EUROPEAN LABORATORY FOR PARTICLE PHYSICS

CERN SL ??-?? (AP)

ZBASE User's Guide Version 1.1 An Impedance Data Base Program

Oliver S. Brüning

Abstract

The development of an impedance data base program was motivated by the changing impedance budgets in the LEP and LHC machines and the resulting need to recalculate the instability thresholds for different parameters. For LEP, the changes are implied by the LEP-II upgrade and for the LHC, the impedance data for most items is still being calculated and continues to change as the geometries of the impedance components converge to a final design. The development of the **ZBASE** program was guided by four goals. First, the data base program should collect the impedance information in a central place and in a standard format which allows easy access. Second, it should include information required for estimating the impact of the impedance on the beam dynamics. Third, the program should provide a user friendly graphical interface and fourth, it should provide an interface to the programs that are used for calculating the impedance data. Collecting not only the impedance data but also the input files for the programs that were used for calculating it and all data necessary for calculating stability threshold currents, greatly facilitates the recalculation of the threshold currents if the parameters change. For LEP-II, for example, the transverse mode coupling instability is expected to limit the maximum current in the machine. As the number of installed impedance components and the optics functions in the machine will change during the upgrade of LEP from 1996 to 1998 it is desirable to provide all the information required for estimating the instability threshold current in the data base and to provide an estimate for any stage of the upgrade. For the LHC, most of the impedance items are still in the design process and feedback from impedance calculations is desirable at this stage. A central data base that allows a semi-automatic recalculation of the impedance data once the geometry of an item changes will facilitate such a feedback.

Geneva, Switzerland

November 5, 1996

Contents

1 Introduction

The data base program is written in the Tcl-Tk [1] scripting language to provide portability between different Unix platforms and to facilitate the programming of a graphical user interface. The program, all sub-programs and all data reside on an afs-file system under the directory

$$\text{afs/cern.ch/user/z/zdata/public/zbase/} \tag{1}$$

In this way the data can be accessed from any machine with an afs-mount and all users have only read-permission on the data files. In addition, there are symbolic links to the data base program and the impedance data under the user **slath**. The data base program can be started from the directory (1) by calling the shell-script

$$\text{afs/cern.ch/user/z/zdata/public/bin/zbase} \tag{2}$$

The shell-script (2) first evaluates the environment variables of the **ZBASE** program ($\rightarrow$ Section 2) before starting the main Tcl-Tk shell-script '**zbase.tk**'. The '**zbase.tk**' Tcl-Tk shell-script resides under the directory (1).

The user interface has a help menu which explains the main features of the data base program. Most of the program options can be explored interactively by using the help pages of the interface and anybody interested in using the **ZBASE** program is encouraged to try this approach. In the following we will discuss the organisation of the data in the data base and describe each menu option of the data base program.

2 Environment Variables

The shell script (2) first sets six environment variables. Four of them, **ZBASE_DIR**, **ZBASE_DATA**, **ZBASE_EDITOR** and **LPDEST** can be specified by the user. The user can change the settings in the **ZBASE** program by setting the corresponding environment variables in his shell. The environment variables, **ZBASE_SOURCE** and **PATH** are set by the initial shell script (2) and can not be modified by the user. We will briefly describe the purpose of each environment variable:

- **ZBASE_SOURCE**:
  The environment variable **ZBASE_SOURCE** specifies the source directory where the program files are located.
- **ZBASE_DIR:**
  The environment variable **ZBASE_DIR** specifies the working directory of the **ZBASE** program. All temporary files will reside in this directory. By default, the program chooses the directory

  $$\text{\$HOME/tmp/zbase} \tag{3}$$

  If the working directory does not exist, the program will create it during the startup. The user can specify a different working directory by setting the **ZBASE_DIR** environment variable.

- **ZBASE_DATA:**
  The environment variable **ZBASE_DATA** specifies the directory where the impedance data resides. By default, the program chooses the **data** directory relative to (1). If the user wants to select a different directory during the program start up he can specify it with the **ZBASE_DATA** environment variable. The user can also change the data base directory once the **ZBASE** program is running (→ Section 7.1).
- **ZBASE_EDITOR:** The environment variable **ZBASE_EDITOR** determines which editor will be used as the default editor in the **ZBASE** program. The default editor can also be changed from within the **ZBASE** program using the **Misc** menu (→ Section 7.9).
- **PATH:** The environment variable **PATH** determines where the program will look for other programs or shell scripts. At startup, the shell script (2) includes the paths for some required programs in front of the original **PATH** name. This change only applies for the **ZBASE** program. For other programs the environment variable **PATH** remains unchanged.
- **LPDEST:** The environment variable **LPDEST** determines which printer will be used as the default destination for printing. The default printer can also be changed from within the **ZBASE** program using the **Misc** menu (→ Section 7.9).

After evaluating these environment variables the Tcl-Tk shell-script is started.

3 Program directories

The directory (1) has the following ten sub-directories:

- **bin** directory:
  Not all tasks and functions of the **ZBASE** program are written in Tcl-Tk. For some procedures, for example searching big files or modifying the input files of the external programs, it is more efficient to find a solution using other scripting languages or c- and Fortran-programs. The compiled c- and Fortran-programs are stored in the **bin** sub-directory.
- **source** directory:
  The **source** directory contains the source code of the compiled binaries.
- **shellscr** directory:
  All shell-script procedures are stored in the **shellscr** sub-directory.
- **data** directory:
  The **data** sub-directory contains the impedance data of the data base and therefore, plays a key role for the **ZBASE** program.
- **archive** directory:
  The **archive** sub-directory contains the backup files of the data base and serves as a data archive. The user can change the default data base interactively to one of the files in the **archive** sub-directory (→ Section 7.1).
- **scratch** directory:
  If the user selects a tar file from the data base **archive**, the compressed tar-file in the **archive** sub-directory must first be expanded. If the user has a **PARC** account, the

ZBASE program will use the 'project/parc/scratch' directory on the CERN-SP cluster for this expansion. If the user has not a PARC-account the **ZBASE** program will expand the compressed data base files in the **scratch** sub-directory. In both cases, the program will remove all temporary files and directories on the **scratch** or 'project/parc/scratch' directories on exit (→ Section 7.1).

- **help** directory:
  The **Help** sub-directory contains the help-text files for the **ZBASE** program (→ Section 7.11).
- **examples** directory:
  ZBASE offers interfaces and example input files to the ABCI, URMEL, MAD and MAFIA programs (→ Section 7.8). The **examples** sub-directory contains the example input files for all external programs.
- **optics** directory:
  The **optics** sub-directory contains information on the position of each item in the **machine** and provides 'include files' for the MAD program for optics calculations (→ Section 7.5 for more details).
- **bitmaps** directory:
  The **bitmaps** sub-directory contains bitmaps and icon symbols used in the **ZBASE** program.

In addition to the sub-directories under the **ZBASE** directory (1), the program expects a **tmp** directory relative to the user's home directory. If the user does not have such a **tmp** directory the **ZBASE** program will create it during the program start-up.

4 Organisation of the data in the data base

All data resides on an afs-file system under the user (1) in the directory **data**. Directly under the **data** directory are the **.info** and **.beam** files. The **.info** file is a **log-file**. Any modification of the data base is recorded in this file. **ZBASE** automatically records the date, the user name, the time and the modified **item** label for each modification. In addition, the user can enter a comment describing the modification (→ Section 7.1). Each entry in the **log-file** terminates with a line of hash symbols ('#'). The **.beam** file contains the default beam parameters that are loaded during the startup of the **ZBASE** program (→ Sections 5 and 7.4).

The impedance data is hierarchically organised in four different levels of sub-directories:

$$\text{data}/\textbf{machine}/\textbf{group}/\textbf{item}/\textbf{mode}/\text{data-files}. \tag{4}$$

The **machine** sub-directory corresponds to a machine for which the data base contains entries. Each **group** sub-directory refers to different impedance items belonging to the same group and the third sub-directory specifies a single impedance item. Here, each **item** sub-directory in the **ZBASE** program corresponds to one element in the Oracle data base. This allows a straight forward link between the Oracle data base and the **ZBASE** program. The fourth sub-directory specifies the mode, e.g. longitudinal or transverse, for which the impedance data entries are calculated. Such a hierarchical directory structure allows an access of the impedance data without the graphical user interface.

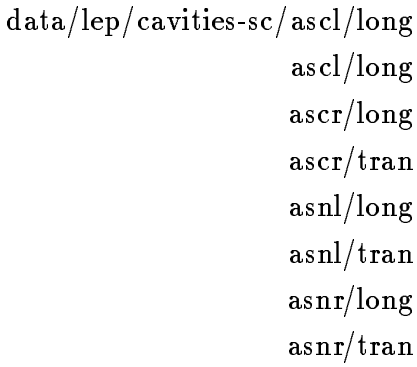


Figure 1: *Example for the data base directory structure. It shows the sub-directories for the* **machine**, **group**, **item** *and* **mode** *(longitudinal or transverse) labels.*

For example, Fig. 1 shows the entries for the LEP super-conducting cavities in the data base. The LEP machine has its own sub-directory **lep** and the entries for all super-conducting cavities are in the sub-directory **cavities-sc**. The Oracle data base has four different elements for the super-conducting cavities. The elements 'ASCR' and 'ASCL' correspond to the Nb-film cavities and the elements 'ASNR' and 'ASNL' to the Nb-sheet cavities. For each element there are two sub-directories: One for the longitudinal impedance data and one for the transverse.

The impedance data entries consist of six different types of data:

- First, there is one file which contains some basic information on the impedance item. In the following and in the **ZBASE** program we will call these entries the **item attributes**. The information is stored in a file named **.zbase**. The first entry is a comment line describing the item. The second entry is the number of impedance components that are in the machine. The third and fourth entries give the average horizontal and vertical betatron values in meters at the location of the impedance component in the machine. The fifth entry is the low frequency inductance (Z_0/n) in Ω and the sixth and seventh entries give the horizontal and vertical low frequency limit of the imaginary part of the transverse impedance ($Z_{x,\perp}$ and $Z_{y,\perp}$) in Ω/m. The eighth entry specifies the location of the item in the machine ($\rightarrow$ Section 7.5). Whenever a new item is entered into the data base, the program will create the **.zbase** file and ask the user to supply the required information.
- The second type of data are the higher-order-mode entries (e.q: 'frequency', 'R/Q' and 'Q-value') which are also stored in the **.zbase** file ($\rightarrow$ Section 4). The mode frequencies must be specified in Hz and the R/Q values in Ω. The program expects all entries in standard **ANSI-C** format.
- The third data type are the loss factors related to an impedance item. They are stored in a file named **.loss**. The file **.loss** contains information on the longitudinal and transverse loss factors for different bunch lengths. The first line gives the bunch length in meter, the second the longitudinal loss factor, the third, fourth and fifth line the horizontal loss factors (transverse, azimuthal, and longitudinal) and the sixth, seventh and eight line the vertical loss factors (transverse, azimuthal, and longitudinal). The loss factors are

```
[cernsp] ~/public/zbase % ls -la data/lep/cavities-sc/ascl/long
.
..
.loss
.wakepot.abci.azimx.0.5e-2.gz
.wakepot.abci.azimx.0.75e-2.gz
.wakepot.abci.azimx.1.0e-2.gz
.wakepot.abci.azimx.1.5e-2.gz
.wakepot.abci.long0.0.5e-2.gz
.wakepot.abci.long0.0.75e-2.gz
.wakepot.abci.long0.1.0e-2.gz
.wakepot.abci.long0.1.5e-2.gz
.wakepot.abci.longx.0.5e-2.gz
.wakepot.abci.longx.0.75e-2.gz
.wakepot.abci.longx.1.0e-2.gz
.wakepot.abci.longx.1.5e-2.gz
.wakepot.abci.tranx.0.5e-2.gz
.wakepot.abci.tranx.0.75e-2.gz
.wakepot.abci.tranx.1.0e-2.gz
.wakepot.abci.tranx.1.5e-2.gz
.zbase
abci.ascl
abci.ascl.ps.gz
abci.ascl.short.ps.gz
urmel.ascl.emode
urmel.ascl.mmode
[cernsp] ~/public/zbase %
```

Figure 2: *Stored files for one of the super-conducting LEP cavities in the ZBASE data base.*

given in V/pC, V/pC/m, V/pC/m and V/pC/m^2 for the longitudinal and the transverse, azimuthal, and longitudinal loss factors of the dipole mode respectively.

