

Database of the Parameter List for the EURISOL Beta-Beam Task

N. Emelianenko, A. Fabich, E. Wildner

The beta-beam task involves colleagues from six European institutes studying different subjects of the beta-beam complex. A parameter list shall provide a single source describing the baseline scenario, such that the coherence of the studies is guaranteed. The aspects of maintenance and user accessibility made us choose the database technology for the storage of the parameter list. Auxiliary tools for the access via the web and from *Wolfram Mathematica* are provided for the convenience of the users. The relational database management system OracleTM was chosen for the implementation since it is a widely used technology, centrally supported at CERN and offers powerful tools for the development and maintenance.

The database and the applications built around it provide an efficient and convenient tool for the beta beam task.

Summary of benefits:

Database technology and the specific design:

- Unique data source
- Easily maintainable and extendable
- Open and ready to use also together with other databases in Oracle
- Data access from several commonly used programs (*Mathematica*, Excel, ...)
- Data export in various formats (ASCII, PDF, Word...)

Oracle & central CERN Oracle support:

- Safe storage
- Oracle ensured security, fine grained data access and roles
- Daily backups
- Assistance (central Oracle support)

This document gives the summary of the experience gained in the development of the beta-beam parameters database and shall describe the concept, set up (administration) and the practical use of the beta beam parameter list with focus on the technology choices and the layout. It is not a user's guide, but shall show the possibilities of the used technologies and give a hint for using it for other applications.

Database Design

The requirements for the storage of the parameters of any system in evolution (or under development) are:

- Extendable set of components (objects)
- Extendable set of parameters for each component
- Automatic logging of changes
- Versioning (fixed releases or system snapshots)

Additional requirements for the beta-beam application are:

- Provide a search engine for the object structure
- Easy report generation
- Access through the Web
- Integration with *Mathematica*

All these requirements are fulfilled with the use of original *generic* database design, Oracle features and Java technologies. It was decided to

- Impose a rigorous systematization of the knowledge about the system,
- Separate the parameter definition (metadata) and parameter value (data),
- Store the system metadata in the database.

The *generic* database is able to store the description of any system having a tree or network structure. To simplify the design, it was decided to limit the network structure in a way such that each node may have only one or two parents. This is sufficient for the beta beam project (see Figure 1).

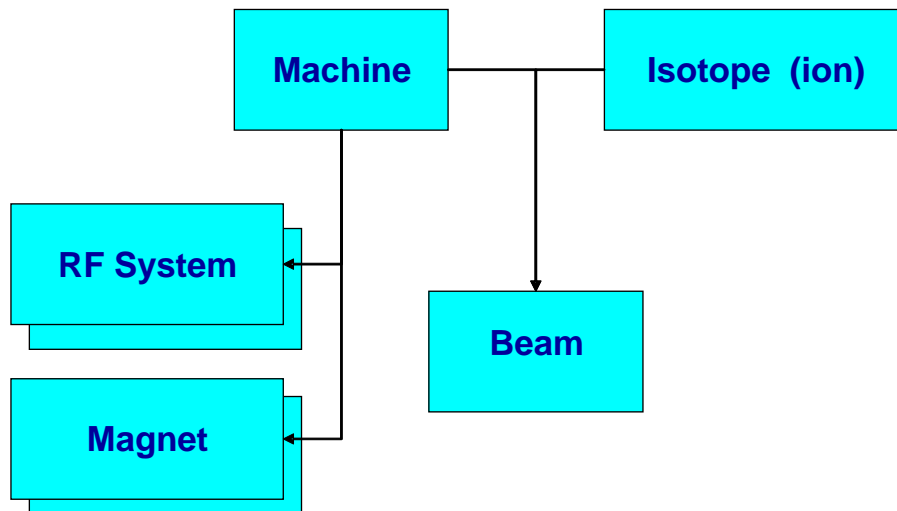


Figure 1: Structure of the objects used in the beta beam facility description. Arrows point from parent to child objects, the *beam* object has two parents: *machine* and *ion*. Each *machine* can have many types of *RF systems* and *magnets* (one-to-many relationship), but each pair of *machine* and *ion* defines a unique *beam*.

