Report** in a viewer window, part of which is shown below. Items with a source conflict (source not found or changed since object creation) will be noted with an asterisk at the left margin of the report. The source file and member will be underlined if the source could not be found. The object creation date will be underlined if the source change date is more recent than the object creation date.

```
 8/04/99 13:34:28                                        ABSTRACT
                                                  Object Exception Report
Object name . . : *ALL
Object type . . : *ALL       Attribute  : *ALL
Object library  : APLSAMPLE
Options . . . . : *NOSRC          *SRCCHG

 Object     Type       Src file   Src lib    Src member Creation   Source     Last use   XRF load
*CUSTRPG    *PGM RPG   QRPGSRC    ZZZ        CUSTRPG   1997/03/11

  28 Objects analyzed        1 Objects have service data that points to wrong source
                             1 Objects have source that was changed after it was created
                            22 Objects have never been used
                            28 Objects are not loaded into cross-reference
                             0 Orphaned objects
                          8192 Bytes used by listed objects
```

Run the ABSTRACT Change Service Data (CHGSVCDTA) command from an iSeries command line to update the object's service description to correct any problems.

Step 7 - Load the Cross-Reference

For more information on loading the Cross-Reference refer to page 37.

Sign on to a PC workstation using security officer (QSECOFR) access authority or equivalent. ABSTRACT requires usage authority to all objects to be loaded into the cross-reference.

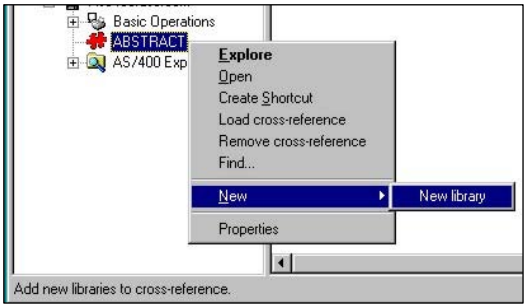
Start iSeries Navigator.

Use either the toolbar or pop-up menu:

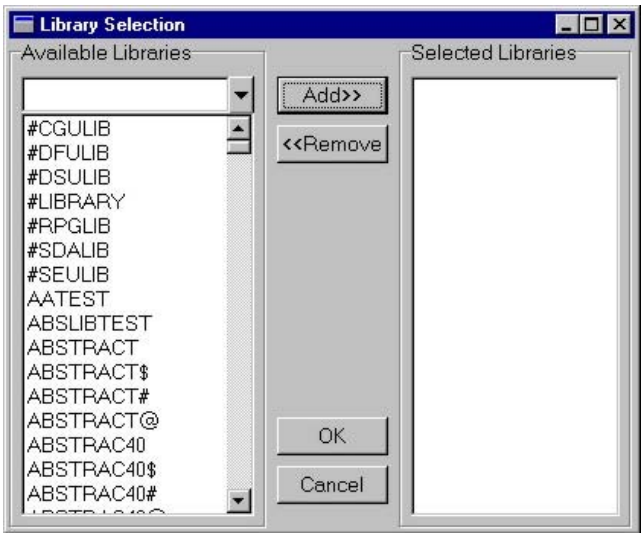
The ABSTRACT **New library** icon appears at the far left on the iSeries iSeries Navigator toolbar. **Click** on this icon.

OR

Right-click on any node of the iSeries Navigator tree to display the pop-up menu shown at right, then **click** on the **New** > **New Library** option.



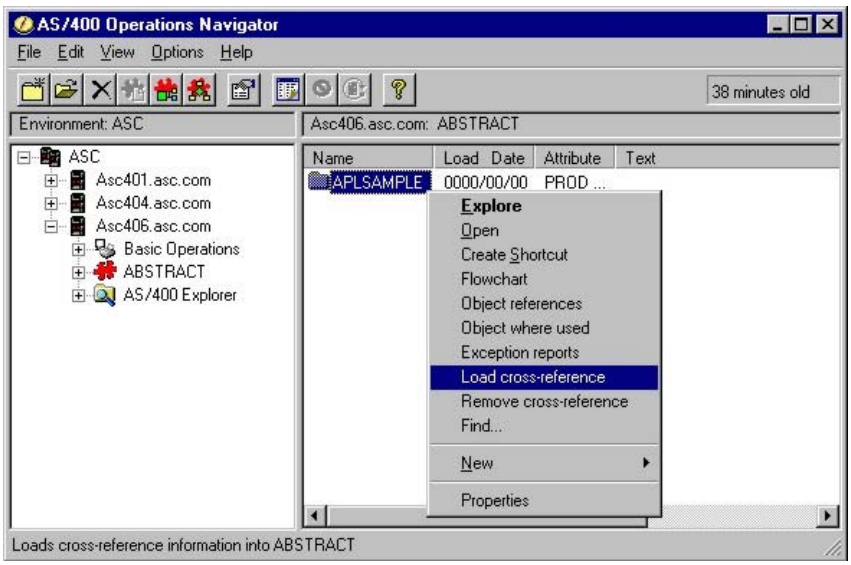
In either case, the ABSTRACT **Library Selection** window will be displayed (see next page).



Click on the drop-down arrow in the **Available Libraries** column (left column) to display a list of libraries on the system.

Use your mouse to highlight libraries in the left-hand column, then click the **Add>>** button to include them in the **Selected Libraries** column. Be sure to include all data, program and source libraries that you need to include in the cross-reference. You may add as many libraries as you need. Remove libraries from the Selected Libraries column by highlighting them and then clicking on the **<<Remove** button.

Click on the **OK** button when your selections are done. The library(s) you selected will now appear in the right half of the iSeries Navigator window (below). Note that the **Load Date** field next to each library name is set to 0000/00/00. This indicates the library has been selected, but has not been loaded into the cross-reference. You will need to submit a load request at this time.

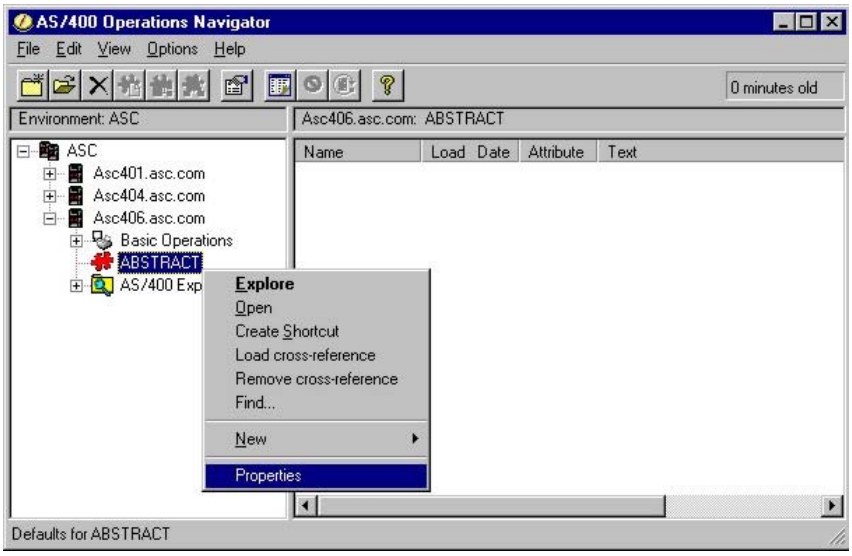


Select the libraries on the right half, then right-click to display the pop-up menu. Click on the **Load cross-reference** option to initiate a load request. (Alternately, you could right-click on the ABSTRACT node in the left half to display the pop-up menu. All libraries on the right half would automatically be included in the load request.)

When the cross-reference load request is completed you will be able to use all of ABSTRACT's powerful capabilities.

ABSTRACT Defaults

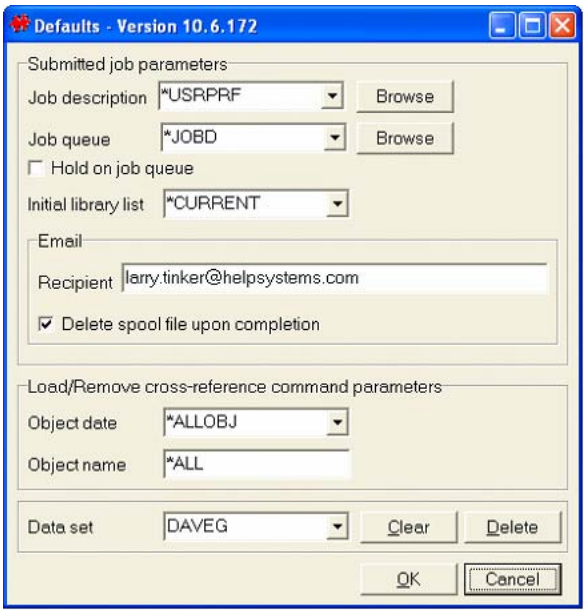
Right-click on the ABSTRACT node or any folder or object within to display the pop-up menu shown below.



Click on **Properties** to display the **Defaults** window. The Defaults window allows you to set the basic job stream and cross-reference defaults for the current user profile.

Job Description

Specify the job description to use for jobs submitted from ABSTRACT functions. You may type in a valid job description or click on the drop-down to display and select from a list of job descriptions available on the system. You may also click on the **Browse** button next to this entry field to display a list of job descriptions (with detailed descriptions) in the **Object Selection** window (page 27).



Job Queue

Specify the job queue to use for jobs submitted from ABSTRACT functions. You may type in a valid job queue or click on the drop-down to display and select from a list of job queues available on the system. You may also click on the **Browse** button next to this entry field to display a list of job queues (with detailed descriptions) in the **iSeries Object Selection** window (page 27).

Email Recipient

Specify an Email Recipient to receive reports generated by the find string feature, recreate database relations, and exception reports.

Hold on job queue

Click this button **ON** to hold ABSTRACT jobs in the queue.

Object date

The object creation date may be used to select objects loaded or removed from the cross-reference. Specify one of the following values:

<u>*ALLOBJ</u> - Load all objects, regardless of their creation date, into the cross-reference files.

***CHG –** Includes objects with a creation date that is more recent than the date of the cross-reference information, or if cross-reference information isn't found. If the object hasn't changed since its cross-reference information was loaded, it will be skipped.

Date - Specify a date in system date format. Objects created on or after this date will be passed to the analysis routines. When `TYPE(*SRCF)` loading is used, the member change date is examined and compared to this date.

Object name

The object name may be used to select objects loaded or removed from the cross-reference. Specify one of the following values:

<u>*ALL</u> - Each object in the library be analyzed.

Name - Specify the name of an object to be analyzed, or specify a generic name (one to nine characters suffixed with an asterisk) to indicate that all objects sharing the prefix will be analyzed.

Data set

ABSTRACT makes it easy to keep the documentation for several application sets independent, using different **data sets**. Whether you want to make a distinction between "production" and "development" applications, or to segregate the documentation for each application into different databases, ABSTRACT can accomplish your objective by allowing you to specify a data set name when the cross-reference load request is made. Information in the various data sets will remain independent throughout the analysis, reporting, and development phases of ABSTRACT use.

Use the drop-down to select one of the following values, or type a name in the entry field to create a new data set**. See the detailed instructions on the following page when working with multiple data sets.**

***FIRST** - All information is stored in ABSTRACT's default data set.

Name – Enter the name of a specific data set.

The active data set is displayed in the entry field. You may clear or delete the data set using the adjacent buttons:

Clear - Removes the contents of a data set but leaves the data set object intact so you can reload the cross-reference.

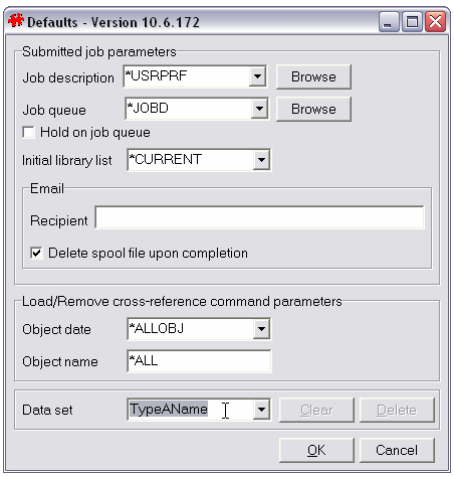
Delete - Deletes the data set and its contents entirely.

Select a Data Set

You will need to specify the active data set at the **Defaults** window. The default data set is ***FIRST**, and you can add, change, clear or delete data sets as your needs dictate. Follow the instructions below for these procedures.

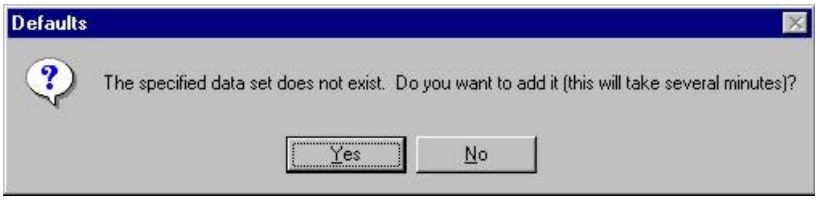
Add a New Data Set

Open the **Defaults** window as described on the previous page.



Position your cursor in the **Data set** text box and type in a new name.

Click **OK** to continue. A confirmation window (below) will be displayed to verify that you want to add the new data set.

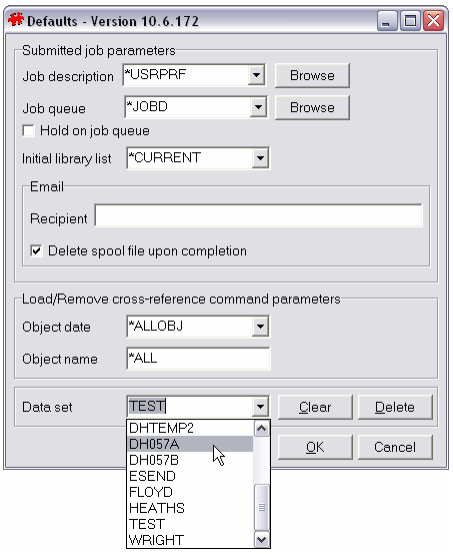


Click on the **Yes** button to continue. This process may take a few minutes as the new data set is created on the iSeries. Upon completion, the new data set will be active and will be redisplayed in the **Data set** text box.

Change the Data Set

Open the **Defaults** window.

Position your mouse arrow over the **Data set** dropdown arrow and click to display the list of available data sets (below).



Use your mouse to highlight the desired data set name within the dropdown list. Single-click to select the data set.

Click the **OK** button. The ABSTRACT hierarchy will be redisplayed to show the contents of this data set. This is now the active data set.

Clear or Delete the Data Set

Open the **Defaults** window.

Use the **Data set** dropdown to select the data set you want to clear or delete.

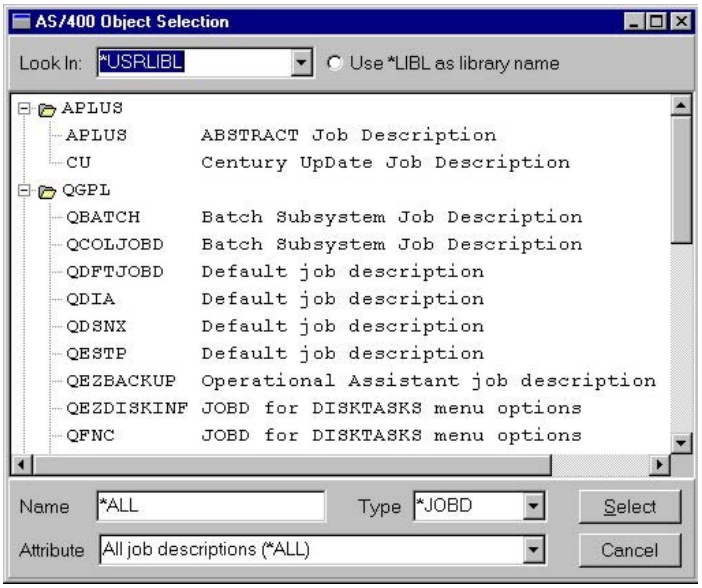
Click on the **Clear** or **Delete** button.

Clearing a data set removes the contents of a data set but leaves the data set object intact so you can reload the cross-reference.

Deleting a data set deletes the data set and its contents entirely.

Object Selection Window

The **Object Selection** window is displayed when you click on one of the **Browse** buttons used within a number of ABSTRACT windows. Its contents will correspond to the object type with which the Browse was associated. You will then be able to view a full selection of objects and select the one you need. The example below shows a list of job descriptions.



Look In:

This drop-down may be used to display and select available library list options. If a new library list is selected, the main window will be redisplayed to show the library(s) in the list.

(Main window)

Click on a new object in this window, then click the **Select** button to confirm the setting.

Name, Type, Attribute

Displays the name, type and attribute of the active setting.

Exception Reports

About Exception Reports

The **Exception Reports** option allows you to create reports that describe potential problems with your application or ABSTRACT cross-reference data. It is recommended to run the report and resolve conflicts before a library is loaded into the cross-reference to ensure correct source for the programs is analyzed and to avoid having to re-run the cross-reference load process.

Exception reports may contain the following information:

Orphaned objects - Objects in an application library that are not referenced by any object loaded into the ABSTRACT cross-reference database. Presumably, if it is not referenced, it is not used by your application.

Undocumented objects - Objects in an application library that have never been loaded into the ABSTRACT cross-reference database, or that haven't been loaded since a given date.

Object-source conflicts - Objects in your application library with service (compile-time) information that references a source member that cannot be located on your system, or that references a source member changed since the object was created.

Unused objects - Application objects that have never been used, or haven't been used since a given date. The OS/400 operating system updates an object's last used date whenever it is used. Refer to the Display Object Description (DSPOBJD) and Change Object Description (CHGOBJD) commands in the CL Reference Manual for a complete description of the object usage information and how it can be managed.

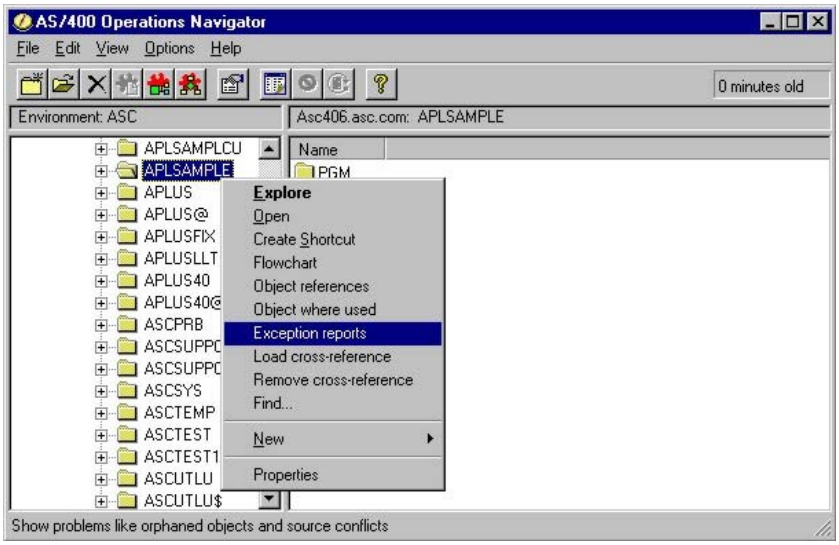
Most object-source conflicts result from poor object/source control methods that call for objects (and source) to be moved to the production library once development in a "test" library has been completed. Once the source code is moved, the corresponding object's service data (which identifies the location of the source) no longer points to the correct location. ABSTRACT will resolve these discrepancies on its own as long as you have loaded the libraries where the actual source resides into the cross-reference. If multiple versions of the same source exist for any given program, then it may be required to run the CHGSVCDTA command. The ABSTRACT Change Service Description (CHGSVCDTA) command can be used to update the object's service description to correct this problem.

Other object-source conflicts occur because of careless source management practices that allow source code to be changed without requiring subsequent compilation. The ABSTRACT program or file recompiler can be used to recreate these objects.

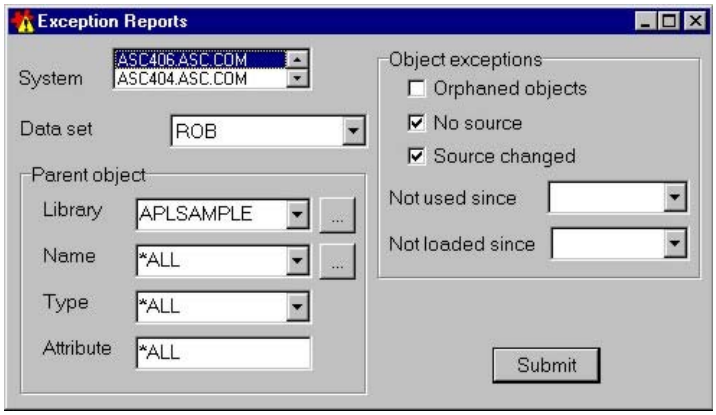
Report requests can be made so that one, some, or all of these types of reports are run at one time.