- Fourth, the data base program stores wake potentials. The longitudinal wake potential is given in V/pC, the transverse and azimuthal wake potentials of the dipole mode in V/pC/m and the longitudinal wake potential of the dipole mode in V/pC/m^2. There is one file for each bunch length and each wake potential type. The files have the following naming convention ($\rightarrow$ Fig. 2). All files begin with '.wakepot'. The second string identifies the program that produced the wake potential, the third identifies the wake potential type (e.q. 'long0', 'tranx', 'azimx', 'longx' for the longitudinal and the horizontal, azimuthal and longitudinal wake potentials of the dipole mode) and the third string the bunch length in meter. The file entries consist of a few comment lines and two data columns. The comment lines can provide additional information to the wake potential data. The first column of the data entries gives the distance from the bunch head and the second column the corresponding wake potential value. The position of the bunch head is defined 5 σ in front of the bunch centre.
- Fifth, the **ZBASE** program stores the **input files** and some of the **output files** of the programs that were used for calculating the impedance data. All input and output files have a prefix that specifies the program that is associated with the files.
- Sixth, the **ZBASE** program stores the **location of the items** in a machine in the **.location** file. The first entry in the **.location** file specifies a reference position in the machine and all other entries specify the location of the items with respect to this reference point in meter. There is one line per position. However, the number of specified locations does not have to agree with the number of installed items in the **attributes** file. While the total number of installed items is specified in the **attributes** file the data in the **.location** file can be used for calculating the average betatron function values for each item. This

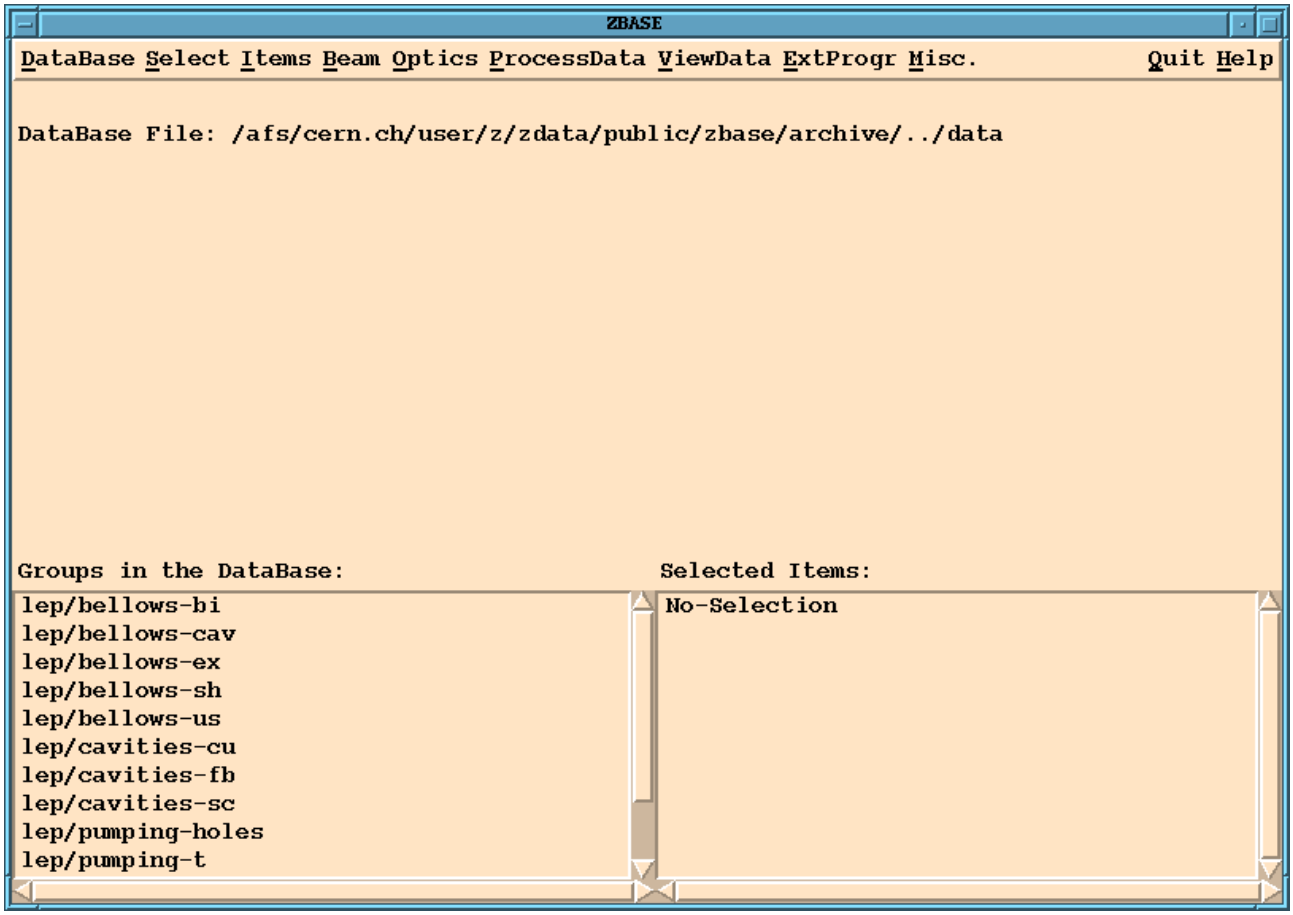


Figure 3: *The graphical user interface for the* **ZBASE** *program.*

separation of the item positions and the entry for the number of installed **items** provides flexibility for changing the number of installed **items** or calculating the average betatron function values. For example, if the optical functions change significantly over the length of an **item** it is convenient to calculate the average betatron function values by distributing several markers in the optics calculation program over the length of the **item**. In this case, the number of entries in the **.location** file does not coincide with the number of installed **items** in the machine.

Fig. 2 shows the data entries for the longitudinal impedance of one of the super-conducting cavities in LEP. There is the **.loss** file with the loss factor entries, 16 different wake potential entries for four different bunch lengths, the **.zbase** attributes file, an input file for ABCI [2], the postscript output file of ABCI, and two URMEL [3] input files. The program expects all entries in standard **ANSI-C** format.

In addition to the data related to one item, each machine directory has a **.beam** file and a **.files** sub-directory. The **.beam** file contains the default beam parameters for each machine in the data base and the **.files** sub-directory contains files with additional information that is not directly associated with the geometry of an item ($\rightarrow$ Sections 7.3 and 7.5).

5 Processing Data

One of the goals for the **ZBASE** data base program in addition to storing impedance information is evaluating the net effect of the impedance on the beam dynamics. For example, it provides not only the HOM parameters for the impedance items but also some estimates for the corresponding multi-bunch instability rise times. To this end, **ZBASE** distinguishes between **primary** and **secondary data**:

- The **primary data** describes all information directly associated with the geometry of an item. This would include the inductance, the loss factors and the HOM parameters for example.
- The **secondary data** describes all information not directly associated with an item. This includes for example the betatron functions and the beam parameters.

In general, the **primary data** can not be modified by the user unless the user has write permissions for the data base directory. (A user can always create a private copy of the data base in a directory for which he has write permissions (→ Section 7.1)). During startup the **ZBASE** program searches the directories in the data base for **secondary data** and stores the values in internal arrays. The entries of these arrays can be modified by any user (→ Sections 7.3, 7.4 and 7.5).

In order to process impedance data, the user must first select **items** from the data base. One can either select individual **items** or **groups** of items or all items belonging to one **machine** in the data base (→ Section 7.2). All selected item names are again stored in an array. When the program processes the impedance data it first goes through this name array and reads the required **primary data** for each item from files in the data base. The required **secondary data** is read from the internal arrays.

Depending on the amount of data, the processes can take up to a few minutes. During this time, the graphical user interface (→ Section 6) will be inactive and the user can not interact with the program until the process has terminated. All processes are described in more detail in Section 7.

6 The Graphical User Interface

Fig. 3 shows the graphical user interface for the **ZBASE** program. The interface consists of a menu bar and three display areas:

- The first display area displays the selected data base file and the **attributes** for the **items** in the data base (→ Sections 7.1, 7.6 and 7.7).
- The second display area in the lower left corner of Fig. 3 lists all impedance **groups** which are available in the data base.
- The third display area in the lower right corner of Fig. 3 lists the **selected items** that have been selected for further processing.
- The menu bar allows the user to modify and process the data in the data base.

The menu bar offers eleven different menus:

- The **DataBase** menu allows the user to open a new data base, create backup files, save, remove or modify data in the data base and to create or remove **items** in the data base.
- The **Select** menu allows the user to select **items** for further processing.
- The **Items** menu allows the user to view and modify the number of installed **items**.
- The **Beam** menu allows the user to view and modify the beam parameters of the **machine**.
- The **Optics** menu allows the user to view, calculate and modify the values of the stored optical functions at the location of the impedance **items** in the machine.
- The **ProcessData** menu allows the user to process the impedance data in the data base. For example, one can calculate the multi-bunch instability rise times or the transverse mode coupling instability threshold currents for the selected parameters.
- The **ViewData** menu allows the user to look at the data of one **item** in the data base.
- The **ExtProgr** menu starts graphical interfaces to external programs (MAFIA, ABCI, URMEL and MAD).
- The **Misc.** menu allows the user to select a printer or editor.
- The **Quit** menu exits the **ZBASE** program.
- The **Help** menu provides help pages for the **ZBASE** program.

We will describe each menu option in more detail in a separate Sub-section of Section 7.

The **ZBASE** program has six generic menu buttons which appear in all windows generated by the program:

- The **O.K.** menu button initiates a procedure.
- The **Export** menu button allows the user either to save or print the contents of a display window. In the first case, a new window opens up and the user is asked to enter a file name. The file will always be saved relative to the users home directory. In the second case, the program prints the contents of the display window on the selected printer. By default, the program uses the printer that is specified in the **LPDEST** environment variable. The user can always select a new printer in the **MISC** menu of the main menu bar ($\rightarrow$ Section 6).
- The **Plot** button produces a plot of the displayed data (using the first data column for the horizontal and the second data column for the vertical axis). The program uses gnuplot for producing a postscript file which is displayed using ghostview. (See the Unix manual pages for more details on the gnuplot and ghostview programs.) Before plotting the data, the program first eliminates all lines not containing data points (comment lines for example).
- The **Search** menu button allows the user to search the text in a display window for a string pattern or to move to the beginning or end of the displayed text.
- The **Quit** menu button closes the display window and aborts the current process.
- The **Help** menu button opens a new window with help information related to the current process.

The user interface is mouse oriented and the following two lists give an overview and a short description of the mouse functions.

General mouse functions:

- **Buttons, menu-options** and **check-boxes** can be activated by moving the pointer over the **button**, **menu-option** or **check-box** and clicking the left mouse button once.
- **Scrollbars** can be either moved by positioning the pointer over the scrollbar and moving the mouse while keeping the left mouse button pressed down or by moving the pointer on one of the triangles at the scrollbar and pressing the left mouse button down.
- **Marking lines in a display area:** By clicking the left mouse button once in the display area one marks the line under the pointer. By dragging the mouse over the display area while keeping the left mouse button pressed down one can mark more than one line in the display area.
- **Un-marking lines in a display area:** By clicking the right mouse button once in the display area one un-marks the highlighted line under the pointer. By dragging the mouse over several highlighted lines in the display area while keeping the right mouse button pressed down one can un-mark more than one line in the display area.
- **Selecting a line in a display area:** By double-clicking the left mouse button in the display area one selects the line under the pointer.

Additional mouse functions in the main program display:

- Double-clicking the middle mouse button in the **Groups in the DataBase** display selects the highlighted groups for further processing.
- Double-clicking the left mouse button in the **Selected Items** display lists the item **attributes** of the item located under the pointer.
- Double-clicking the middle mouse button in the **Selected Items** display un-selects the highlighted items from the **Selected items** list.
- Clicking first the left and then the right mouse button lists the files belonging to the **item** under the pointer in a new display window.

7 Menu options of the ZBASE program

The menu bar of the main program display offers eleven different menus. In the following we will describe each menu in a separate Sub-section.

7.1 The 'DataBase' menu

The **DataBase** menu allows the user to open a new data base, create backup files, save data in the data base and to create or remove **items** in the data base. In the following we will describe each sub-menu of the **DataBase** menu in a separate Sub-section.

