IMPLEMENT AN INTERFACE FOR CONTROL SYSTEM TO INTERACT WITH ORACLE DATABASE AT SSC-LINAC

Shi An, Wei Zhang, Kewei GU, Junqi Wu, Xiaojun Liu, IMP, Lanzhou, China

Abstract

SSC-LINAC control system is based on EPICS architecture. The control system includes ion sources, vacuum, digital power supplies, etc. In these subsystems, some of those need to interact with the Oracle database, such as power supplies' control subsystems, which need to get some parameters while power supplies are running and also need to store some data with Oracle. So we design and implement an interface for EPICS IOC to interact with the Oracle database. The interface is a soft IOC which is also based on EPICS architecture, so other IOCs and OPIs can use the soft IOC interactive with Oracle via Channel Access protocol.

INTRODUCTION

SSC is a separated-sector cyclotron. To improve the efficiency of HIRFL, a linear accelerator is considered as a new injector for SSC of HIRFL [1]. The SSC-LINAC control system is based on EPICS. Also we use Oracle to store some history data and some parameters for control system. For these reasons we design a soft IOC (see Fig. 1) be an interface between EPICS and Oracle.

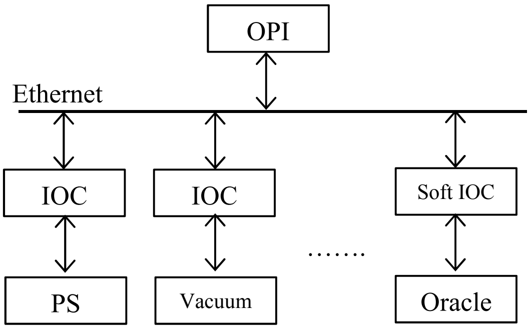


Figure 1: Structure of SSC-LINAC control system.

All of the other IOCs can interact with Oracle via the interface. For the OPI (or other IOC) they don't need to know the detail of how to connect to Oracle database, just access the PV to interact with it.

DESCRIPTIONS

This part will describe design and implementation of the soft IOC. Descriptions include four sections and each section is introduced in detail as following. The structure of the soft IOC is shown in Figure 2.

Structure of the Soft IOC

The bottom of the structure is Oracle database; above Oracle is OCI, which provides necessary APIs for the database. In the middle layer is device support of EPICS architecture. In this case Oracle is also a soft device for the EPICS. The device support includes a very useful and convenient C++ library, OTL 4 [2]. All of function of write and read the database is implementation via the OTL. The top of the structure is record support. SSC-LINAC control system creates some record support for each different subsystem.

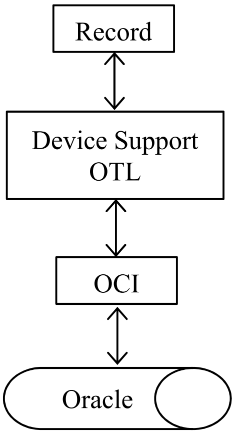


Figure 2: Structure of the soft IOC.

Construction, Development, and Runtime Environment of the Soft IOC

To development the soft IOC, a computer who runs the soft IOC need to construct development (or runtime) environment.

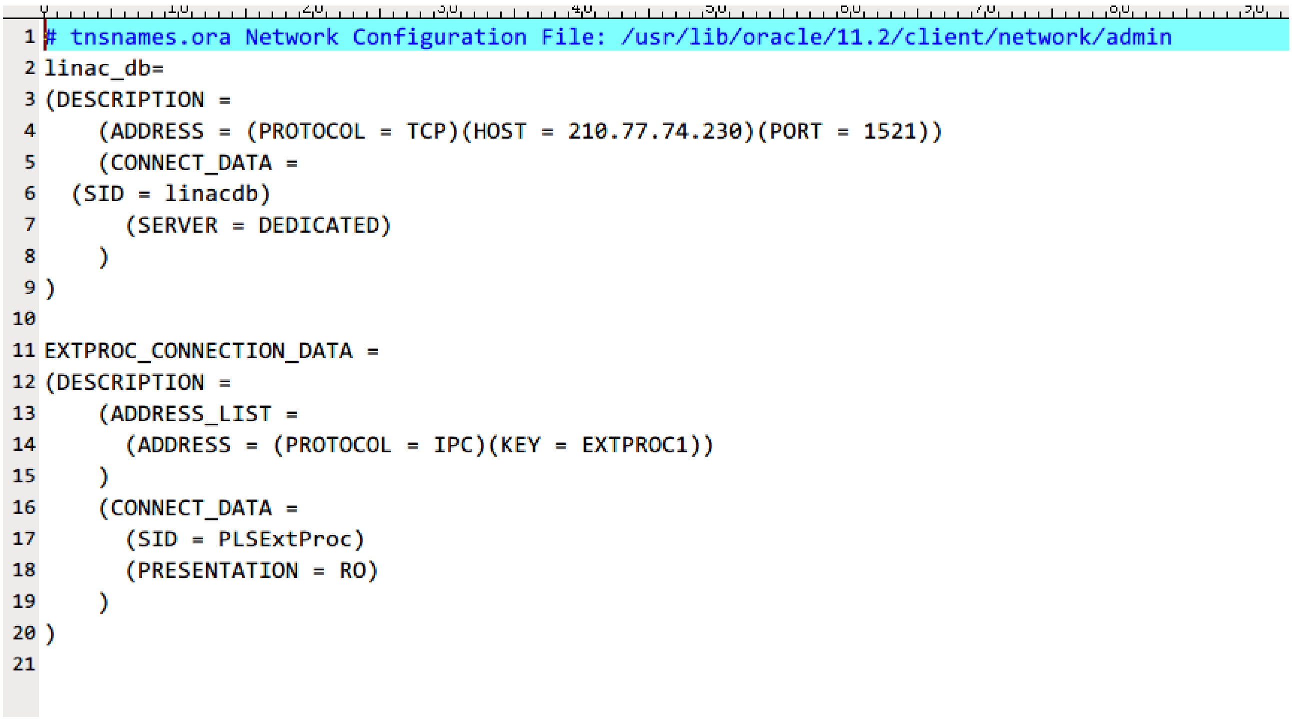
First need to download the OCI from Oracle support web site. There are two packages are necessary, they are Client_Bascic (library) and Client_SDK (C/C++ header file). After downloaded and installed packages also need to set the environment variable LD_LIARARY_PATH as the path where the location of OCI library.