Submit the Object Exception Reports

Expand the **iSeries Explorer** node to display a list of all libraries on the iSeries. Scroll down and select the library for which you want to run reports. **Right-click** on the library folder to display the pop-up menu shown below.



Select the **Exception Reports** option at this menu to display the Exception Reports window. This window lets you specify the reporting options to be used in the request.



The **System** and **Library** settings default to those selected from the iSeries iSeries Navigator window. The **Data Set** originates from the Defaults setting. You may change any of these settings at this time.

Name

The ***Name*** entry lets you search the library for objects meeting the specified name values. If no library qualifier is specified, *LIBL is assumed and all libraries in the job's library list are searched for the objects.

<u>*ALL</u> - All objects in the specified library(s) are selected.

object-name - Only objects with this specific name will be included in the list.

***generic*-name** - Specify a partial object name qualified by an asterisk (*) to select several objects meeting the criteria. The following generic forms are allowed: `ABC*, *ABC, *ABC*, A*B, "abc*", **ALL`.

Type

The ***Type*** entry allows you to generate a report for all object types or a specific object type. An expanded description of the OBJTYPE parameter and a list of valid OS/400 object types can be found in Appendix A, Expanded Parameter Descriptions, of Volume 1 of the <u>CL Reference</u> manual.

<u>*ALL</u> - All objects regardless of their type are included.

object-type - Specify any valid system object type to display a list of all objects of that particular type.

***DBF** - All data base files will be included. These are *FILE objects with an attribute beginning with `PF`, `LF`, or `DDMF`.

Attribute

The ***Attribute*** entry can be used to include all object attributes or only specific ones in the exception report. An expanded description of the OBJATR parameter and a list of valid OS/400 object attributes can be found in Appendix A, Expanded Parameter Descriptions, of Volume 1 of the CL Reference manual.

***ALL** - All objects in the specified library(s) are selected regardless of their attribute definition.

***BLANK** - Only objects with no attribute will be included.

object-attribute - Only objects with this specific attribute will be included in the list. `PF-DTA` can be used to select only non-source physical files. `PF-SRC` can be used to select only source physical files.

***generic*-name** - Specify a partial object attribute qualified by an asterisk (*) to select several objects meeting the criteria. The following generic forms are allowed: `ABC*, *ABC, *ABC*, A*B, "abc*", **ALL.` For example, `*RPG*` will select RPG, RPG36, RPG38, and SQLRPG objects. Refer to Appendix G, General Functions, in Volume 1 of the CL Reference manual for more information about generic names.

Orphaned Objects

The ***Orphaned Objects*** entry includes objects in an application library that are not referenced by any object loaded into the ABSTRACT cross-reference database. Presumably, if it is not referenced, it is not used by your application.

No Source

The ***No Source*** entry includes objects with service information that indicates a source file, library, or member does not exist on the system. ABSTRACT cross-reference information is not used in preparing this report.

Source Changed

The ***Source Changed*** entry is used to include objects where the member change date is subsequent to the date indicated by the service

information. Cross-reference information is not used to prepare this report.

Not Used Since

The ***Not used since*** entry specifies additional selection criteria for the objects to be included on the report. Current object usage information will be examined to find objects that qualify.

***NOUSE:** objects that have never been used will be included on the report in addition to those already selected due to the default parameter values.

date-value: objects that have not been used since the specified date will be included on the report in addition to those already selected due to the default parameter values.

Not Loaded Since

The ***Not loaded since*** entry allows you to apply selection criteria based on the date that ABSTRACT cross-reference information was created. Objects that have been changed, or have not been loaded into the cross-reference since the date specified on the LSTLOD parameter will be listed on the report in addition to those already selected due to the default parameter values.

***NOLOD:** objects that have never been loaded into the ABSTRACT cross-reference files will be listed.

date-value: objects that have not been loaded into the ABSTRACT cross-reference database since the specified date will be included.

Report Information

The **Object Exception** report will include the following information:

- Object name, type, and attribute
- Source file, library, and member used to create the object
- Object creation date
- Source change date (if source can be located)
- Date the object was last used
- Cross-reference load date (if loaded)
- Object text
- Object size
- Object source change date (last recompile date)

Items with a source conflict (source not found or changed since object creation) will be noted with an asterisk at the left margin of the report. The source file and member will be underlined if the source could not be found. The object creation date will be underlined if the source change date is more recent than the object creation date.

Orphaned objects (unreferenced within the cross-reference database) will be noted with an asterisk next to the cross-reference load date.

The examples on the following pages show the different types of exceptions that can appear on the report.

Source Conflicts

An asterisk at the left margin indicates that either the source could not be located, or it has changed since the object was created.

Source Conflicts

```
  Object     Type       Src file   Src lib    Src member Creation Source   Last use XRF load  Object text
  ADDTIME    *PGM RPG   QRPGSRC    PILOT#     ADDTIME    11/07/88 11/07/88
* AUTOSCH    *PGM RPG   QRPGSRC    PILOT#     AUTOSCH    05/13/91 02/19/92          02/13/92
* AUTOSCHC   *PGM CLP   QCLSRC     PILOT#     AUTOSCHC   05/13/91 02/19/92          02/24/92  AutoSch command
        .
        .
        .
* MCR010     *PGM MI    QIRPSRC    PILOT#     MCR010     11/13/89          03/04/92
* MCR030     *PGM MI    QIRPSRC    PILOT#     MCR030     11/09/88                          *
        .
        .
        .
* PJOBLOG    *FILE PRTF QDDSSRC    PILOT#     PJOBLOG    03/17/86          06/11/90
```

Programs AUTOSCH and AUTOSCHC demonstrate a conflict between the creation date of the program and the last change date of the source member. These conflicts will be highlighted by an underlined creation date.

Programs MCR010 and MCR030 show that the original source code can no longer be found on the system. The source member used to create the PJOBLOG file couldn't be found either, though its source file could. The underlined source file and member names indicate that source could not be located. Either it does not exist, or you are not authorized to it.

Orphaned Objects

An asterisk to the right of the load date informs you of an orphaned object exception. There are no references to the object in any of the ABSTRACT cross-references. Although this would be expected for the first program in a job stream, most objects that are really used in your application should have at least one reference to them.

```
  Object     Type       Src file   Src lib    Src member Creation Source   Last use XRF load  Object text
  SCH001     *PGM RPG   QRPGSRC    PILOT#     SCH001     11/15/90 11/15/90 03/04/92 02/13/92  Schedule loader
  SEC900     *PGM RPG   QRPGSRC    PILOT#     SEC900     10/17/89 10/17/89 03/04/92 02/13/92*
  SEC901     *PGM CLP   QCLSRC     PILOT#     SEC901     10/26/88 10/26/88                  *
  SNDMSG     *PGM CLP   QCLSRC     PILOT#     SNDMSG     10/26/88 10/26/88 03/04/92         *
* WCBT01     *PGM MI    QIRPSRC    PILOT#     WCBT01     11/07/88          03/04/92           Work control bl
  PILOT      *OUTQ                                       02/06/85          03/04/92         *
  DOCMSG     *MSGF                                       07/31/90                           *
```

Orphaned Objects

The sample report indicates that no ABSTRACT references could be found for programs SEC900, SEC901 and SNDMSG (among others).

Unused Objects

A missing last use date indicates that the OS400 operating system has never detected this object being referenced.

Unused Objects

	Object	Type	Src file	Src lib	Src member	Creation	Source	Last use	XRF load	Object text
	ADDTIME	*PGM RPG	QRPGSRC	PILOT#	ADDTIME	11/07/88	11/07/88			
*	AUTOSCH	*PGM RPG	QRPGSRC	PILOT#	AUTOSCH	05/13/91	02/19/92		02/13/92	
*	AUTOSCHC	*PGM CLP	QCLSRC	PILOT#	AUTOSCHC	05/13/91	02/19/92		02/24/92	AutoSch command
	CHKJOBC	*PGM CLP	QCLSRC	PILOT#	CHKJOBC	10/26/88	10/26/88	03/04/92		Validate jobq,j
	CHKUSRC	*PGM CLP	QCLSRC	PILOT#	CHKUSRC	10/26/88	10/26/88	03/04/92		Check user prof

Undocumented Objects

A missing cross-reference load date indicates that the object has never been processed by the cross-reference load.

	Object	Type	Src file	Src lib	Src member	Creation	Source	Last use	XRF load	Object text
	ADDTIME	*PGM RPG	QRPGSRC	PILOT#	ADDTIME	11/07/88	11/07/88			
*	AUTOSCH	*PGM RPG	QRPGSRC	PILOT#	AUTOSCH	05/13/91	02/19/92		02/13/92	
*	AUTOSCHC	*PGM CLP	QCLSRC	PILOT#	AUTOSCHC	05/13/91	02/19/92		02/24/92	AutoSch command
	CHKJOBC	*PGM CLP	QCLSRC	PILOT#	CHKJOBC	10/26/88	10/26/88	03/04/92		Validate jobq,j
	CHKUSRC	*PGM CLP	QCLSRC	PILOT#	CHKUSRC	10/26/88	10/26/88	03/04/92		Check user prof

Undocumented Objects

Loading the Cross-Reference Database

Overview

The ABSTRACT cross-reference database is created and maintained through a batch process known as *initialization*. The complete initialization process involves many individual steps. The initialization function can be performed on an entire library or on specific objects - the scope of each initialization request can be specified when it is made.

In addition, ABSTRACT makes it easy to keep the documentation for several application sets independent. Whether you want to make a distinction between “production” and “development” applications, or to segregate the documentation for each application into different databases, ABSTRACT can accomplish your objective by allowing you to specify a “data set” name when the initialization request is made. Information in the various data sets will remain independent throughout the analysis, reporting, and development phases of ABSTRACT use.

The initialization process uses a combination of iSeries system functions and ABSTRACT routines to document the objects in your applications. Some of these functions analyze iSeries objects and the information contained in them - files, programs, commands, menus, etc. Other functions require access to the program source code for your applications. ABSTRACT can determine the source code used to create a program by referencing the service data in the program object description. An initialization option will direct ABSTRACT to locate the source code for the program and automatically analyze it after the object level information has been documented. Once the source code has been located, ABSTRACT can determine the HLL program usage of data fields and program calls that will complete the documentation database. If the object’s service data is incorrect, the library containing the actual location of the source needs to be included in the cross-reference.

After the initialization phase is complete, the ABSTRACT cross-reference database will contain a wealth of information that’s available to you through its wide array of on-line analysis and reporting functions.

Object Authority Requirements

All ABSTRACT initialization functions respect the object authority constraints in place in your environment. As a result, ABSTRACT cannot create cross-reference information for objects on which it does not have proper authority. The ABSTRACT initialization procedure can only load objects if the user profile under which the initialization occurs

has operational rights (*USE) to objects and operational and read rights (also *USE) for the application source files.

Unfortunately, it is possible to *believe* that objects are being analyzed and placed into the cross- references when, in fact, they are not. Fortunately, you may review which objects are loaded into the cross-references by exploring the ABSTRACT node of the iSeries Navigator window.

One of the easiest methods to avoid accidental exclusion of application objects from the cross-reference database is to run the initialization procedure from the security officer (QSECOFR) user profile.

If you determine that some of your application objects have not been placed into the cross- reference files, you can take one of two actions. You can have someone with proper authority grant the necessary rights for the objects which are currently restricted to the user profile that will be running the initialization procedure. Alternatively, you can run the initialization procedure under a different user profile; one having the necessary authority to the application objects and source code.

Cross-Reference Initialization Steps

The following actions take place when the ABSTRACT cross-reference is initialized, or loaded:

Load database relationships

Load file descriptions

Cross-reference hysical files and trigger programs

Cross-reference commands and their CPPs

Load program-file relationships

Analyze CL program objects (refer to source if ALWRTVSRC(*NO))

Load service programs

Load module objects

Load binding directories

Load job descriptions

Analyze menu objects

Analyze OS/400 query definitions

Analyze subsystem descriptions

Analyze SEQUEL views

Locate HLL source and determine field usage and HLL CALLs

Information in the Cross-reference

When the initialization process is complete, the cross-reference database will contain the following information:

- Database structure
- Physical/logical relationships
- Format definitions
- Field definitions
- Flow of control
- Program-program transfers among CL and HLL programs/modules
- User profile initial program
- Command processing program (CPP) access
- Menu access to programs and commands
- Subsystem routing entry programs
- Trigger programs
- Data flow
- File object manipulations
- Program-file relationships
- Program-field relationships
- Database use by SEQUEL views
- Database use by OS/400 query definitions
- Object usage
- Command parameter references within programs
- Command parameter references within job descriptions (user profile, queues, etc.)
- Repository Information
- Binding directories
- Service programs

Library Selection

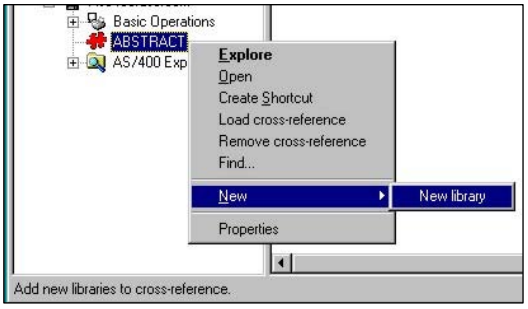
This is the recommended library selection process, as all the selected libraries are loaded in the same initialization request and ABSTRACT will analyze all objects in the proper sequence: (1) data files, (2) program objects, and (3) source files.

Use either the toolbar or pop-up menu:

The ABSTRACT **New library** icon appears at the far left on the iSeries iSeries Navigator toolbar. **Click** on this icon.

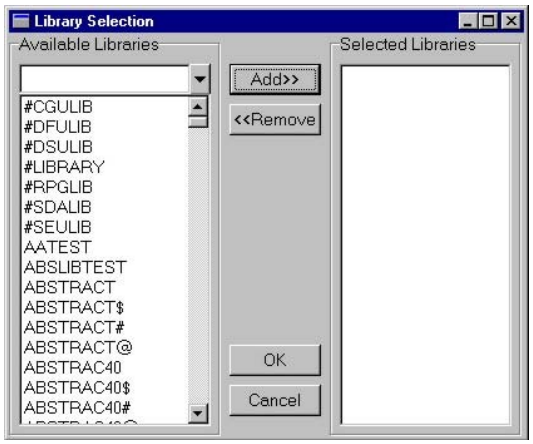
OR

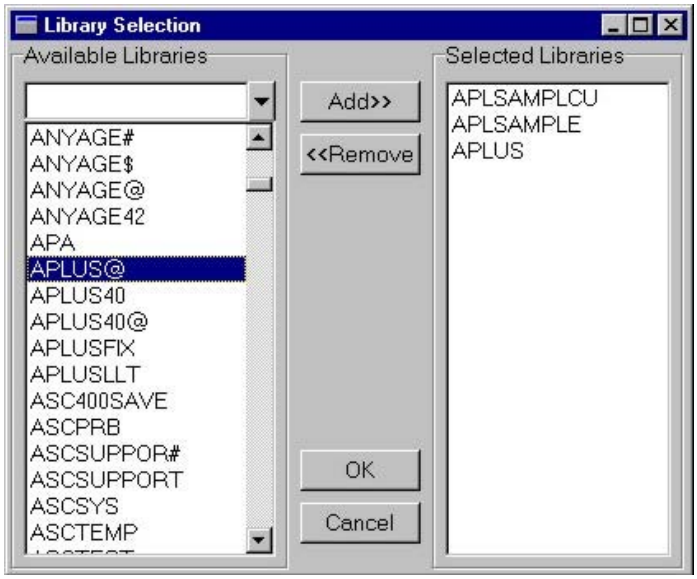
Right-click on any node of the iSeries Navigator tree to display the pop-up menu shown at right, then **click** on the **New > New Library** option.



In either case, the ABSTRACT **Library Selection** window will be displayed (below).

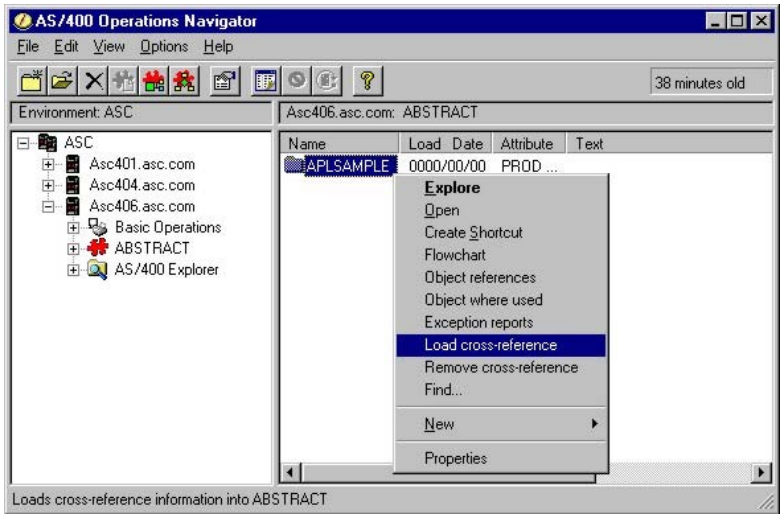
Click on the drop-down arrow in the **Available Libraries** column (left column) to display a list of libraries on the system.





Use your mouse to highlight libraries in the left-hand column, then click the **Add>>** button to include them in the **Selected Libraries** column. You may use the **Ctrl + Click** and **Shift + Click** conventions to select multiple libraries at once. Be sure to include all data, program and source libraries that you need to include in the cross-reference. You may add as many libraries as you need. Remove libraries from the Selected Libraries column by highlighting them and then clicking on the **<<Remove** button.

Click on the **OK** button when your selections are done. The library(s) you selected will appear in the right half of the iSeries Navigator window inside the ABSTRACT node. Note that the **Load Date** field next to each library name is set to 0000/00/00. This indicates the library has been selected, but has not been loaded into the cross-reference. You will need to submit a load request at this time.

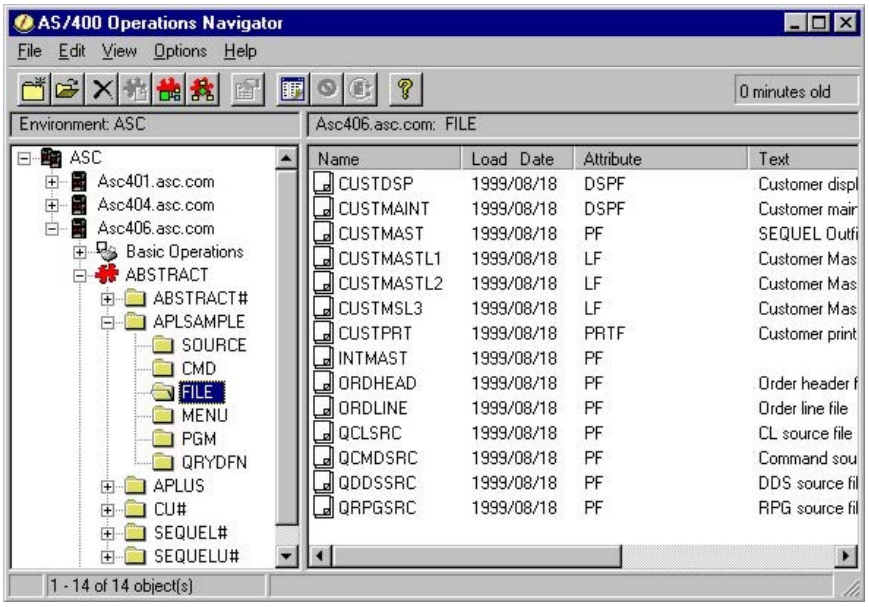


Right-click on the ABSTRACT node to display the pop-up menu, then click on **the Load cross-reference** option. (Selecting the ABSTRACT

node ensures that all the libraries are loaded in the proper sequence.) Alternately, you can select libraries on the right-hand side and load them individually. Keep in mind that if you are loading libraries one at a time, libraries that depend on other libraries will need to be reloaded as well. If you have any doubts about this process, it a good idea to reload the entire cross-reference to make sure all libraries are loaded in the proper sequence.

Work with Cross-Reference Data

When the **ABSTRACT** node is expanded within the iSeries Navigator window, the list of libraries loaded into the cross-reference will be displayed as folders.



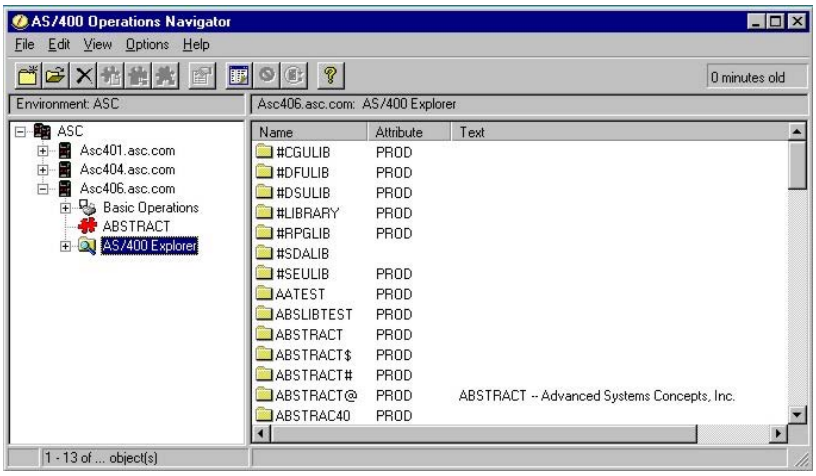
Double-click any of the library folders to list object sub-folders representing the different object types loaded into the library.