7.1.1 View LogFile

This sub-menu opens the **.info log-file** (→ page 4) and displays its contents in a new window.

7.1.2 Save Secondary Parameters in DataBase

This sub-menu writes the **secondary data** (→ Section 5) to files in the data base. The **secondary data** is stored in internal arrays and can be modified by the user (→ Section 4). Before initiating the process, the program first displays a message box that informs the user that the process will modify the data base files and asks the user to confirm the action. The program will then ask the user to enter a comment to the **log-file** (→ page 4). As this menu will modify the files in the data base the user needs to have write permission to the data base directory. If the user does not have write permission to the data base directory the program will display an error message and abort the process without modifying the data base entries.

7.1.3 Create a private Copy of the DataBase

This sub-menu allows the user to copy the current data base into his own home directory (→ Section 7.1.4). Upon selection, the program opens a new window with two entry displays. The **Path name** entry display shows the users home directory. The **Directory name** entry display shows the default directory name to which the data will be copied. Both entries can be modified by the user. However, the user must have write permission on the directory where the data will be stored.

7.1.4 Open a new DataBase Directory

This sub-menu allows the user to open a new data base file. By default, the program allocates the standard data base directory (1) under the user **zdata**. Upon selection, the program opens a new window with one entry and one display area.

- The **Source Directory** entry display shows the path where the program will look for the new data base.
- The **display area** shows the contents of the selected **Source Directory**.

By default, the program shows the contents of the **ZBASE** archive directory (→ Section 3). The **Source Directory** entry can be modified by the user and upon pressing the **Enter** key on the keyboard the program will display the contents of the new directory in the **display area**. The user can select a new data base by double clicking with the left mouse button on one of the displayed lines:

- If the selected file has the extension '.tgz' the program will automatically uncompress and un-tar the selected file into a **scratch** directory.
- If the user has a **PARC** account, the program will extract the data on the 'project/parc/scratch' directory on the CERN-SP cluster.
- If the user has no **PARC** account, the program will use the **scratch** directory in the data base home directory (1) (→ Section 3).
- If the selected line does not have the extension '.tgz' the program assumes that it is a **ZBASE** data base directory and it will link the data base directly to the selected directory.

In any case, the program checks whether the selected directory has a **ZBASE** data base format by searching for the **.zbase** files in the data base. If the selected directory is not a **ZBASE**

data base directory the program displays an error message and aborts the process. The current data base directory is always displayed in the main display area of the **ZBASE** user interface (→ Fig. 3). (See Section 7.1.3 for how to create a private copy of the data base directory.)

7.1.5 Save DataBase in a tar-File

This sub-menu allows the user to store all information of the current data base in a compressed tar file (extension '.tgz', see the manual pages on gnu-tar for more information on this format). Upon selection, the program opens a new window with two entry displays.

- The **Path name** entry display shows the directory path where the compressed tar file will be stored.
- The **File name** entry display shows the file name for the compressed tar file.

By default, the program chooses the users home directory for the **Path name** entry and a **File name** which consists of the prefix '**zbase.**' and the current date as an extension. Both entries can be modified by the user. However, the user must have write permission on the directory where the compressed tar file will be stored. The back-up process might take some time depending on the size of the data base and the **ZBASE** program will open a message box with this warning.

7.1.6 Save Item Data in a tar-File

This sub-menu allows the user to store all information belonging to one **item** in the data base in a compressed tar file (extension '.tgz', see the manual pages on gnu-tar for more information on this format). This is useful if the user has created item entries in a private copy of the data base and wants to export the data to another data base. When initiated, the program opens a new window with three entry displays where the user must specify the **machine**, **group** and **item** labels for the item to be saved (→ Section 4 for more details on **machine**, **group** and **item** labels).

After pressing the **O.K.** button the program opens a new window with two entry displays.

- The **Path name** entry display shows the directory path where the compressed tar file will be stored.
- The **File name** entry display shows the file name for the compressed tar file.

By default, the program chooses the users home directory for the **Path name** entry and a **File name** which consists of the prefix '**zbase.machine.group.item.**' and the current date as an extension. Both entries can be modified by the user. However, the user must have write permission on the directory where the compressed tar file will be stored.

Depending on the size of the data base the back-up process can take a few minutes and the **ZBASE** program will open a message box with this warning. Before storing the information in the tar file, the program will search the current **log-file** (→ page 4) for entries on the selected **item** and will copy the relevant entries to a new **log-file** which will also be stored in the tar file. (See also Section 7.1.7)

7.1.7 Load Item Data from a compressed tar-File

This sub-menu allows the user to install **item** data from a compressed tar-file in the data base (→ Section 7.1.6). It has the same functionality as the **Open A New DataBase Directory** menu (→ Section 7.1.4). If the **item** label in the tar-file corresponds to an existing entry in the data base, the process will overwrite all existing information in the data base and the program opens a message box with this warning. Once the user confirms the action the program first creates a backup tar-file of the old information in the **ZBASE_DIR** directory (→ Section 2) before extracting the information from the specified tar-file. If the new data has not a **ZBASE** data base format the program discards it and replaces it by the old backup information.

7.1.8 Create New Item

This sub-menu allows the user to create a new **item** in the current data base. When initiated, the program opens a new window with three entry displays where the user must specify the **machine**, **group** and **item** labels for the item to be created (→ Section 4). If one of the specified labels does not correspond to a directory in the data base, the program will create the corresponding directory.

After creating the directories, the program will ask the user to enter a comment to the **log-file** (→ page 4) and to specify the item **attributes** (→ page 4). All other information must be entered with the sub-menu **Change Item Data** in the **DataBase** menu of the main menu-bar (→ Section 7.1.9). The user must have write permission for the current data base. If the user does not have write permission, the program will display an error message and abort the procedure.

7.1.9 Change Item Data

This sub-menu allows the user to modify the **item** entries. The user must have write permission for the data base directory. The program first opens a new window where the user can select six different modification types:

- **AddFile**: The **AddFile** menu allows the user to add additional files to an item. Upon selection, the program opens a new window with four entry displays and five check-boxes. In the entry displays the user must specify the **machine**, **group** and **item** labels and the **target file name** for the new file in the data base (→ Section 4). Furthermore, the user must specify whether the new file will correspond to a longitudinal or transverse impedance calculation by selecting the corresponding check-button. In the same way, the user must specify whether the calculation was done using the MAFIA, ABCI or URMEL program. Depending on these selections, the file will get the prefix 'abci.', 'urmel.', or 'mafia.' and will be stored in a sub-directory for the longitudinal or transverse impedance data (→ Section 4). Postscript files will be automatically compressed before storing the files in the data base and all compressed postscript files get the new extension **.gz**.

  The **O.K.** menu button opens a new window where the user can browse his/her directories for the new file. After selecting a file, the program will ask the user to enter a comment to the **log-file** of the data base (→ page 4).
- **RemoveFile**: The **RemoveFile** sub-menu allows the user to remove files from the data base. Upon selection, the program opens a new window with four entry displays, three

menu buttons and two check-boxes. In the entry displays the user must specify the **machine**, **group** and **item** labels and the **target file name** for the file in the data base (→ Section 4). Furthermore, the user must specify whether the selected file corresponds to the longitudinal or transverse impedance data by selecting the corresponding check-button.

- **Attributes:** The **Attributes** sub-menu (→ page 4) allows the user to modify the **attributes** of an **item**. Upon selection, the program opens a new window with three entry displays for the **machine**, **group** and **item** labels of the item. Pressing the **Enter** key moves the cursor from one entry display to the next. The new window has three menu buttons. The **O.K.** menu button will first ask the user to confirm the action before opening another window where the user can enter a comment for the **ZBASE log-file** (→ page 4). After closing the window the program opens a new window with an entry display for each **attribute** (→ page 4) of the **item**. Pressing the **Enter** key moves the cursor from one entry display to the next.
- **HOM**: The **HOM** sub-menu allows the user to save new data for the higher order modes of an item. The program asks the user to specify a file witch contains the mode frequencies, R/Q, and Q-values of the higher order modes. The mode frequencies must be specified in Hz and the R/Q values in Ω. The program expects entries in standard **ANSI-C** format and one line for each **HOM**. The parameters must appear in the above order. Upon selection, the program opens a new window with three entry displays for the **machine**, **group** and **item** labels (→ page 4) of the **item** and four check-boxes. Pressing the **Enter** key moves the cursor from one entry display to the next. With the check-boxes the user must specify whether the **HOM** parameters belong to the transverse or longitudinal impedance and whether the new information should be added to the existing entries or whether it should replace the existing information. The **O.K.** button will open a new window where the user can browse through directories and select a file containing the new **HOM** parameters.
- **Location**: The **Location** sub-menu allows the user to save new data for the locations of the **item** in the **machine**. The program will ask the user to specify a file witch contains the locations in meters. The program expects entries in standard **ANSI-C** format and one line with one number for each position. Upon selection, the program opens a new window with three entry displays for the **machine**, **group** and **item** labels (→ page 4) of the **item**. Pressing the **Enter** key moves the cursor from one entry display to the next. The **O.K.** button opens a new window where the user can browse through directories and select a file containing the new **location** parameters.
- **LossFactor**: The **LossFactor** sub-menu allows the user to add new entries to the **.loss** file which contains the loss factors for different bunch lengths. Upon selection, the program opens a new window with eleven entry displays. Three displays are for the **machine**, **group** and **item** labels of the **item**, one for the bunch length and seven for the horizontal and vertical loss factors (→ Section 4). Pressing the **Enter** key moves the cursor from one entry display to the next.
- **Wake-pot**: The **Wake-pot** sub-menu allows the user to save new wake potentials for an item. Upon selection, the program opens a new window with four entry displays and nine check-boxes. Three entry displays are for the **machine**, **group** and **item** labels of the **item** and one for the bunch length. Pressing the **Enter** key moves the cursor from

one entry display to the next. The check-boxes specify the type of the wake potential (e.g. longitudinal, and transverse, azimuthal and longitudinal wake potentials of the dipole mode with horizontal or vertical offset). All wake potential files are automatically compressed before the files are stored in the data base and all files get the extension '**.gz**'. (See Section 4 for the naming convention and data structure of the wake potential files.)

7.1.10 Remove Machine

This sub-menu allows the user to remove all data belonging to one **machine** from the data base. When initiated, the program opens a new window with three entry displays. In the first entry display the user must specify the **machine** label for the machine for which the data will be removed (→ Section 4). If the specified machine label does not correspond to a directory in the data base, the program will prompt an error message.

7.1.11 Remove Group

This sub-menu allows the user to remove all data belonging to one **group** from the data base. When initiated, the program opens a new window with three entry displays. In the first two entry displays the user must specify the **machine** and **group** labels for the group who's data is supposed to be removed (→ Section 4). If the specified machine or group labels do not correspond to a directory in the data base, the program will prompt an error message.

7.1.12 Remove Item

This sub-menu allows the user to remove all data belonging to one **item** from the data base. When initiated, the program opens a new window with three entry displays. In these entry displays the user must specify the **machine** and **group** and **item** labels for the data which is supposed to be removed (→ Section 4). If the specified **machine**, **group** or **item** labels do not correspond to a directory in the data base, the program will prompt an error message.

7.2 Select

The **Select** sub-menu allows the user to select and un-select **items** for further processing of the impedance data (→ Section 7.6). In the following we will describe each sub-menu of the **Select** menu in a separate Sub-section.

7.2.1 Select Machine for Processing Data

This sub-menu selects all **items** corresponding to one **machine** for further processing. The menu has a cascade of sub-menu buttons with one button for each **machine** in the data base (→ Section 4). Moving the pointer over one of the sub-menu buttons and pressing the left mouse button once will select all data of the corresponding machine. The selected items will appear in the **Selected Items** display of the program user interface and the name of the selected machine will be posted in the main display area of the user interface (→ Section 6).

7.2.2 Select Group for Processing Data

This sub-menu allows the user to select all **items** corresponding to one impedance **group** in the data base for further processing (→ Section 4). The menu opens a new window with all **machine** names in the data base. After selecting one **machine**, the program displays all impedance **groups** for the selected **machine** for selection (→ Section 6 on using the mouse).

7.2.3 Select Item for Processing Data

This sub-menu allows the user to select one **item** in the data base for further processing (→ Section 4). The menu opens a new window with all **machine** names in the data base. After selecting one **machine**, the program first displays all impedance **groups** for the selected **machine** for selection and then all **items** belonging to the selected **group** (See Section 6 on using the mouse).

