

Invoking XRef GUI Client

This section covers the following topics:

- Plug-In Manager Settings
 - Invoking XRef GUI Client From the Main Menu
 - Invoking XRef GUI Client From the Context Menu
 - Cross References
 - Active Cross References
 - Passive Cross References
 - List Children
 - Influence of Environmental and User-specific Settings
-

Plug-In Manager Settings

Before any action can be taken, the XRef GUI Client plug-in has to be activated in your Plug-In Manager. Detailed information on the activation procedure can be found in section Plug-In Manager.

Once it is activated, you can invoke XRef GUI Client either from the main menu, toolbar buttons or by right clicking on an object within the tree view where XRef data is stored.

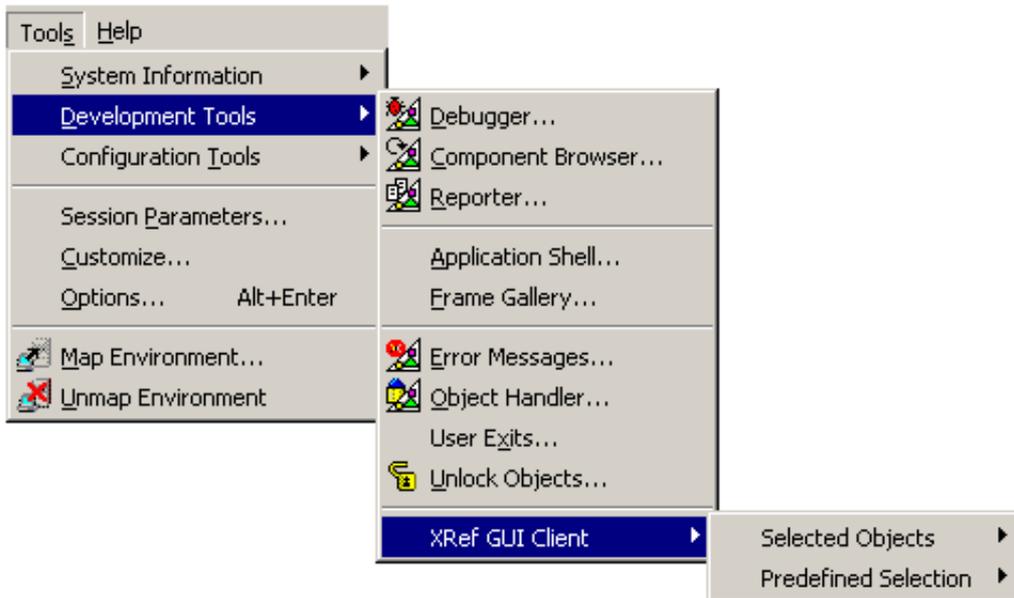
Note:

The tree view on XRef data that is presented by XRef GUI Client always depends on the LOGON library and Steplib settings that are active during initial invocation/instanciation of the corresponding cross reference tree view. For that reason, the results of the same XRef data retrieval can vary due to different environmental or user-specific settings. See Influence of Environmental and User-specific Settings for further information.

Invoking XRef GUI Client From the Main Menu

To invoke XRef GUI Client from the main menu:

Choose a library or an object within a library by left clicking on the library or object. From the main menu, choose Tools > Development Tools > XRef GUI Client. See the sample menu below.



You can choose from two options now. If you choose Selected Objects, only the active or passive cross references of the object you selected are displayed.

Note:

This option is only available, if you have selected an object within a library (for example a program, subprogram or subroutine).

Alternatively to invoking XRef GUI Client from the main menu, you can also use one of the following buttons:

-  for Active Cross References
-  for Passive Cross References.

However this option is also only available if you have selected an object within a library.