Double-click any of the object sub-folders to display a list of those objects in the right panel of the iSeries Navigator window.

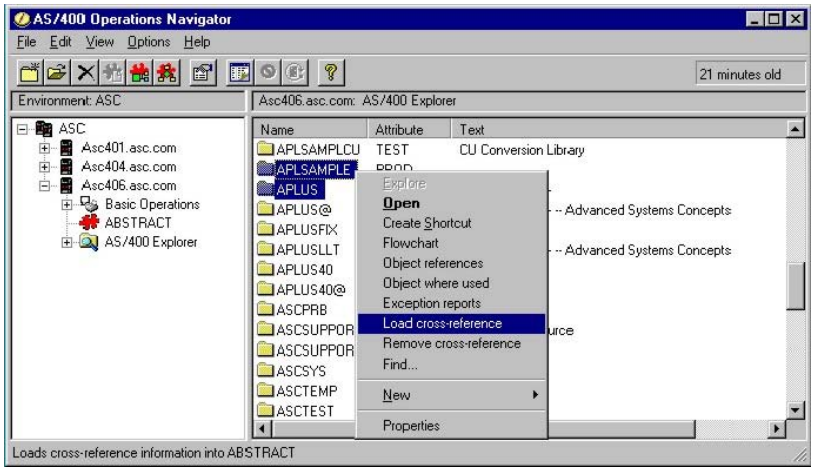
Select Library(s) or Objects from AS/400 Explorer

Load Entire Library(s)

Start iSeries iSeries Navigator and click on the **AS/400 Explorer** node. This node is installed with ABSTRACT and contains a complete list of libraries on your system. The libraries will be displayed in the right panel of the window.



Scroll down the list of libraries on the right and use your mouse to select the ones you want to include in the cross-reference. Again, you need to load the libraries in the proper sequence: (1) data files, (2) program objects, and (3) source files. You may use the standard **Shift+click** or **Ctrl+click** conventions to select multiple libraries. After you make your selection, right click to display the pop-up menu shown below.



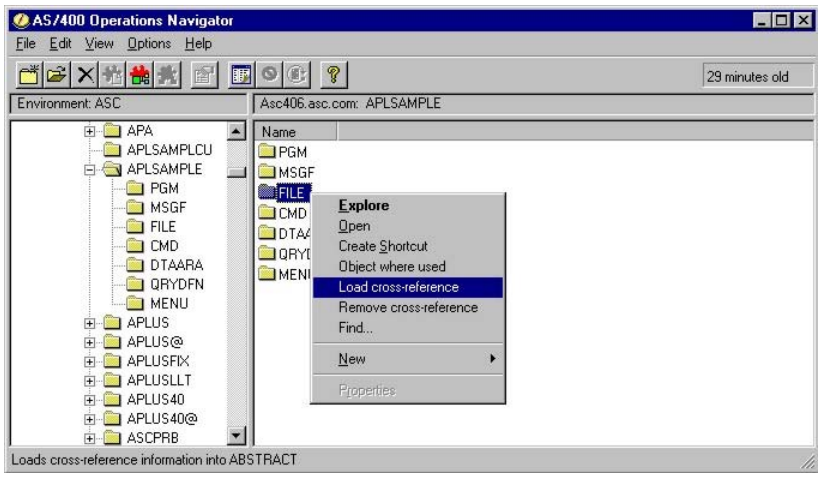
Single-click the **Load cross-reference** option. This submits a job that loads the library(s) into the cross-reference using the parameters you entered earlier in the **Defaults** window.

Load Specific Object Types

Alternately, you may wish to load only certain object types within a library. This can be done as follows:

Start iSeries iSeries Navigator and click on the **iSeries Explorer** node. Expand the iSeries Explorer node by clicking on the ⊞.

Scroll down the list of libraries on the left column and use your mouse to click on the library containing the objects you want to load. A list of object type folders will appear on the right column.

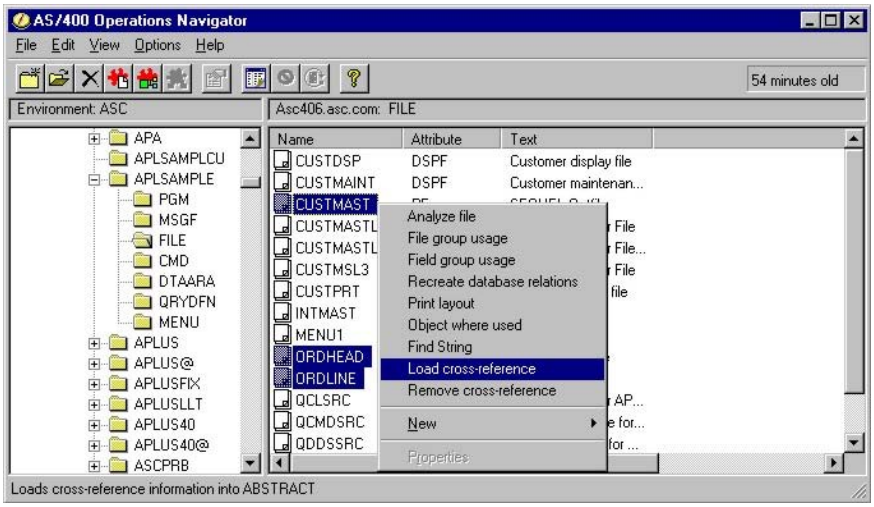


Click on the object types that you want to include in the cross-reference. You may use the standard **Shift+click** or **Ctrl+click** conventions to select multiple object types. After you make your selection, right click to display the pop-up menu shown above.

Click on the **Load cross-reference** option to submit the load request.

Load Specific Objects

You may also load specific objects into the cross-reference by navigating through the iSeries Explorer node and making your choices at the object level. The example below shows three files in the APLSAMPLE library being loaded into the cross-reference.



Remove Libraries from the Cross-reference

Right click on a library folder in the left panel of the iSeries Navigator window to display a pop-up menu, then select **Remove Cross-reference**

–or-

From the iSeries Navigator Menu, Select File > Remove Cross-reference.

(To be more specific when removing cross-reference data, choose any of the object subfolders before using the steps above.)

Reload Existing Cross-reference Information

Right-click on the ABSTRACT Node or any library folder and select **Load Cross-reference** from the pop-up menu

–or-

Click the **Load Cross-reference Button**  on the iSeries Navigator tool bar**.**

Object Relations

Overview

ABSTRACT provides two major cross-referencing functions:

- **Object references -** Objects used by another object
- **Object usage** – Object "where used"

Object reference and object usage information are also referred to as object relation functions.

Object relationships are presented on displays using a hierarchical 'tree' approach, similar to other iSeries Navigator displays. The object relation displays provide you with a true object-action mechanism of working with the various elements of your software applications.

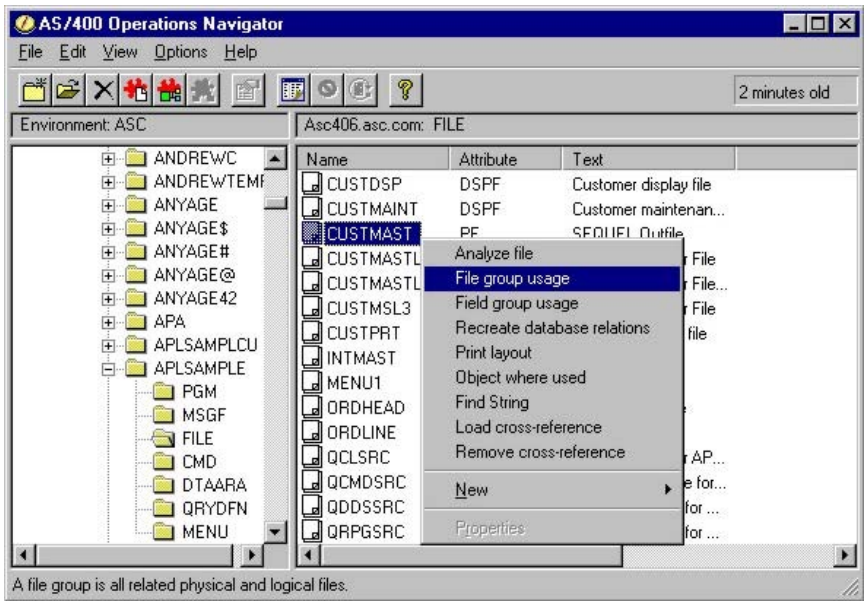
Every object relation display functions in a similar manner. A consistent set of options and functions are available regardless of the type of information presented on the display. This makes ABSTRACT easy to use - once you master the concepts involved with the object relation displays, you will feel "at home" on any of the ABSTRACT windows.

This section will explain the features and functions of the object relation displays. The examples use displays generated using object references, although the other object relationship displays work in a similar fashion.

How to View Object Relations

Object relationship information may be displayed for any objects and fields loaded into the cross-reference from the File Systems and Database nodes within iSeries Navigator or ABSTRACT windows. These options may be initiated from pop-up menus available by right-clicking on an object. The options allowed will depend on the object type selected.

For example, to display file group usage for a file in the cross-reference, expand the Explorer node and right-click on the file to display the pop-up window shown below.



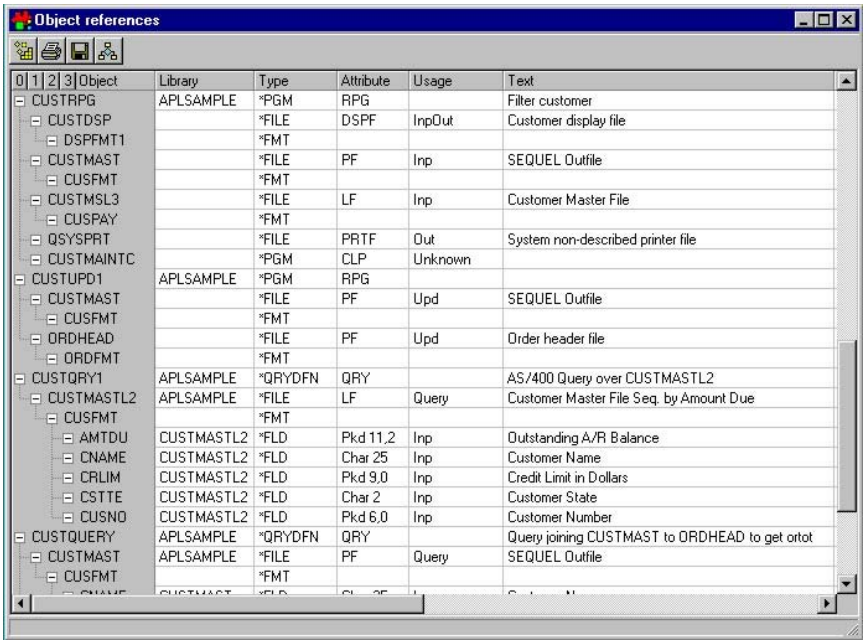
Click on the **File Group Usage** option. After a slight pause the **File Group Where Used** window will be displayed for the file. In this example, the relationships for the CUSTMAST file are displayed.

File group where used

0 | 1 | 2 Object	Library	Type	Attribute	Usage	Text	Extended
CUSTMAST		*FILE	PF		SEQUEL Outfile	Lib(*LIBL)
CUSTMSL3	APLSAMPLE	*FILE	LF		Customer Master File	
CUSTMASTL1	APLSAMPLE	*FILE	LF		Customer Master File	
CUSTMASTL2	APLSAMPLE	*FILE	LF		Customer Master File Seq. by Amount Due	
CUSNOCHGC	APLSAMPLE	*PGM	CLP	OVRDBF		Seq#(13)
CUSTCOPY	APLSAMPLE	*PGM	RPG	InpOut		
CUSTLIST	APLSAMPLE	*PGM	RPG	Inp		
CUSTRPG	APLSAMPLE	*PGM	RPG	Inp	Filter customer	
CUSTUPD1	APLSAMPLE	*PGM	RPG	Upd		
CUSTMAST	APLSAMPLE	*FILE	PF		SEQUEL Outfile	
CUSTMAST		*FILE	PF		SEQUEL Outfile	Lib(*LIBL)
CUSNOCHGC	APLSAMPLE	*PGM	CLP	OVRDBF		Src(APLSAMPLE/CUSTMA
CUSTQUERY	APLSAMPLE	*QRYDFN	QRY	Query	Query joining CUSTMAST to ORDHEAD to get ortot	
CUSTMASTL1	APLSAMPLE	*FILE	LF		Customer Master File	
CUSTMASTL2	APLSAMPLE	*FILE	LF		Customer Master File Seq. by Amount Due	
CUSTQRY1	APLSAMPLE	*QRYDFN	QRY	Query	AS/400 Query over CUSTMASTL2	
CUSTMSL3		*FILE	LF		Customer Master File	Lib(*LIBL)
CUSTRPG	APLSAMPLE	*PGM	RPG	Inp	Filter customer	

Elements of the Object Relation Display

The object relation display shows the objects included in your selection criteria and any related information from the cross-references that were built during the ABSTRACT initialization phase. The example below shows a display presented by an object reference request.



Object references

0 1 2 3 Object	Library	Type	Attribute	Usage	Text
CUSTRPG	APLSAMPLE	*PGM	RPG		Filter customer
CUSTDSP		*FILE	DSPF	InpOut	Customer display file
DSPFMT1		*FMT			
CUSTMAST		*FILE	PF	Inp	SEQUEL Outfile
CUSFMT		*FMT			
CUSTMSL3		*FILE	LF	Inp	Customer Master File
CUSPAY		*FMT			
QSYSPRT		*FILE	PRTF	Out	System non-described printer file
CUSTMAINTC		*PGM	CLP	Unknown	
CUSTUPD1	APLSAMPLE	*PGM	RPG		
CUSTMAST		*FILE	PF	Upd	SEQUEL Outfile
CUSFMT		*FMT			
ORDHEAD		*FILE	PF	Upd	Order header file
ORDFMT		*FMT			
CUSTQRY1	APLSAMPLE	*QRYDFN	QRY		AS/400 Query over CUSTMASTL2
CUSTMASTL2	APLSAMPLE	*FILE	LF	Query	Customer Master File Seq. by Amount Due
CUSFMT		*FMT			
AMTDU	CUSTMASTL2	*FLD	Pkd 11,2	Inp	Outstanding A/R Balance
CNAME	CUSTMASTL2	*FLD	Char 25	Inp	Customer Name
CRLIM	CUSTMASTL2	*FLD	Pkd 9,0	Inp	Credit Limit in Dollars
CSTTE	CUSTMASTL2	*FLD	Char 2	Inp	Customer State
CUSNO	CUSTMASTL2	*FLD	Pkd 6,0	Inp	Customer Number
CUSTQUERY	APLSAMPLE	*QRYDFN	QRY		Query joining CUSTMAST to ORDHEAD to get ortot
CUSTMAST	APLSAMPLE	*FILE	PF	Query	SEQUEL Outfile
CUSFMT		*FMT			

Parent objects

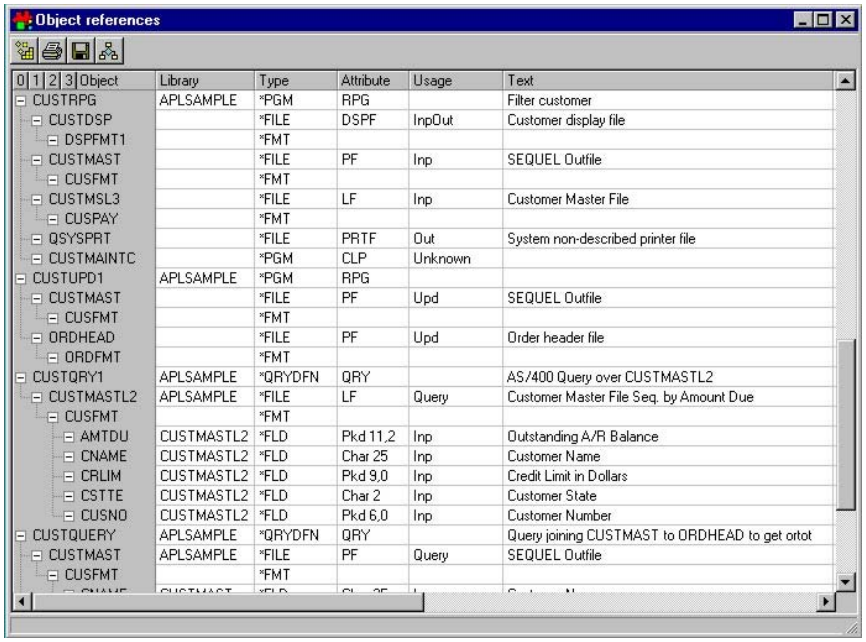
The body of the display presents object relationship information in alphabetical order, by type, for each object in the list specified by the LIB, OBJ, OBJTYPE, and OBJATR values of your request. These “parent” objects are listed at the leftmost (not indented) positions of the list.

Indented objects

Items related to the objects within the selection criteria are shown (indented) on branches beneath their “parent” object. You may expand or collapse the branches by clicking on the ‘-‘ or ‘+’ boxes. Indented objects are ordered alphabetically by type, just as in a DSPLIB listing. The object type, usage, and text are also shown for each parent and related item.

Object reference displays (like the one above) show the objects that are referenced by parent objects in the list. **The indented object is used by the parent**.

Object usage displays show the counterpart to the reference display - the indented objects are those that reference the parent object**. The parent objects are used by those indented beneath them**.



Object references

0 1 2 3 Object	Library	Type	Attribute	Usage	Text
CUSTRPG	APLSAMPLE	*PGM	RPG		Filter customer
CUSTDSP		*FILE	DSPF	InpOut	Customer display file
DSPFMT1		*FMT			
CUSTMAST		*FILE	PF	Inp	SEQUEL Outfile
CUSFMT		*FMT			
CUSTMSL3		*FILE	LF	Inp	Customer Master File
CUSPAY		*FMT			
QSYSPRT		*FILE	PRTF	Out	System non-described printer file
CUSTMAINTC		*PGM	CLP	Unknown	
CUSTUPD1	APLSAMPLE	*PGM	RPG		
CUSTMAST		*FILE	PF	Upd	SEQUEL Outfile
CUSFMT		*FMT			
ORDHEAD		*FILE	PF	Upd	Order header file
ORDFMT		*FMT			
CUSTQRY1	APLSAMPLE	*QRYDFN	QRY		AS/400 Query over CUSTMASTL2
CUSTMASTL2	APLSAMPLE	*FILE	LF	Query	Customer Master File Seq. by Amount Due
CUSFMT		*FMT			
AMTDU	CUSTMASTL2	*FLD	Pkd 11,2	Inp	Outstanding A/R Balance
CNAME	CUSTMASTL2	*FLD	Char 25	Inp	Customer Name
CRLIM	CUSTMASTL2	*FLD	Pkd 9,0	Inp	Credit Limit in Dollars
CSTTE	CUSTMASTL2	*FLD	Char 2	Inp	Customer State
CUSNO	CUSTMASTL2	*FLD	Pkd 6,0	Inp	Customer Number
CUSTQUERY	APLSAMPLE	*QRYDFN	QRY		Query joining CUSTMAST to ORDHEAD to get ortot
CUSTMAST	APLSAMPLE	*FILE	PF	Query	SEQUEL Outfile
CUSFMT		*FMT			

For instance, the object reference window above shows that an RPG program named CUSTRPG in the APLSAMPLE library references files named CUSTDSP, CUSTMAST, CUSTMSL3, and QSYSPRT. For each of these files, the display also shows what fields and formats are explicitly referenced by CUSTRPG. Other fields and/or formats that exist in those files but are not referenced will not be shown. The display also shows that program CUSTRPG makes an unqualified call to a program named CUSTMAINTC.

Type

Indicates the object type.

Attribute

Programs, files and queries will display the specific type of program or file. For fields, the type and length of the field will be displayed.

