

## THE USE OF ORACLE FOR DEVELOPMENT AND MAINTENANCE OF EPICS DATABASES

A. Lukyantsev, A. Khvorostianov, IHEP, Protvino, Russia  
M. Clausen, Chr. Gerke, V. Klinger, DESY, Hamburg, Germany

### Abstract

The use of EPICS [1] in a part of the DESY accelerator controls has initialized the development of special tools to create and maintain IOC (Input Output Controller) databases containing tens of thousands of EPICS records. This paper presents the process of designing IOC db's and the relevant tools for supporting IOC db's by ORACLE[2] facilities developed using ORACLE FORMS[3]. The process of designing a concrete IOC database includes the definition of a set of EPICS records, the assignment of names to records according to the certain rules and the setting of values to fields of records distinct from initial values determined by types of EPICS records. As a number of IOC db's can have groups of records, which differ only by their names and by the values of individual fields of records, individual prototypes for such groups are defined. The requirements on the tools to create and maintain IOC db's are:

- The satisfaction of the various users of accelerator controls;
- The storage all IOC db in a unified base, which allows to coordinate the information in the developed IOC db (for example, guarantee of uniqueness of record names in all IOC db);
- The formalization of several procedures in order to simplify the process of designing IOC db's, large amounts of information brought into IOC db's;
- The parallel maintenance of IOC db for different EPICS releases (i.e. translation of IOC db from one EPICS release to another);
- The development of data structures which would support both old and new definitions of EPICS databases.

We created a new form in ORACLE called EpicsOra in which the basic principles of how we design an IOC db are realized.

### 1 STAGES THE DESIGNING PROCESS

In the offered process of designing IOC db's it is possible to run through three basic stages. Each stage

corresponds to a tool in the program module and a data structure:

- EPICS dbd (database definition): Stage of database definitions for EPICS releases in ORACLE. The appropriate section of the data structure includes the descriptions of EPICS record types, menus etc. At this stage the textual files (\*.dbd files) contain the database definitions for the concrete EPICS release, the descriptions for each record type, menus, table of breakpoints etc. are imported into ORACLE. The data input procedures support the database definition files and the ASCII files for the Epics R3.12XXX releases, and the "dbd" and database ("db") files - for the Epics R3.13XXX release.
- PROTOTYPES: Design stage of the groups and records prototypes. The appropriate section of the data structure includes the descriptions of the groups prototypes and of EPICS records prototypes which are under construction on the basis of EPICS dbd objects. Hereinafter objects we shall understand types of records, prototypes of groups, projects, group, IOC databases and other elements of the data structure.
- PROJECTS: Design stage of IOC db's of the projects. The appropriate section of the data structure includes a set of the projects, each of which contains a number of IOC db, consisting of concrete EPICS records. The descriptions of the objects at this level are formed on the basis of the object descriptions of the previous levels.

The levels of object's designing and their priorities are illustrated in Fig. 1.

#### 1.1 Creation of Groups and Records Prototypes

At the stage PROTOTYPES the prototypes for a number of objects are created. Groups of EPICS records with identical record types and the same values for the properties can be shared between several IOC databases if they are collected in prototypes. They must be created beforehand in order to use these prototypes for new IOC databases. On the basis of these concrete groups IOC db can be projected.

