THE IMPORTANCE OF SETUP VALUES
Documenting Multiple Environments

Most iSeries (AS/400, Power Systems, System i) sites separate their system into different environments. Some sites maintain two environments, production and test, while others may maintain many different environments. This article will explain the set up of PATHFINDER for individual environments, as well as the set up necessary for multiple environments.

How the iSeries handles environments

First, a discussion of how the iSeries handles environments. When a user signs on to the iSeries, a specific library list is assigned. This library list defines the environment for the user. As the user executes programs, the library list is used to qualify the library name for any unqualified object, or for any object qualified by *LIBL. The operating system searches the job's library list for files used by a program and selects only the first occurrence of a file. If a file is not located within a library on the library list, an error will occur.

How PATHFINDER handles environments

PATHFINDER creates X-ref data by emulating the operating system's execution of your programs. During this emulation, PATHFINDER refers to the DOCLIBL (Documentation Library List) to qualify object references in the same manner that the operating system uses the job's library list. PATHFINDER users create their own DOCLIBLs through option "1. DOCLIBL – your values" on the Setup Menu. These DOCLIBLs enable users to create library lists which reflect their actual environments. When generating X-ref data, a DOCLIBL is used to library qualify object references, just as the operating system would use the job's library list during execution. By providing a DOCLIBL which resembles the end-user's library list, PATHFINDER will qualify object references appropriately for the environment defined for that end-user. Thus, the DOCLIBL is the key to defining environments to PATHFINDER.

What is the purpose of the DOCLIBL?

The DOCLIBL is an important part of PATHFINDER. It has three main purposes:

1. As defined in the previous paragraph, the DOCLIBL is used to library qualify any objects referenced without a library qualifier, in the same manner as the operating system uses the job's library list.
2. The "Build/refresh" is the process which creates and updates PATHFINDER's X-ref (cross reference) information. A specific DOCLIBL can be designated by you during the Object and Field Build/refresh to serve as the list of libraries to be documented. Generally, when a DOCLIBL is created to define a particular environment, it will contain all the libraries which need to be documented. Therefore, when submitting the X-ref builds, "*DOCLIBL" can be specified for the "Library to X-ref" value.

3. “*DOCLIBL” is also a valid entry for the library parameter on many of the options within PATHFINDER, allowing you to direct your searches to objects residing in the libraries on the DOCLIBL specified in your Defaults.

In order for PATHFINDER to properly qualify object references within environments, it is necessary to build X-ref data along environmental lines. For the same reasons you would not include test libraries on a production user’s library list, you would not want to include test libraries on a DOCLIBL used to document the production environment. You will need to create a separate DOCLIBL for each environment to be documented.

What is the purpose of the Source List?

PATHFINDER must access source for the programs being documented when building both the Object and Field X-ref data. By default, the build routines will look for the source member shown in the program object description. If this source member has been moved, the Source List (see option “2. Source list – your values” on the Setup Menu) must be used to tell the Build/refresh where this source member now resides. The Source List also accepts entry of DDM source files for those sites which store source on a separate system. Finding the correct source to document a program is vital to the accuracy of the cross reference data. Without program source, all object types can not be documented in the Object X-ref, and no field usage can be documented in the Field X-ref. If either the Object or Field Build/refresh has problems locating source, it will be noted on the Problem Log Reports generated when the Build/refresh runs.

“*SRCL” is also a valid entry for the library parameter on many PATHFINDER options. The Source list to be used is specified in your Defaults.

How to setup PATHFINDER for your system

Tailoring PATHFINDER to your site’s needs is accomplished through options on the Setup Menu. Start by taking option “1. DOCLIBL – your values”. This will display the DOCLIBL associated with your user profile. Your “user DOCLIBL” will be helpful to you for PATHFINDER options, however, there is a potential danger in using your “user DOCLIBL” to build X-ref data. If another user (with a different “user DOCLIBL”) updates X-ref information that you created with your “user DOCLIBL”, the integrity of the X-ref could be endangered. For this reason, we recommend using an application specific “Alternate DOCLIBL” to create and update Object and Field X-ref.

To access Alternate Setup Values, select “F8=Work with” while displaying any Setup Value. “F8=Work with” from the DOCLIBL screen will display a list of all the DOCLIBLs defined on your system. Previously created DOCLIBLs can be displayed, maintained, or deleted via the subfile options provided. To create an Alternate DOCLIBL, select “F6=Create”. When creating new DOCLIBLs, you will be prompted for a DOCLIBL name and text description. The name you assign will be used to identify the DOCLIBL created, so should be meaningful to all staff members. An Alternate DOCLIBL should be created for each environment to be documented.