7.2.4 Unselect Group

This sub-menu allows the user to un-select all **items** corresponding to one impedance **group** in the data base from the **Selected Item** list (→ Section 4). It has the same functionality as the **Select Group** menu (→ Section 7.2.2).

7.2.5 Unselect Item

This sub-menu allows the user to un-select one **item** in the data base from the **Selected Item** list (→ Section 4). It has the same functionality as the **Select Item** menu (→ Section 7.2.3).

7.2.6 Unselect All

This sub-menu un-selects all **items** from the **Selected Items** list.

7.2.7 Save Current Selection in File

This sub-menu allows the user to store the current **Selected Item** list in an external file. Upon selection, the program opens a new window where the user can enter a file name. The file will be stored relative to the users **$HOME** directory and the program attaches the suffix **.itemselection** to the file name (→ Section 7.2.8).

7.2.8 Read Item Selection from File

This sub-menu allows the user to read a **Selected Item** list from file. Upon selection, the program opens a new window with all **machine** names in the data base and the users **$HOME** and **$HOME/tmp** directories. The user can browse through the directories and select a file containing a list of **items** (→ Section 7.2.7).

7.3 Items

This sub-menu allows the user to display and modify the number of installed **items** in the machine (→ Section 5). Each **item** in the data base has an entry in the **attributes** file (→ page 4) which specifies the number of installed **items** in the **machine**.

7.3.1 List Number of Installed Items

This sub-menu displays the number of installed items for all impedance **items** in the data base. The program opens a new window displaying the number of installed **items**, the **item** names (**machine**/**group**/**item**) and the comment line from the **attribute** files (→ page 4). Fig. 4 shows an example of such a listing.

7.3.2 Edit Number of Installed Items

This menu allows the user to modify the number of installed **items** from the **attributes** file (→ page 4). The menu opens a new window with one display area, one entry area and four buttons. The window is shown in Fig. 5.

- The **display area** shows the name of the **item** and the **entry display** the current entry for the number of installed **items** in the **machine**.
- The **Reset** button jumps back to the first **item** in the item list.
- The **Prev** button moves backwards to the previous **item** in the item list.
- The **Next** button moves to the next **item** in the item list.
- The **Search String** button allows the user to search the internal array for a string pattern. If the program finds a match, it will display the first **item** in the array that matches the string pattern.

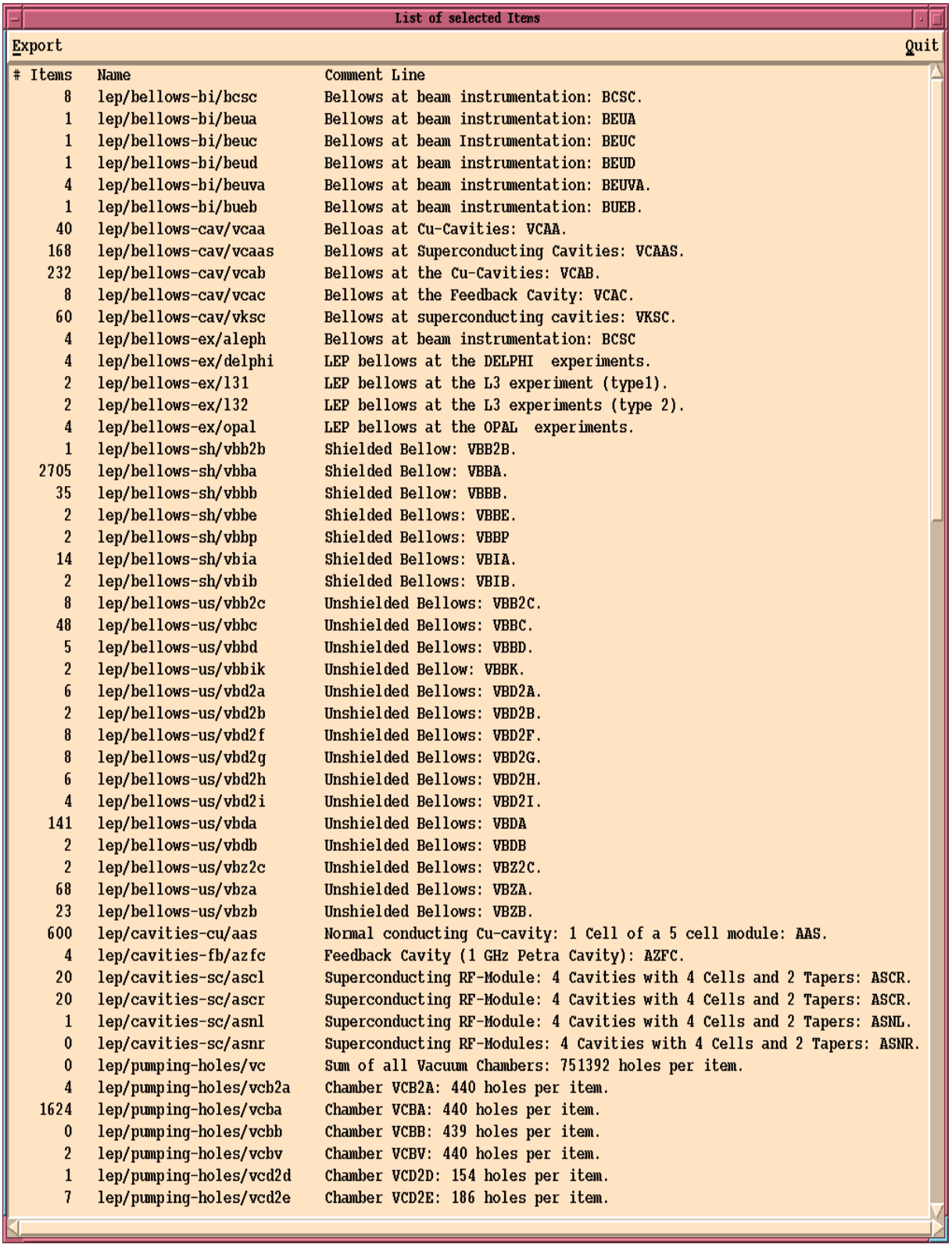
After changing the entry in the entry display area the user confirms the modification by pressing the **Enter** key on the keyboard. Pressing the **Enter** key on the keyboard also takes the user to the next **item** in the item list.

7.3.3 Load Number of Installed Items from File

This sub-menu allows the user to load new values for the number of installed **items** from a file. The menu opens a new window where the user can browse through directories and can select a file containing the new values (→ Section 7.3.4). For each **machine** in the data base, the corresponding data base files are stored in the **.files** sub-directory (→ Section 4).

7.3.4 Save Number of Installed Items to File

This sub-menu allows the user to save the current values for the number of installed **items** in a file. The menu opens a new window where the user can enter a file name relative to his home directory (→ Section 7.3.3). In addition to the specified name the file will get the extension **nitem.data**.



# Items	Name	Comment Line
8	lep/bellows-bi/bcsc	Bellows at beam instrumentation: BCSC.
1	lep/bellows-bi/beua	Bellows at beam instrumentation: BEUA
1	lep/bellows-bi/beuc	Bellows at beam Instrumentation: BEUC
1	lep/bellows-bi/beud	Bellows at beam instrumentation: BEUD
4	lep/bellows-bi/beuva	Bellows at beam instrumentation: BEUVA.
1	lep/bellows-bi/bueb	Bellows at beam instrumentation: BUEB.
40	lep/bellows-cav/vcaa	Belloas at Cu-Cavities: VCAA.
168	lep/bellows-cav/vcaas	Bellows at Superconducting Cavities: VCAAS.
232	lep/bellows-cav/vcab	Bellows at the Cu-Cavities: VCAB.
8	lep/bellows-cav/vcac	Bellows at the Feedback Cavity: VCAC.
60	lep/bellows-cav/vksc	Bellows at superconducting cavities: VKSC.
4	lep/bellows-ex/aleph	Bellows at beam instrumentation: BCSC
4	lep/bellows-ex/delphi	LEP bellows at the DELPHI experiments.
2	lep/bellows-ex/l31	LEP bellows at the L3 experiment (type1).
2	lep/bellows-ex/l32	LEP bellows at the L3 experiments (type 2).
4	lep/bellows-ex/opal	LEP bellows at the OPAL experiments.
1	lep/bellows-sh/vbb2b	Shielded Bellow: VBB2B.
2705	lep/bellows-sh/vbba	Shielded Bellow: VBBA.
35	lep/bellows-sh/vbbb	Shielded Bellow: VBBB.
2	lep/bellows-sh/vbbe	Shielded Bellows: VBBE.
2	lep/bellows-sh/vbbp	Shielded Bellows: VBBP
14	lep/bellows-sh/vbia	Shielded Bellows: VBIA.
2	lep/bellows-sh/vbib	Shielded Bellows: VBIB.
8	lep/bellows-us/vbb2c	Unshielded Bellows: VBB2C.
48	lep/bellows-us/vbbc	Unshielded Bellows: VBBC.
5	lep/bellows-us/vbbd	Unshielded Bellows: VBBD.
2	lep/bellows-us/vbbik	Unshielded Bellow: VBBK.
6	lep/bellows-us/vbd2a	Unshielded Bellows: VBD2A.
2	lep/bellows-us/vbd2b	Unshielded Bellows: VBD2B.
8	lep/bellows-us/vbd2f	Unshielded Bellows: VBD2F.
8	lep/bellows-us/vbd2g	Unshielded Bellows: VBD2G.
6	lep/bellows-us/vbd2h	Unshielded Bellows: VBD2H.
4	lep/bellows-us/vbd2i	Unshielded Bellows: VBD2I.
141	lep/bellows-us/vbda	Unshielded Bellows: VBDA
2	lep/bellows-us/vbdb	Unshielded Bellows: VBDB
2	lep/bellows-us/vbz2c	Unshielded Bellows: VBZ2C.
68	lep/bellows-us/vbza	Unshielded Bellows: VBZA.
23	lep/bellows-us/vbzb	Unshielded Bellows: VBZB.
600	lep/cavities-cu/aas	Normal conducting Cu-cavity: 1 Cell of a 5 cell module: AAS.
4	lep/cavities-fb/azfc	Feedback Cavity (1 GHz Petra Cavity): AZFC.
20	lep/cavities-sc/ascl	Superconducting RF-Module: 4 Cavities with 4 Cells and 2 Tapers: ASCR.
20	lep/cavities-sc/ascr	Superconducting RF-Module: 4 Cavities with 4 Cells and 2 Tapers: ASCR.
1	lep/cavities-sc/asnl	Superconducting RF-Module: 4 Cavities with 4 Cells and 2 Tapers: ASNL.
0	lep/cavities-sc/asnr	Superconducting RF-Modules: 4 Cavities with 4 Cells and 2 Tapers: ASNR.
0	lep/pumping-holes/vc	Sum of all Vacuum Chambers: 751392 holes per item.
4	lep/pumping-holes/vcb2a	Chamber VCB2A: 440 holes per item.
1624	lep/pumping-holes/vcba	Chamber VCBA: 440 holes per item.
0	lep/pumping-holes/vcbb	Chamber VCBB: 439 holes per item.
2	lep/pumping-holes/vcbv	Chamber VCBV: 440 holes per item.
1	lep/pumping-holes/vcd2d	Chamber VCD2D: 154 holes per item.
7	lep/pumping-holes/vcd2e	Chamber VCD2E: 186 holes per item.

Figure 4: *Listing of the number of installed items in the* **ZBASE** *data base.*

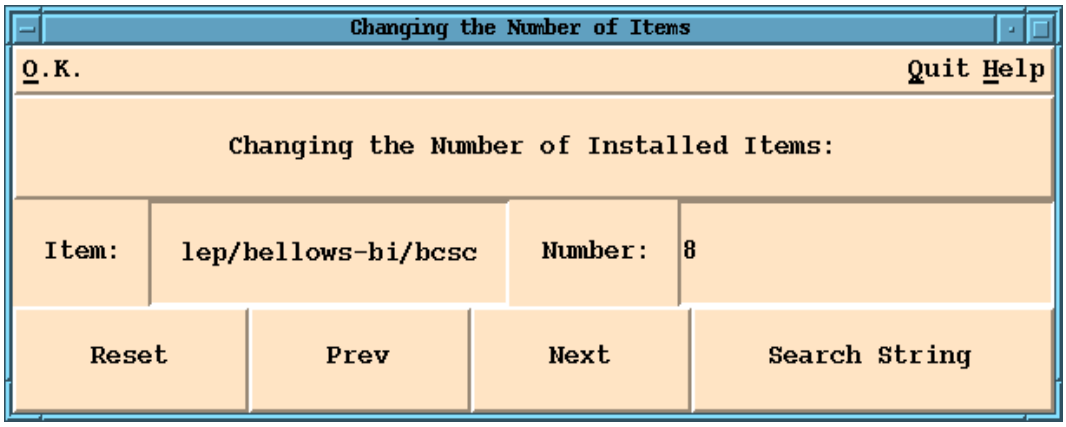


Figure 5: *Graphical user interface for editing the number of installed* **items** *in the machine.*

7.4 Beam

This menu allows the user to display and modify the stored beam parameters in the data base (→ Section 5).

