Subject: Link IDL to Oracle (VMS Case)
Posted by George Constantinides on Fri, 24 Nov 2000 01:53:10 GMT
View Forum Message <> Reply to Message

Hi,
Following is a method to link IDL to ORACLE in VMS with the
CALL_EXTERNAL method.
This tip along with the discussion on CALL EXTERNAL in the
ADVANCED DEVELOPMENT GUIDE (chapter 4) should be
adequate for most VMS developers.

PS. Since Kodak is not going to support VMS after IDL 5.4, I am posting
this mainly as a curiosity. Who knows? If this post, and others
like
it, generate enough interest Kodak might reconsider.

LINKING IDL TO ORACLE PRO*FORTRAN (OpenVMS Case)

PROBLEM:

Access of ORACLE database data from IDL in a OpenVMS environment.

ENVIRONMENT:

RSI Product: IDL V5.3
OS Platform: OpenVMS V7.2
Layered Products: ORACLE V8.05, PRO*FORTRAN V1.8 - DIGITAL Fortran 90
HW Platform: COMPAQ ALPHA

DISCUSSION:

To compile Oracle PRO*FORTRAN code into a sharable image that can be
used with CALL_EXTERNAL do the following:

1) Precompile the PRO*FORTRAN file to generate a FORTRAN .for source:

```
$ PROFOR file.pfo
```

2) Look at the first few lines of the generated "file.for" and make a
note
of the function name and ALL the COMMON blocks generated by
PRO*FORTRAN.
Typically the function name appears at the beginning of the file,
followed by variable declarations and two or more COMMON blocks.

The names of the ORACLE PRO*FORTRAN generated COMMON blocks will be something like "fileI", "fileC"... Example:

```
REAL*4 FUNCTION test(VAR1,VAR2)
EXTERNAL testD
INTEGER*4 SQLITR
INTEGER*4 SQHSTV(1)
 ....
COMMON / testI / IAPSUC,IAPFAI,IAPFTL,SQLCTX,SQC000
COMMON / testC / SQLFNM
```

Note:
PRO*FORTRAN uses the function name to construct its own internal COMMON blocks. If the function name is longer that five characters it truncates the name of its COMMON blocks to five characters and appends the letters "I","C",etc.. at the end of it.

3) Create a linker OPTION file, "file.OPT" and setup the function name as a universal symbol. Make COMMON blocks non-sharable PSECTS.

Example:

```
SYMBOL_VECTOR=(test=PROCEDURE)
PSECT_ATTR=testI,noshr
PSECT_ATTR=testC,noshr
```

4) Use the PRO*FORTRAN link procedure LNPROFOR to create a sharable image from "file.OBJ". Include in your link command any other
objects
(and/or object libraries) that you are linking with, and include the

Option file with the /OPT/SHARE keywords. Example:

```
$ LNPROFOR test -
DISK:[OBJECT_DIRECTORY]test,-
DISK:[OTHER_OBJECTS_AND_LIBRARIES]objects,library/LIB,-
DISK:[OPTION_FILE_DIRECTORY]test/OPT/SHARE
```

5) You have now created a shareable image that is usable with the CALL_EXTERNAL method in IDL.
Define a VMS logical that points to the name on the sharable image.
Example:

```
$ DEFINE  test_EXE  DISK:[DIRECTORY]test.EXE
```

6) Start up IDL and use CALL_EXTERNAL to invoke the sharable image.
Example:

```
IDL> OUT = CALL_EXTERNAL(/F_VALUE, 'test_EXE', 'test', 0L, 0L)
IDL> PRINT, OUT
```

--
Regards,

George Constantinides
GeorgeC@mhl.nsw.gov.au
URL   http://www.mhl.nsw.gov.au