In addition to the DOCLIBL, Alternate Setup Values can be created and maintained for the Source List, Build Values, CALLING List, and Omit List. These Alternate Setup Values are maintained in the same manner as the Alternate DOCLIBL. For consistency, the Alternate Source List, Build Values, etc. can be assigned the same name as the Alternate DOCLIBL. To aid in the creation of multiple Setup Values, any Setup Value can be copied to a new or existing value. This can be used for Standard, User, or Alternate values.

Building X-ref data

PATHFINDER's Object X-ref is built and refreshed with option "1. Build/refresh object X-ref" from the Object X-ref Menu. The Field X-ref is built and refreshed with option "1. Build/refresh field X-ref" from the Field X-ref Menu. When using these interactive screens to submit Build/refresh jobs, select "F6=Assign setup values" to designate which Setup Values the Build/refresh routine should use. By default, your user Setup Values will be used. Using "F6=Assign setup values" will allow you to enter an Alternate Setup Value you have created. Each X-ref may also be built or refreshed through a CL procedure. If you are executing a Build/refresh from a CL procedure, there are parameters on both the BLDOBJ (Build/refresh object X-ref) and BLDFLD (Build/refresh field X-ref) commands to specify which Setup Values should be used.

Where to store your cross reference

When generating X-ref data for multiple environments, an important consideration is where to store the X-ref data. The library in which X-ref data is to be stored is specified when submitting the X-ref build. X-ref data for multiple environments can be stored within the same X-ref library, or it can be stored in different libraries.

You may wish to use one library for multiple environments if one person is responsible for maintaining the X-ref data for all environments. To store X-ref data for multiple environments in the same X-ref library, submit multiple build jobs which specify the same X-ref library. The first build would specify "Y" for "Clear existing X-ref before saving new" (SAVE(*CLEAR)) to clear out any old X-ref data before saving the newly generated data. NOTE: This value will only be displayed if the library specified for "Library to store X-ref data in" already contains X-ref information. Builds for subsequent environments would identify different Setup Values but the same X-ref library, specifying "N" for "Clear existing X-ref before saving new" and then select option "2. Add x-ref data" (SAVE(*ADD)) to add the data generated to the X-ref already existing in the library.

Most sites choose to store X-ref data for each environment within its own X-ref library. This method works well for sites where different programmers or groups of programmers are responsible for maintaining particular environments. By storing the X-ref data separately, each programmer can be responsible for the maintenance of their own X-ref data, without affecting other X-ref libraries. There is no limit to the number of different X-ref libraries which can exist on your system.

Where is this information coming from?

For sites using multiple X-ref libraries, controlling which library is used when retrieving X-ref data is easy. When making X-ref inquiries, the X-ref library name is displayed in an input-capable field in the upper right corner of the inquiry screen. Information can be extracted from a different X-ref library by changing the X-ref library name and pressing enter. As an added feature, PATHFINDER stores the name of the last X-ref library used by each user in their "Default values" (see Setup Menu) and automatically defaults each user to the last X-ref library they used. This eliminates the need to indicate to PATHFINDER which X-ref library to use every time an inquiry is made.

Keeping X-ref up-to-date

After initially building X-ref for your environments, it is very important to create a strategy for the ongoing maintenance of the X-ref data. PATHFINDER's Refresh portion of the Build/refresh options is designed to easily keep your X-ref data up-to-date with any changes that have taken place within your environments. Parameters on the refresh commands allow specification of Alternate Setup Values (DOCLIBL, Source list, etc.), as well as the library in which X-ref data is to be refreshed. We recommend creating a procedure which will refresh X-ref data for all environments necessary. By scheduling the periodic execution of this procedure, sites can easily insure that X-ref data remains up-to-date and accurate for all environments. PATHFINDER has a built-in feature which will actually help write this procedure for you. For further information on PATHFINDER's Refresh, how it works, and how to automate it, please refer to the Tips and Techniques article entitled "Up-to-date Effortlessly" (attached).

Review your history

If X-ref data is already built on your system, you can review all the details of how it was built by accessing history. History can be viewed by selecting "F22=History" when making an X-ref inquiry, or by prompting the DSPHST command. The X-ref history will detail each build for a particular X-ref library. Information such as who submitted the build, when it ran, how many programs were documented, as well as each of the Setup Values (DOCLIBL, Source List, Build Values, etc.) is captured at the time the build is executed. If you question how or when X-ref data was built or refreshed, your answer can be found in the history.

Once PATHFINDER is set up appropriately for your system, the entire staff can take full advantage of having on-line X-ref data available. The built-in flexibility will enable every site to build and store X-ref data in a manner which works best for them.

For further information, please contact our Technical Services department. We can be reached by email at info.hawkinfo.com or call us Monday-Thursday, 7 a.m. to 5 p.m. (MST) and Friday 7 a.m. to 3 p.m., VOICE (970) 498-9000 or FAX (970) 498-9096.