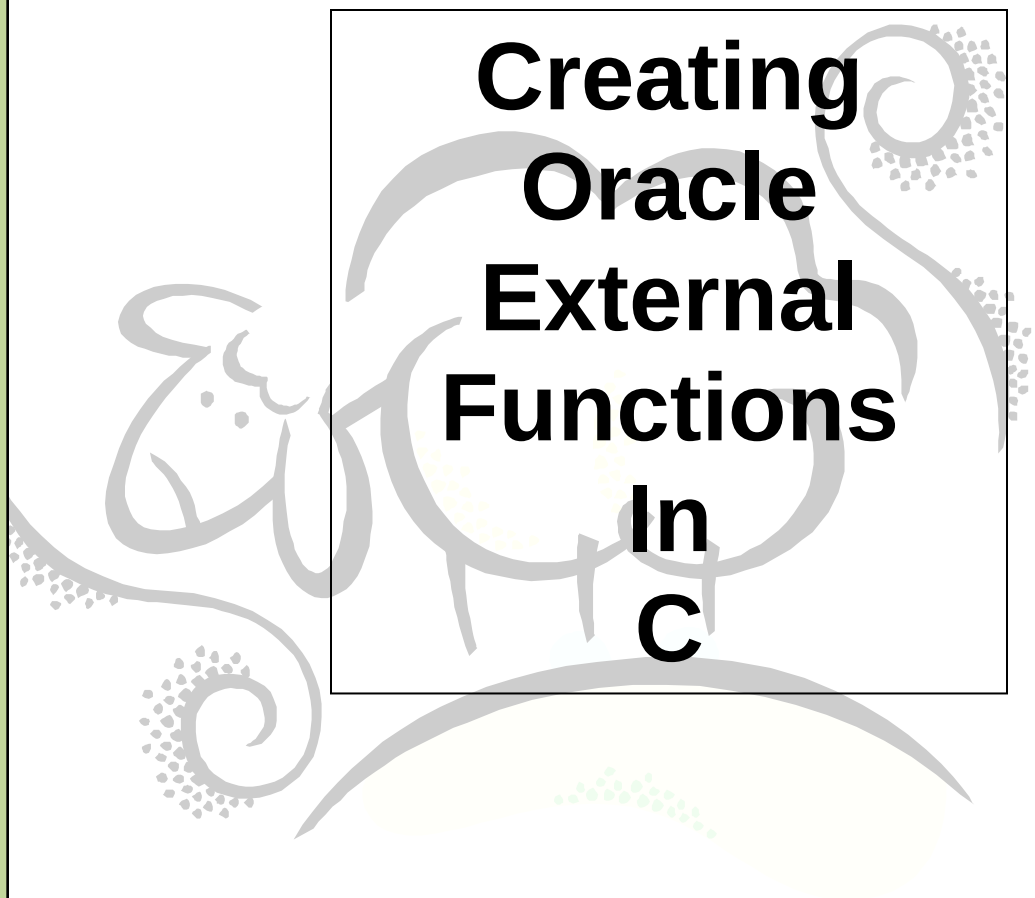


Creating Oracle External Functions In C

This presentation as well as other presentations and documents found on the sheepsqueezers.com website may contain quoted material from outside sources such as books, articles and websites. It is our intention to diligently reference all outside sources. Occasionally, though, a reference may be missed. No copyright infringement whatsoever is intended, and all outside source materials are copyright of their respective author(s).

Table of Contents

Introduction

This is a basic introduction about how to create a C function on the *Linux* operating system that will be used in PL/SQL or SQL. That is, you can create a C "DLL" or "Shared Object" called, say, `GetItHere()`, that can be used like this:

```
SELECT GetItHere(123)
     FROM DUAL;
```

or

```
SET SERVEROUTPUT ON
DECLARE

 result NUMBER;

 BEGIN

  dbms_output.enable;

  result := sys.GetItHere(19);
  dbms_output.put_line(result);

END;
/
```

There is also a section at the end on how to compile Pro*C applications.

