

Xref Analysis Tool

Scott Auge
sauge@amduus.com

Amduus Information Works, Inc.
<http://www.amduus.com>

Table of Contents

License.....	3
Introduction.....	4
Installation.....	5
Install PostgreSQL.....	5
Install PHP.....	5
Install Xref Analysis.....	5
Creating the XREF database in PostgreSQL.....	6
Configuring DB Connection for Xref Analysis.....	7
Loading The Database.....	9
Querying The Database.....	10

License

XRef Analysis

Copyright (C) 2007 Scott Auge and contributors

sauge@amduus.com scott_auge@yahoo.com

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Introduction

One of the overlooked tools available to the Progress ABL (4GL) programmer is the xref file that can be created by the COMPILE statement. It can help identify table scanning searches, include files used, tables used, etc. in an easy to read manner.

When combined with the xref files of other programs in an application, a bigger picture can appear of what is happening. For example, if you change an include file – which programs use that include file which will require re-compiling? If you are making changes to how a table works – which programs use that table? Making changes to a program? Which programs call that program so a testing scheme can be worked out?

When you look at this tool, you will immediately realize that it is not a Progress Database nor a Progress language but the PostgreSQL¹ open source database and the PHP² open source language. Why didn't I choose Progress? One reason is that I wanted it to be web based as a centralized repository and not everyone has Webspeed available for use in that way. Another reason is that I use an Apple most of the time and it is convenient to have tools available on it when Progress apps don't run on it. (I remote desktop³ into a Windows computer to access Progress.)

1 <http://www.postgresql.org/> This is a pretty popular DB and one can see some of it's users at <http://www.postgresql.org/about/users>
2 <http://www.php.net>
3 <http://www.rdesktop.org>

Installation

Install PostgreSQL

First you need to install PostgreSQL if you have not already done so. This is actually pretty quick if you are familiar with it and might take half a day or so if you need to learn how to do it. It is pretty forgiving on configurations. One should see the documentation that is available for PostgreSQL for these instructions.

Install PHP

Once you have PostgreSQL installed, you will need to insure PHP is installed. PHP does not automatically come linked with the PostgreSQL clients. Here is how I configured PHP on my machine (you will see it is very light):

```
'./configure' '--with-apxs2=/usr/local/apache2/bin/apxs' '--with-curl' '--enable-ftp' '--with-mysql=/usr/local/mysql-test' '--with-pgsql' '--enable-soap' '--enable-sockets' '--with-pear'
```

The important parts are `--with-apxs` which identifies which entry points in the version of Apache you have available as well as `--with-pgsql` which assumes it is installed in `/usr/local/pgsql`.

Important: When you are compiling PHP or installing PHP binaries, be sure the command line tools are made available since we will be running PHP scripts from the command line!

All of these tools are available for Windows but I cannot give such instructions since I run UNIX.

Install Xref Analysis

Once you have those configured, download the zip file containing the application⁴. It

⁴ Often you can find this in <http://www.amduus.com/OpenSrc/SrcLib/XRefAnalysis>

will be named xref.yyyyjjjhhmmss.zip where:

yyyy represents the year of the distribution

jjj represents the day of the year of the distribution

hh represents the hour of the distribution

mm represents the minute of the distribution

ss represents the seconds of the distribution.

Unzip this file in a directory of your web server's DocumentRoot.

Creating the XREF database in PostgreSQL

Once you have the PostgreSQL database server up and running, you will need to create a database and load the schema SQL into it. Here are some commands to do so:

```
postgres@gaius:/appl/php/xref/sql> /usr/local/pgsql/bin/createdb -O scottauge  
-h gaius -p 5432 xref1
```

which should reply:

```
CREATE DATABASE
```

This creates a database named xref1 on a machine named gaius for a server running on port 5432 which is owned by scottauge.

Next we load a schema into the database:

```
postgres@gaius:/appl/php/xref/sql> /usr/local/pgsql/bin/psql -d xref1 -h  
gaius -p 5432 < postgresql.sql
```

which should reply:

```
SET  
SET  
SET  
SET  
SET  
SET  
SET  
CREATE TABLE  
ALTER TABLE  
COMMENT  
COMMENT
```

```
COMMENT
COMMENT
COMMENT
COMMENT
COMMENT
COMMENT
ALTER TABLE
CREATE INDEX
```

As a check, logging into the DB⁵ via

```
postgres@gaius:/appl/php/xref/sql> /usr/local/pgsql/bin/psql -d xref1 -h
gaius -p 5432
```

which says log into database xref1 found on server gaius on port 5432 (aka mpro -db xref1 -H gaius -S 5432 -N TCP in the Progress world) and then issuing the command

```
xref1-# \d
```

You should see:

```
                List of relations
 Schema | Name   | Type  | Owner
-----+-----+-----+-----
 public | xref_raw | table | scottauge
(1 row)
```

The database should be good to go, now we need to tell the program how to talk to it.