7.4.1 Show Current Parameters

This sub-menu opens a new window which displays the current beam parameters in the program. Pressing the **Enter** key on the keyboard moves the cursor from one entry display to the next. The default values for the beam parameters are stored in the file **.beam** under the **data** directory of the data base and are read during the startup of the **ZBASE** program (→ Section 4). The **Save** button saves the modifications in the **.beam** file (→ Section 4). Without saving the data, all modifications of the beam parameters affect only the entries on the internal working arrays.

7.4.2 'Machine' Parameters

The program has one sub-menu for each **machine** in the data base which displays the default beam parameters for each **machine** (→ page 4). The values in this display can be modified by any user. Pressing the **Enter** key on the keyboard moves the cursor from one entry display to the next. The default values for the beam parameters of a **machine** are stored in the **.beam** file under the **data/machine** directory (→ Section 4). The **Select** button accepts the displayed beam parameters as the current parameters in the data base (→ Section 5) and the **Save** button saves the modifications in the **.beam** file of the corresponding **machine** (→ Section 4). If the displayed parameters were not selected as the current parameters in the data base, all changes are discarded. Fig. 6 shows an example display of the beam parameters for one machine in the data base.

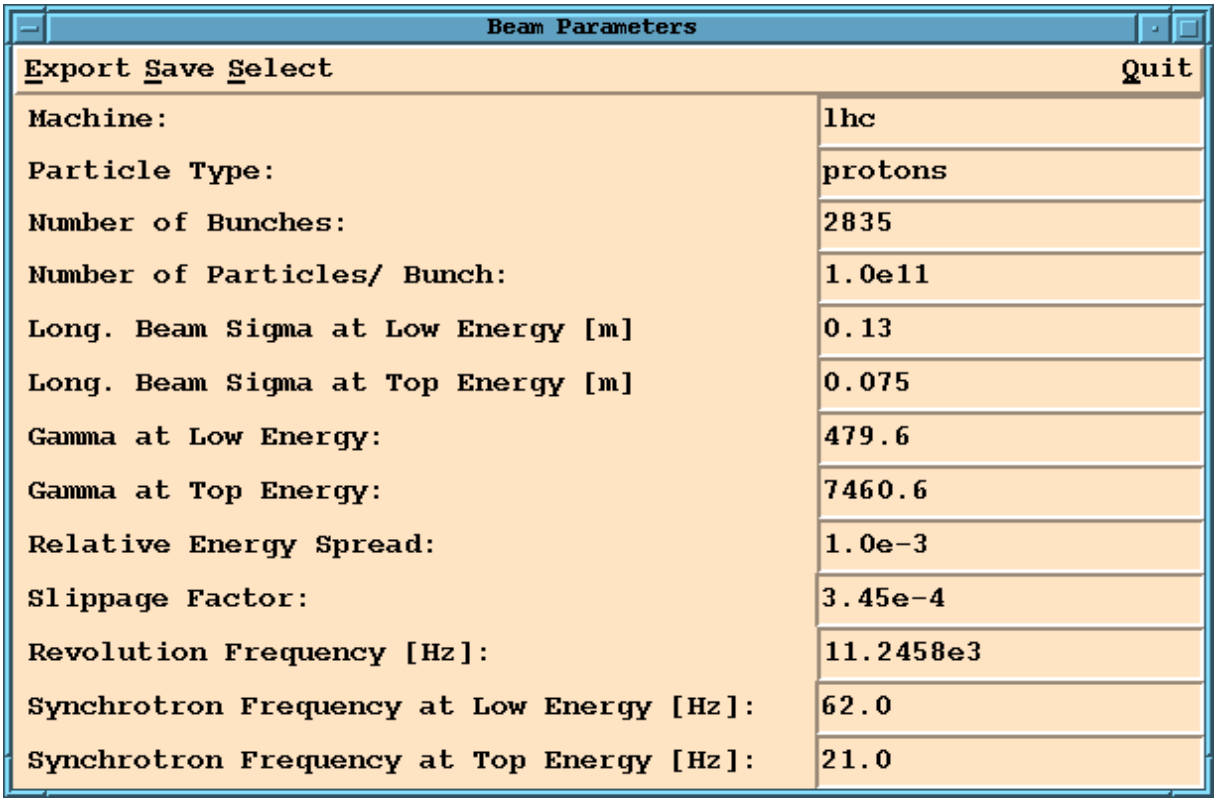


Figure 6: *Listing of the default parameters for one machine in the* **ZBASE** *data base.*

7.5 Optics

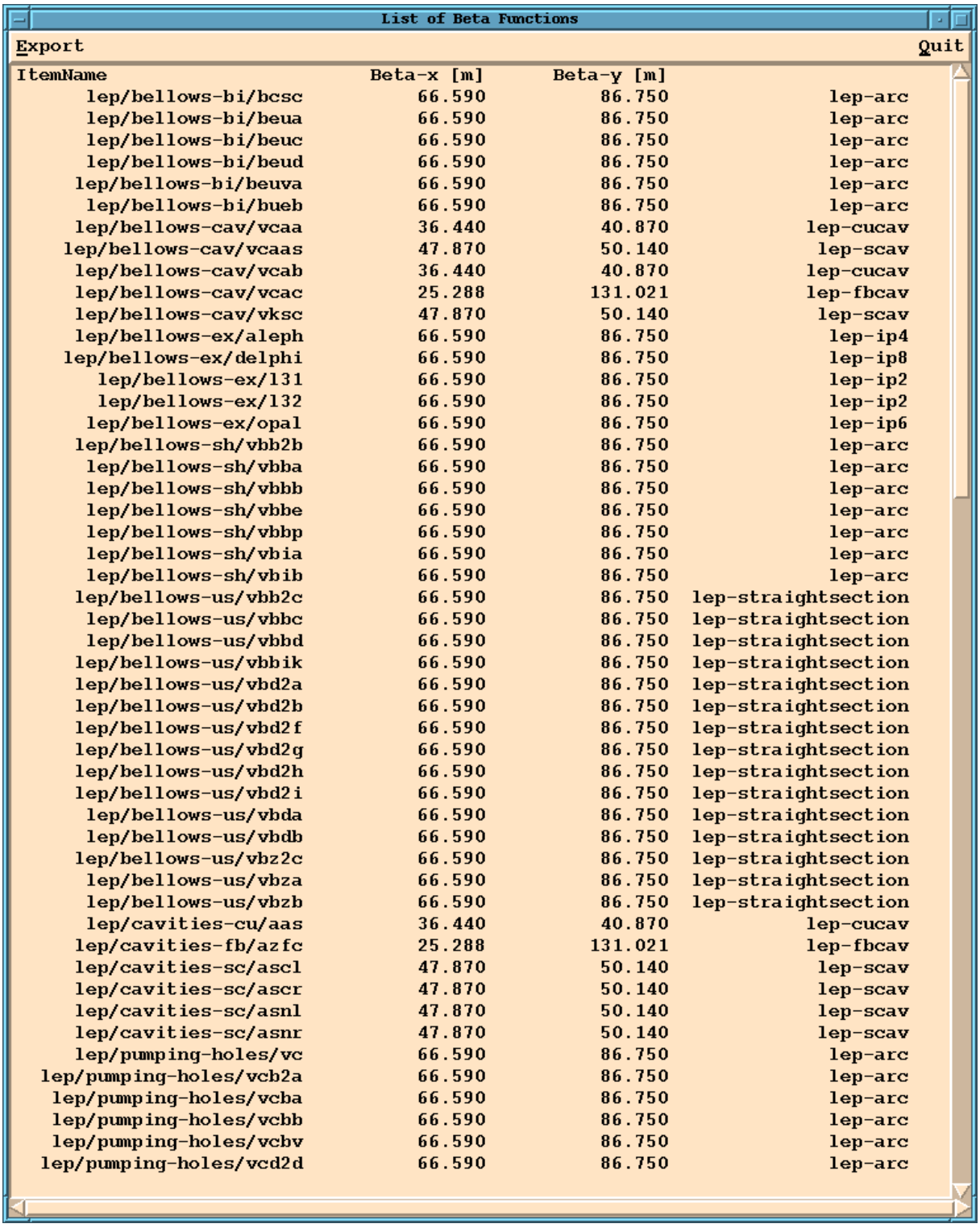
This menu allows the user to display and modify the stored optics parameters at the positions of the impedance **items** in the data base. During the program startup the **ZBASE** program reads the beta function values and **location labels** from the **attributes** file in the data base (→ page 4). After reading the parameters from the **attributes** file they are stored in internal arrays. This menu allows the user to manipulate the entries in these internal arrays.

7.5.1 Show Current Parameters

This sub-menu opens a new window which displays the current optics parameters in the internal arrays of the **ZBASE** program (→ Section 5). The program displays the **item** name, the horizontal and vertical betatron functions and a **location** label which can be used for simultaneously changing the betatron function values of all **items** with the same **location** label. Fig. 7 shows an example of this display.

7.5.2 Reload Parameters from DataBase

This sub-menu reloads the optics parameters for the selected **items** from the **attributes** file (→ page 4) in the data base.



ItemName	Beta-x [m]	Beta-y [m]	
lep/bellows-bi/bcsc	66.590	86.750	lep-arc
lep/bellows-bi/beua	66.590	86.750	lep-arc
lep/bellows-bi/beuc	66.590	86.750	lep-arc
lep/bellows-bi/beud	66.590	86.750	lep-arc
lep/bellows-bi/beuva	66.590	86.750	lep-arc
lep/bellows-bi/bueb	66.590	86.750	lep-arc
lep/bellows-cav/vcaa	36.440	40.870	lep-cucav
lep/bellows-cav/vcaas	47.870	50.140	lep-scav
lep/bellows-cav/vcab	36.440	40.870	lep-cucav
lep/bellows-cav/vcac	25.288	131.021	lep-fbcav
lep/bellows-cav/vksc	47.870	50.140	lep-scav
lep/bellows-ex/aleph	66.590	86.750	lep-ip4
lep/bellows-ex/delphi	66.590	86.750	lep-ip8
lep/bellows-ex/l31	66.590	86.750	lep-ip2
lep/bellows-ex/l32	66.590	86.750	lep-ip2
lep/bellows-ex/opal	66.590	86.750	lep-ip6
lep/bellows-sh/vbb2b	66.590	86.750	lep-arc
lep/bellows-sh/vbba	66.590	86.750	lep-arc
lep/bellows-sh/vbbb	66.590	86.750	lep-arc
lep/bellows-sh/vbbe	66.590	86.750	lep-arc
lep/bellows-sh/vbbp	66.590	86.750	lep-arc
lep/bellows-sh/vbia	66.590	86.750	lep-arc
lep/bellows-sh/vbib	66.590	86.750	lep-arc
lep/bellows-us/vbb2c	66.590	86.750	lep-straightsection
lep/bellows-us/vbbc	66.590	86.750	lep-straightsection
lep/bellows-us/vbbd	66.590	86.750	lep-straightsection
lep/bellows-us/vbbik	66.590	86.750	lep-straightsection
lep/bellows-us/vbd2a	66.590	86.750	lep-straightsection
lep/bellows-us/vbd2b	66.590	86.750	lep-straightsection
lep/bellows-us/vbd2f	66.590	86.750	lep-straightsection
lep/bellows-us/vbd2g	66.590	86.750	lep-straightsection
lep/bellows-us/vbd2h	66.590	86.750	lep-straightsection
lep/bellows-us/vbd2i	66.590	86.750	lep-straightsection
lep/bellows-us/vbda	66.590	86.750	lep-straightsection
lep/bellows-us/vbdb	66.590	86.750	lep-straightsection
lep/bellows-us/vbz2c	66.590	86.750	lep-straightsection
lep/bellows-us/vbza	66.590	86.750	lep-straightsection
lep/bellows-us/vbzb	66.590	86.750	lep-straightsection
lep/cavities-cu/aas	36.440	40.870	lep-cucav
lep/cavities-fb/azfc	25.288	131.021	lep-fbcav
lep/cavities-sc/ascl	47.870	50.140	lep-scav
lep/cavities-sc/ascr	47.870	50.140	lep-scav
lep/cavities-sc/asnl	47.870	50.140	lep-scav
lep/cavities-sc/asnr	47.870	50.140	lep-scav
lep/pumping-holes/vc	66.590	86.750	lep-arc
lep/pumping-holes/vcb2a	66.590	86.750	lep-arc
lep/pumping-holes/vcba	66.590	86.750	lep-arc
lep/pumping-holes/vcbb	66.590	86.750	lep-arc
lep/pumping-holes/vcbv	66.590	86.750	lep-arc
lep/pumping-holes/vcd2d	66.590	86.750	lep-arc