The Database Set Up and Administration

Responsibility

Write-access to the database should be given to a limited number of people by means of special Oracle security management. For the database of the beta-beam parameters two persons have the write access, one being a database engineer and *system analyst* and another one a physicist having deep knowledge of the project, working in close collaboration. Suggestions given by all task members are reviewed by a committee and implemented accordingly. This avoids incoherence of the parameter list.

The technology being used allows giving the full or partial access rights to larger number of people whatever is their location. In this way the data base can be updated and maintained, not only from inside CERN.

Partial access includes the following rights which can be granted to particular users or user groups (only the first three are implemented for the beta-beam task):

- Read all data or a part of the data,
- Register / modify / delete objects, parameter categories or parameters,
- Insert / modify / delete the parameter values,
- Modify the value of a parameter, all parameters of an object etc.

The Database Set Up

The database set-up consists of the definition of the types of the task components (objects) and their templates (lists of parameters) and of the manual input of these formalized definitions in the database. The overall success of the database project implementation and the convenience of its use depend on the correct and efficient choice of the objects and templates.

The Origin and Type of the Parameter Values

The *values* of a given parameter can have various sources:

- Given value. The value has to be inserted manually, and for each value the exact source (published document, presentation, web site etc.) must be specified. All sources are registered in the detailed *Data Sources list*.
- Automated input from e.g. *Mathematica* (for the parameters derived from other requiring complex calculations).
- Calculated on the fly upon parameter access (the formulae are defined and stored in the database along with the parameter definition).

The **type of the source** for a parameter values and **unit** for the numeric values are defined (fixed) in the parameter *definition*. The *value* itself can be either numeric or alpha-numeric.

Practical Use

Accessing the parameter list

For the whole community there are 3 ways to access the data stored in the parameter database:

- Direct access in Oracle (through any tool allowing the connection to the CERN **ACCDB** node) from the account **BBUSER**,
- Web interface, including the report(s) in PDF format,
- Read access from Wolfram *Mathematica* (or any other program using ODBC or able to talk to a SOAP application).

Referencing the Parameter Values

With the generic database design the referencing to the parameter values is straightforward. To find a parameter value one should specify the name of the object to which the parameter belongs and the name of the parameter. Special functions are defined for this purpose in the database and in *Mathematica*.

Web Interface

The web interface provides the navigation through the task objects and allows reading the contents of the parameter database in a structured way. The link to the beta-beam web-site can be found at <http://cern.ch/beta-beam/task/diverse/docs.htm>. Each page of the site is generated on the fly at the time of the page call. The pages are built from the data (and metadata) stored in the database. On the welcome page there is a link to a PDF-report, which is also built on the fly with the use of the data stored in the database.

The Web interface is a set of JSP and servlets and is hosted within central CERN public J2EE services.

Direct Access from Mathematica with DatabaseLink

Here we describe the two possibilities to access the database from *Mathematica*.

Access via the DatabaseLink

If the access to the CERN ACCDB node with *DatabaseLink* is possible, the user can choose the option of the direct Oracle access. We provide a package which defines in *Mathematica* a set of functions to work with the database (these functions have shorter aliases, long aliases are given here to explain the function purpose, **id** stands for the object name, **par** for the parameter name):

```
getParameterValue [id_String, par_String]
putParameterValue [id_String, par_String, value_String]
delParameterValue [id_String, par_String]
```

These functions are to be used by the physicists, responsible for the parameter values input and calculations. For the whole community the restricted version (only the `getParameterValue` function) is proposed.

Access via the Web Service

When the user does not need the write access to the database, this option is much more preferable. It does not require a database connection and is easier and faster. The access is provided via the Web service, which is deployed as JWS in the Apache Axis application (the same framework is used by *Mathematica*), within CERN central J2EE service.

With the use of the Web service the database connection is not needed, so no passwords have to be specified (the access is open to any user worldwide). One only requires:

- Load the standard *Mathematica* package `WebServices`,
- Install the web service, which loads the definition of the function which allows to get the parameter values as strings.
- Optionally use the BB Parameter Browser — the palette including the button to insert the name of the function and the button “menu” to navigate through the objects and choose the parameter names.