Configuring DB Connection for Xref Analysis

Under the /src directory of the unzipped file for xref analysis you should see a config.php file. It will have a line similar to:

```
$dsn = 'pgsql://username:password@tcp+hostname:port/dbname';
```

For example, using our examples, it would read

```
$dsn = 'pgsql://scottauge:secret@tcp+gaius:5432/xref1';
```

5 Using a GUI tool like <http://www.pgadmin.org/screenshots/> or a web based tool like <http://phppgadmin.sourceforge.net/index.php> can make this much easier if you are unfamiliar with a command line tool.

This basically instructs the PHP code how to connect to the proper database you have set up.

Once you have the database set up and the Xref Analysis PHP code configured to talk with it, you can start loading it up.

Loading The Database

The first thing you will need to do is create your XREF files. This can be done manually or by a tool like Zammi⁶ which walks the propath given generating r-code, listings, and xref files as told to.

Once you have your list of xref files, you need to load them into the database. Under the /script directory is a command to do so. An example invocation is:

```
./ldxref -delete -appl test -propath "/appl/schdeve/work:/appl/schdeve/source"
```

Which basically says, delete all entries in the database for an application named test and load with xrefs found in the propath /appl/schedeve/work:/appl/schdeve/source.

If you do not have php as part of your path, you will need to change the line

```
php ldxref.php $@
```

in ldxref to use the absolute path to the php command line interpreter.

You must cd into the /scripts directory for any scripts that might call this automatically (such as cron.)

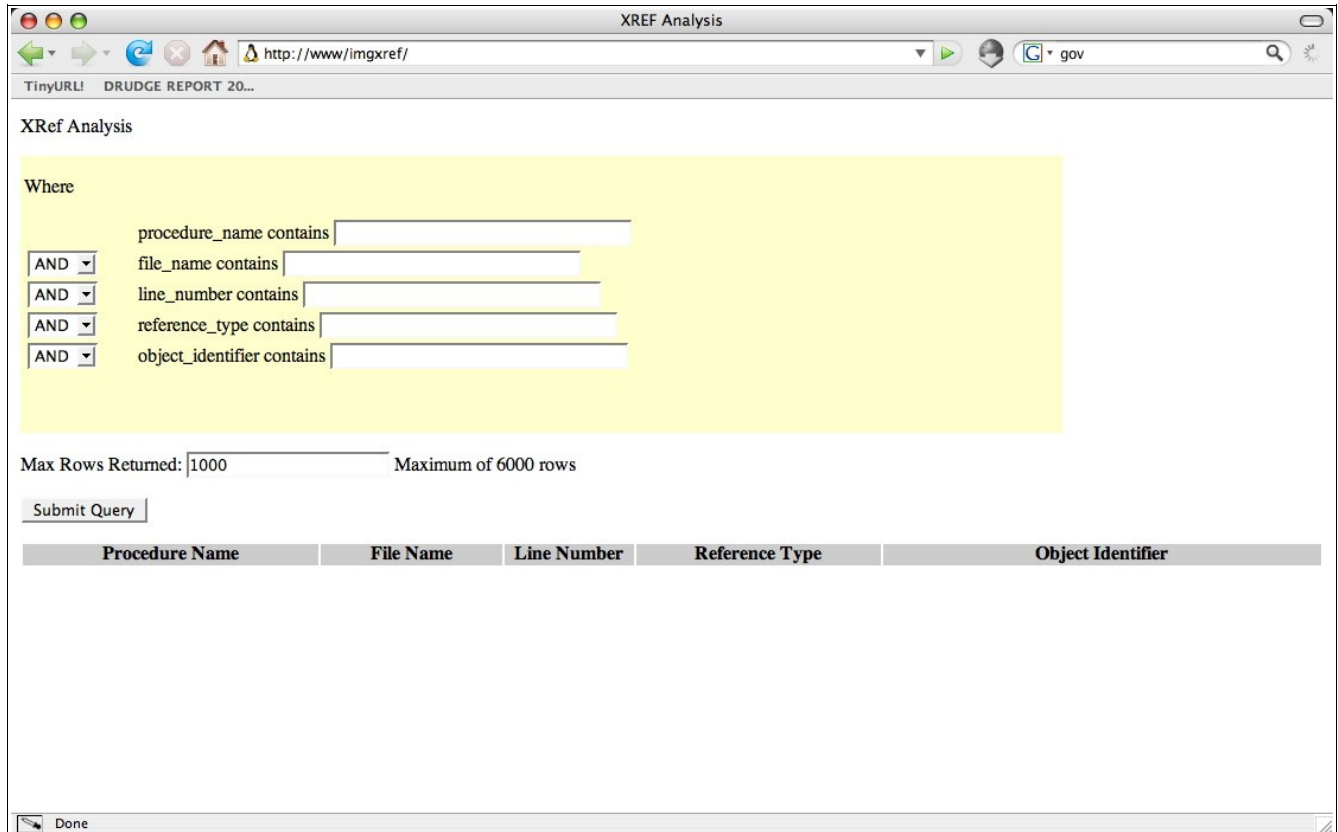
Be aware that loading large applications may take a couple of hours. A 771,201 line application took one and half hours to load on a dual 3MHz Pentium 4 CPU.