Figure 7: *Listing of the optic parameters in the* **ZBASE** *data base.*

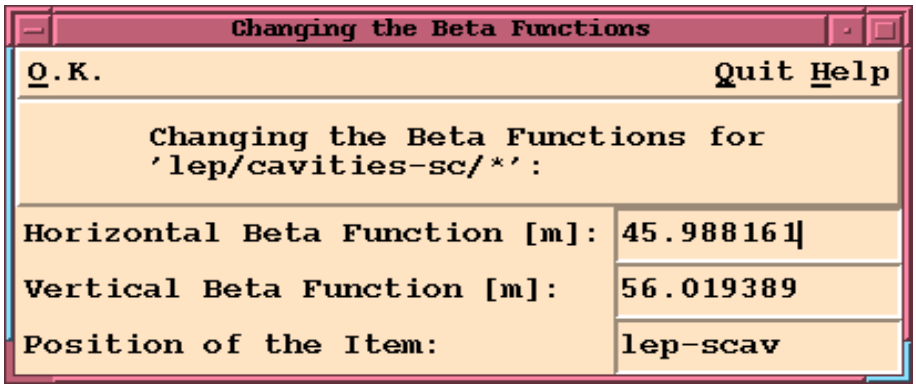


Figure 8: *Graphical user interface for changing the betatron function values at the position of the* **items**.

7.5.3 Get New Beta Function Values

This sub-menu allows the user to get new values for the beta functions from optics calculations. The user can get new values either by using existing optics calculations or by generating a new optics calculation. The menu has a sub-menu for each option:

- **Using existing optics calculations:** Upon selection the menu will open a new window with all machine names in the data base. After selecting one machine, the program displays all optics files in the **.files** directory for the selected **machine** (→ Section 4). Each file has three columns:
  - The first column is an identifier string consisting of the first three and last two characters of the **group** label and the **item** label.
  - The second and third column specify the horizontal and vertical betatron function values respectively in meter (the program expects all data entries in standard **ANSI-C** format).

  After selecting a file, the program reads the data in the selected file and modifies the betatron function values for all selected **items** in the **Selected Items** display. The betatron function values of all other **items** remain unchanged. Depending on the number of selected **items** in the **Selected Items** display this process can take a few minutes.
- **Generating new optics calculations:** This sub-menu opens an editor with a new MAD input file where the user can enter a new optics input file. At the beginning of this input file is one include file for each group in the data base. The include files place a marker at each position of an **item** in the MAD output file. The include files are located in the **optics** directory (→ page 4) of the data base. After closing the editor, the program opens a window where the user is asked if the MAD-run should be started. If the MAD-run is started, the program waits until MAD has finished the optics calculation and reads the new optics values from the **beta.data** output file of the MAD-run. The program only modifies the betatron function values for the selected **items** in the **Selected Items** display and those **items** which have an entry in the **beta.data** output file. The betatron function values of all other **items** remain unchanged. Depending on the number of selected **items** in the **Selected Items** display this process can take a few minutes. (A new feature of the

MAD program allows the calculation of the betatron function values at any given position in the machine. This allows also a direct calculation of the average betatron function values using the **.location**) file entries.)

7.5.4 Change the Beta Functions for a Group

This sub-menu allows the user to modify the betatron function entries of all **items** belonging to one impedance **group** in the data base (→ Section 4).

The menu opens a new window with all **machine** names in the data base. After selecting one **machine**, the program displays all impedance **groups** for the selected **machine** for selection. After selecting one impedance **group** from the display, the program opens a new window with three entry displays:

- The first two entry displays show the horizontal and vertical betatron function values of the first **item** in the selected impedance **group** (→ Section 4).
- The third entry display shows the **position label** for the first **item** in the selected impedance **group** (→ Section 4).

All three entries can be modified by the user. The **Enter** key on the keyboard moves the cursor to the next entry display. Fig. 8 shows the corresponding graphical user interface.

7.5.5 Change the Beta Functions for an Item

This menu allows the user to modify the betatron function entries of one **item** in the data base. The menu opens a new window with all **machine** names in the data base. After selecting one **machine**, the program displays first all impedance **groups** for the selected **machine** for selection and then, after selecting one **group**, all impedance **items** belonging to the selected **group** (→ Section 4). After selecting one impedance **item** from the display, the program opens a new window with three entry displays.

- The first two entry displays shows the horizontal and vertical betatron function values of the first item in the selected impedance group.
- The third entry display shows the position label for the first item in the selected impedance group (→ page 4).

All three entries can be modified by the user. The **Enter** key on the keyboard moves the cursor to the next entry display.

7.5.6 Change the Beta Functions at one Location

This sub-menu allows the user to modify the betatron function entries of all **items** with the same **location label**. The entries of these arrays can be modified with this menu. The menu opens a new window which displays all **location labels** in the array. After selecting one **location label** from the display, the program opens a new window with three entry displays:

- The first two entry displays shows the horizontal and vertical betatron function values of the first **item** in the selected impedance **group**.
- The third entry display shows the **position label** for the first **item** in the selected impedance **group** (→ Section 4).

All three entries can be modified by the user. The **Enter** key on the keyboard moves the cursor to the next entry display.

7.5.7 Read new Parameters from File

This sub-menu allows the user to load new values for the optics parameters from a file. The menu opens a new window where the user can browse though directories and can select a file containing the new values. For each **machine** in the data base, the files are stored in the **.files** sub-directory (→ Section 4). (See Section 7.5.8 on storing the optics parameters in a file.)

7.5.8 Save Current Parameters in File

This sub-menu allows the user to save the current values of the optics parameters in a file. The menu opens a new window where the user can enter a file name relative to his home directory. In addition to the specified name the file will get the extension 'optics.data'. (See Section 7.5.7 on reading optics parameters from a file.)

7.5.9 View the '.location' file entries for an Item

This sub-menu allows the user to look at the entries in the **.location** file of an **item** (→ Section 4).

7.5.10 Edit the '.location' file entries for an Item

This sub-menu allows the user to create or modify the entries in the **.location** file of an **item** (→ page 4). The user must have write permission for the current data base directory. The program creates automatically an entry in the **log-file** of the data base.

7.6 ProcessData

This menu allows the user to process the selected impedance data. The menu offers five different sub-menus.

7.6.1 List Selected Items

This sub-menu displays the number of installed **items** for all selected **items**. The program opens a new window, displaying the number of installed items, the **item** name (**machine/group/item**) and the comment line from the **attributes** file (→ page 4).

7.6.2 Attributes

This sub-menu reads the **attribute** files (→ page 4) of all selected **items** and displays the sum of the longitudinal inductances and the low frequency limit of the imaginary part of the transverse impedances. For the latter, the program weights the contributions of the individual impedance items by the betatron function values at the position of the **items** and divides the total sum by the average betatron function value. Furthermore, the program displays the total number of installed items and the average horizontal and vertical betatron function values.

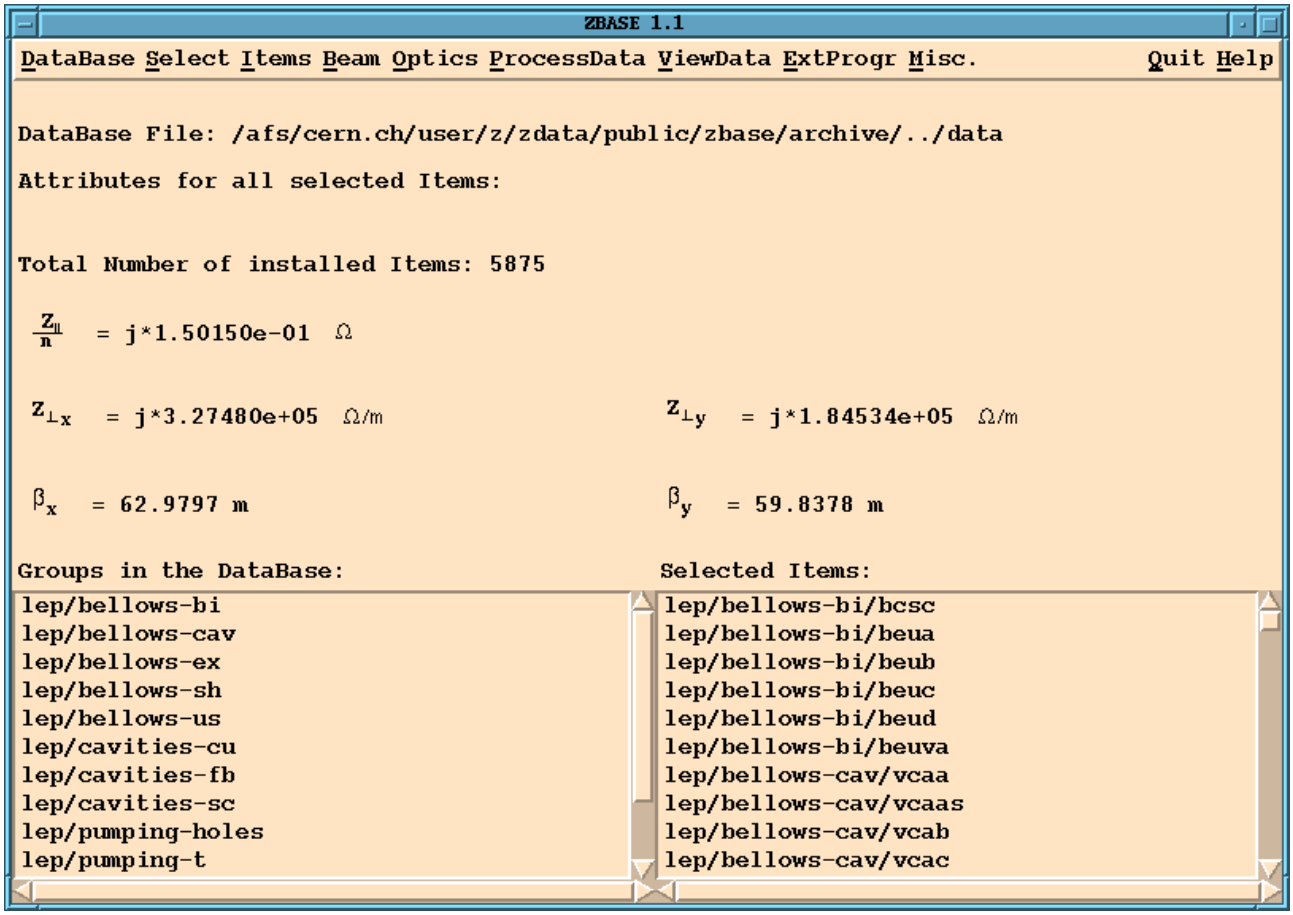


Figure 9: *The graphical user interface for the* **ZBASE** *program showing the net impedance* **attributes** *for all LEP* **items** *but the RF-cavities.*

All data is displayed in the main display area of the graphical user interface (→ Fig. 3). Fig. 9 shows the processed **attributes** of all the LEP impedance **items** but the RF-cavities.