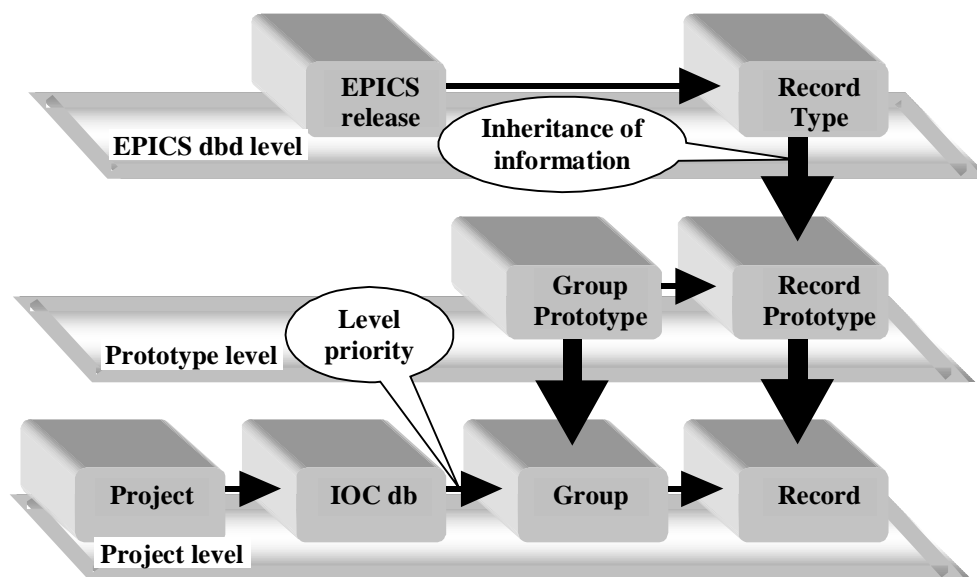


Figure 1: The scheme of object's design levels and priorities.

Some of the fields of EPICS records have initial values which are defined in the description of the given record type. These initial values are simply inherited. Other fields of the records in a group have values which are distinct from the initial values, but identical to all IOC db's, in which the given group is used. The values of such fields are set at creation time of the prototype of such a group and then are inherited at the design time of the concrete group of EPICS records for an IOC db.

Other fields of records have values that depend on various conditions or on the relations between projected objects. For example, whether the record belongs to the project, the IOC db or to the group. In this case the definition of values of such fields is formalized by means of the use of variables. When instantiating a concrete object (project, IOC db, group etc.) the values of required variables are determined and assigned to the object. These values of variables are further substituted in fields of records containing these variables and having relations to this object. The values of variables in concrete records are inherited on a priority and are substituted in fields of this record. For example, let's create a project containing some IOC db's each of which includes its own set of groups and records. The value of a field in a record contains a variable, it is defined at first as the value of the variable which is an attribute of the group. If the variable is not determined on group level, the IOC db is searched, and then the project. If the value of a variable which itself is an attribute of an object changes, it will result in a change of the values of all fields of concrete records, into which the given variable enters, and to which (EPICS records) relate for the given object.

Those fields of records in a group which shall have different values for different IOC db are marked at the

design time of the group prototype as fields which require obligatory setting of individual values. During the design of the concrete group of IOC db records they are viewed as a list for all group record/fields. This is a user-friendly interface where only those record/field entries that need individual data entries are shown.

Let's say some words about how we form unique names for records and the fields of the records prototypes. The mnemonics of a record name contains the information about I/O controllers, about a subsystem, which it serves, about a record type and functionality, about the location of the device to which the record corresponds etc. Uniqueness of records prototypes names is guaranteed within the framework of the group prototype. For the names of the record prototypes masks have been implemented that include variables. The names of the variables consist of two symbols: % and a letter or figure. Two masks are now determined: %F%D%L \_ of %T - for records such as "device" and %I%G%P \_ %T - for records of all other types. The semantics of variable is, that it bears the information on any object, to which the given record relates. So in above mentioned masks variables concern accordingly: %I - IOC, %G - group, %P - group prototype, %T - record type, %F - facility, %D - device, %L - location. At creation of the record prototype the appropriate mask is taken, the value of variable %T is substituted only and thus the name of the record prototype is formed which in turn is a mask for the formation of a name of a concrete record constructed on the given prototype. So, at creation of the group prototype the following operations are carried out:

- the set of the records prototypes which are included in the group prototype, is specified;

- the names of the records prototypes are specified which are masks for names of concrete records;
- the values of record fields are set which are inherited in fields of concrete EPICS records;
- those fields of records are marked, whose values must be defined at creation of the concrete EPICS record.