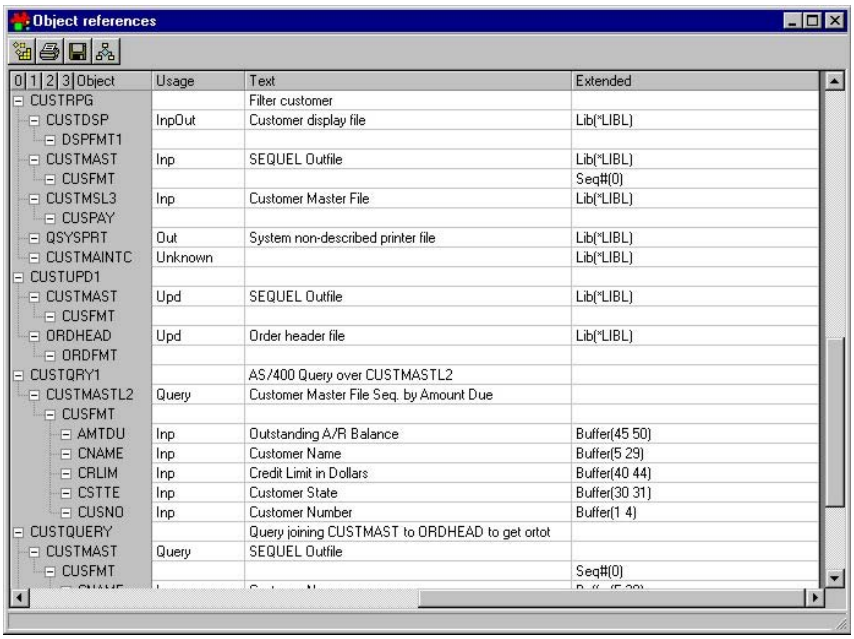
Usage

The source analysis determines how a program is using a file or field, whether for input, output or both.

Text

If an object contains descriptive text, it will appear in this column.

Extended Description



Object references

0 1 2 3 Object	Usage	Text	Extended
CUSTRPG		Filter customer	
CUSTDSP	InpOut	Customer display file	Lib(*LIBL)
DSPFMT1			
CUSTMAST	Inp	SEQUEL Outfile	Lib(*LIBL)
CUSFMT			Seq#(0)
CUSTMSL3	Inp	Customer Master File	Lib(*LIBL)
CUSPAY			
QSYSPRT	Out	System non-described printer file	Lib(*LIBL)
CUSTMAINTC	Unknown		Lib(*LIBL)
CUSTUPD1			
CUSTMAST	Upd	SEQUEL Outfile	Lib(*LIBL)
CUSFMT			
ORDHEAD	Upd	Order header file	Lib(*LIBL)
ORDFMT			
CUSTQRY1		AS/400 Query over CUSTMASTL2	
CUSTMASTL2	Query	Customer Master File Seq. by Amount Due	
CUSFMT			
AMTDU	Inp	Outstanding A/R Balance	Buffer(45 50)
CNAME	Inp	Customer Name	Buffer(5 29)
CRLIM	Inp	Credit Limit in Dollars	Buffer(40 44)
CSTTE	Inp	Customer State	Buffer(30 31)
CUSNO	Inp	Customer Number	Buffer(1 4)
CUSTQUERY		Query joining CUSTMAST to ORDHEAD to get ortot	
CUSTMAST	Query	SEQUEL Outfile	
CUSFMT			Seq#(0)

The object reference display includes a column labeled "**Extended**" (extended description) to the far right of the window. It contains information that is specific to the type of item referenced, as follows:

Lib( ) - If the ABSTRACT object resolution located an object in a library other than that indicated by the CL source, the original library qualifier will be listed

Seq( ) - The source sequence number in the CL source for the item reference. If the reference occurs several times, multiple sequence numbers will be listed.

Src( ) - The source file ABSTRACT analyzed to acquire the information.

Mbr( ) - The member named in the CL command (ADDPFM, FMTDTA, OVRDBF, ...) or the HLL source member used during the analysis.

CblFld( ) - Long field name for fields used in COBOL programs

FldAtr( ) - Alternate descriptions for the field if program described and there is more than one form of the field.

Buffer( ) - The starting and ending positions for a field within the indicated format.

Object Resolution

The object subtype (RPG, PF, etc.), text, and the library shown in the **Library** column for the referenced items on the display is acquired on a real-time basis according to the following search algorithm:

If the cross-reference item is qualified, search the **qualifying library** name.

If the item cannot be found in the indicated library, or if it is unqualified or qualified by *LIBL, search the library of the **parent object** shown on the display.

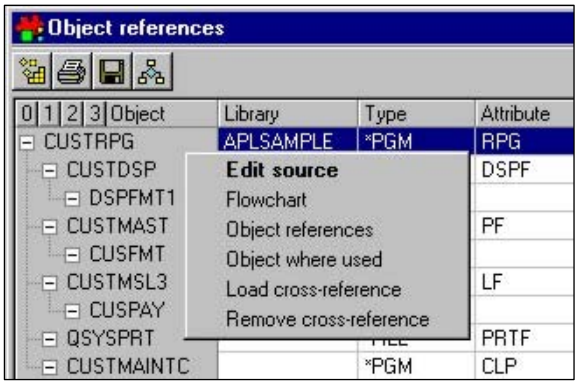
If not found, search the current job's **library list**.

Finally, search the **cross-reference files** for any qualified reference to the named object. Search the library indicated by the qualifier listed in the cross-reference file.

If an existing object still cannot be found following this search, the library name and text columns are left blank on the display and the object type column will not include an attribute.

Object Relation Options

Each object in an object relation display has context-dependent options that can be run against it. Right-click on an object to display and select



the available options.

The following options are available for **all** object types:

- Edit Source (if it's an editable object and you have a compatible edit tool. See the Edit Source section on page 94 for more information on allowed edit tools.)
- Object where used (page 56)
- Load cross-reference (page 37)
- Remove cross-reference (page 47)

These options are also available for **programs**, **commands**, and **menus**:

- Flowchart (page 69)
- Object references (page 59)

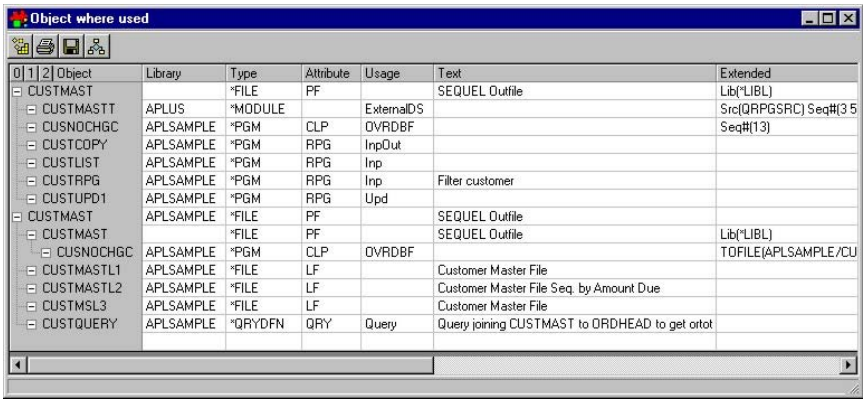
These options are also available for **files**:

- Analyze file (page 73)
- File group usage (page 57)
- Field group usage (page 58)

Types of Object Relation Displays

Object Where Used

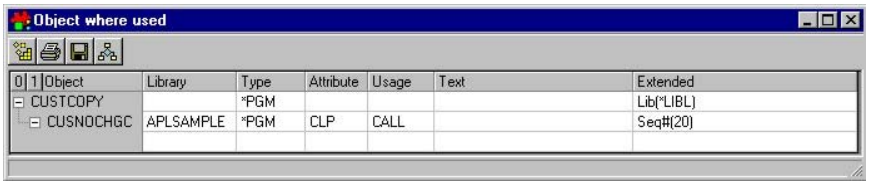
As the name of this option readily suggests, an **Object Where Used** display shows wherever a selected object is used. The first example below shows the usage for a physical file named CUSTMAST. There are two primary branches for the CUSTMAST file. The first branch shows all usage within the library list of the user profile that submitted the initialization process, as indicated by the value **Lib(*LIBL)** in the **Extended** column. The second branch shows usage only within the APLSAMPLE library, which was the only library actually loaded into the cross-reference during the initialization process.



Object where used

0 1 2 Object	Library	Type	Attribute	Usage	Text	Extended
CUSTMAST		*FILE	PF		SEQUEL Outfile	Lib(*LIBL)
CUSTMASTT	APLUS	*MODULE		ExternalDS		Src(QRPGSRC) Seq#(3 5
CUSNOCHGC	APLSAMPLE	*PGM	CLP	OVRDBF		Seq#(13)
CUSTCOPY	APLSAMPLE	*PGM	RPG	InpOut		
CUSTLIST	APLSAMPLE	*PGM	RPG	Inp		
CUSTRPG	APLSAMPLE	*PGM	RPG	Inp	Filter customer	
CUSTUPD1	APLSAMPLE	*PGM	RPG	Upd		
CUSTMAST	APLSAMPLE	*FILE	PF		SEQUEL Outfile	
CUSTMAST		*FILE	PF		SEQUEL Outfile	Lib(*LIBL)
CUSNOCHGC	APLSAMPLE	*PGM	CLP	OVRDBF		TOFILE(APLSAMPLE/CU
CUSTMASTL1	APLSAMPLE	*FILE	LF		Customer Master File	
CUSTMASTL2	APLSAMPLE	*FILE	LF		Customer Master File Seq. by Amount Due	
CUSTMSL3	APLSAMPLE	*FILE	LF		Customer Master File	
CUSTQUERY	APLSAMPLE	*QRYDFN	QRY	Query	Query joining CUSTMAST to ORDHEAD to get ortot	

The next example shows that the RPG program CUSTCOPY is used once, in a call from a CL program called CUSNOCHGC.

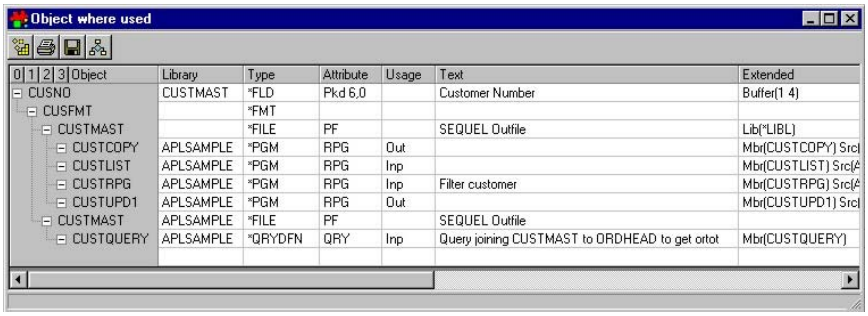


Object where used

0 1 Object	Library	Type	Attribute	Usage	Text	Extended
CUSTCOPY		*PGM				Lib(*LIBL)
CUSNOCHGC	APLSAMPLE	*PGM	CLP	CALL		Seq#(20)

Field Where Used

A **Field Where Used** display shows where a selected field is used, as in



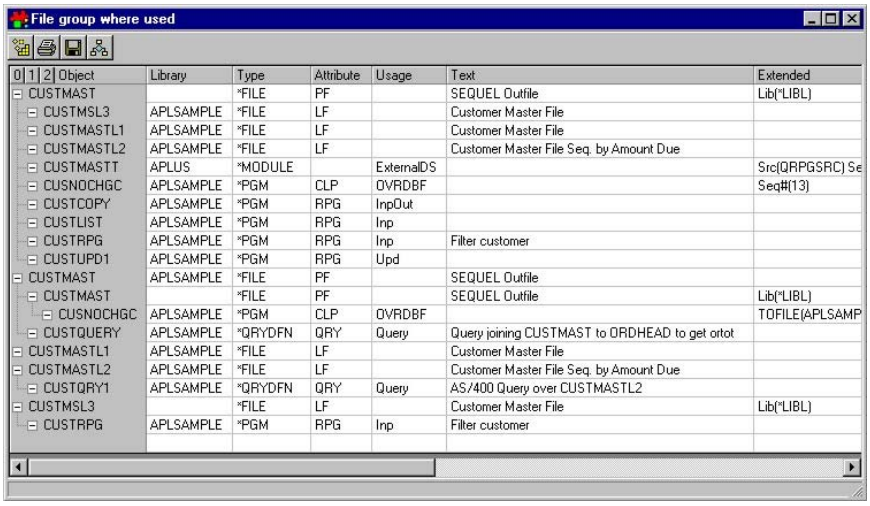
Object where used

0 1 2 3 Object	Library	Type	Attribute	Usage	Text	Extended
CUSNO	CUSTMAST	*FLD	Pkd 6,0		Customer Number	Buffer(1 4)
CUSFMT		*FMT				
CUSTMAST		*FILE	PF		SEQUEL Outfile	Lib(*LIBL)
CUSTCOPY	APLSAMPLE	*PGM	RPG	Out		Mbr(CUSTCOPY) Srcl
CUSTLIST	APLSAMPLE	*PGM	RPG	Inp		Mbr(CUSTLIST) Srcl(A
CUSTRPG	APLSAMPLE	*PGM	RPG	Inp	Filter customer	Mbr(CUSTRPG) Srcl(A
CUSTUPD1	APLSAMPLE	*PGM	RPG	Out		Mbr(CUSTUPD1) Srcl
CUSTMAST	APLSAMPLE	*FILE	PF		SEQUEL Outfile	
CUSTQUERY	APLSAMPLE	*QRYDFN	QRY	Inp	Query joining CUSTMAST to ORDHEAD to get ortot	Mbr(CUSTQUERY)

the example below for a field named CUSNO.

File Group Usage

Usage information for a physical file and its related logical files is shown in a **File Group Usage** display, as the following example shows for the CUSTMAST file.

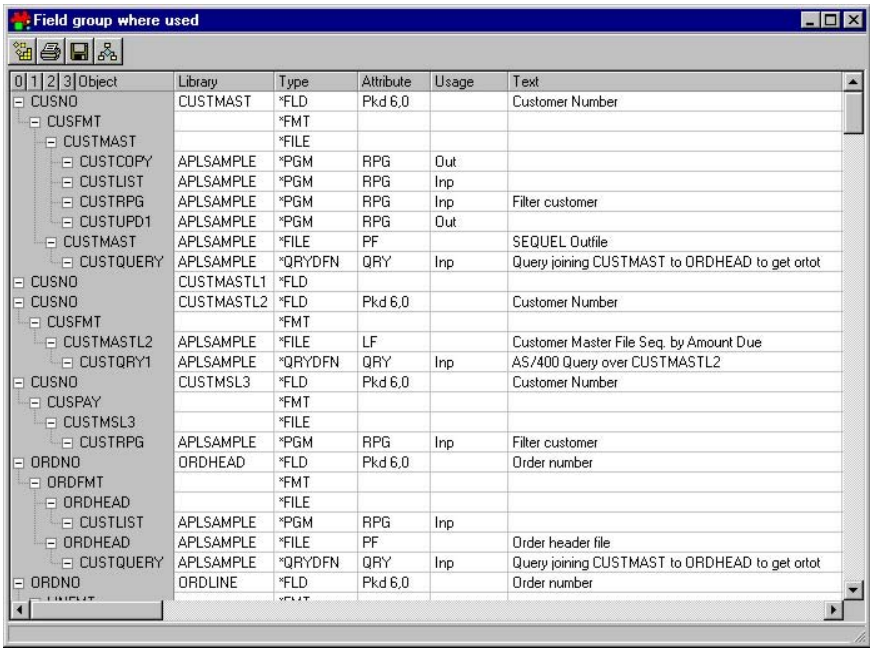


File group where used

0 1 2 Object	Library	Type	Attribute	Usage	Text	Extended
CUSTMAST		*FILE	PF		SEQUEL Outfile	Lib(*LIBL)
CUSTMSL3	APLSAMPLE	*FILE	LF		Customer Master File	
CUSTMASTL1	APLSAMPLE	*FILE	LF		Customer Master File	
CUSTMASTL2	APLSAMPLE	*FILE	LF		Customer Master File Seq. by Amount Due	
CUSTMASTT	APLUS	*MODULE		ExternalDS		Src(QRPGSRC) Se
CUSNOCHGC	APLSAMPLE	*PGM	CLP	OVRDBF		Seq#(13)
CUSTCOPY	APLSAMPLE	*PGM	RPG	InpOut		
CUSTLIST	APLSAMPLE	*PGM	RPG	Inp		
CUSTRPG	APLSAMPLE	*PGM	RPG	Inp	Filter customer	
CUSTUPD1	APLSAMPLE	*PGM	RPG	Upd		
CUSTMAST	APLSAMPLE	*FILE	PF		SEQUEL Outfile	
CUSTMAST		*FILE	PF		SEQUEL Outfile	Lib(*LIBL)
CUSNOCHGC	APLSAMPLE	*PGM	CLP	OVRDBF		TOFILE(APLSAMP
CUSTQUERY	APLSAMPLE	*QRYDFN	QRY	Query	Query joining CUSTMAST to ORDHEAD to get ortot	
CUSTMASTL1	APLSAMPLE	*FILE	LF		Customer Master File	
CUSTMASTL2	APLSAMPLE	*FILE	LF		Customer Master File Seq. by Amount Due	
CUSTQRY1	APLSAMPLE	*QRYDFN	QRY	Query	AS/400 Query over CUSTMASTL2	
CUSTMSL3		*FILE	LF		Customer Master File	Lib(*LIBL)
CUSTRPG	APLSAMPLE	*PGM	RPG	Inp	Filter customer	

Field Group Usage

A **Field Group Usage** display, similar to File Group Usage, shows usage information for a physical file and its related logical files, only at a more detailed field level.



Field group where used

0 1 2 3 Object	Library	Type	Attribute	Usage	Text
CUSNO	CUSTMAST	*FLD	Pkd 6,0		Customer Number
CUSFMT		*FMT			
CUSTMAST		*FILE			
CUSTCOPY	APLSAMPLE	*PGM	RPG	Out	
CUSTLIST	APLSAMPLE	*PGM	RPG	Inp	
CUSTRPG	APLSAMPLE	*PGM	RPG	Inp	Filter customer
CUSTUPD1	APLSAMPLE	*PGM	RPG	Out	
CUSTMAST	APLSAMPLE	*FILE	PF		SEQUEL Outfile
CUSTQUERY	APLSAMPLE	*QRYDFN	QRY	Inp	Query joining CUSTMAST to ORDHEAD to get ortot
CUSNO	CUSTMASTL1	*FLD			
CUSNO	CUSTMASTL2	*FLD	Pkd 6,0		Customer Number
CUSFMT		*FMT			
CUSTMASTL2	APLSAMPLE	*FILE	LF		Customer Master File Seq. by Amount Due
CUSTQRY1	APLSAMPLE	*QRYDFN	QRY	Inp	AS/400 Query over CUSTMASTL2
CUSNO	CUSTMSL3	*FLD	Pkd 6,0		Customer Number
CUSPAY		*FMT			
CUSTMSL3		*FILE			
CUSTRPG	APLSAMPLE	*PGM	RPG	Inp	Filter customer
ORDNO	ORDHEAD	*FLD	Pkd 6,0		Order number
ORDFMT		*FMT			
ORDHEAD		*FILE			
CUSTLIST	APLSAMPLE	*PGM	RPG	Inp	
ORDHEAD	APLSAMPLE	*FILE	PF		Order header file
CUSTQUERY	APLSAMPLE	*QRYDFN	QRY	Inp	Query joining CUSTMAST to ORDHEAD to get ortot
ORDNO	ORDLINE	*FLD	Pkd 6,0		Order number

Object References

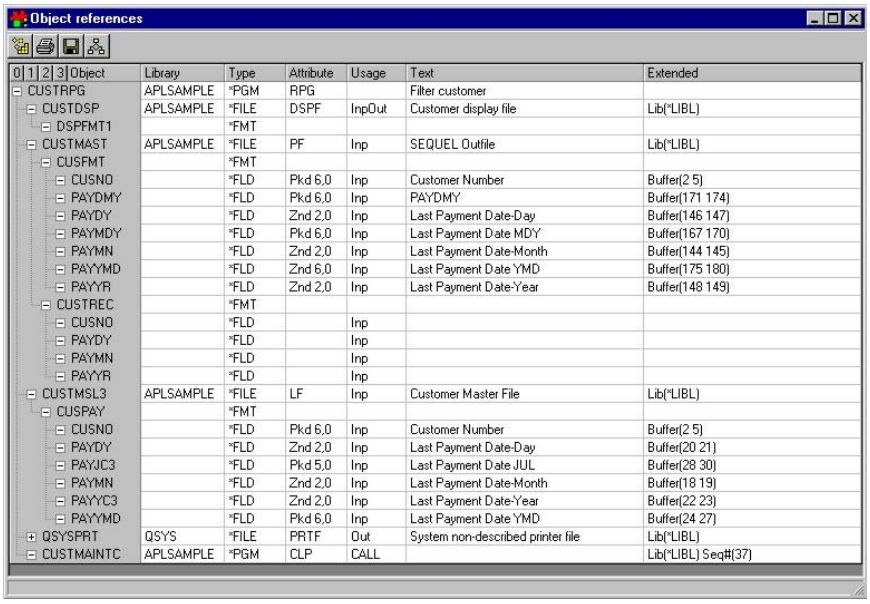
Object references displays show the objects used by a selected object. The display below shows the files and fields used by an RPG program named CUSTCOPY. Note that field usage within the program (input/output) is also identified.