If you choose Predefined Selection, you can select a specific type of object to have its cross references displayed. The data displayed will then be referred to as "Predefined Selection Objects of Type X", where "X" represents one of the objects from the menu below:



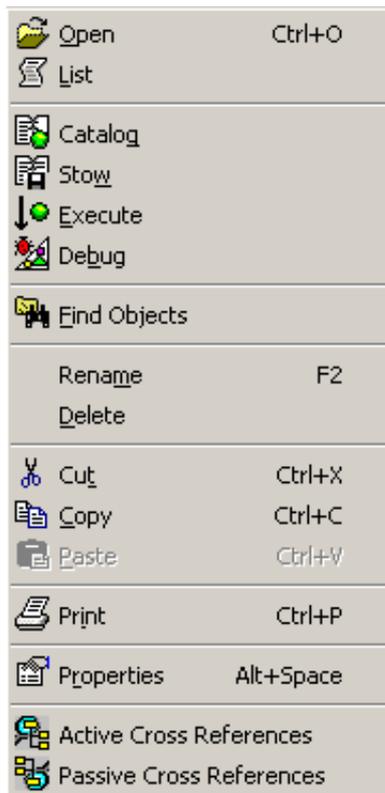
The root object in the resulting tree view is also called "Predefined Selection Objects of Type X", where "X" represents the chosen object. The source objects, for example programs and their cross references (either active or passive) are represented on the next level in the tree view (one level below the root object "Predefined Selection").

Note:

This option is only available if you are logged on a library.

Invoking XRef GUI Client From the Context Menu

Select an object within a library by left clicking on it. Right click on your selection to display the context menu. Select either Active Cross References or Passive Cross References. See the sample menu.

**Note:**

It is not possible to define a Predefined Selection (i.e. displaying the cross references of a specific object type, such as program or class) from the context menu.

Cross References

There are two kinds of cross references which you can display separately in the XRef GUI Client: active and passive cross references.

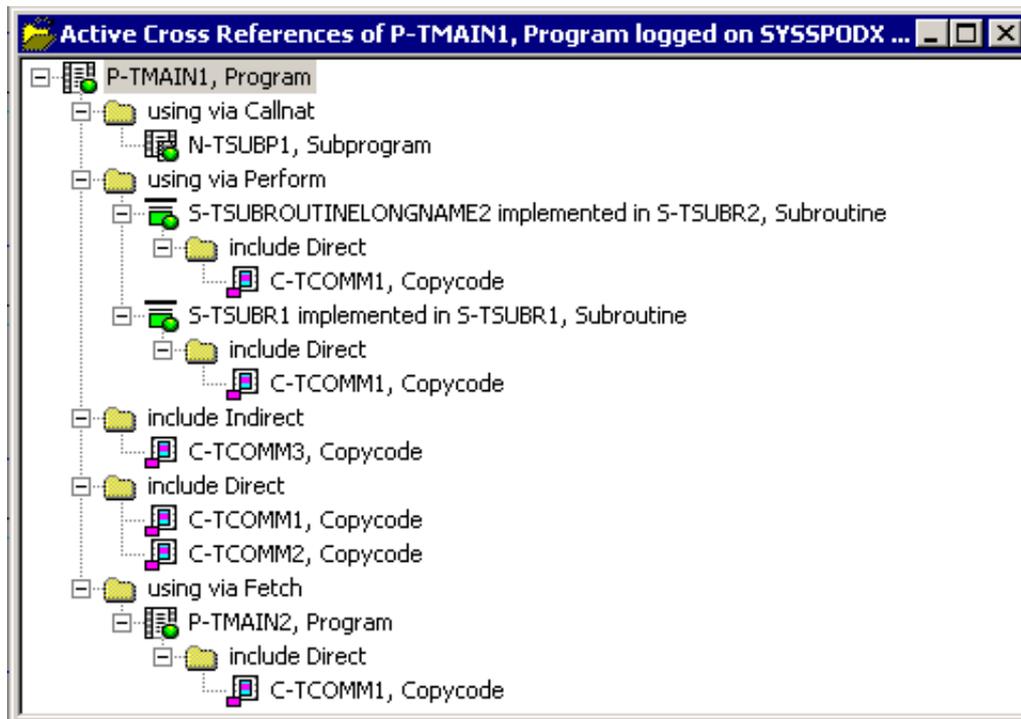
Active Cross References

If you select active cross references of an object, XRef GUI Client generates a tree view with the referencing ("using") object (for example a program) on top of the tree. Below this object, all objects which are referenced ("used") by the top level object are shown (for example other programs, subprograms or subroutines). For a more detailed explanation of the term object, see Contents of XRef Data.