The realization of the two-level structure for the groups prototypes allows the group prototype to be built on the base of already created ones. This is useful when the prototype created for one project becomes a General Prototype. This way it is possible to reuse the General Prototype with small changes for another Project Prototype. In this Project Prototype the information concerning the concrete project is already set.

### *1.2 Creation of the IOC db projects*

At the PROJECTS stage, the concrete projects are built based on objects defined in the previous stages. First, the project is created, then within the framework of the project, the IOC db's are created, and further in the framework of IOC db's, the groups of records (or simply single records, not concerning to groups) are created.

Variables are introduced in the IOC db in order to structure and formalize the design. The variables are used in the names of the EPICS prototype records and in the fields of the records. The semantics of the variable is related, as a rule, to the created object. To define a new variable it is necessary to assign a variable name, to choose the object it will be related to and to describe the semantics of the variable. The following objects now are determined which the user can choose from: a record (1), a prototype of groups (2), a prototype of a record (3), a project (4), an IOC db (5), groups (6) or records (7). The priority of inheritance in ascending order is specified in brackets. For example, let variable %X is determined and the objects 4,5,6 are chosen. Let's design the group prototype P1, in which the value of one of the fields of a record will be defined by means of the variable %X. Let's create the project A, consisting from several IOC db - I1, I2, I3, each of which contains groups G1, G2, G3, built up on the basis of the group prototype, P1. While the variable is not assigned in one of the objects A, I1, I2, I3, G1, G2, G3, the value of the fields determined by the variable %X in records of the project A will not be given and will be empty. If %X=value1 for the project A, then for all records of the project A the value of the fields determined by variable %X, becomes equal value1. If %X=value2 for I1, only for the records in IOC db I1 the value of the appropriate fields becomes equal value2. If %X=value3 for G1 from I1, only for the records of group G1 in IOC I1 the value of the appropriate fields will change to value3.

## **2 EPICSORA FORM**

According to the IOC db design process a specific Oracle Form has been developed which corresponds to the design stage: EPICS dbd – maintenance of the Epics releases, PROTOTYPES - designing of the groups prototypes, PROJECTS - designing IOC db of the projects. At the startup of the form the user has to choose the EPICS release, which he will work with, then the main window of the form is displayed from which it is possible to start one of the above listed design stages.

In the functional menu of the form the calls of common procedures are assembled which can be called at the various moments of work with the form. Among them are the export and import procedures from/to the ORACLE database in ASCII files.

Each section of the form starts with the catalogue of objects which can be chosen from. It is possible to overlook its attributes. If no useful objects can be found, new ones can be created. Existing objects can be edited or removed from the current object, or one can proceed to a sub catalogue of the objects components - if the object is a compound one.

## **3 OUTLOOK**

The current implementation of EpicsOra is a basic foundation to organize the work with several different EPICS databases for several EPICS releases and a big number of IOC's. Using Oracle as the basis to implement and store EPICS databases has the advantage to be able to perform relations to other tools which have been implemented in Oracle – like the logbook. Besides this also the documentation of the databases is much easier and the conversion of databases on release changes is simplified.

The next stage of this project will be to combine the Oracle implementation with a graphical representation of the databases. This can be used for documentation purposes, but also to specify for instance the prototypes.

## **REFERENCES**

1. L.R.Dalesio, J.O.Hill, M.Kraimer etc. "The Experimental physics and industrial control system architecture: past, present, and future", Proc. Of ICALEPCS-93, Oct.18-23,1993, Berlin, Germany.
2. E.Armstrong, C.Closkey, B.Linden, M.Pratt "Oracle7 (TM) Server Concepts Release 7.3 ", Belmont, California, USA, Nov. 1996.
3. D.Brown, K.Chu, J.Magee "Oracle Forms Developer's Guide, Release 4.5", Belmont, California, USA.