7.6.3 HOM-LONG & MBIRT

This sub-menu reads the **attribute** files (→ page 4) of all selected **items** and displays the sum of the longitudinal higher-order-mode parameters (HOM) in a new window. Furthermore, the sub-menu allows the calculation of the multi-bunch instability rise times (MBIRT). The process is initiated by double clicking with the left mouse button on one of the displayed HOM entries. Upon initialisation, the program reads the **secondary data** for the **selected items** from the internal arrays of the **ZBASE** program and calculates the corresponding interaction matrix for the selected HOM parameters assuming a resonator impedance model [5]. The interaction matrix is calculated for the first 20 longitudinal modes. For each mode, the matrix is calculated considering the first 20 radial modes. The eigenvalues are calculated using standard routines from the EISPAC package [6] of the CERN GENLIB library. The corresponding MBIRT are displayed in a new window. Fig. 10 shows examples for the two display windows. (See also Section 7.6.4.) Double clicking the right mouse button in the display window initiates the calculation of the net inductance resulting from the HOM parameters.

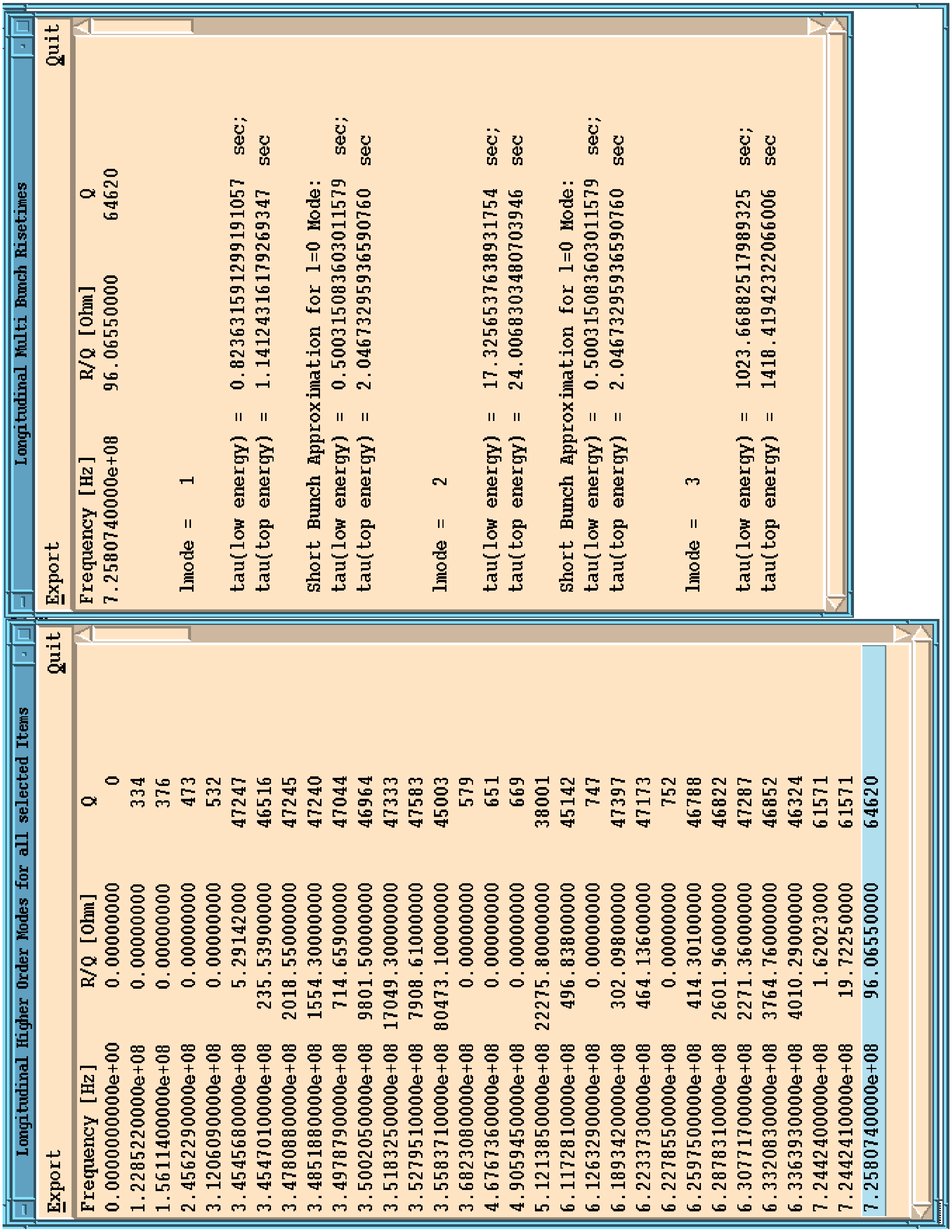


Figure 10: *The two display windows for the* **'HOM-long & MBIRT'** *sub-menu of the* **ProcessData** *menu.*

7.6.4 HOM-TRAN & MBIRT

This sub-menu reads the **attribute** files (→ page 4) of all selected **items** and displays the sum of the transverse higher-order-mode parameters (HOM) in a new window. It offers the same functionality as the **'HOM-long & MBIRT'** sub-menu (→ Section 7.6.3).

7.6.5 LOSS FACTOR & TMCI THRESHOLD

This sub-menu reads the **.loss** files (→ Section 4) of all selected **items** and displays the available bunch length entries in a new window. After selecting a bunch length, the program reads again the **.loss** files (→ Section 4) of all selected **items** and calculates the sum of the loss factors for the selected bunch length. At the end of the calculation, the program opens two new windows:

- One window displays all **items** from the **Selected Items** list which did not have loss factor entries for the selected bunch length.
- The other window displays the total loss factor, the relative contributions of the selected **groups**, the **TMCI** threshold currents, a list of the current beam parameters (→ Section 7.4) and a list of all selected **items** with their average betatron function values. The required **secondary data** for the **TMCI** threshold currents is read from the internal arrays of the **ZBASE** program (→ Section 5).

Fig. 11 shows the two windows for the LEP data.

7.6.6 SUMMARY OF SECONDARY PARAMETERS

This sub-menu displays all **secondary data** in a new display window (→ Section 5).

7.7 ViewData

This menu allows the user to look at the impedance entries of a single **item**. The menu offers seven different sub-menus. Except for the first sub-menu, **Clear**, all sub-menus require the selection of an impedance **item** from the data base. In all cases, the selection process consists of the following procedure: The menu first opens a new window with all **machine** names in the data base. After selecting one **machine**, the program displays all impedance **groups** for the selected **machine** for selection and then all **items** belonging to the selected **group** (See Section 6 on using the mouse). After selecting one **item** from the display window the selected sub-menu is initiated for the selected **item**.

7.7.1 CLEAR

This sub-menu clears the main display area of the graphical user interface of the main program (→ Section 6 and Fig. 3).

7.7.2 LOSS FACTOR

This sub-menu displays all loss factor entries in the **.loss** file (→ Section 4) for the selected impedance **item** in a new display window.

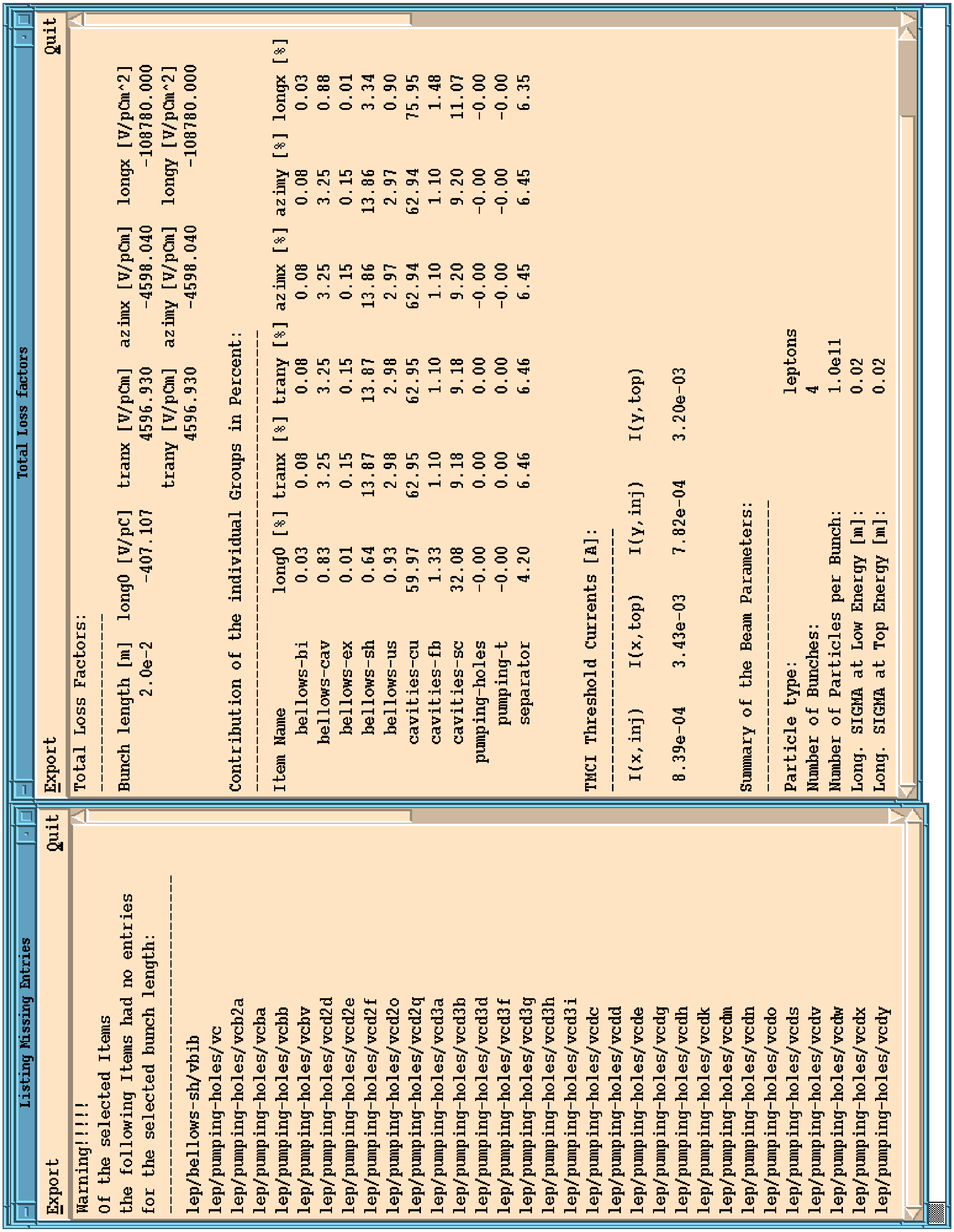


Figure 11: *The two display windows for the* **'LossFactor & TMCI Threshold'** *sub-menu of the* **ProcessData** *menu.*

7.7.3 Wake Potential

This sub-menu displays all wake potential entries in the data base (→ Section 4) for the selected impedance **item** in a new display window. The first column indicates the type of the wake potential and the second column specifies the corresponding bunch length. After selecting a wake potential from the display window the program opens a new window displaying the entries in the wake potential file.

7.7.4 Z-long

This sub-menu displays the longitudinal higher order mode (HOM) parameters of the selected **item** in a new display window.

7.7.5 Z-trans

This sub-menu displays the transverse higher order mode (HOM) parameters of the selected **item** in a new display window.

7.7.6 Attributes

This sub-menu reads the **attribute** files (→ page 4) of the selected **item** and displays the **attributes** in the main display area of the graphical user interface (→ Section 6 and Fig. 3). (See also Section 6 on how to use the mouse in the graphical user interface.)

7.7.7 File

This sub-menu shows a listing of all files in the **long** and **tran** sub-directories of the selected **item** (→ Section 4). After selecting a file from the display the program displays its contents in a new display window. If the selected file is a postscript file, the program displays the file using the ghostview program (see the Unix manual pages for more details on ghostview). Compressed postscript files are automatically un-compressed before being displayed.

7.8 ExtProgr

ZBASE offers graphical user interfaces and example input files to ABCI, URMEL, MAD and MAFIA. (All example input files are stored in the **examples** (→ page 3) sub-directory of the **ZBASE** directory (1).) All interfaces have the same layout. Fig. 12 shows the interface for the ABCI program. Each interface has one display area and one menu bar with seven menu buttons:

- **Submit**
  This menu allows the user to submit batch jobs on the CERN-SP cluster. It has two sub-menus: The **ShowSelection** sub-menu shows the selected options and input files and the **Submit** sub-menu submits the job to the load leveller on the CERN-SP.
- **JobStatus**
  This menu allows the user to inquire the status of batch jobs. It has two sub-menus: The **Status** sub-menu lists information of all running jobs belonging to the user and the **StatusAll** sub-menu lists information of all running jobs on the CERN-SP cluster.

