APPLICATION OF A SIMPLE TEXT FORMAT AS A DEVICE CONFIGURATION FILE

T. T. Nakamura, KEK, Tsukuba, Japan

Abstract

In the KEKB magnet control system, relational database management system (RDBMS) has been used for the management of the device configuration. Most of the device parameters, such as address of the interfaces, constant parameters of the magnet power supplies, excitation functions of the magnets, limit values for the operation, etc. are stored in the RDBMS. These parameters are used mainly for the macro expansion to generate the runtime EPICS database files. They are also used to configure the high level application programs. Although this system is flexible, it is heavy to maintain with little man power in the KEKB control group. To reduce the maintenance cost, restructuring of this system is planned. The plan consists of two major changes. One is the reducing data and making data structure simpler. Another is the replacing tools with simpler ones. As the part of the restructuring, the replacement of the RDBMS with simple configuration files is planned. The configuration file has simple text format, which is designed easy to read and easy to modify only using text editor. The design of the format and the development of the tools have been almost completed. Its application to the KEKB magnet control system is also in progress.

INTRODUCTION

KEKB is an asymmetric electron-positron collider at 8 × 3.5 GeV/c, which is dedicated to B-meson physics. Its operation was started in December 1998. The KEKB accelerator control system has been constructed based on EPICS (Experimental Physics and Industrial Control System) tool kit [1]. EPICS provides core mechanism for the distributed control system. EPICS runtime database is running on a local control computer called IOC (Input/Output Controller). More than 100 VME/VxWorks computers are installed as IOC in the KEKB accelerator control system. The workstations of 4 kinds of platform (PA-RISC/HP-UX, Alpha/OSF1, PC-AT/Linux and Macintosh/OSX) are also installed. Most of the high level application programs run in these workstations. The runtime database is downloaded from the central server workstation when the IOC starts up. In the KEKB storage rings and the injection beam transport lines, about 2500 magnet power supplies are installed and controlled by 11 IOCs through ARCNET [2].

In the KEKB accelerator control system, the data management system based on the RDBMS has been developed [3]. Oracle is chosen as the RDBMS. Especially the magnet control system uses this system intensively. Figure 1 shows the overview of the information flow of this system.

The system was designed to keep parameters of various devices and to provide them to the all kinds of control software. Especially automatic generation of the EPICS runtime database is the major task of this system. The EPICS runtime database files (db files) are generated from the template database files and the EPICS parameter files (dbprm files). The dbprm files are generated from the information in the RDBMS. The generation process of the dbprm is described in PL/SQL, which is the extended SQL by Oracle Corporation. The major parts of them are defined inside Oracle as the stored procedures.

Some control parameters in the dbprm files are derived from the primary data in Oracle. For example, coefficients of the magnetic field excitation functions are calculated from the field measurement data and the information of the magnets and the power supplies in Oracle. Such calculation is also done in the Oracle PL/SQL scripts as the part of the generation process.

In the KEKB most of the high level application programs are written in SAD or Python [4]. Some of the Python applications access the RDBMS to get device information. For this purpose, Python library module "rdbtool" has been developed. It defines "rdb" class, which is designed to handle tabular data on memory and has simple interface to the RDBMS to retrieve data.

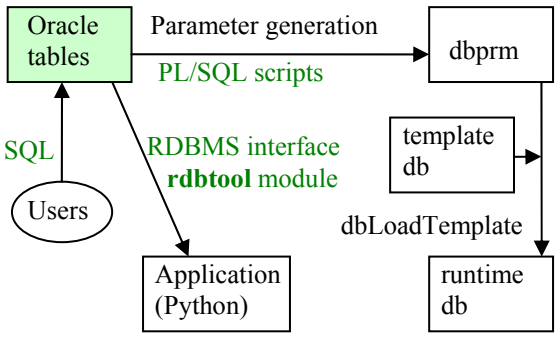

Figure 1: Information flow (current system).