A user guide for this option together with the small *Mathematica* package to perform the first two steps, and the BB Parameter Browser palette can be found on the beta-beam documentation web-page.

BB Parameter Browser Palette

This palette-style notebook is generated by a specially developed package in *Mathematica* from the beta-beam database metadata. With the use of this palette one does not need to remember the parameters or objects names – just navigate through the structure similar to those on the web site (but showing all object types on the first level) and click on buttons with short parameter descriptions. The Browser also provides the information on the units in which numerical values are expressed.

Text Format and Encoding Issues

We have experienced various difficulties with special characters, precision of numbers, string and number operation. They all have been solved, but they are not described here in detail.

Perspective Implementation for Other EURISOL Tasks

The developed database structures and software, in principle, allow the straightforward implementation for any similar project (with relatively small number of objects and parameters and only scalar parameter values).

The main implementation efforts will be put on the systematization of the data and on the choice of object types. This step requires the tight collaboration of the physicist and the system analyst.

For each new project the best choice is to create a separate database (new Oracle account with new tables) if

- ✚ it deals with different objects
- ✚ most of its parameters are independent from those of other projects and
- ✚ different people will manage the data

In each case the decision on whether to join or to separate the parameter databases is a kind of art, when one should foresee all future needs, cross-references, redundancy etc.

Appendix A. Database Structure

The generic concept of the database was implemented with the following database tables:

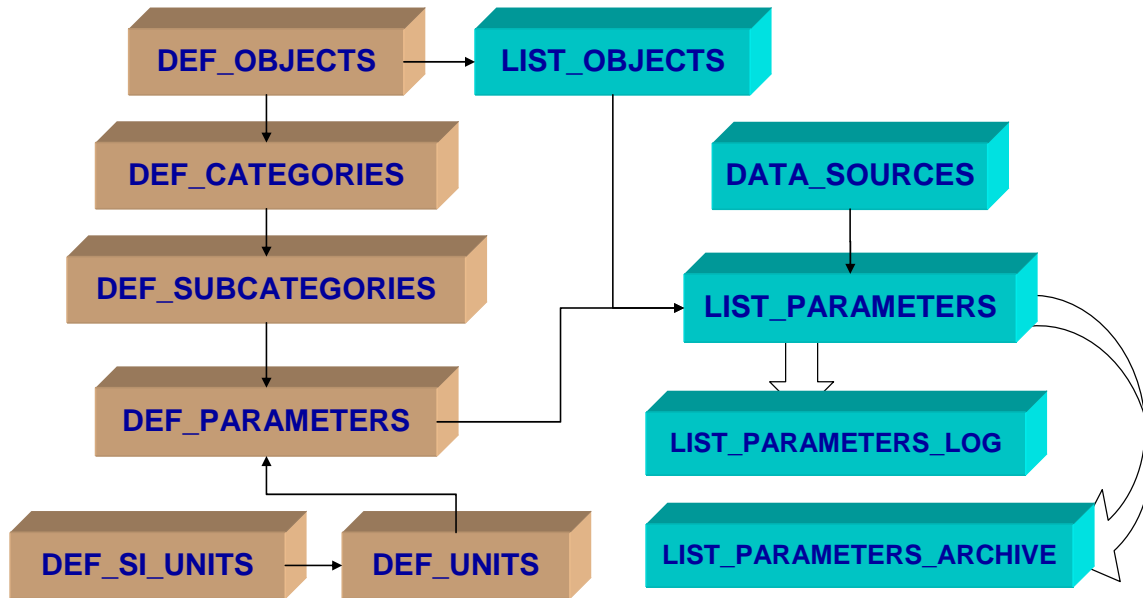


Figure 2. Database Structure.

Thin arrows represent the referential integrity constraints, link from parent to child tables, which means, for example, that in the table `LIST_OBJECTS` one can register objects only of types defined in the table `DEF_OBJECTS`; big arrows represent the logging of the changes and deletions.

The system metadata is stored in the tables which names start with `DEF_`. The lists of the system components (objects) and their parameters are stored in tables which names start with `LIST_`.