As you can imagine, using Zammi and Xref Analysis, you can nightly compile and load xrefs via a scheduler such as cron or what have you.

⁶ <http://www.amduus.com/OpenSrc/SrcLib/Zammi/>

Querying The Database

When you first access the query tool, you will receive a page similar to the following:



The screenshot shows a web browser window titled "XREF Analysis" with the URL "http://www/imgxref/". The page content includes a search form with the following fields:

- procedure_name contains
- AND file_name contains
- AND line_number contains
- AND reference_type contains
- AND object_identifier contains

Below the search form, there is a "Max Rows Returned:" field with a value of "1000" and a note "Maximum of 6000 rows". A "Submit Query" button is located below the form.

Procedure Name	File Name	Line Number	Reference Type	Object Identifier
----------------	-----------	-------------	----------------	-------------------

The status bar at the bottom of the browser window shows "Done".

This screen allows you to search on the major components of an Xref file. If you leave the entry blank, it is ignored. If you wish to combine some entries do so by filling in the blank and choosing the conjunction to use.

Sometimes you can get a lot of entries returned – you can specify the maximum number⁷ of rows to return so you don't over-run your web browser.

⁷ If you need more than 6000 rows to be returned, update the line 189 (or thereabouts) in index.php that reads “if (\$Limit > 5000) \$Limit = 6000;” to the number you wish to max out at.

XREF Analysis

TinyURL! DRUDGE REPORT 20...

Where

procedure_name contains _____

AND file_name contains activityleader

AND line_number contains _____

AND reference_type contains include

AND object_identifier contains whole-index

Max Rows Returned: 1000 Maximum of 6000 rows

Submit Query

Issuing SQL: SELECT * FROM xref_raw WHERE UPPER(file_name) LIKE UPPER('%activityleader%') AND UPPER(reference_type) LIKE UPPER('%include%') OR UPPER(object_identifier) LIKE UPPER('%whole-index%') ORDER BY procedure_name LIMIT 1000

Procedure Name	File Name	Line Number	Reference Type	Object Identifier
/appl/schdeve/source/actactivityleader.p	/appl/schdeve/source/actactivityleader.p	1	INCLUDE	src/web/method/e4gl.i
/appl/schdeve/source/actactivityleader.p	/appl/schdeve/source/actactivityleader.p	39	INCLUDE	sysmain.i
/appl/schdeve/source/actactivityleader.p	/appl/schdeve/source/actactivityleader.p	43	SEARCH	school.ActivityStatus ActivityStatusDescr WHOLE-INDEX
/appl/schdeve/source/actactivityleader.p	/appl/schdeve/source/actactivityleader.p	251	INCLUDE	looksetup.i
/appl/schdeve/source/actactivityleader.p	/appl/schdeve/source/actactivityleader.p	259	INCLUDE	pagescript.i
/appl/schdeve/source/actactivityleader.p	/appl/schdeve/source/actactivityleader.p	260	INCLUDE	sysheader2.i
/appl/schdeve/source/actactivityleader.p	/appl/schdeve/source/actactivityleader.p	261	INCLUDE	mnutabheader2.i

Done

In this example, for the program named activityleader, tell us the INCLUDE files used as well as denote any table scans occurring (the whole-index keyword.)

XREF Analysis

Where

procedure_name contains

AND file_name contains

AND line_number contains

AND reference_type contains

AND object_identifier contains

Max Rows Returned: Maximum of 6000 rows

Issuing SQL: SELECT * FROM xref_raw WHERE UPPER(reference_type) LIKE UPPER('%search%') AND UPPER(object_identifier) LIKE UPPER('%webstate%') ORDER BY procedure_name LIMIT 1000

Procedure Name	File Name	Line Number	Reference Type	Object Identifier
/appl/schdeve/source/actactivity.p	/appl/schdeve/source/sysmain.i	60	SEARCH	school.WebState sessionPageData
/appl/schdeve/source/actactivity.p	/appl/schdeve/source/sysmain.i	93	SEARCH	school.WebState SessionID
/appl/schdeve/source/actactivity.p	/appl/schdeve/work/sysheader2.i	68	SEARCH	school.WebState sessionPageData
/appl/schdeve/source/actactivity.p	/appl/schdeve/source/sysmain.i	76	SEARCH	school.WebState sessionPageData
/appl/schdeve/source/actactivity.p	/appl/schdeve/source/actactivity.p	566	SEARCH	school.WebState sessionPageData
/appl/schdeve/source/actactivityadult.p	/appl/schdeve/source/sysmain.i	60	SEARCH	school.WebState sessionPageData
/appl/schdeve/source/actactivityadult.p	/appl/schdeve/source/sysmain.i	76	SEARCH	school.WebState sessionPageData
/appl/schdeve/source/actactivityadult.p	/appl/schdeve/source/sysmain.i	93	SEARCH	school.WebState SessionID

Done

In the above example, we examine all the programs that do a search (FIND or FOR EACH or Dynamic Query) on the webstate table in the application.

Examine the COMPILER statement in the Language Reference Guide for additional terms to search on in the Reference Type and Object Identifier areas.