The KEKB accelerators have been operated more than 10 years. Although the data management system worked well at the beginning, the maintenance difficulty arises for these years. Unexpected modifications have been accumulated both in the data and the PL/SQL scripts. Some of them have been done in quick and dirty manner. When new type of magnets or power supplies are installed, irregular modification is often necessary, which was unexpected when it was designed. The wiring database is designed being flexible but too complex to understand the whole structure. Sometimes many lines of SQL script are required to accomplish the modification. Version up of the Oracle is also complicated task. It needs some modifications and careful test to keep compatibility. On the other hand, the man power of the control group is decreasing during these 10 years.

To overcome such maintenance complexity, the restructuring of the system is planned. Main part of the plan is the reducing data and making data structure simpler. Another part is the replacement of the Oracle RDBMS with a simple data management system, which consists of the text files with new format. In the following sections the design of the new format system is discussed.

CONCEPTUAL DESIGN OF THE TXDB

The new format is called TXDB, which stands for the **t**ext **d**ata**b**ase. The text format has following advantages.

- User can read the contents directly. Neither special tools nor servers are necessary.
- User can roughly but quickly explore the contents using general text processing tools such as grep or sed.

The TXDB is designed with the following guidelines.

- Simple syntax
- Compact notation
- Human-friendly and good readability
- User can directly edit the files with a text editor.
- Not aim at general purpose but rather special purpose for the device configuration

Figure 2 shows the information flow of the device configuration using the TXDB system. Main differences compared to the current system are followings.

- All data are resident in the TXDB files, not in the RDBMS.
- The generation process of the dbprm files are described in Python scripts, not in PL/SQL scripts.
- Instead of the SQL, text editors can be used to modify the primary data.

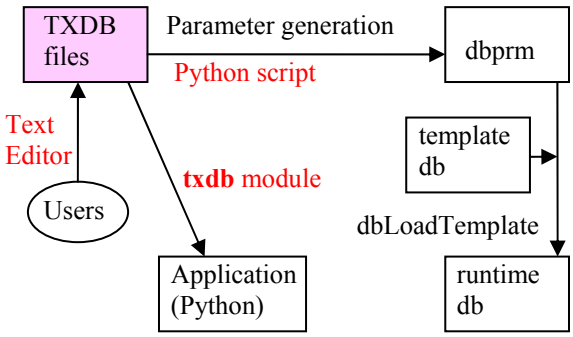

Figure 2: Information flow (TXDB system).

FORMAT OF THE TXDB FILE

Basic Format

The basic structure of the RDBMS is the table, which is the 2-dimensional array of rows × columns. The TXDB also has the similar structure. Figure 3a shows the basic format of the TXDB file. The TXDB file consists of a set of records. The record is described in one line of the file. Each record consists of a set of fields. Each field is identified by name (field name) and described with the form "***name*=*value***" in the line. The records and the fields are corresponding to the rows and columns of the RDBMS table respectively.

a) Basic format

```
name=QF loc=D08 port=11 SN=L15-001 table=MagnetPS
name=QD loc=D08 port=12 SN=L03-002 table=MagnetPS
name=QC1 loc=D08 port=13 SN=S07-012 table=MagnetPS bp=Yes
name=QC2 loc=D08 port=14 SN=S08-015 table=MagnetPS
name=QR1 loc=D11 port=26 SN=S21-103 table=MagnetPS
name=QR2 loc=D11 port=27 SN=S21-118 table=MagnetPS limit=380
name=QR3 loc=D11 port=28 SN=S21-073 table=MagnetPS
```

b) Block notation

```
%block  loc=D08  table=MagnetPS
 name=QF port=11 SN=L15-001
 name=QD port=12 SN=L03-002
 name=QC1 port=13 SN=S07-012 bp=Yes
 name=QC2 port=14 SN=S08-015
%block  loc=D11  table=MagnetPS
 name=QR1 port=26 SN=S21-103
 name=QR2 port=27 SN=S21-118 limit=380
 name=QR3 port=28 SN=S21-073
```

block

block

c) Column list notation

```
%columns  name  port  SN
%block  loc=D08  table=MagnetPS
 QF    11  L15-001
 QD    12  L03-002
 QC1  13  S07-012  bp=Yes
 QC2  14  S08-015
%block  loc=D11  table=MagnetPS
 QR1  26  S21-103
 QR2  27  S21-118  limit=380
 QR3  28  S21-073
```

field names

field values

Figure 3: Examples of the TXDB file.