Second step is to construct runtime environment of the soft IOC. Environment variable ORACLE_HOME will be set as the Oracle base path and TNS_ADMIN must be set as the path where locations of tnsnames.ora file. Figure 3 shows a sample of tnsnames.ora file. You must correct configure the file to connect Oracle database.

OTL

The soft IOC use OTL to access OCI. First is get the OTL from http://otl.sourceforge.net/otl3_down.htm. The OTL library just is a head file. Put the head file into EPICS IOC source directory after downloaded it. After that you can use the function of the OTL. Figure 4 shows add OTL support into device support source code.

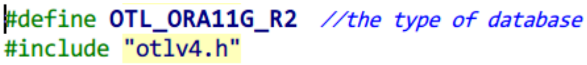
ISBN 978-3-95450-139-7



```
# tnsnames.ora Network Configuration File: /usr/lib/oracle/11.2/client/network/admin
linac_db=
(DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 210.77.74.230)(PORT = 1521))
    (CONNECT_DATA =
  (SID = linacdb)
      (SERVER = DEDICATED)
    )
)

EXTPROC_CONNECTION_DATA =
(DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
    )
    (CONNECT_DATA =
      (SID = PLSExtProc)
      (PRESENTATION = RO)
    )
)
```

Figure 3: A sample of tnsnames.ora.



```
#define OTL_ORA11G_R2  //the type of database
#include "otlv4.h"
```

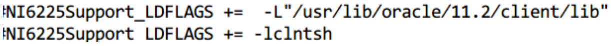
Figure 4: Add OTL library support into device support source code.

The type of database must correct define depend what type of database you are using.

Device Support

A device support routine has knowledge of the record definition. It also knows how to talk to the hardware directly or how to call a device driver which interfaces to the hardware. Thus device support routines are the interface between hardware specific fields in a database record and device drivers or the hardware itself [3]. In this case Oracle is a 'hardware' which device support will access. Device support will convert request of OPI (or other IOC) into correct SQL statement. If convert is success, device support will commit the SQL statement via the OTL and OTL will return the result of the operation. After that device support will store the result into specific EPICS fields in a database record.

To compile the device support that use the OCI must add some statement into Makefile of the soft IOC source. These statements tell the location of Oracle library to linker. Figure 5 shows these statements.



```
NI6225Support_LDFLAGS +=  -L"/usr/lib/oracle/11.2/client/lib"
NI6225Support LDFLAGS += -lclntsh
```

Figure 5: A fragment of a Makefile for the soft IOC.

CONCLUSION

The design uses a soft IOC to be an interface between EPICS and Oracle.

The first advantage of this structure is not every IOC must install the OCI (Oracle Call Interface) who want to access Oracle. We just need install the OCI on a computer who run the soft IOC.

Second advantage is for the OPI (also other IOC) the Oracle is transparent just like a common IOC, so the function of write and read the Oracle database is same as other IOC.

Third, if made some changes of the Oracle database don't need to modify every relational IOC but just need to modify the relational soft IOC.

REFERENCES

[1] He Yuan, Wang Zhijun et al., "6-14 Design of Linear Injector for SSC", IMP & HIRFL Annual Report (2009), pp. 256-257.

[2] OTL 4.0, Introduction. http://otl.sourceforge.net/otl3_intro.htm

[3] Martin R. Kraimer, Janet B. Anderson, etc. "EPICS Application Developer's Guide". http://www.aps.anl.gov/epics/base/R3-14/12-docs/AppDevGuide/