Active cross references will provide you with an answer to the question: "Which parts of my application depend on the object I have currently selected?"

Example

In the following sample tree view the result of a request for active cross references is displayed. The referencing object P-TMAIN1 (a program) stands on top of the tree view. This is the object for which the request was performed:

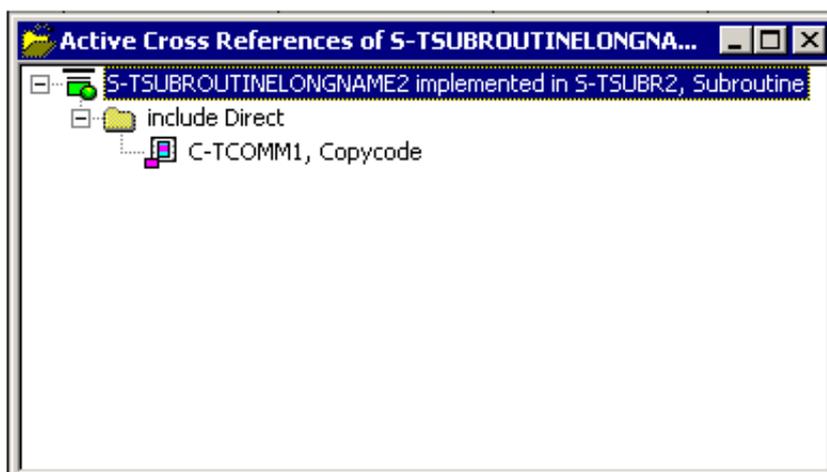


Note:

Subroutines with long names are also supported by the XREF GUI Client. Subroutines with long names are represented as follows:

<long name> implemented in <member name>, Subroutine

See sample tree view below.



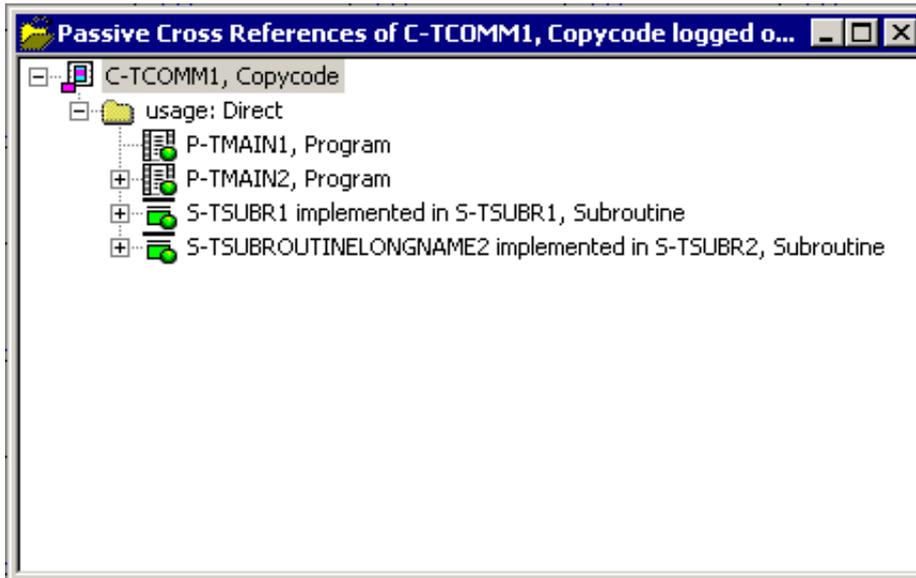
Passive Cross References

If you select passive cross references of an object, the XRef GUI Client generates a tree view with the referenced ("used") object on top of the tree. Below this object, all objects which reference ("use") the top level object are shown. For an explanation of the term object, see Contents of XRef Data.

Select passive cross references if you want to know which other objects use a certain object. For example, if you have a copycode you might want to know in which parts of your application it is included. There are some types of objects which by default only have passive cross references, such as copycodes, DDMs and methods.