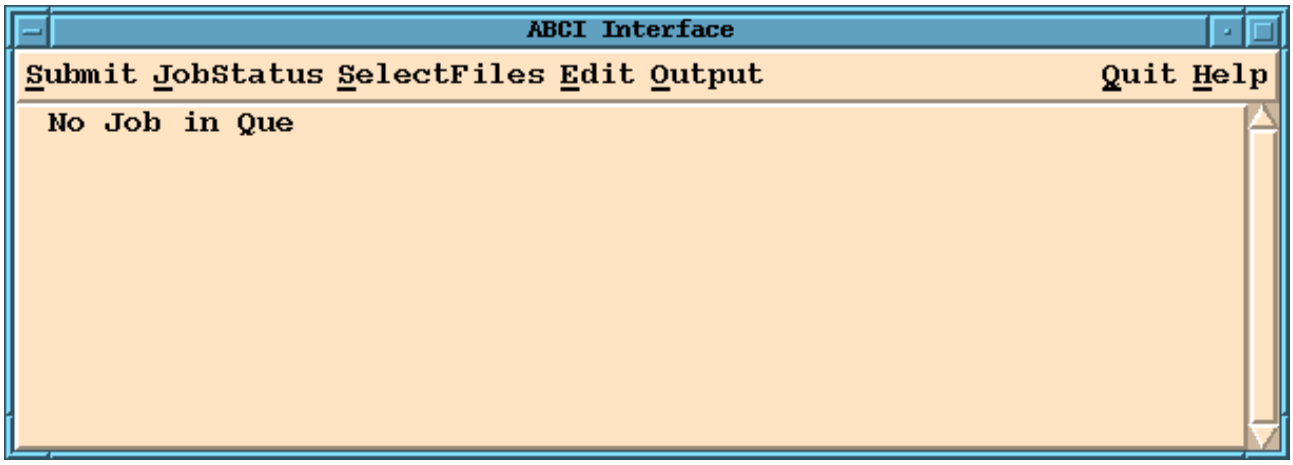


Figure 12: *The graphical user interface for the* **ABCI** *program.*

- **Select**
  This menu allows the user to select input files and job classes for the batch job. The detailed functionality is slightly different for the individual interfaces and we will describe the **Select** menu in more detail in a separate Sub-section for each interface.
- **Edit**
  This menu button allows the user to edit the selected input files before submitting the batch-job. For example, the user can select an input file from the data base and modify some parameters before submitting the job on the CERN-SP.
- **Output**
  This menu button allows the user to look at the program output and to remove directories of the batch jobs. Each batch job has a separate directory where the job output is located. The directory names are of the form 'program-name-run-number', where 'program-name' stands for 'abci', 'urmel', 'mafia' and 'mad, and 'number' is a run number index which distinguishes different batch jobs. Depending on the selected options in the **Select** menu, the output files are in the users **tmp** or **PARC** scratch directory (→ Section 3).
  - The **View Output** sub-menu first displays all batch job directories in the display window of the interface. After selecting a file from the display, the program will display the contents of the selected file in a new display window. Postscript files are displayed with the ghostview program.
  - The **Remove a Directory** sub-menu allows the user to remove one of the job directories. Upon selection the program first displays all job directories in the display window of the interface. After selecting a directory from the display, the program asks the user to confirm the removal of the directory before deleting the directory.
- **Quit**
  This menu button closes the interface.
- **Help**
  The **Help** displays related help information.

In the following we will describe the individual functionalities of the **Select** menu for each interface in a separate Sub-section.

7.8.1 ABCI

The **Select** menu allows the user to select an input file and the job-class for the batch-job submission.

- The **SelectInput** sub-menu first shows a selection of different sources for the input file. The source directories are displayed in the display area of the interface (→ Fig. 12). The first lines show the **machine** labels in the data base. One line offers the selection of '**All selected items in the data base**', one line shows the **examples** directory of the **ZBASE** data base, one the users home directory and one the users **tmp** directory (→ Section 3). By double clicking with the left mouse button on one of the lines in the display, the user can browse through the listed directories and select an input file.

  By selecting the '**All selected items in the data base**', the program will submit two batch-jobs for each **selected item** in the data base (one for the longitudinal and one for the transverse impedance). If there are more then one ABCI input file per **item** in the data base, the program will choose the first file in the directory. Upon selection, the program opens a new window where the user can modify some of the parameters of the input files. The new window has three entry displays, two check-buttons and one **O.K.** button:

  - the first entry display allows the user to enter a new bunch length for the ABCI run.
  - the second entry display allows the user to modify the longitudinal mesh size
  - the third entry display allows the user to change the length over which the ABCI program calculates the wake potential
  - the first check-button determines whether the program will automatically add the calculated loss factors in the data base (the user needs write permissions on the data base directory for this option)
  - the second check-button determines whether the program will automatically add the calculated wake potentials in the data base (the user needs write permissions on the data base directory for this option)
  - the **O.K.** button accepts the entries and closes the window. However, the jobs are not yet submitted and the user still has to submit the batch-jobs using the **Submit** menu of the ABCI interface. After closing the window, the program displays the selected options in the display area of the ABCI interface.
- The **SelectJob** sub-menu allows the user to select different job-classes for the batch-job. Each job-class is listed with the corresponding cpu-time limit. All job-classes which start with an 'e' are reserved for users with a **PARC** account and have smaller ques than the normal job-classes.

7.8.2 URMEL

The **Select** menu allows the user to select an input file and the job-class for the batch-job submission.

- The **SelectInput** sub-menu first shows a selection of different sources for the input file. The source directories are displayed in the display area of the interface (→ Fig. 12). The first lines show the **machine** labels in the data base. One line shows the **examples**

directory of the **ZBASE** data base, one the users home directory and one the users **tmp** directory (→ Section 3). By double clicking with the left mouse button on one of the lines in the display, the user can browse through the listed directories and select an input file.

- The **SelectJob** sub-menu allows the user to select different job-classes for the batch-job. Each job-class is listed with the corresponding cpu-time limit. All job-classes which start with an 'e' are reserved for users with a **PARC** account and have smaller ques than the normal job-classes.

7.8.3 MAFIA

The **SelectFiles** menu allows the user to select input files and job parameters for the MAFIA run. The **SelectFiles** menu has seven sub-menus:

- The **SelectStep** sub-menu allows the user to select different steps in the MAFIA run. MAFIA is a modular program which uses different modules and input files for the different steps of a calculation. For example, calculating the longitudinal loss factor in a structure consists of three individual steps. First, the mesh generator creates a mesh for the structure. Second, a time domain module solves Maxwell's equations on the mesh, and finally, in a third step, a post processor calculates the loss factor from the wake potentials. Each step in the MAFIA calculation requires its own input file and parameters. The **SelectStep** sub-menu selects one step in the MAFIA calculation (the default is 1). After that selection, all additional selections refer to this step in the MAFIA calculations.
- The **SelectModule** sub-menu allows the user to select different modules for each step in the MAFIA run.
- The **SelectInputFile** sub-menu allows the selection of an input file. It first shows a selection of different sources for the input file. The source directories are displayed in the display area of the interface (→ Fig. 12). The first lines show the **machine** labels in the data base. One line shows the **examples** directory of the **ZBASE** data base, one the users home directory, one the users **tmp** directory and one the users **PARC** scratch directory (→ Section 3).
- The **SelectProfFile** sub-menu allows the selection of a MAFIA profile file. It first shows a selection of different sources for the input file. The source directories are displayed in the display area of the interface (→ Fig. 12). The first lines show the **machine** labels in the data base. One line shows the **examples** directory of the **ZBASE** data base, one the users home directory, one the users **tmp** directory and one the users **PARC** scratch directory (→ Section 3).
- The **SelectDiracFile** sub-menu allows the selection of a MAFIA dirac file. It first lists all the **machines** in the **ZBASE** data base, the users home directory, the users **tmp** directory and the users **PARC** scratch directory (→ Section 3). The user can browse through the displayed directories and select a dirac file for the batch-job. If the mesh generator is selected for the first step of the MAFIA calculation, the program assumes that the dirac file will be created by the mesh generator and no selection is necessary.
- The **SelectMemory** sub-menu allows the user to select different memory values for the different steps in the MAFIA run. The selection will be interpreted by the dynamic memory allocation procedure of the MAFIA modules.

- The **SelectBigOption** sub-menu allows the user to choose whether the output data will be stored in the '$HOME/tmp' or 'project/parc/scratch/$GROUP/$USER' directory. The 'project/parc/scratch/$GROUP/$USER' has up to 2 GByte storage space, but all data will be automatically deleted after five days.
- The **SelectJob** sub-menu allows the user to select the job-class for the MAFIA run. Here, all listed job-classes are **PARC** classes.

7.8.4 MAD

The **Select** menu allows the user to select an input file and the job-class for the batch-job submission. It has the same functionality as in the URMEL interface (→ Section 7.8.2).

7.9 Misc.

The **Misc.** menu allows the user to select a new printer or editor for the manipulation of input files.

7.10 Quit

The **Quit** menu exits the **ZBASE** program. When exiting the program, it will automatically delete all scratch directories that were created during the run time (→ Section 3).

7.11 Help

The **Help** menu offers information on the concept of the **ZBASE** program and the organisation of the data in the data base, the main menu bar of the graphical user interface and the functionality of the mouse in the user interface.

8 Summary

The main aim of the **ZBASE** impedance data base is to provide the impedance data of one or more machines in a standard format and in a central place which is accessible to everybody. The program is written in the Tcl-Tk scripting language to provide portability between different Unix platforms and to facilitate the installation of additional procedures which evaluate the impact of the machine impedance on the beam dynamics. For example, the current version of the program offers estimates for the transverse mode coupling instability threshold current and multi-bunch instability rise times. Additional procedures can be easily added to the program at any stage.

Furthermore, the program offers interfaces to the programs that were used for calculating the impedance data. For example, the program contains interfaces to the **ABCI**, **URMEL** and **MAFIA** programs. In addition to the impedance data, the data base stores the input and some of the output files of these programs. Collecting not only the impedance data but also the input files for the programs that were used for calculating the impedance data, greatly facilitates the recalculation of the impedance data if the parameters change. For example, for the LHC, most of the impedance items are still in the design process and feedback from

the impedance calculations is desirable at this stage. A central data base that allows a semi-automatic recalculation of the impedance data once the geometry of an item changes will facilitate such a feedback.

The current data base contains impedance data for LEP and LHC. However, the data base could easily be extended to other machines like the SPS for example. The impedance data consists of the longitudinal inductance, the horizontal and vertical low frequency limits of the imaginary impedance, loss factors and wake potentials. Therefore, an analysis of the impact of the impedance on the beam dynamics is currently limited to broad-band and narrow-band approximations for the impedance. However, for a future version of the data base it is planned to store the impedance as a function of frequency, allowing the modelling of any type of impedance in the machine.

Acknowledgements

The author would like to thank F. Ruggiero for initiating this project and for many helpful and stimulating discussions. Furthermore I would like to thank M. Böge for his help and continuous support with the programming.

References

[1] John K. Ousterhout, 'Tcl and the Tk Toolkit' Addison-Wesley Professional Computing Series (1994).

[2] Y. H. Chin 'User's Guide for ABCI Version 8.7', CERN SL/94-02 (AP).

[3] C. Palm, U. van Rienen, T. Weiland 'URMEL and URMEL-T USER GUIDE', DESY M-85-11, December 1985

[4] R. Klatt, F. Krawczyk, W.R. Novender, C. Palm, T. Weiland (DESY), B. Steffen (Julich,Forschungszentrum), T. Barts, M.J. Browman, R. Cooper, C.T. Mottershead, G. Rodenz, S.G. Wipf 'MAFIA - A THREE-DIMENSIONAL ELECTROMAGNETIC CAD SYSTEM FOR MAGNETS, RF STRUCTURES, AND TRANSIENT WAKE FIELD CALCULATIONS', DESY M-86-07, August 1986

[5] Alexander Wu Chao, 'Physics of Collective Beam Instabilities in High Energy Accelerators' Wiley Series in Beam Physics and Accelerator Technology (1993)

[6] See CERN GENLIB.