- **DEF_OBJECTS** — list of object types with the types of their “parent” object if any.
- **DEF_CATEGORIES** and **DEF_SUBCATEGORIES** — define the grouping of the parameters for each object type.
- **DEF_PARAMETERS** — list of the parameters that can be specified for the objects of each object type:
 - Parameter category
 - Subcategory
 - Parameter ID
 - Name, symbol
 - Unit, preferred unit for displaying (for the numeric parameters only)
 - Status (given / calculated / calculated in *Mathematica*)
 - Formula (for the calculated parameters only)
 - Preferred number format for displaying (for the numeric parameters only).

- **DEF_SI_UNITS** and **DEF_UNITS** hold the lists of SI units, used for the stored values of the parameters, and the units used to express the parameters values on display or to show them on the Web interface (“preferred units”). E.g. the value of the stored beam energy is recorded in Joule, but automatically converted to and displayed in kJ.

Example of object definitions (**table LIST OBJECTS**):

<i>Column name</i>	OBJECT_TYPE	ID	PARENT1_ID	PARENT2_ID
<i>Values in different records of the table</i>	machine	PS		
	machine	SPS		
	RF	RF1-SPS	SPS	
	RF	RF2-SPS	SPS	
	ion	6He		
	ion	18Ne		
	beam	PS-6He	PS	6He
	beam	SPS-6He	SPS	6He
	beam	PS-18He	PS	18He
	beam	SPS-18Ne	SPS	18Ne

Below is an example of a parameter definition (this is **one record from the table DEF PARAMETERS**) — the total beam energy lost per cycle. The parameter concerns the object type “beam” and belongs to the subcategory “Total” of the category “Intensity”. Its ordinal number within this subcategory is 2 (this number is used while building the reports). Preferred format is not defined; the default number format will be used (“scientific”). Preferred unit is Joule. The parameter status (CALCULATED) equals 2, i.e. it is calculated in *Mathematica*.

<i>Column name</i>	<i>Value</i>
OBJECT_TYPE	Beam
CATEGORY	Intensity
SUBCATEGORY	Total
PAR_ID	beam_int_tot_energy_lost
ORDER_NU	2
NAME	Energy lost/bb cycle
DESCRIPTION	Total energy lost/beta-beam cycle
SYMBOL	
PREF_FORMAT	
PREF_UNIT	J
FORMULA	
CALCULATED	2
COMMENTS	

The example of real values for a parameter, **table LIST PARAMETERS** (decimal digits truncated to fit the table to the page):

<i>Column name</i>	ID	PAR_ID	VALUE_NUMBER	VALUE_TEXT
<i>Records</i>	PS-6He	beam_int_tot_energy_lost	8024.4334928	8024.4334928
	SPS-6He	beam_int_tot_energy_lost	16755.314753	16755.314753
	PS-18Ne	beam_int_tot_energy_lost	6074.7697313	6074.7697313
	SPS-18Ne	beam_int_tot_energy_lost	6308.8658197	6308.8658197

The parameters can be alphanumeric (like machine status = “design”) or numeric. There are two columns in the table LIST_PARAMETERS used to store the values of the parameters — VALUE_NUMBER and VALUE_TEXT. For the alphanumeric values the column VALUE_NUMBER is empty. To secure the value of the numeric parameters, their values are also stored in text form (column VALUE_TEXT).

The table LIST_PARAMETERS contains also the columns SOURCE to indicate the origin of the parameter value (in case of beam_int_tot_energy_lost the source is “mathematica nb”) to trace back additional information, and INSERT_DATE where the date of insertion or last update is stored automatically.

Some of the parameters are defined as *calculated* (CALCULATED = 1). This means that they can be directly derived from the given parameters (like ion Q/A is derived from ion charge Q and ion atomic number A). The formulae for such parameters are written in PL/SQL (Oracle proprietary language), stored in the column FORMULA and are used to calculate the parameter values on the fly. Like that, there is no need to store the values for these parameters in the table LIST_PARAMETERS.