Object references

0 1 2 3 Object	Library	Type	Attribute	Usage	Text	Extended
CUSTCOPY	APLSAMPLE	*PGM	RPG			
CUSTMAST		*FILE	PF	InpOut	SEQUEL Outfile	Lib(*LIBL)
CUSFMT		*FMT				
AMTDU	CUSTMAST	*FLD	Pkd 11,2	Out	Outstanding A/R Balance	Buffer(121 126)
CADD1	CUSTMAST	*FLD	Char 25	Out	Customer Address Line 1	Buffer(30 54)
CADD2	CUSTMAST	*FLD	Char 25	Out	Customer Address Line 2	Buffer(55 79)
CADD3	CUSTMAST	*FLD	Char 16	Out	Customer Address Line 3	Buffer(80 95)
CNAME	CUSTMAST	*FLD	Char 25	Out	Customer Name	Buffer(5 29)
CPHON	CUSTMAST	*FLD	Pkd 10,0	Out	Phone Number	Buffer(108 113)
CRLIM	CUSTMAST	*FLD	Pkd 9,0	Out	Credit Limit in Dollars	Buffer(116 120)
CSTTE	CUSTMAST	*FLD	Char 2	Out	Customer State	Buffer(96 97)
CTYPE	CUSTMAST	*FLD	Char 2	Out	Customer Type	Buffer(114 115)
CUSNO	CUSTMAST	*FLD	Pkd 6,0	Out	Customer Number	Buffer(1 4)
CZIPC	CUSTMAST	*FLD	Char 10	Out	Customer Zip Code	Buffer(98 107)
DFTWH	CUSTMAST	*FLD	Char 2	Out	Default Warehouse	Buffer(164 165)
HIGHB	CUSTMAST	*FLD	Pkd 9,0	Out	Highext A/R Balance	Buffer(132 136)
MTD$C	CUSTMAST	*FLD	Pkd 9,2	Out	Month to Date Sales	Buffer(149 153)
OROPN	CUSTMAST	*FLD	Pkd 9,2	Out	Total Open Orders in Dollars	Buffer(127 131)
PAYAM	CUSTMAST	*FLD	Pkd 11,2	Out	Last Payment Amount	Buffer(137 142)
PAYDMY	CUSTMAST	*FLD	Pkd 6,0	Out		Buffer(170 173)
PAYDY	CUSTMAST	*FLD	Znd 2,0	Out	Last Payment Date-Day	Buffer(145 146)
PAYJUL	CUSTMAST	*FLD	Pkd 5,0	Out	Last Payment Date JUL	Buffer(180 182)
PAYMDY	CUSTMAST	*FLD	Pkd 6,0	Out	Last Payment Date MDY	Buffer(166 169)
PAYMN	CUSTMAST	*FLD	Znd 2,0	Out	Last Payment Date-Month	Buffer(143 144)
PAYYMD	CUSTMAST	*FLD	Znd 6,0	Out	Last Payment Date YMD	Buffer(174 179)

Expand or Collapse the List Outline

You can selectively expand or collapse individual nodes in the list to show only the information you want to see. For example, if you weren't interested in what fields QSYSPRT (a print file) was using, but did what want to see field usage for the database files, you could click on the minus sign next to QSYSPRT to collapse all information under that node.



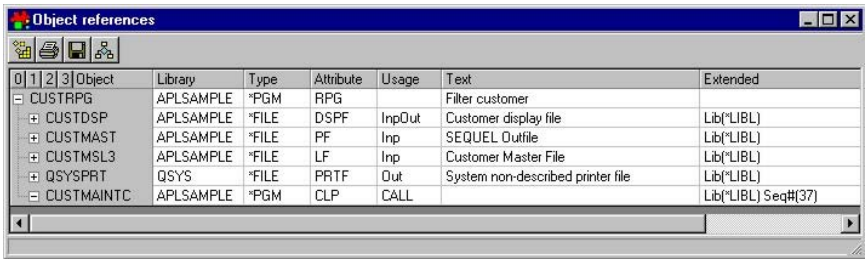
Object references

0 1 2 3 Object	Library	Type	Attribute	Usage	Text	Extended
CUSTRPG	APLSAMPLE	*PGM	RPG		Filter customer	
CUSTDSP	APLSAMPLE	*FILE	DSPF	InpOut	Customer display file	Lib(*LIBL)
DSPFMT1		*FMT				
CUSTMAST	APLSAMPLE	*FILE	PF	Inp	SEQUEL Outfile	Lib(*LIBL)
CUSFMT		*FMT				
CUSNO		*FLD	Pkd 6,0	Inp	Customer Number	Buffer(2 5)
PAYDMY		*FLD	Pkd 6,0	Inp	PAYDMY	Buffer(171 174)
PAYDY		*FLD	Znd 2,0	Inp	Last Payment Date-Day	Buffer(146 147)
PAYMDY		*FLD	Pkd 6,0	Inp	Last Payment Date MDY	Buffer(167 170)
PAYMN		*FLD	Znd 2,0	Inp	Last Payment Date-Month	Buffer(144 145)
PAYYMD		*FLD	Znd 6,0	Inp	Last Payment Date YMD	Buffer(175 180)
PAYYR		*FLD	Znd 2,0	Inp	Last Payment Date-Year	Buffer(148 149)
CUSTREC		*FMT				
CUSNO		*FLD		Inp		
PAYDY		*FLD		Inp		
PAYMN		*FLD		Inp		
PAYYR		*FLD		Inp		
CUSTMSL3	APLSAMPLE	*FILE	LF	Inp	Customer Master File	Lib(*LIBL)
CUSPAY		*FMT				
CUSNO		*FLD	Pkd 6,0	Inp	Customer Number	Buffer(2 5)
PAYDY		*FLD	Znd 2,0	Inp	Last Payment Date-Day	Buffer(20 21)
PAYJC3		*FLD	Pkd 5,0	Inp	Last Payment Date JUL	Buffer(28 30)
PAYMN		*FLD	Znd 2,0	Inp	Last Payment Date-Month	Buffer(18 19)
PAYYC3		*FLD	Znd 2,0	Inp	Last Payment Date-Year	Buffer(22 23)
PAYYMD		*FLD	Pkd 6,0	Inp	Last Payment Date YMD	Buffer(24 27)
QSYSPRT	QSYS	*FILE	PRTF	Out	System non-described printer file	Lib(*LIBL)
CUSTMAINTC	APLSAMPLE	*PGM	CLP	CALL		Lib(*LIBL) Seq#(37)

The QSYSPRT node (near the bottom) then shows a plus sign next to it to indicate that cross-reference information is available that has been hidden. Subsequently clicking on the node will redisplay the hidden information.

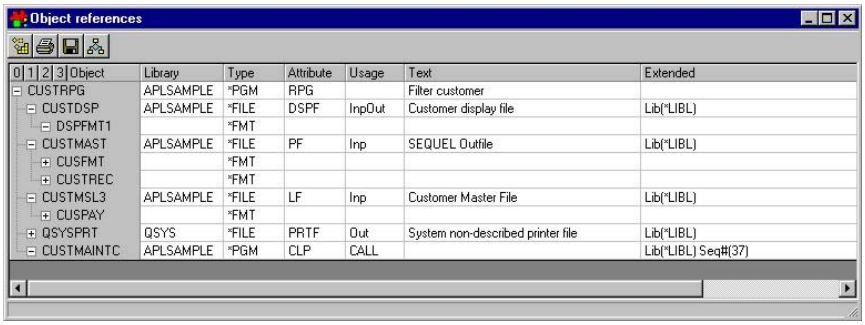
Level Buttons

You can also subset the object relations by using the **level buttons** that appear above the object names in the list. Clicking on any of the numbered buttons collapses or expands all nodes at that level. For example, clicking on the level 1 button will first collapse all of the file references (CUSTDSP, CUSTMAST, and QSYSPRT), hiding the format information at level 2 and the field information at level 3.



0 1 2 3 Object	Library	Type	Attribute	Usage	Text	Extended
CUSTRPG	APLSAMPLE	*PGM	RPG		Filter customer	
CUSTDSP	APLSAMPLE	*FILE	DSPF	InpOut	Customer display file	Lib(*LIBL)
CUSTMAST	APLSAMPLE	*FILE	PF	Inp	SEQUEL Outfile	Lib(*LIBL)
CUSTMSL3	APLSAMPLE	*FILE	LF	Inp	Customer Master File	Lib(*LIBL)
QSYSPRT	QSYS	*FILE	PRTF	Out	System non-described printer file	Lib(*LIBL)
CUSTMAINTC	APLSAMPLE	*PGM	CLP	CALL		Lib(*LIBL) Seq#(37)

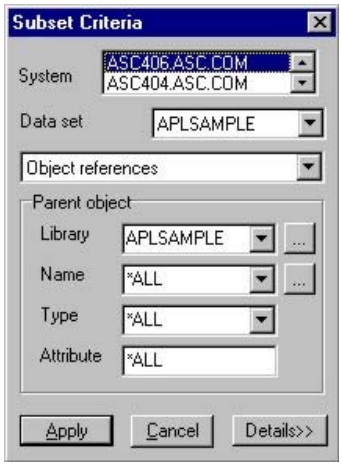
Clicking on the level 2 button will collapse all of the format references and hide all of the field information at level 3 for the database files.



0 1 2 3 Object	Library	Type	Attribute	Usage	Text	Extended
CUSTRPG	APLSAMPLE	*PGM	RPG		Filter customer	
CUSTDSP	APLSAMPLE	*FILE	DSPF	InpOut	Customer display file	Lib(*LIBL)
DSPFMT1		*FMT				
CUSTMAST	APLSAMPLE	*FILE	PF	Inp	SEQUEL Outfile	Lib(*LIBL)
CUSFMT		*FMT				
CUSTREC		*FMT				
CUSTMSL3	APLSAMPLE	*FILE	LF	Inp	Customer Master File	Lib(*LIBL)
CUSPAY		*FMT				
QSYSPRT	QSYS	*FILE	PRTF	Out	System non-described printer file	Lib(*LIBL)
CUSTMAINTC	APLSAMPLE	*PGM	CLP	CALL		Lib(*LIBL) Seq#(37)

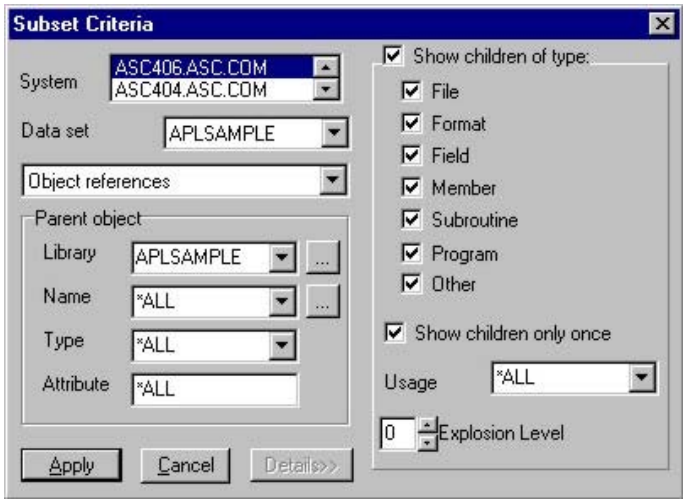
Object Relation Subset

The contents of the object list can be changed through the subset window. Click the **New Subset Button** to view and/or change the subset criteria.



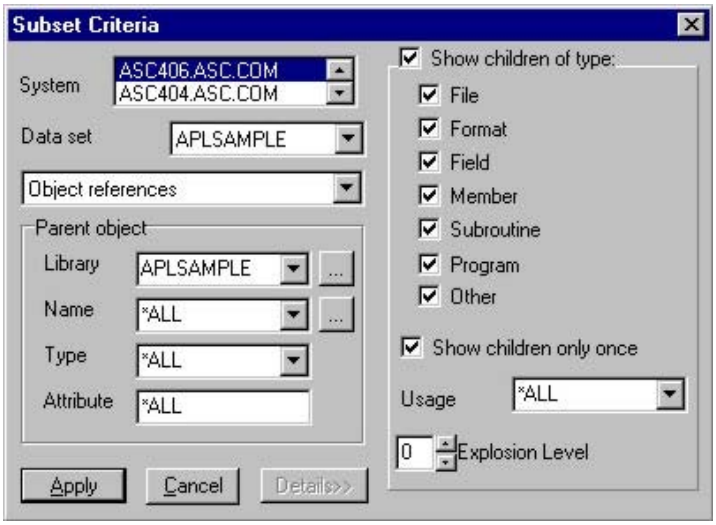
Specify the objects to be included in the list by setting the appropriate values for library, name, type, and attribute at the top of the display.

The subset window above shows the current system, data set, type of object relationship, and object name criteria used in selecting the "parent" objects on the display. Click on the **Details>>** button to show additional options, below, that allow you customize the contents of the display.

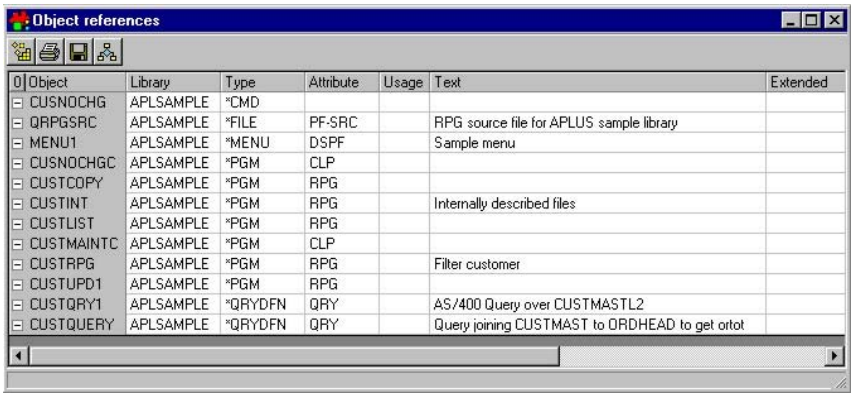


Show Children of Type

After clicking the **Details>>** button, you can specify additional criteria that control what indented objects are shown.



The **Show Children of Type** check box controls what object relations are shown. Unchecking this option shows a list display of parent objects only. You can also control what specific object types are shown as child references by checking or unchecking the File, Format, Field, Member, Subroutine, and Program options. Checking the 'Other' option controls whether any other object types other than those explicitly listed (like commands or data areas) are shown.



0 Object	Library	Type	Attribute	Usage	Text	Extended
CUSNOCHG	APLSAMPLE	*CMD				
QRPGSRC	APLSAMPLE	*FILE	PF-SRC		RPG source file for APLUS sample library	
MENU1	APLSAMPLE	*MENU	DSPF		Sample menu	
CUSNOCHGC	APLSAMPLE	*PGM	CLP			
CUSTCOPY	APLSAMPLE	*PGM	RPG			
CUSTINT	APLSAMPLE	*PGM	RPG		Internally described files	
CUSTLIST	APLSAMPLE	*PGM	RPG			
CUSTMAINTC	APLSAMPLE	*PGM	CLP			
CUSTRPG	APLSAMPLE	*PGM	RPG		Filter customer	
CUSTUPD1	APLSAMPLE	*PGM	RPG			
CUSTQRY1	APLSAMPLE	*QRYDFN	QRY		AS/400 Query over CUSTMASTL2	
CUSTQUERY	APLSAMPLE	*QRYDFN	QRY		Query joining CUSTMAST to ORDHEAD to get ortot	

Show children only once

A parent object may reference a child object more than one time. For instance, an RPG program may reference the same file once for input, and then in a separate F-spec reference the file once for output. Check

the ‘Show children only once’ box to include only the first reference, or uncheck the box to include all references.

Usage

You may further subset the list by showing children that are used a certain way. For instance, you may want to see only those files that are referenced for input. Click on the **Usage** drop down to select one of the special usages that ABSTRACT recognizes. Naming one of the special usage types below will exclude other objects from the display (with the exception of the *ALL value).

***ALL -** All object types will be shown.

CALL - Objects that are called by another

***CMD -** Objects referenced through CL commands

CPP - Command processing programs

EntryMod - Entry point module

ExternalDS - External data structure

INLMNU - Initial menu

INLPGM - Initial programs

***INP -** Files with input usage

***IO -** Files with input, output, or update (including delete) usage

JobdJOBQ - Job description job queues

JobdOUTQ - Job description output queues

JobdPRTDEV - Job description print devices

JobdUSRPRF - Job description user profiles

***NONCMD -** Command references in CL programs will be suppressed from the list. Only non-command (“Unknown” and input/output types) references will be included.

***OUT -** Files with output usage

PrfJOBD - User profile job descriptions

PrfMSGQ - User profile message queues

PrfOUTQ - User profile output queues

PrfPRTDEV - User profile print devices

Query - QRYDFN objects

RTGE - Routing entry

SEQUELView - User spaces for ASC's SEQUEL objects.

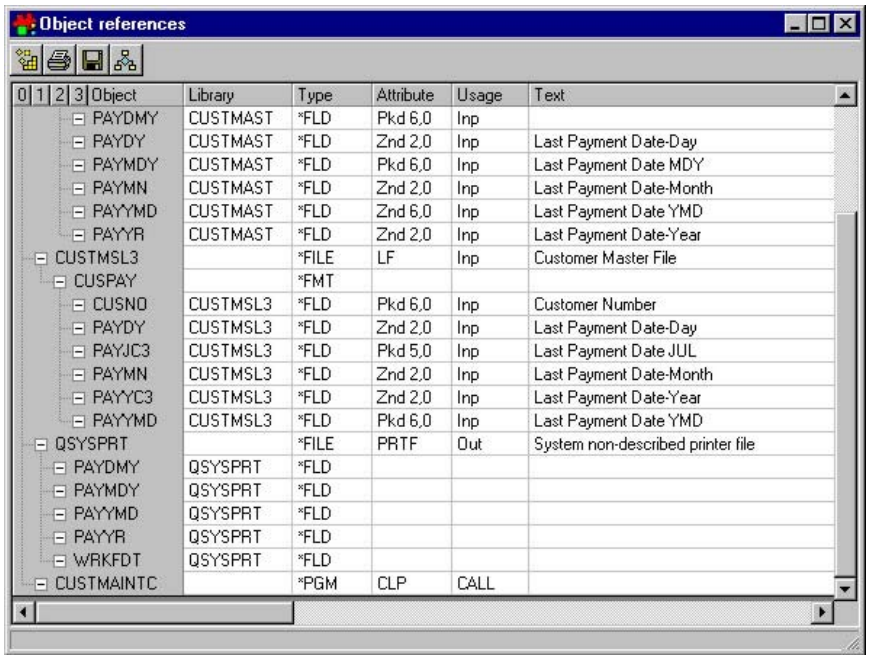
***UPD -** Files with update (or delete) usage

***ValidityCP -** Validity checking programs

Explosion Level

Use the explosion level to specify the maximum number of times that an additional object relation explosion should occur for children objects in the list. This value controls the number of recursive iterations for the list explosion. If the explosion level is greater than zero, the object relations explosion will be attempted for each indented item in the list until no additional relations can be found, or the explosion level is reached.

For example, setting the explosion level to 0 shows the following display for program CUSTRPG:

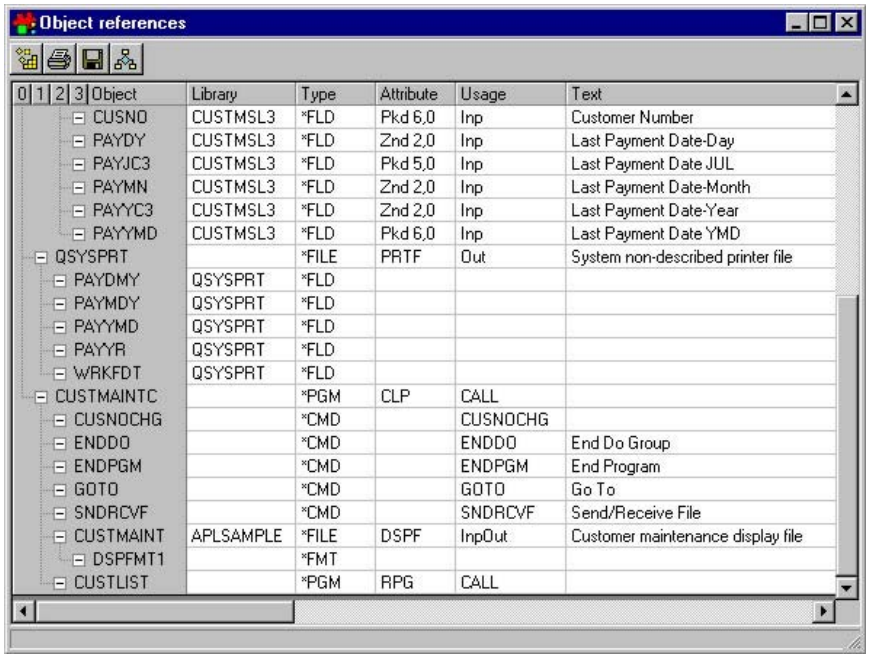


Object references

0 1 2 3 Object	Library	Type	Attribute	Usage	Text
PAYDMY	CUSTMAST	*FLD	Pkd 6,0	Inp	
PAYDY	CUSTMAST	*FLD	Znd 2,0	Inp	Last Payment Date-Day
PAYMDY	CUSTMAST	*FLD	Pkd 6,0	Inp	Last Payment Date MDY
PAYMN	CUSTMAST	*FLD	Znd 2,0	Inp	Last Payment Date-Month
PAYYMD	CUSTMAST	*FLD	Znd 6,0	Inp	Last Payment Date YMD
PAYYR	CUSTMAST	*FLD	Znd 2,0	Inp	Last Payment Date-Year
CUSTMSL3		*FILE	LF	Inp	Customer Master File
CUSPAY		*FMT			
CUSNO	CUSTMSL3	*FLD	Pkd 6,0	Inp	Customer Number
PAYDY	CUSTMSL3	*FLD	Znd 2,0	Inp	Last Payment Date-Day
PAYJC3	CUSTMSL3	*FLD	Pkd 5,0	Inp	Last Payment Date JUL
PAYMN	CUSTMSL3	*FLD	Znd 2,0	Inp	Last Payment Date-Month
PAYYC3	CUSTMSL3	*FLD	Znd 2,0	Inp	Last Payment Date-Year
PAYYMD	CUSTMSL3	*FLD	Pkd 6,0	Inp	Last Payment Date YMD
QSYSPRT		*FILE	PRTF	Out	System non-described printer file
PAYDMY	QSYSPRT	*FLD			
PAYMDY	QSYSPRT	*FLD			
PAYYMD	QSYSPRT	*FLD			
PAYYR	QSYSPRT	*FLD			
WRKFDT	QSYSPRT	*FLD			
CUSTMAINTC		*PGM	CLP	CALL	

Note that the CUSTMAINTC program at the bottom of the window appears with no additional object references. Changing the changing the explosion level to 1 shows additional objects referenced by CUSTMAINTC (next page).