Steps to Creating a C External Function with Example

Step1: Create your C program containing the functions you want to use in SQL or PL/SQL

```
//*---------------------------------------------------------------------------*
//* Program:      mylib.c                                                     *
//* Program Type: C                                                           *
//* Authors:                                                                  *
//* Date:                                                                     *
//*                                                                           *
//* Application:  Test                                                        *
//*                                                                           *
//* Abstract:     This C-Program contains a function which is used by an Oracle*
//*               PL/SQL function via a LIBNAME created with CREATE LIBRARY.  *
//*                                                                           *
//* Invocation:   None.                                                       *
//*                                                                           *
//* Assumptions:  Users must have EXECUTE privilege on the library.           *
//*                                                                           *
//* Parameters:   N/A                                                         *
//*                                                                           *
//* Input:        N/A                                                         *
//*                                                                           *
//* Output:       mylib.so                                                    *
//*                                                                           *
//* Example:      None.                                                       *
//*                                                                           *
//* Notes:        None.                                                       *
//*                                                                           *
//* Modification History:                                                     *
//* Date          Prog    Mod #   Reason                                      *
//* ---------     ----    -----   ------------------------------------------- *
//*---------------------------------------------------------------------------*


//*---------------------------------------------------------------------------*
//* Includes                                                                  *
//*---------------------------------------------------------------------------*
#include "stdio.h"
#include "stdlib.h"
#include "string.h"

//*---------------------------------------------------------------------------*
//* Function Prototypes                                                       *
//*---------------------------------------------------------------------------*
void cmdexec(char*);
short int addone(short int);

//*---------------------------------------------------------------------------*
//* Name: cmdexec                                                             *
//* Purpose: Executes a command at the Linux command prompt.                  *
//*---------------------------------------------------------------------------*
void cmdexec(char *cmd) {

 int iRC;

 iRC=system(cmd);
}

//*---------------------------------------------------------------------------*
//* Name: addone                                                              *
//* Purpose: Adds one to an incoming integer.                                 *
//*---------------------------------------------------------------------------*
short int addone(short int anInt) {
 return ++anInt;
}
```

Step2: Create a test C Program to Test that the Function Works Properly

```
//*-------------------------------------------------------------------------*
//* Program:      test1.c                                                   *
//* Program Type: C                                                         *
//* Authors:                                                                *
//* Date:                                                                   *
//*                                                                         *
//* Application:  Test                                                      *
//*                                                                         *
//* Abstract:     Test Program                                              *
//*                                                                         *
//* Invocation:   None.                                                     *
//*                                                                         *
//* Assumptions:                                                            *
//*                                                                         *
//* Parameters:   N/A                                                       *
//*                                                                         *
//* Input:        N/A                                                       *
//*                                                                         *
//* Output:       test1                                                     *
//*                                                                         *
//* Example:      None.                                                     *
//*                                                                         *
//* Notes:                                                                  *
//*                                                                         *
//* Modification History:                                                   *
//* Date          Prog    Mod #   Reason                                    *
//* ---------     ----    -----   ------------------------------------------*
//*-------------------------------------------------------------------------*

extern int cmdline(char *);
extern int addone(int);

int main(void) {

 char astr[9]={"ls -alF\0"};

 printf("STRING FROM PROGRAM===>%s\n",astr);
 printf("RETURN CODE===>%d\n",cmdexec(astr));
 printf("RETURN CODE===>%d\n",addone(5));

 return 0;
}
```

Step3: Run the Following Script to Compile Your Functions and the Test Program

```
#!/bin/bash

# Set up the Load Library Path to the location where your C Program is located as well as Oracle libraries
LD_LIBRARY_PATH=/home/oracle/t1d/cprogs/OracleCProgs:/usr/local/app/oracle/product/9.2.0/lib:/lib:/usr/lib:/usr/local/lib
export LD_LIBRARY_PATH

echo "$LD_LIBRARY_PATH=" $LD_LIBRARY_PATH

# Remove the previous Shared Object mylib.so
rm -f test1 mylib.so

# Compile the C program containing your functions.  Note that –shared keyword to make a .so file and not a stand-alone
executable.
gcc -o /home/oracle/t1d/cprogs/OracleCProgs/mylib.so -shared -L/usr/local/app/oracle/product/9.2.0/rdbms/demo mylib.c

# Print out some useful information about the shared object using the file command.
file mylib.so

# Compile the test C program.
gcc -o test1 -L/home/oracle/t1d/cprogs/OracleCProgs mylib.so test1.c

# Print out some useful information about the test program.
file test1

echo "*-----------------------------------------------*"
echo "* BEGIN TEST PROGRAM NOW!!                      *"
echo "*-----------------------------------------------*"

# Run the test program
./test1

echo "*-----------------------------------------------*"
echo "* END TEST PROGRAM NOW!!                        *"
echo "*-----------------------------------------------*"

# Copy the Shared Object to a general location…here I put it with the other Oracle libraries.
cp -v mylib.so /usr/local/app/oracle/product/9.2.0/lib/mylib.so

exit
```