There is the special field named "table". Usually each record has a table field. The set of the records that have same table field can be treated as the table, which is corresponding to the table of the RDBMS. One TXDB file can contain multiple tables and one table can span multiple files.

In contrast to the RDBMS, the TXDB has no explicit declaration of the fields. Each record can have any field without declaration. Even in the same table, each record can have different fields.

The value of the field has no type. It is always treated as a character string. It also can be NULL, which is the special value meaning that the field is undefined. Usually NULL is noted by "-" (hyphen) in the TXDB files.

Block Notation

The basic structure of the TXDB is very simple, but not so human-friendly, because every field values are

explicitly written in the file. In some case the same values are often repeated many times. To enhance the readability the TXDB format allows some compact notations.

Figure 3b shows the "block notation". The special line that begins with "%" is the statement to control the TXDB format itself. The %block statement supplies the implicit values of fields. It declares the common field values for the multiple records in the block scope, which spans from the %block statement to the line just before the next %block statement or the end of file.

Column List Notation

Figure 3c shows the "column list notation", using which the user does not have to write the field names in every line. The "%columns" statement specifies the field names to be omitted. In this example each record has the list of field values without field name. Each value is associated to the field name in the %columns statement by its order. The extra fields with the form "*name*=*value*" can also appear after the value list.

Other Features

The %shortcut statement defines the abbreviations of the field values. %default statement defines the global default values within the file. %set statements set some options that control the lexical syntax of the TXDB itself.

Each record can have a special field named "COMMENTS" with the special notation. It begins after the special character "#" and ends at the end of line.

DATA PROCESSING TOOL FOR THE TXDB

To handle the TXDB format file, the library module named "txdb" in Python has been developed. The library is designed in the object oriented manner. It defines the "txdb" class, whose instance is the container of the TXDB records on memory. Most of the functions are implemented as the methods of this class.

Users can write their own data processing programs using this library in Python. The txdb class is designed being similar to the rdb class in the rdbtool module. Most of the applications that read the Oracle tables are expected to be able to read the TXDB with little modification.

APPLICATION TO THE MAGNET CONTROL SYSTEM

To apply the TXDB system to the magnet control system, following tasks are in progress.

(1) Development of the data conversion programs from Oracle to TXDB is in progress. Currently almost all of the data that are necessary for the EPICS macro expansion are ready to convert. Remaining work is the conversion of the address table of the PLC for the magnet interlocks.

(2) Almost all active scripts in Oracle for the EPICS macro expansion have been translated from PL/SQL to Python.

(3) The high level application programs that access Oracle need some modifications to introduce txdb module. The modification due to the change of the table structure of the magnet control system is also necessary for these applications. The modifications are in progress.

CONCLUSION

The modern control system consists of large number of subsystems and each of them has many software components. The management of the device configuration for all components is important but heavy task. In the case of little man power, the simple tools such as the TXDB are rather suitable. The human readable text format is easy to handle and gives a good prospect to the users.

REFERENCES

[1] N. Yamamoto et al., "KEKB Control System: The Present and the Future", PAC-99, New York, 29 Mar.-2 Apr. 1999, p. 343.

[2] T. T. Nakamura et al., "Magnet Power Supply Control System in KEKB Accelerators", ICALEPCS'99, Trieste, 4-8 Oct. 1999, p. 406.

[3] M. Kaji et al., "Relational Database System in the KEKB Accelerator Control System", APAC98, Tsukuba, 23-27 Mar. 1998.

[4] N. Yamamoto et al., "Use of Object Oriented Interpretive Languages in an Accelerator Control System", ICALEPCS'99, Trieste, 4-8 Oct. 1999, p. 600.