In this case, we can see that CUSTMAINTC uses command CUSNOCHG, file CUSTMAINT, and it in turn calls program CUSTLIST at source sequence 7. Changing the explosion level to 2 would show object references for program CUSTLIST. Changing the explosion level to 3 would show object references for any program that CUSTLIST called, and so on.



Object references

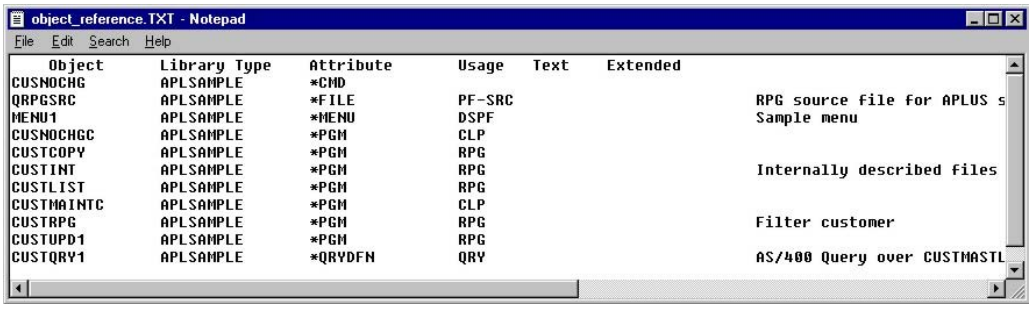
0 1 2 3 Object	Library	Type	Attribute	Usage	Text
CUSNO	CUSTMSL3	*FLD	Pkd 6,0	Inp	Customer Number
PAYDY	CUSTMSL3	*FLD	Znd 2,0	Inp	Last Payment Date-Day
PAYJC3	CUSTMSL3	*FLD	Pkd 5,0	Inp	Last Payment Date JUL
PAYMN	CUSTMSL3	*FLD	Znd 2,0	Inp	Last Payment Date-Month
PAYYC3	CUSTMSL3	*FLD	Znd 2,0	Inp	Last Payment Date-Year
PAYYMD	CUSTMSL3	*FLD	Pkd 6,0	Inp	Last Payment Date YMD
QSYSPRT		*FILE	PRTF	Out	System non-described printer file
PAYDMY	QSYSPRT	*FLD			
PAYMDY	QSYSPRT	*FLD			
PAYYMD	QSYSPRT	*FLD			
PAYYR	QSYSPRT	*FLD			
WRKFDT	QSYSPRT	*FLD			
CUSTMAINTC		*PGM	CLP	CALL	
CUSNOCHG		*CMD		CUSNOCHG	
ENDDO		*CMD		ENDDO	End Do Group
ENDPGM		*CMD		ENDPGM	End Program
GOTO		*CMD		GOTO	Go To
SNDRCVF		*CMD		SNDRCVF	Send/Receive File
CUSTMAINT	APLSAMPLE	*FILE	DSPF	InpOut	Customer maintenance display file
DSPFMT1		*FMT			
CUSTLIST		*PGM	RPG	CALL	

Print the Object Relation List

Click the **Print Button** to print the list at a local printer.

Save the Object Relation List

Click the **Save Button** to save the list to a comma-separated format that could be opened in any word processing or spreadsheet program.



object_reference.TXT - Notepad

File Edit Search Help

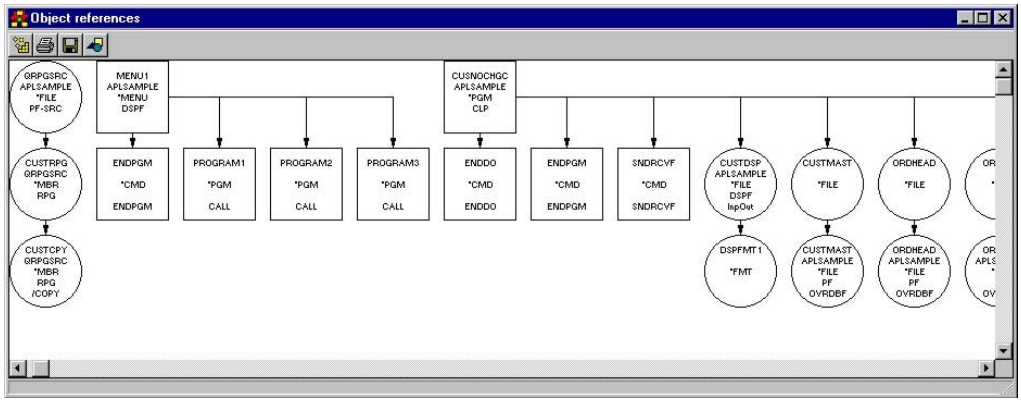
Object	Library Type	Attribute	Usage	Text	Extended
CUSNOCHG	APLSAMPLE	*CMD			
QRPGSRC	APLSAMPLE	*FILE	PF-SRC		RPG source file for APLUS s
MENU1	APLSAMPLE	*MENU	DSPF		Sample menu
CUSNOCHGC	APLSAMPLE	*PGM	CLP		
CUSTCOPY	APLSAMPLE	*PGM	RPG		
CUSTINT	APLSAMPLE	*PGM	RPG		Internally described files
CUSTLIST	APLSAMPLE	*PGM	RPG		
CUSTMAINTC	APLSAMPLE	*PGM	CLP		
CUSTRPG	APLSAMPLE	*PGM	RPG		Filter customer
CUSTUPD1	APLSAMPLE	*PGM	RPG		
CUSTQRY1	APLSAMPLE	*QRYDFN	QRY		AS/400 Query over CUSTMASTL

Flowcharting

Flowcharting is a function within Object Relations. ABSTRACT has its own built-in controls for creating graphical flowcharts. Many people have their own preferences for creating and managing flowchart diagrams, so HELP/SYSTEMS added support for Microsoft Visio, a Windows-based diagramming tool (ABSTRACT requires Visio 2002 or newer). Once installed, the flowcharts will simply appear in Visio rather than ABSTRACT's standard flowcharting control. If you have questions, please contact your HELP/SYSTEMS representative for more details.

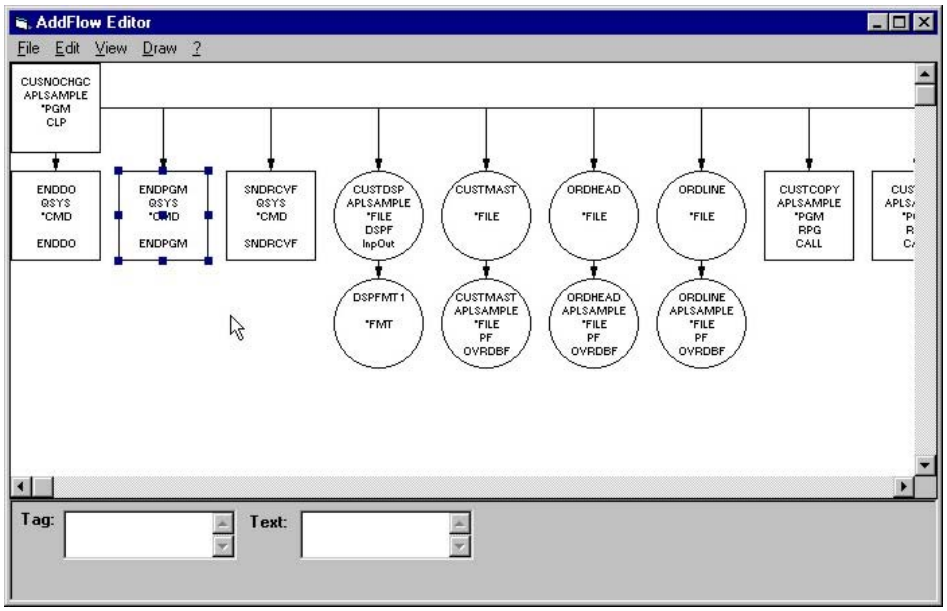
Flowchart the Object Relation List

Click the **Flowchart Button** to generate a flowchart of the object reference list.



Edit the Flowchart

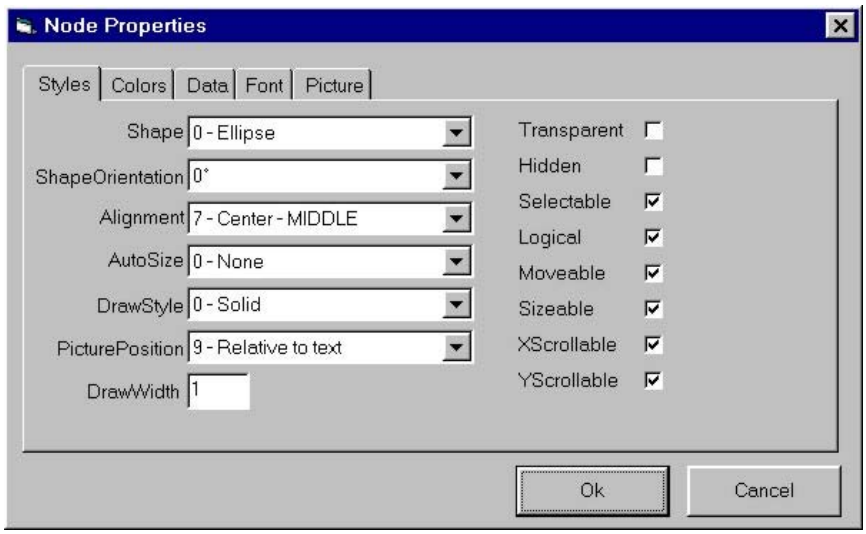
Click on the flowchart edit button  to initiate an edit session.



You may now edit or change the flowchart with the iSeries noted on the following pages.

Select an Object

Click on any object in the flowchart to select for editing. An object is active (may be edited) when the resize handles (9 little squares) are displayed. When an object is active, you may right click on it and select the **Properties** option. This window allows you to change any of the object's attributes.



Draw a new object

Bring the mouse cursor into the flowchart window, press the left button, move the mouse and release the left button. This creates a new object.

The handle at the center of the object is used to draw a link. The 8 others allow to resize the object. If you want to move the object, bring the mouse cursor into the object, press the left mouse button, move the mouse and release the left button.

Draw a link

Bring the mouse cursor into the handle at the center of the selected object, press the left button, move the mouse towards the other node. When the mouse cursor is into the other object, release the left button. The link has been created. And it is selected since a handle is displayed at the center of this link.

Stretch a link

Bring the mouse cursor into the link handle, press the left button, move the mouse and release the left button. You have created a new link segment. It has 3 handles allowing you to add or remove segments. (The handle at the intersection of two segments allows you to remove a segment : you move it with the mouse so that the two segments are aligned and when these two segments are approximately aligned, release the left button).

Draw a reflexive link

Select a node by clicking on it. Then bring the mouse cursor into the handle at the center of the selected object. Press the left button, move the mouse and release the left button when the mouse is still inside the selected object. You have created a reflexive link, i.e. a link whose origin and destination are the same.

File Analysis

Overview

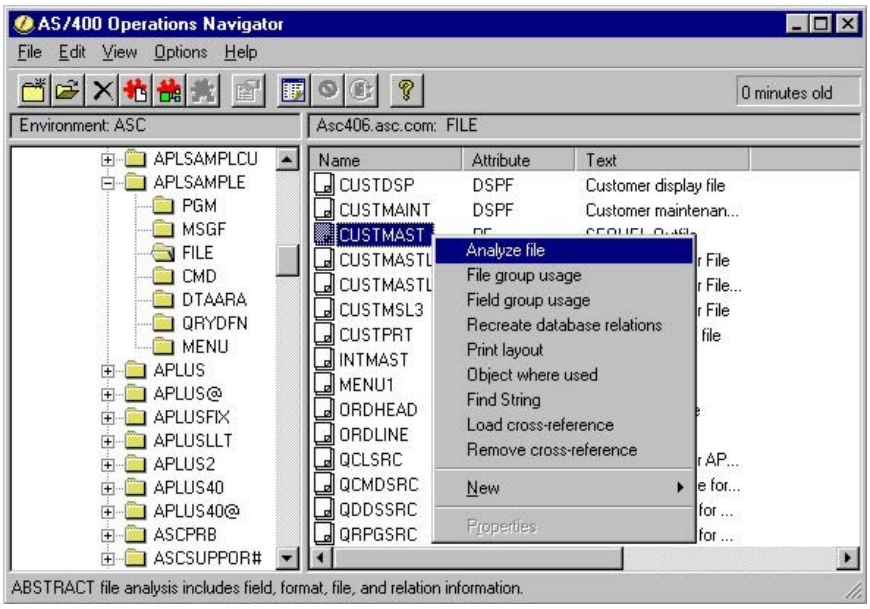
ABSTRACT file analysis will access the current file definition for database and device files on your system. You can use it to display the following information:

- Field descriptions and buffer locations for each record format in the file
- Member information
- Database relationships
- Access path descriptions
- File attribute information
- Program usage of fields
- Where-used information for the file and its fields.

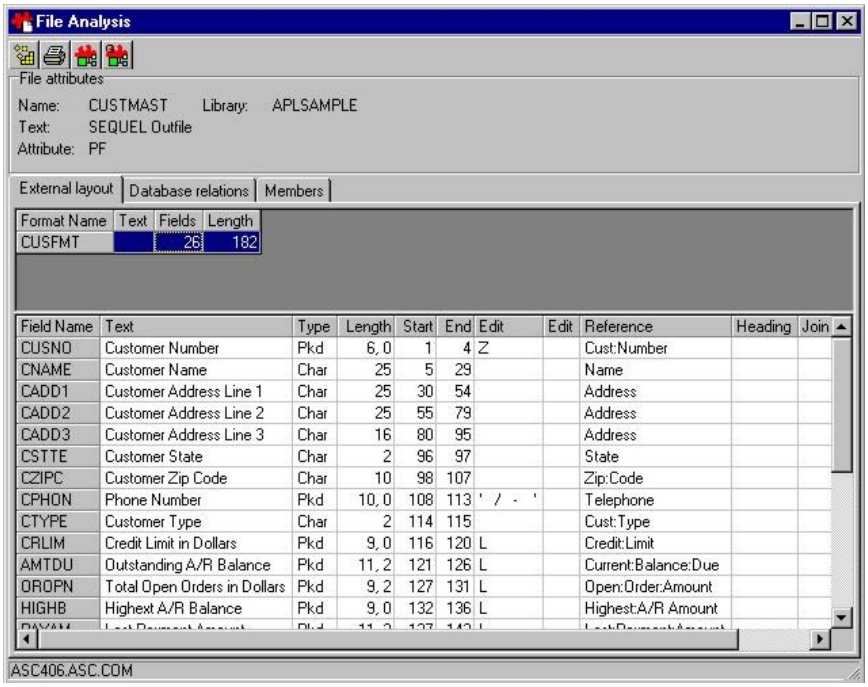
File Analysis functions do not use the ABSTRACT cross-reference database except to deliver “where-used” information. You do not need to load the cross-reference prior to running file analysis.

View File Analysis Information

Expand either the ABSTRACT or iSeries Explorer node, navigate to the file you want to analyze, then right-click on the file name to display the pop-up menu shown below.



Click on the **Analyze File** option to display the window shown below.



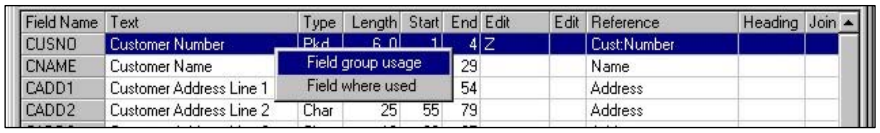
File Analysis displays a record layout through a tabbed interface. If you have selected a Distributed Data Management (DDM) file, ABSTRACT will attempt to reference the remote definition for the file. If you have chosen a program described file linked to an IDDU definition, ABSTRACT will display the external definition for the file.

External Layout

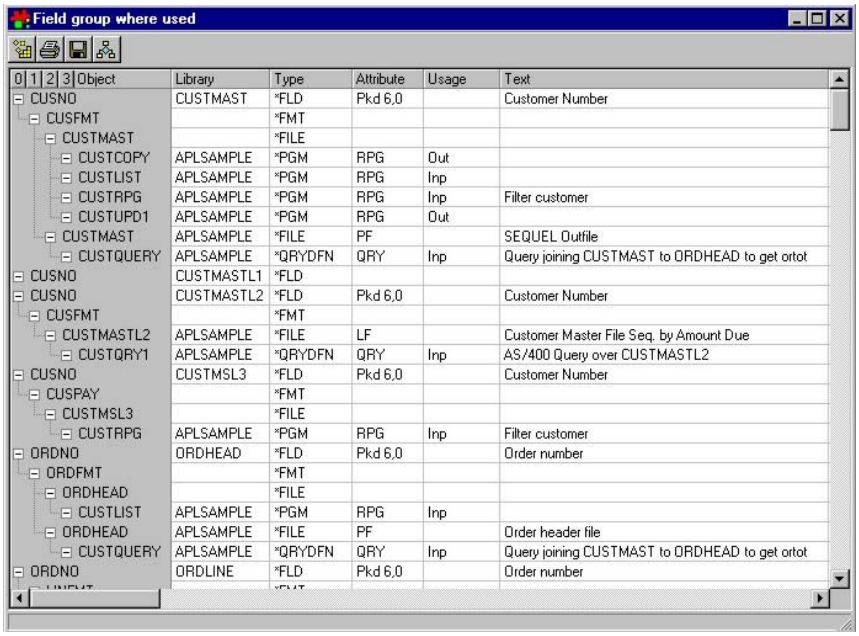
The external layout tab shows the external definition of the record format. If the file has more then one format, click on the name of the format for which you want to see information.

Each field in the format will be shown in positional order along with its attributes, length and text, if any.

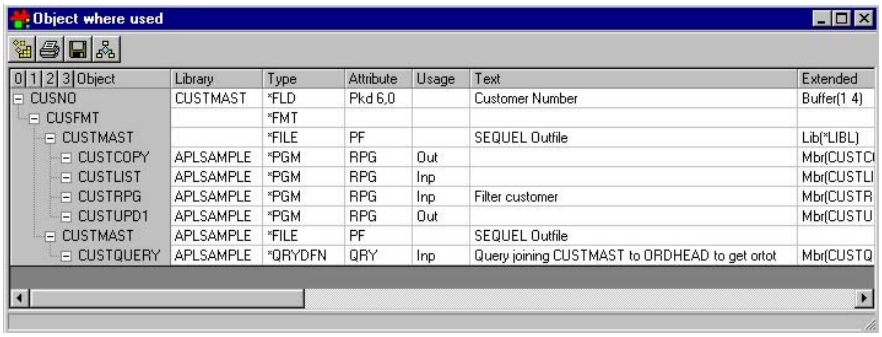
Right-click on a field name to show options available for that field, below. In order to obtain field level information, the file containing the field along with any programs that use it must be loaded into the cross-reference.



Field group usage will show field usage through the physical file, logical files built on it, and files overridden in the **Field Group Where Used** window. The example below shows the option for the CUSNO field selected above.



Field where used can be used to display the usage of the selected field through this file <u>only</u> in the **Object Where-Used** window. Choosing this option restricts the search to usage made through the file listed at the top of the display, once again shown for the CUSNO field.