Step4: Log into the database as SYS and tell Oracle Where the External Functions Are Located

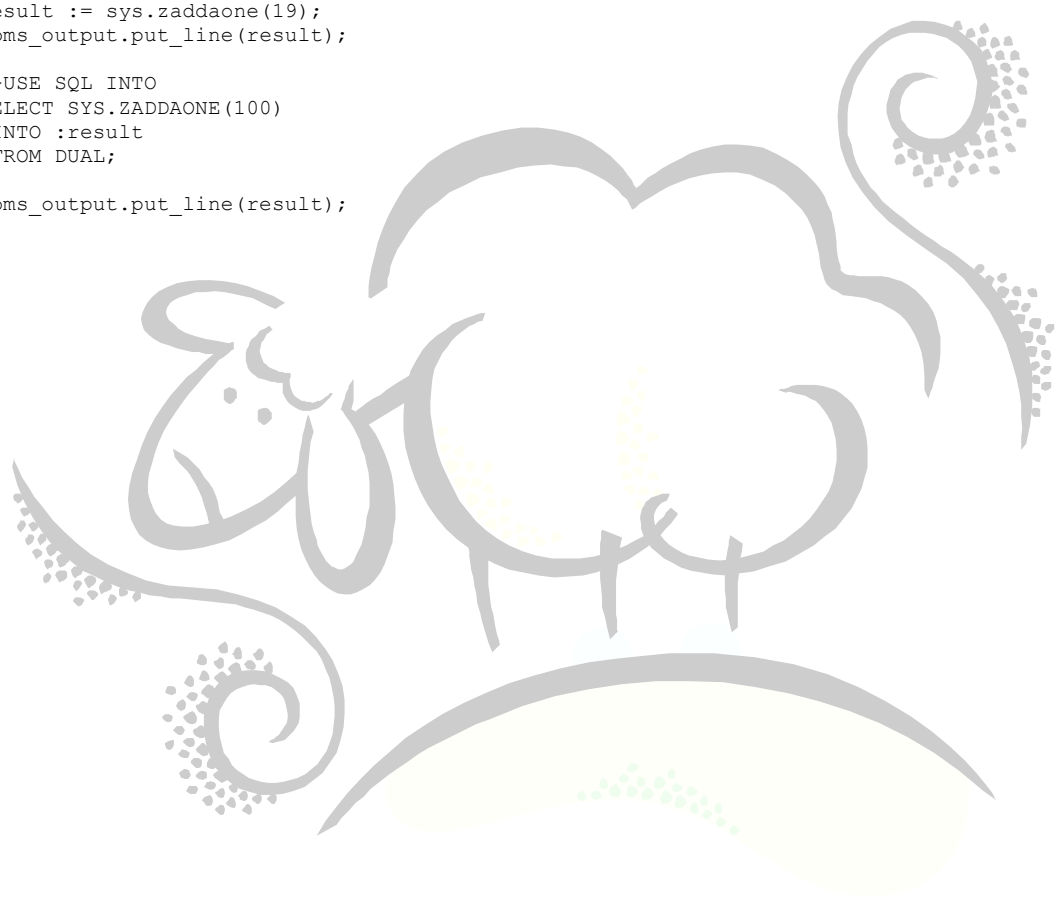
Note that you may have to GRANT CREATE LIBRARY access to the user who is running the following DDL.

```
--Place the exact location of the Shared Object in tick marks.
CREATE OR REPLACE LIBRARY mylib AS '/usr/local/app/oracle/product/9.2.0/lib/mylib.so'
/
SHOW ERRORS
/
--Create a Procedure to Reference the C Procedure "cmdexec"
CREATE OR REPLACE PROCEDURE CMDEXEC(cmd IN CHAR)
 AS EXTERNAL
 NAME "cmdexec"
 LIBRARY mylib
 LANGUAGE C
 PARAMETERS (cmd STRING);
/
SHOW ERRORS
/
--Create a Function to Reference the C Function ADDAONE
CREATE OR REPLACE FUNCTION ZADDAONE(inint IN BINARY_INTEGER) RETURN BINARY_INTEGER
 AS LANGUAGE C
 NAME "addone"
 LIBRARY mylib
 PARAMETERS (inint short,RETURN short);
/
SHOW ERRORS
/
```