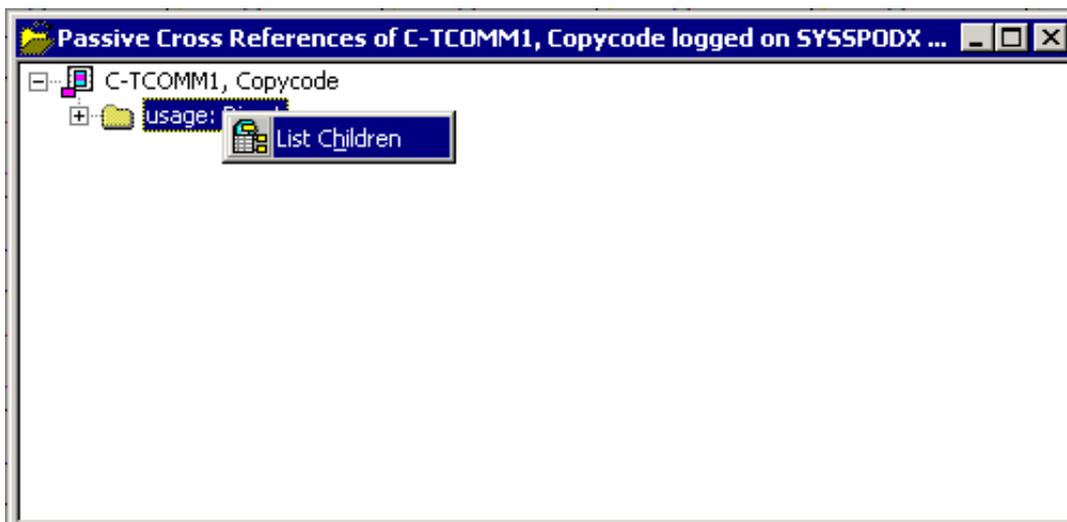
Example

In the following sample tree view, the result of a request for passive cross references is displayed. The referenced object C-TCOMM1 (a copycode) stands on top of the tree view. This is the object for which the request was performed:

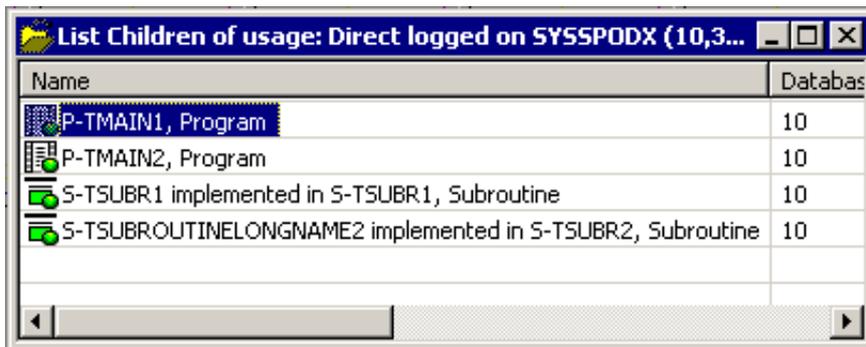


List Children

The option List Children is only available within the tree view on XRef data. This tree view is the result of your request processed by the XRef GUI Client. See the sample below.



The result of the List Children option is a list view of the children of the selected object (for example a program or subprogram) or node (for example *using via Perform* or *include Direct*). See the sample below.



Name	Databas
P-TMAIN1, Program	10
P-TMAIN2, Program	10
S-TSUBR1 implemented in S-TSUBR1, Subroutine	10
S-TSUBROUTINELONGNAME2 implemented in S-TSUBR2, Subroutine	10

In contrast to the tree view, you can select more than just one object within the list view and perform an action such as Catalog or Stow. This is very helpful when performing mass updates is necessary.

Influence of Environmental and User-specific Settings

The tree view on XRef data that is presented by XRef GUI Client always depends on the LOGON library and Steplib settings that are active during initial invocation/instanciation of the corresponding cross-reference tree view.

For example, if you switch to a different library while XRef GUI Client processes a request, the presented result will always refer to the library and Steplib settings that were active when you started your request. This may (but not necessarily has to) lead to different results if you start the same request on the same object, but with different LOGON libraries and/or Steplib settings.