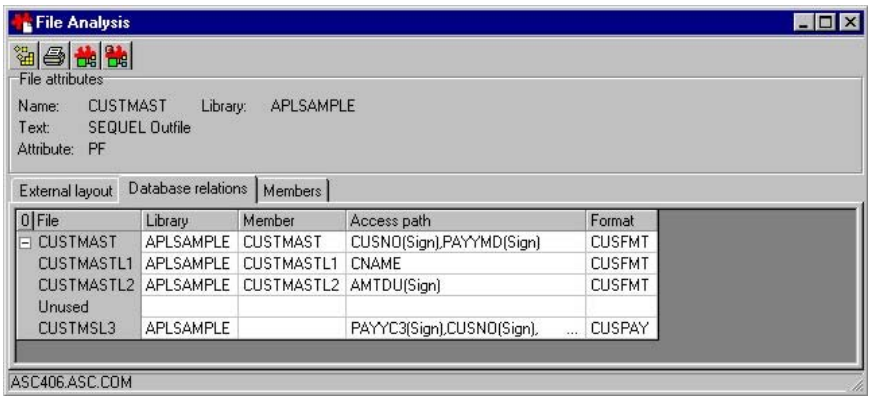


Object where used

0 1 2 3 Object	Library	Type	Attribute	Usage	Text	Extended
CUSNO	CUSTMAST	*FLD	Pkd 6,0		Customer Number	Buffer(1 4)
CUSFMT		*FMT				
CUSTMAST		*FILE	PF		SEQUEL Outfile	Lib(*LIBL)
CUSTCOPY	APLSAMPLE	*PGM	RPG	Out		Mbr(CUSTC
CUSTLIST	APLSAMPLE	*PGM	RPG	Inp		Mbr(CUSTLI
CUSTRPG	APLSAMPLE	*PGM	RPG	Inp	Filter customer	Mbr(CUSTR
CUSTUPD1	APLSAMPLE	*PGM	RPG	Out		Mbr(CUSTU
CUSTMAST	APLSAMPLE	*FILE	PF		SEQUEL Outfile	
CUSTQUERY	APLSAMPLE	*QRYDFN	QRY	Inp	Query joining CUSTMAST to ORDHEAD to get ortot	Mbr(CUSTQ

Database Relations

The **Database Relations** tab depicts database structure and access path information. Use it to determine the relationships among the physical and logical database files in your application. In acquiring the database relationship structure, ABSTRACT begins by locating the physical file(s) that contain the data you have selected from a node within iSeries Navigator. If you have selected a logical file, ABSTRACT locates each of its underlying physical files.



Once the physical file(s) are identified, ABSTRACT acquires the current member list for each file, and proceeds to show the logical files (and members) built over each of them. Access path information is also shown for each file represented on the display.

The example above lists the database structure in which the CUSTMAST file participates. Each member of the physical files in the structure is represented at the leftmost position of the file name column. Logical file members that are built on the physical file member are listed underneath it and indented to the right. Access path information is listed to the right of the file and member name.

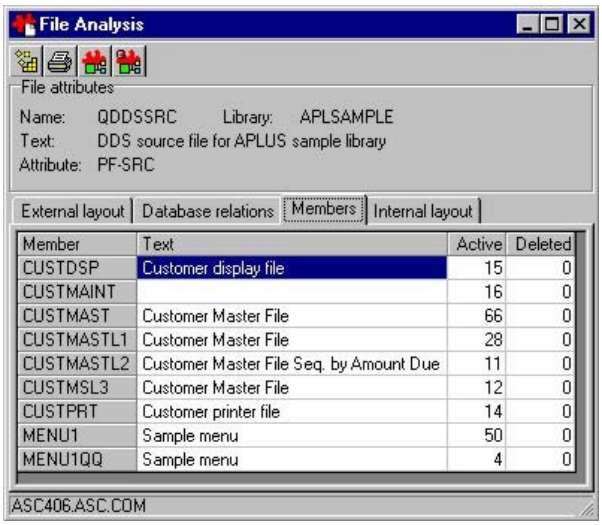
Some files may appear under a heading of “Unused”. These logical files are built over a physical file but do not currently use any of its members.

The display also shows that CUSTMAST has two key fields **(CUSNO, PAYYMD)**. Files that are not keyed would be indicated by **Arrival Sequence** notation in the access path column.

The CUSTMSL3 file show an ellipsis (...) at the right. This indicates that the complete access path description is not shown and includes more information than can be presented on a single line. Right-click on the file and choose **Access path description** to see the entire access path definition.

Member Information

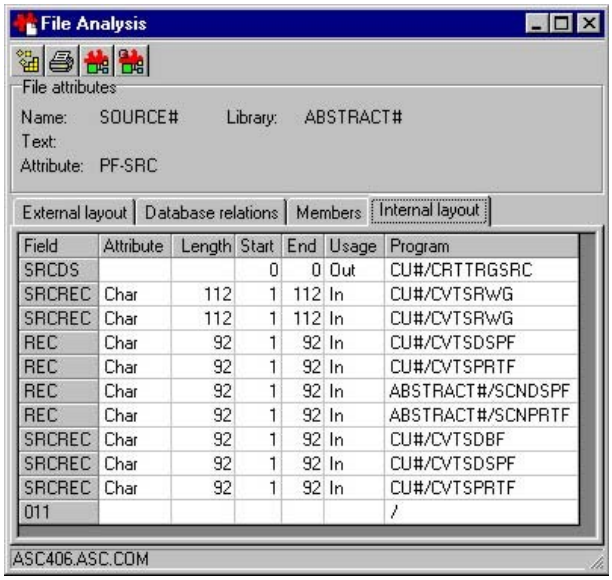
The member tab shows information for each database file member (this option is not available for device files). For each member, a count of active and deleted records is shows, as well as member text.



Member	Text	Active	Deleted
CUSTDSP	Customer display file	15	0
CUSTMAINT		16	0
CUSTMAST	Customer Master File	66	0
CUSTMASTL1	Customer Master File	28	0
CUSTMASTL2	Customer Master File Seq. by Amount Due	11	0
CUSTMSL3	Customer Master File	12	0
CUSTPRT	Customer printer file	14	0
MENU1	Sample menu	50	0
MENU1QQ	Sample menu	4	0

Internal Layout

The internal record layout tab provides information about your application's use and definition of program described files. If programs in your application use record or field definitions that are different from the external definitions in the database, ABSTRACT will acquire and



Field	Attribute	Length	Start	End	Usage	Program
SRCDS			0	0	Out	CU#/CRTTRGSRC
SRCREC	Char	112	1	112	In	CU#/CVTSRWG
SRCREC	Char	112	1	112	In	CU#/CVTSRWG
REC	Char	92	1	92	In	CU#/CVTSDSPF
REC	Char	92	1	92	In	CU#/CVTSPRTF
REC	Char	92	1	92	In	ABSTRACT#/SCNDSPF
REC	Char	92	1	92	In	ABSTRACT#/SCNPRTF
SRCREC	Char	92	1	92	In	CU#/CVTSDBF
SRCREC	Char	92	1	92	In	CU#/CVTSDSPF
SRCREC	Char	92	1	92	In	CU#/CVTSPRTF
011						/

store the program descriptions in its database.

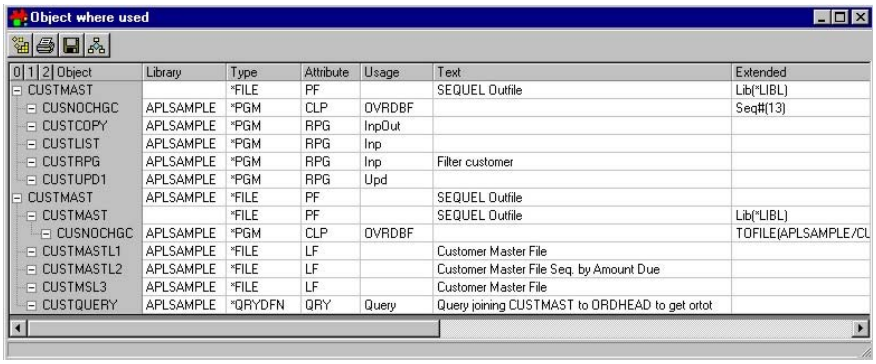
Fields are displayed in positional order. Buffer positions as well as length and attributes are displayed for each field. The program which provides the description is shown at the right side of the display, and its use of the field is also indicated.

ABSTRACT attempts to point out possible conflicts in definition of the record by displaying apparent conflicts in high intensity. Overlapping fields will have buffer positions high lighted and conflicting field attributes (Ex.: Character and Zoned) will have the attributes or decimal positions highlighted.

COBOL group level data names are displayed along with elementary items. Group items have attributes 'GRP'.

File Usage

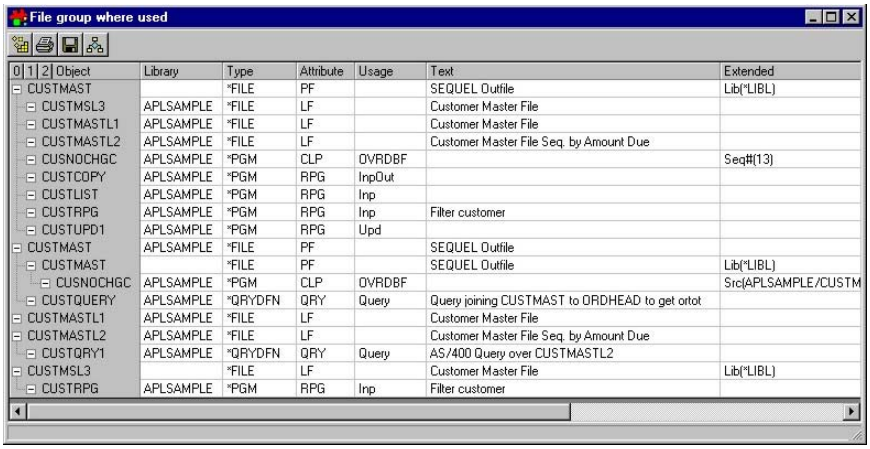
The **File Usage** button shows usage for the selected file in an **Object Where Used** window.



0 1 2 Object	Library	Type	Attribute	Usage	Text	Extended
CUSTMAST		*FILE	PF		SEQUEL Outfile	Lib(*LIBL)
CUSNOCHGC	APLSAMPLE	*PGM	CLP	OVRDBF		Seq#(13)
CUSTCOPY	APLSAMPLE	*PGM	RPG	InpOut		
CUSTLIST	APLSAMPLE	*PGM	RPG	Inp		
CUSTRPG	APLSAMPLE	*PGM	RPG	Inp	Filter customer	
CUSTUPD1	APLSAMPLE	*PGM	RPG	Upd		
CUSTMAST	APLSAMPLE	*FILE	PF		SEQUEL Outfile	
CUSTMAST		*FILE	PF		SEQUEL Outfile	Lib(*LIBL)
CUSNOCHGC	APLSAMPLE	*PGM	CLP	OVRDBF		TOFILE(APLSAMPLE/CU
CUSTMASTL1	APLSAMPLE	*FILE	LF		Customer Master File	
CUSTMASTL2	APLSAMPLE	*FILE	LF		Customer Master File Seq. by Amount Due	
CUSTMSL3	APLSAMPLE	*FILE	LF		Customer Master File	
CUSTQUERY	APLSAMPLE	*QRYDFN	QRY	Query	Query joining CUSTMAST to ORDHEAD to get ortot	

File Group Usage

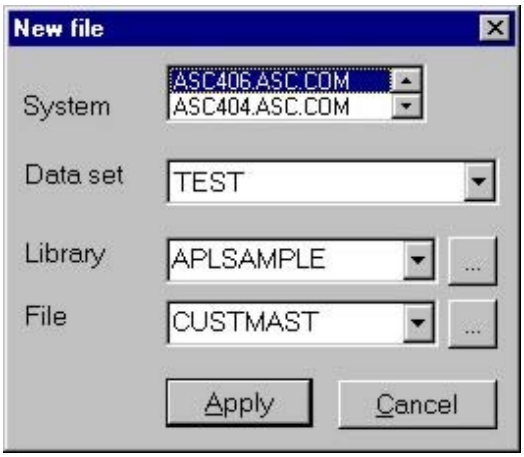
The **File Group Usage** button shows the relationship between associated physical and logical files in a **File Group Where Used** window.



0 1 2 Object	Library	Type	Attribute	Usage	Text	Extended
CUSTMAST		*FILE	PF		SEQUEL Outfile	Lib(*LIBL)
CUSTMSL3	APLSAMPLE	*FILE	LF		Customer Master File	
CUSTMASTL1	APLSAMPLE	*FILE	LF		Customer Master File	
CUSTMASTL2	APLSAMPLE	*FILE	LF		Customer Master File Seq. by Amount Due	
CUSNOCHGC	APLSAMPLE	*PGM	CLP	OVRDBF		Seq#(13)
CUSTCOPY	APLSAMPLE	*PGM	RPG	InpOut		
CUSTLIST	APLSAMPLE	*PGM	RPG	Inp		
CUSTRPG	APLSAMPLE	*PGM	RPG	Inp	Filter customer	
CUSTUPD1	APLSAMPLE	*PGM	RPG	Upd		
CUSTMAST	APLSAMPLE	*FILE	PF		SEQUEL Outfile	
CUSTMAST		*FILE	PF		SEQUEL Outfile	Lib(*LIBL)
CUSNOCHGC	APLSAMPLE	*PGM	CLP	OVRDBF		Src(APLSAMPLE/CUSTM
CUSTQUERY	APLSAMPLE	*QRYDFN	QRY	Query	Query joining CUSTMAST to ORDHEAD to get ortot	
CUSTMASTL1	APLSAMPLE	*FILE	LF		Customer Master File	
CUSTMASTL2	APLSAMPLE	*FILE	LF		Customer Master File Seq. by Amount Due	
CUSTQRY1	APLSAMPLE	*QRYDFN	QRY	Query	AS/400 Query over CUSTMASTL2	
CUSTMSL3		*FILE	LF		Customer Master File	Lib(*LIBL)
CUSTRPG	APLSAMPLE	*PGM	RPG	Inp	Filter customer	

Analyze a Different File

Click on the **New Subset** button to specify a new file to analyze.



Print File Analysis Information

Click on the **Print File Analysis** button to send information from each of the tabs to a local printer

Recreate Database Relations

About Recreate Database Relations

The **Recreate Database Relations** option will greatly reduce the time and effort required to recreate all affected system objects when changes have been made to a physical file record layout. All current data is preserved along with physical and logical structure and other file attributes. Current data base structure and file attributes are analyzed when the option is run. If you choose to compile dependant programs, the affected programs are determined based on information previously loaded in the cross reference.

IMPORTANT NOTE: The default 'compile to' library for files and programs processed by this command is the library or libraries loaded into the cross reference. If you build your ABSTRACT cross reference over your production libraries, you should establish controls and practices that insure against inadvertent replacement of production objects, and understand the use of the Target Library parameter, which is described later in this section.

The recreate database relations option performs the following steps automatically:

- Renames the current physical file to preserve its data
- Creates the new physical file in using the current file attributes
- Creates all logical files built on the "old" physical into QTEMP
- Adds members to the newly created physical
- Adds members to the newly created logicals, preserving original DTAMBRS structure
- Copies data from "old" physical file to new file
- Deletes logical files from "old" physical file
- Deletes "old" physical
- Moves the new logicals from QTEMP to their original library(s)
- Optionally recompiles programs that use any file affected by the compile.

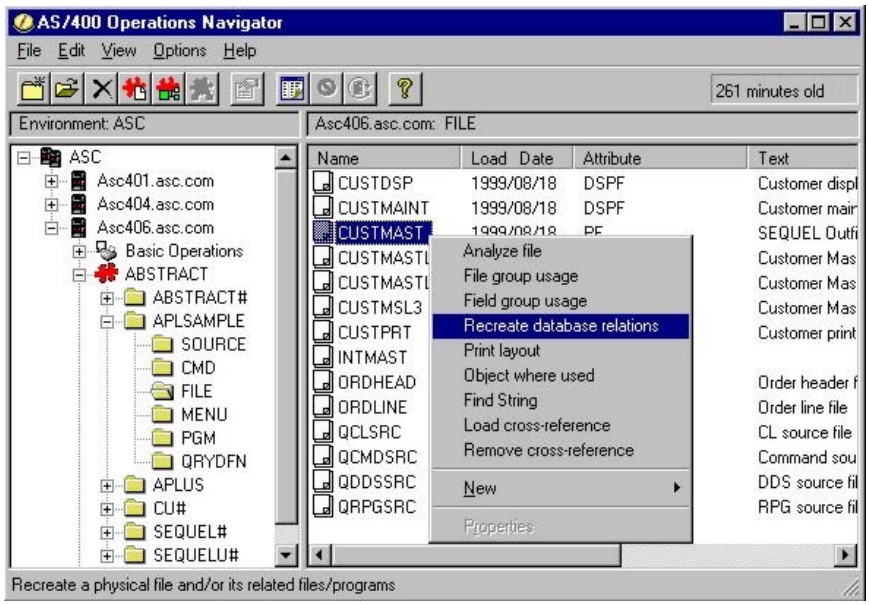
When device files are recompiled, the following attributes are specifically retained:

`PAGESIZE, LPI, CPI, OVRFLW, LVLCHK, PRTTXT, SHARE, DUPLEX, OUTQ, FORMTYPE, HOLD, SAVE, DEV, PRTQLTY and MAXRCDS`

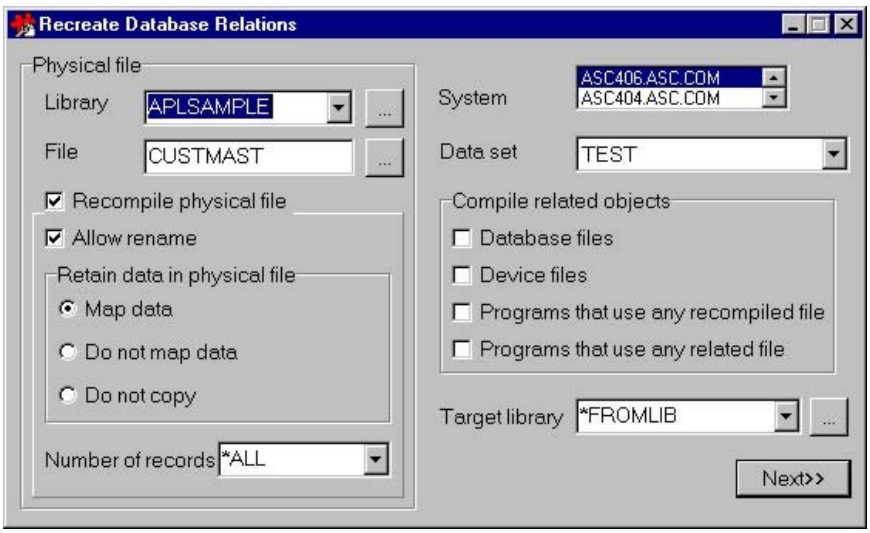
The recreate database relations process will provide an audit report showing which steps were taken and their outcomes. Built-in error recovery automatically restores the database to its original status if an error is encountered while compiling the new objects.

Recreate Options

Select the file to recreate, then right-click to display the pop-up menu shown below.



Click the **Recreate database relations** option to display the window of the same name. The **Library**, **File**, **System** and **Data set** entry fields will default to that which was selected.



You may specify other file recreation options on the other fields on this window:

Recompile Physical File

This option is ‘checked’ as a default and is used to recreate the physical file. Should you only want to recompile affected programs, for instance, you may uncheck the option. If unchecked, the file is not recompiled and the **Allow Rename** and **Retain data in physical file** fields, below, will be disabled.

Allow Rename

When the TOLIB is not *FROMLIB, this parameter specifies whether or not the base physical file or files can be temporarily renamed. This can be important if compiling from a production library. It is necessary to rename a base physical file if the source code for any of the logical files specifies the library name for the based on physical file. In that case, the original physical is renamed and the new physical file is moved into the base library and the new logicals are compiled into the target library. The new physical file is moved to the target library and the old physical file is renamed back to its original name.

<u>*NO</u> - Do not rename the base physical file. ABSTRACT will add the target library to the library list of the compile job before compiling logical files. This option requires that the source code for all logical files reference the related physical file/s without naming the library for the physical file/s.

***YES -** Allow temporary rename of the base physical file. ABSTRACT will rename each physical file while compiling the related logical files. Renaming takes place whether or not the source code for the logical files specifies the library name for the physical.

Retain Data in Physical File

The **Retain data in physical file** radio buttons allow you to specify whether data from the physical file should be retained and if so, how it should be copied into the new file.

Map Data: Retain the data and use the FMTOPT(*MAP) parameter when copying it into the file to be created. Refer to the Copy File (CPYF) command for a complete description of this parameter.

Do not map data: Retain the data and use the FMTOPT(*NOCHK) parameter when copying it into the file to be created. Refer to the Copy File (CPYF) command for a complete description of this parameter.

Do not copy: Do not retain the current file data. All data in the physical file will be lost.