Step5: Test Your Function Using PL/SQL and/or SQL

Note that you may have to GRANT `EXECUTE ANY PROCEDURE` access to the procedure cmdexec and the function `zaddaone`.

```
SET SERVEROUTPUT ON
DECLARE

 result NUMBER;

 BEGIN

  dbms_output.enable;

  --rc := sys.cmdexec('ls -alF');
  --sys.cmdexec('ls -alF');

  result := sys.zaddaone(19);
  dbms_output.put_line(result);

  --USE SQL INTO
  SELECT SYS.ZADDAONE(100)
   INTO :result
   FROM DUAL;

  dbms_output.put_line(result);

END;
/
```



Compiling Pro*C Applications on Linux

Here are the step needed to compile a Pro*C program on a Linux machine:

1. Ensure the Pro*C software is installed. You can find the proc binary in

   /usr/local/app/oracle/product/9.2.0/bin/proc

   If you cannot find it, then you need to run the OUI to install the Pro*C software.

2. Update the file /usr/local/app/oracle/product/9.2.0/precomp/admin/pcscfg.cfg. Here is an Example pcscfg.cfg file:

```
[oracle@marlin CPrograms]$ cat /usr/local/app/oracle/product/9.2.0/precomp/admin/pcscfg.cfg
sys_include=(/usr/lib/gcc-lib/i386-redhat-linux/3.2.3/include,/usr/include,/usr/lib/gcc-lib/i486-suse-
linux/2.95.3/include,/usr/lib/gcc-lib/i386-redhat-linux/2.96/include)
ltype=short
include=(/usr/lib/gcc-lib/i386-redhat-linux/3.2.3/include)
include=(/usr/include/linux)
```

3. Write your Pro*C program, and make sure the extention is .pc and not .c.
4. Run proc on your Pro*C program: `proc program.pc`. This creates a C program called `program.c`
5. Next, compile the C program: `make -f demo_proc.mk build OBJS=`*`program`*`.o EXE=`*`program`*
6. You now have an executable called *program*.

You can find demo_proc.mk in /usr/local/app/oracle/product/9.2.0/precomp/demo/proc.

Here is a sample script to compile your Pro*C program:

```
[oracle@marlin CPrograms]$ cat makePROCEXE
#!/bin/bash
#LD_LIBRARY_PATH=/home/oracle/t1d/cprogs/OracleCProgs:/usr/local/app/oracle/product/9.2.0/lib:/lib:/usr/lib:
/usr/local/lib
#export LD_LIBRARY_PATH
#echo "$LD_LIBRARY_PATH=" $LD_LIBRARY_PATH

echo "Compiling==>$1"
#rm -f $1
proc $1.pc
#gcc -O3 -frerun-loop-opt -o $1 $1.c
make -f demo_proc.mk build OBJS=$1.o EXE=$1
exit
```

Support sheepsqueezers.com

If you found this information helpful, please consider supporting sheepsqueezers.com. There are several ways to support our site:

- ☐ Buy me a cup of coffee by clicking on the following link and donate to my PayPal account: Buy Me A Cup Of Coffee?.
- ☐ Visit my Amazon.com Wish list at the following link and purchase an item: http://amzn.com/w/3OBK1K4EIWIR6

Please let me know if this document was useful by e-mailing me at comments@sheepsqueezers.com.