Working with the Parameters in the Database

With the generic design the referencing to the parameter values is straightforward. For example, to retrieve the status of the LINAC and PS, the following SQL statement can be used:

```
SQL> select ID, VALUE_TEXT, VALUE_NUMBER
2   from LIST_PARAMETERS
3   where ID in ('LINAC', 'PS')
4   and   PAR_ID = 'machine_status';
```

```
ID      VALUE_TEXT      VALUE_NUMBER
-----
LINAC  design
PS     existing
```

To retrieve the value of a calculated parameter, the query has to be more advanced. To facilitate the retrieval of the parameter values the special functions were defined in the database (stored PL/SQL package). With the use of these functions we do not need to know whether the parameter is calculated or not. Example for PVALUE:

```
SQL> select bb.pvalue('PS', 'machine_status') val
2   from dual;

VAL
-----
existing
```

or

```
SQL> select ID, bb.pvalue(ID, 'machine_status') val
2   from LIST_PARAMETERS
3   where ID in ('LINAC', 'PS');

ID      VAL
-----  -----
LINAC   design
PS      existing
```

To insert or update a value of a parameter the INSERT and UPDATE statements are used. In the example below, we update the value for a numeric parameter in the column VALUE_TEXT; a special database trigger will automatically update the column VALUE_NUMBER (if the inserted value is numeric, otherwise it will be empty):

```
SQL> update LIST_PARAMETERS
2   set VALUE_TEXT = '1.6'
3   where ID      = 'PS'
4   and   PAR_ID = 'machine_dB_dT_max';

1 record updated.
```

Appendix B. Web Service

For the whole community we propose a Web interface (see the picture below), reports in PDF format and the read access via a Web Service.

The screenshot shows a web browser window titled "Beta Beam Baseline Parameters: ion - Microsoft Internet Explorer provided by CERN". The address bar shows the URL: `http://beta-beam-parameters.web.cern.ch/beta-beam-parameters/servlet/rootObjectsData?object_type=ion`. The page content includes a navigation sidebar on the left with links for Constants, Ions, Machines, and Neutrino beams. The main content area features a table of parameters for an ion, with a header row: **Parameter**, **Symbol**, **Unit**, **Calculated**, **6He**, **18Ne**, **19Ne**, and **Proton equivalent**. The table is divided into two sections: **Ion** and **Target**.

Parameter	Symbol	Unit	Calculated	6He	18Ne	19Ne	Proton equivalent
Ion							
charge	q	e	no	2	10	10	1
A		nucleons	no	6	18	19	1
Q/A			on the fly	0.33	0.56	0.53	1.00
Equivalent mass		amu	no	6.019	18.006	19.002	1.007
lifetime at rest	$t_{1/2}$	s	no	0.81	1.67	17.30	∞
decay mode			no	b ⁻ to ⁶ Li	EC to ¹⁸ F	EC to ¹⁹ F	
Q-value		eV	no	3.51E+06	3.30E+06	2.20E+06	
nuclear spin			no	0	0	1/2	1/2
rest mass		eV	no	5.61E+09	1.68E+10	1.77E+10	9.39E+08
rest mass/nucleon		eV/nucleon	on the fly	9.343E+08	9.315E+08	9.313E+08	9.393E+08
Target							
primary proton energy		GeV	no	2.2	2.2		
average current		mA	no	0.10	0.10		
average power		kW	no	220	220		
target method			no	converter	direct	direct	

Page generated on Sun Apr 02 16:49:54 CEST 2006
 Database & Web design: Natalia Emelianenko, CERN
 Content: Adrian Fabich, CERN

The Web interface is a set of JSP and servlets and the Web service is deployed as JWS in the Apache Axis application within CERN central J2EE service.