Number of Records

You may control the number of records placed into the output file with this field . This parameter is ignored if the **Retain data/ Do not copy** option is specified above.

***ALL -** the entire result will be placed in the outfile member.

Number - Specifies the maximum number of records to be placed in the output member.

Compile Related Objects

The ***Compile related objects*** panel allows you to also specify what file relations are to be created. Check any of the following boxes:

Database files - Recreates the physical file along with all associated logical files. Unless this value is specified, the physical file will not be recreated.

Device files - Recreate all device (printer or display) files that depend (via the REF or REFFLD keyword) on the physical file and all associated logical files.

Programs that use any recompiled file - Recreates all programs that use any files that will be created by the file selection criteria, above.

Programs that use any related file - Recreates all programs that use any files that will be created by the file selection criteria, above, as well as any additional files that refer to those recompiled files via REF or REFFLD.

When the program recompilation is completed ABSTRACT will print an audit report showing the attempted compiles and the outcome of each. The audit report will precede the compile listings in the output queue.

The columns on the report identify the program name and the source used to compile it. The result of the compile operation is listed as a Yes ('Y') or No ('N') indicator. The end of the report provides a total of successful and unsuccessful compile counts.

Target Library

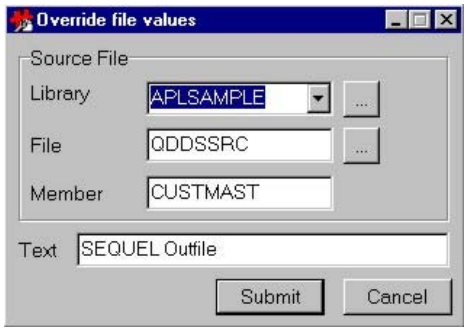
<u>*FROMLIB</u> - Recompile the files and programs back into the original library where each object is found. For program objects, these will be libraries that are loaded in the ABSTRACT cross reference.

***CVT -** Compile the objects into the 'test' library defined within WRKLIBX (usually the library name with 'CU' appended to it).

Name – Type in a valid library name. Files and programs will all be compiled into this library.

Next>> Button

Click this button to continue to the **Override file values** window, where you may change additional settings and submit the request.



Source File

This field defaults to the file previously selected. When the default is overridden, this parameter provides the ability to recreate a single file when the source code cannot be found by normal means. This parameter provides information used by the compile process and is ignored if the request does not involve a file compile. If the FILE parameter includes more than a single file, the SRCFILE parameter provides information only for the first file to be compiled.

***SAME** – ABSTRACT will use service data to obtain the source file name and source library. If source code cannot be located using service data , ABSTRACT will search the libraries loaded in the cross-reference. If source cannot be located by this second method, an error message is recorded in the job log and the summary report will indicate that the file was not compiled.

Name - Enter the name of the source file containing the Data Description Specifications for the file named in the FILE parameter.

Library-name - Specifies the name of the library containing the source file.

Text

This field includes a description of the selected object. The possible values are:

***SRCMBRTXT -** If the source file is a database file, the text is taken from the source file member used to create the file. If the source file is an inline file or a device file, the text is blank.

***BLANK -** No text is specified.

'description' - Specify up to 50 characters of descriptive text enclosed in apostrophes.

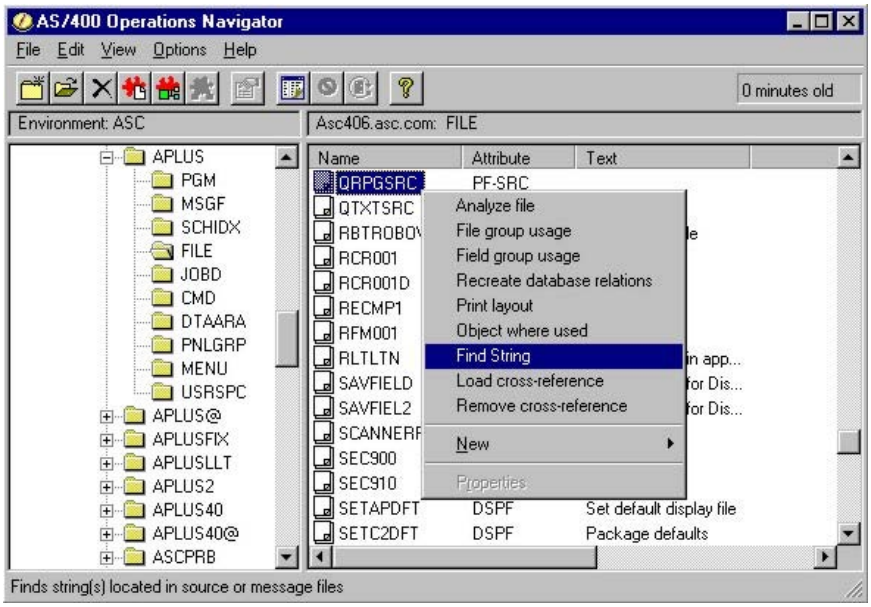
Audit Report

When the recreate database relations process completes, a report will be generated that shows all of the activity that has taken place.

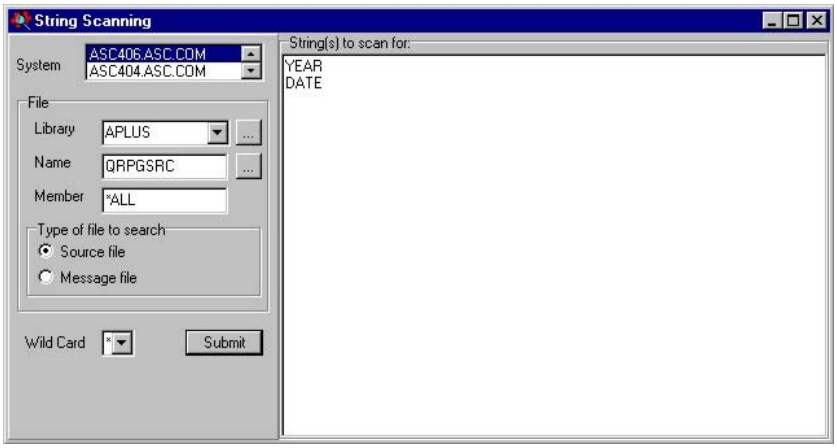
Find String

The **Find String** option lets you search source and message files for occurrences of one or more text strings. ABSTRACT will create a report showing the locations of each string. The find string option does not rely upon cross-reference information to perform the search.

Expand either the ABSTRACT or Explorer nodes, then **right-click** on the source file in which you want to search to display the pop-up menu shown below.



Select and click on the **Find String** option to display the String Scanning window, below.



The values in the text boxes in the left column default to the file selected on the previous window. You may change these and the following settings to meet your needs

Wild Card

Search strings may include a **wild card** character in the body of the string(s). The default wild card character is an asterisk '*', but it can be changed by typing any valid character into the entry field. If a match is found for all characters in the string except the wild card character(s), the source will be included on the report. For example, a search string of `OVR***F` and a CL source file name will cause ABSTRACT to find all occurrences of `OVR***F, OVR***F, OVR***F,`. without returning occurrences of `OVRDBF`.

String(s) to scan for

Key in the strings that you want ABSTRACT to search into the **String(s) to scan for** text box. Each search string may be up to 79 characters in length. You can specify multiple search strings by pressing Enter between each request. ABSTRACT will search each member or file for each of the strings you specify. Text in the source or message files will be converted to uppercase so that case sensitivity will not be a problem.

As mentioned above, you may use wild card characters within the search string.

The String Scanning Report

The report will examine the indicated file(s) and list the results similar to the examples below and on the following pages.

The first example shows the result of a message file search. The FNDSTR request searched the QCPFMSG message file in the QSYS library. It indicated that the entire message (no position specification via function key 22) should be searched for the word “Domain”.

The report shows the five messages that include the word “Domain” and their message identifiers. It concludes by indicating the number of messages searched, and the number of occurrences located.

```
13:51:33                    ABSTRACT                                        Page   1
 3/26/92               String Scanning Utility

String(s)  : From   1 to  80  DOMAIN

Message File: QSYS/QCPFMSG          Wild card character: ?

Msgid   Message
CPD0578 Object domain error for program &1 in library &2.
CPF1989 Domain violation occurred for variable &3.
CPI2247 Domain failure by program &28/&27 for object &22/&21 type &23.
CPPAA2A Time domain reflectometry limit reached
MCH6801 Object Domain error for object &1.

  17316 messages searched
      5 occurrences of requested string found
```

No wild card characters were indicated, although “DO?AIN” and “??MAIN” searches would have located the same messages (and possibly others as well).

The next example (following page) shows the output from a source scan. The entire source file (APLUS#/QDDSSRC) was searched for instances of PNLGRP. No position restrictions or wild card characters were used. As before, the end of the report indicates the results of the search.

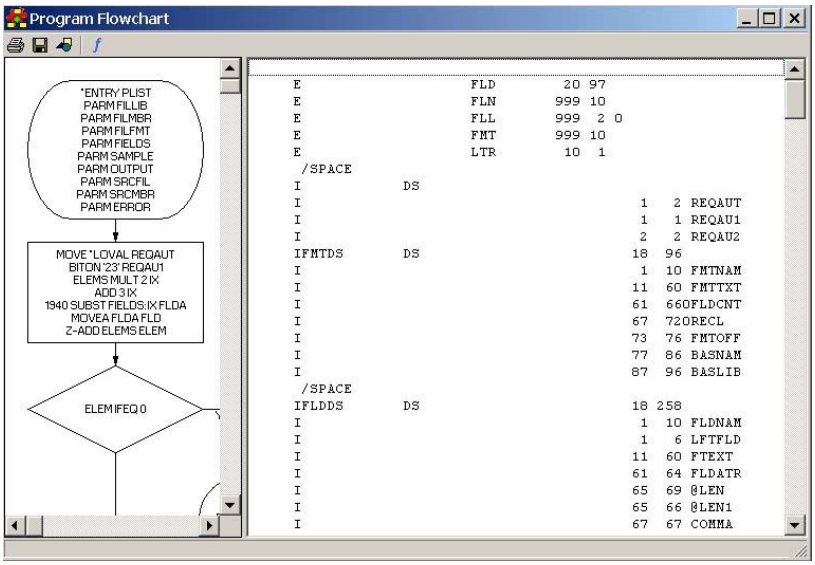
```
 15:36:07                                    ABSTRACT                                        Page
1
  3/26/92                                String Scanning Utility

 String     : From   1 to 256  PNLGRP
 Source File: APLUS#/QDDSSRC        Member: *ALL       Wild card character: ?

 Member     Sequence Source statement
 APCRTPGMS      .08       A                                        HLPPNLGRP(‘APCRTPGMS’ APUSR6)
               5.21       A          H                             HLPPNLGRP(‘CRTXXXPGMS’ APUSR6)
               5.23       A          H                             HLPPNLGRP(‘MENU/CMDLINE’ APUSRINT)
               5.25       A          H                             HLPPNLGRP(‘MENU/MENUFKY’ APUSRINT)
 APEXCP         .06       A                                        HLPPNLGRP(‘APEXCP’ APUSR5)
               5.21       A          H                             HLPPNLGRP(‘APEXCP/ONE’ APUSR5)
               5.23       A          H                             HLPPNLGRP(‘APEXCP/TWO’ APUSR5)
               5.25       A          H                             HLPPNLGRP(‘APEXCP/THREE’ APUSR5)
               5.37       A          H                             HLPPNLGRP(‘MENU/CMDLINE’ APUSRINT)
               5.39       A          H                             HLPPNLGRP(‘MENU/MENUFKY’ APUSRINT)
 APFILE         .05       A                                        HLPPNLGRP(‘APFILE’ APUSR4)
               5.21       A          H                             HLPPNLGRP(‘APFILE/ONE’ APUSR4)
               5.23       A          H                             HLPPNLGRP(‘APFILE/TWO’ APUSR4)
               5.25       A          H                             HLPPNLGRP(‘APFILE/THREE’ APUSR4)
               5.27       A          H                             HLPPNLGRP(‘APFILE/FOUR’ APUSR4)
               5.29       A          H                             HLPPNLGRP(‘APFILE/FIVE’ APUSR4)
               5.33       A          H                             HLPPNLGRP(‘MENU/CMDLINE’ APUSRINT)
               5.35       A          H                             HLPPNLGRP(‘MENU/MENUFKY’ APUSRINT)
 APINZ          .90       A                                        HLPPNLGRP(‘APINZ’ APUSR1)
               5.60       A          H                             HLPPNLGRP(‘APINZ/ONE’ APUSR1)
               5.90       A          H                             HLPPNLGRP(‘APINZ/TWO’ APUSR1)
               6.20       A          H                             HLPPNLGRP(‘APINZ/THREE’ APUSR1)
               6.50       A          H                             HLPPNLGRP(‘APINZ/FOUR’ APUSR1)
               6.80       A          H                             HLPPNLGRP(‘APINZ/FIVE’ APUSR1)
               8.83       A          H                             HLPPNLGRP(‘MENU/CMDLINE’ APUSRINT)
               8.90       A          H                             HLPPNLGRP(‘APINZ/MENUFKY’ APUSR1)
 APLUS          .90       A                                        HLPPNLGRP(‘MENU’ APUSRINT)
               5.70       A          H                             HLPPNLGRP(‘APINZ’ APUSR1)
               6.00       A          H                             HLPPNLGRP(‘APOBJR’ APUSR2)
               6.30       A          H                             HLPPNLGRP(‘APOBJU’ APUSR3)
               6.60       A          H                             HLPPNLGRP(‘APFILE’ APUSR4)
               6.90       A          H                             HLPPNLGRP(‘APEXCP’ APUSR5)
               7.20       A          H                             HLPPNLGRP(‘APTOOL’ APUSR6)
               7.80       A          H                             HLPPNLGRP(‘MENU/CMDLINE’ APUSRINT)
               8.10       A          H                             HLPPNLGRP(‘MENU/MENUFKY’ APUSRINT)
 APOBJR         .07       A                                        HLPPNLGRP(‘APOBJR’ APUSR2)
               5.22       A          H                             HLPPNLGRP(‘WRKOBJR’ APUSR2)
               5.25       A          H                             HLPPNLGRP(‘WRKOBJRS’ APUSR3)
               5.28       A          H                             HLPPNLGRP(‘WRKOBJR/PGM’ APUSR2)
        .
        .
        .
    135 members searched
  22280 source lines scanned
    522 occurrences of requested string found
```

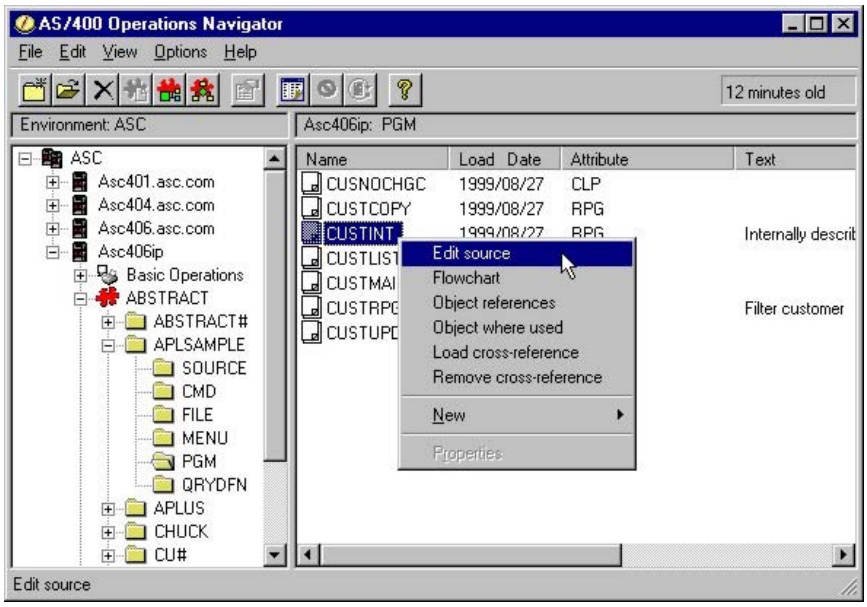
Source Flowchart

The source flowchart is only accessible via the WDSc plugin. It attempts to diagram the logic flow within a single RPG source member. The left hand window is reserved for the diagram, while the right hand side shows the actual code associated with the member.

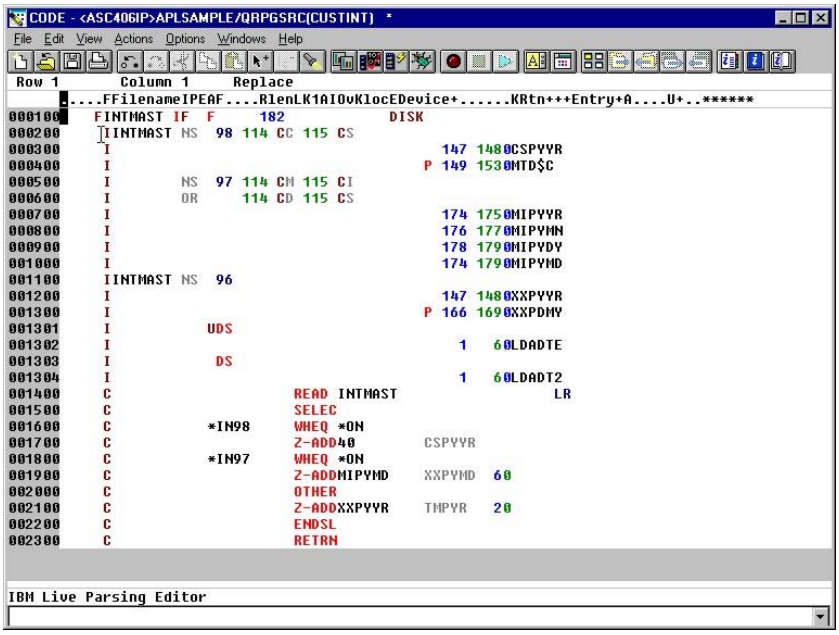


Edit Source

This option is available if you have IBM's CODE product already installed on your system. You can right-click on an object shown within an ABSTRACT window, then take the **Edit Source** option to initiate an edit session. This CODE edit option will work via the Navigator or WDSc, but will not work with the LPEX editor in WDSc.



CODE Edit Window



Index

A

ABSTRACT Features 5
Allow Rename 85
Analyze CL 38
Analyze File 74
Analyze OS/400 39
Analyze SEQUEL 39
Analyzing a Different File 81
AS/400 Object Selection Window 27
Audit Report 88

C

Compile Related Objects 86
Controlling Displayed Object Types 63
CPPs 38
Cross-Reference 37
- About 37
- Information 40
- Initialization Steps 38
- Load 18
- Load Object Types 46
- Load Specific Objects 47
- Reload Existing Information 47
- Remove Libraries 47
- Select Libraries 41
- Select Libraries from AS/400 Explorer 45
- Working with Data 44

D

Data Set 23
- Add 25
- Change 26
- Clear 26

Database Relations 78
Database Relationships 38, 73
Defaults 21
Details button 62

E

Edit Source 94
Elements of the Object Relation Display 51
Email Recipient 22
Exception Reports 15
Expand outline 60
Explosion Level 66, 67
Extended Description 53
External Layout 76

F

Field Group Usage 58, 76
Field Usage Options 76

Field Where Used 57, 77
File Analysis
About 73
Display.....75
View.....74
File Group Usage.....57, 81
File Usage.....80
Find String.....89
Flowcharting the Object Relation List.....69

G

Get Started.....11

H

HLL CALLs.....40
Hold on Job Queue 22

I

Indented Objects.....51
Initialization
About 37
Internal Layout 79
Introduction 1

J

Job Description 22
Job Queue 22

L

Level buttons.....61
Library Selection window 18, 41
Load AS/400 Software 12
Load Client Software.....12
LOADJOBD.....38
LOADQRY 38
Locate HLL.....39

M

Member Information.....73, 79

O

Object ‘where-used’ 48
Object Authority Requirements 37
Object Date.....23
Object Exception Reports 15, 28
Report Information.....34
Submit 30
Object Name.....23
Object Reference Displays.....59
Object Relations
About 48
How to view 49
Options 55
Object Where Used.....56

Objects used by another object....48
Operations Navigator 1
Orphaned Objects....35
Override file values window 87

P

Parent Objects 51
Printing File Analysis Information....81
Printing the Object Relation List....68
Printing Unloaded Objects 33
Printing Unsed Objects 31, 32, 33

R

Recompile Physical File....85
Recreate Database Relations
    About 82
    Options 84
Retain Data in a Physical File 85

S

Saving the Object Relation List....68
Select a Data Set....25
Set Defaults 13
Show Children
    Of Type....63
    Only once 63
Showing Unique Relations....63
Source Conflicts....35
Source Scan Report....93
String Scanning Report....91
String(s) to scan for....90
Subset....62
Subsetting by Object Usage....64

T

Target Library....87
Types of Object Relation Displays 56

U

Undocumented Objects 36
Unused Objects 36
Using Level Buttons....61

W

Wild card....90
Working With Cross-reference Data 44