With the use of the Web service the database connection is not needed:

```

In[1]:= Needs["WebServices`"]

In[2]:= WebServices`InstallService[
  "http://beta-beam-parameters-ws.web.cern.ch/beta-beam-
  parameters-ws/getParameterValue.jws?wsdl"];

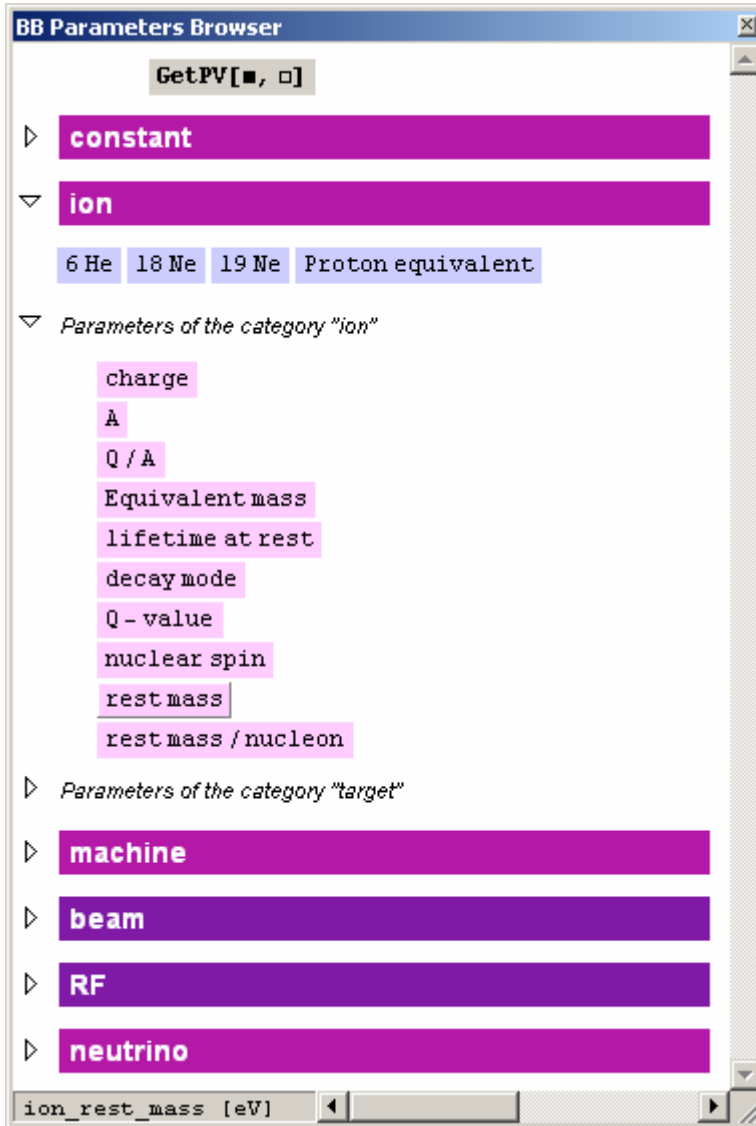
In[3]:= GetPV[id_String, par_String] :=
  If[NumericQ[ToExpression[getString[id, par]]],
    ToExpression[getString[id, par]], getString[id, par]]

In[4]:= GetPV["6He", "ion_half"]

Out[4]= 0.81
  
```

Appendix C. BB Parameters Browser

To ease the input of the object and parameter identifiers we have created a special palette (GUI) where the objects and parameters can be searched by their common names.



The palette notebook itself is generated in *Mathematica*; the current definitions of the objects and parameters are selected from the database and transformed into the button boxes.

When a user clicks on the object (blue) or parameter (rose) buttons, their identifiers are pasted into the user's active notebook.

The GetPV function template button is also provided. When the user inserts the object identifier (for the first placeholder), the second placeholder is selected automatically.

When the user points on a button, the value going to be pasted and the unit for the parameter (if it is the button for a parameter) is displayed in the status bar.

Appendix D. Text Format and Encoding Issues

The parameter names, units and parameter values can contain the special characters and symbols. The data in the database is stored as ASCII strings (character set WE8ISO8859P1), and the HTML notation is used to add special character entities, sub- or superscript. This makes the data input easy and the values readable in usual database interfaces.

While it is straightforward to display the data on the Web, the problem arise when transferring to *Mathematica* or PDF. Therefore, the special characters like “β” have to be translated into *Mathematica* definitions like “[beta]” to be nicely displayed on the BB Parameters Browser palette buttons or treated by the ToExpression[] function. The <SUB> and <SUP> HTML tags are removed or transformed into PDF chunks for the PDF reports. The translation is made by a PL/SQL function.

For example, the proton half life-time (infinity) is stored in the database like

```
VALUE_TEXT = '&infin;'
```

And in *Mathematica* we get:

```
In[5]:= GetPV["proton", "ion_half"]
Out[5]= \[Infinity]

In[6]:= ToExpression[%]
Out[6]= ∞

In[7]:= a = 1/%
Out[7]= 0
```