THE APPLICATION Mk4Tcl FOR WORK WITH DATA STRUCTURES

Boneva Ani[1], Ivanova Veronika[2] and Boneva Yordanka[1]

[1] *Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, Bulgaria,*
Sofia 1113, Acad. G. Bonchev str., bl.2,
a.boneva@isdip.bas.bg, *yordanka.boneva@hsi.iccs.bas.bg*

[2]*Institute of Robotics - Bulgarian Academy of Sciences, Bulgaria,*
Sofia 1113, Acad. G. Bonchev str., bl.1,
iwanowa.w@abv.bg

Abstract
The article discusses an application for working and processing databases - MetaKit for Tcl (Mk4Tcl). It is a standalone application for the Tcl language and has the ability to work in operating environments Windows, Linux and Unix. One of the advantages of this application is that it does not require a special installation, but is "loaded" into the program code and then compiled with the entire program. The article gives the basic rules for working with the package and some basic commands. Some applications using Mk4Tcl are also considered.

Keywords**:** *Mk4Tcl, Tcl/Tk. Control, Software Package. Computer Program*

1. Introduction

Increasingly, in application programs, as well as in programs for management and monitoring of external devices, there is a need to manage large data sets. To work with this data, either standard variables (arrays, lists, etc.) must be used, or they must be transferred to one of the known packages for working with databases [1]. Both options are cumbersome and disrupt the connection with the control program. The article offers a solution using a built-in module for working with databases. This approach is very efficient and flexible and allows for complex software solutions.

In this case, we consider an extension to the language Tcl (MetaKit for Tcl) and intended for use by scripts written in this language.

MetaKit [2, 3, 4, 5] is a multi-platform dynamic database library that allows the operation and management of memory for mid-range databases (several hundred Mb). Data structures can be defined, resolved, and managed using data integrity at all times. Data files are available across all platforms (UNIX, Windows, Mac, VMS, DOS, etc.).

In data model terminology, MetaKit for Tcl, the mean of RDBMS and OODBMS [4].

MetaKit for Tcl (Mk4Tcl) is a self-contained extension for Tcl 8.x for use by Tcl language scripts. Compiled into a mk4tcl.dll file, it takes up only 184K and can be embedded and used in any Tcl script.

Like any other database library, MetaKit manages the data stored on disk. Of course, there are some things that are different from many other database libraries and provide unique benefits for connecting to scripting languages such as Tcl. To use the application, it is enough for the command [1, 3] to be present in the respective script [1, 3]:

load Mk4Tcl.dll
or
package requireMk4Tcl.

That is all it takes: you do not need to install or set up a special database (somewhere in the system). The extension brings everything it needs.

Some of the main advantages of this application are:

- **easy to learn** - Unlike "standard" database concepts, a programmer does not need to be specifically profiled to work with databases (to use MetaKit). In practice, the application supports almost the same complex queries as SQL;
- **extensibility and dynamism** - allows easy monitoring of the structure of the data file and allows the addition of fields with additional variables;
- **easy to install** - MetaKit has the ability to work on various platforms, including UNIX, Windows, Mac, Linux and Solaris. As mentioned above, it is not necessary to install a special application for working with databases, in addition, the MetaKit library connects or dynamically loads in the respective application. The library itself is small enough - 184 K.

2. Basic concepts in Mk4 for Tcl

There are several concepts you need to know to work with MetaKit. Most of them are used in other mechanisms for working with databases, but under different names [1, 3, 4]:

- **Data file** - MetaKit stores all data in one or more files on disk. They are supported like all other files in the file system. They can be opened with MetaKit commands when access to the data in them is needed and then closed. When a file is opened, a "tag" must be specified, which is associated with the open file (or in other systems, called a "descriptor"). In this case, the following commands are used:

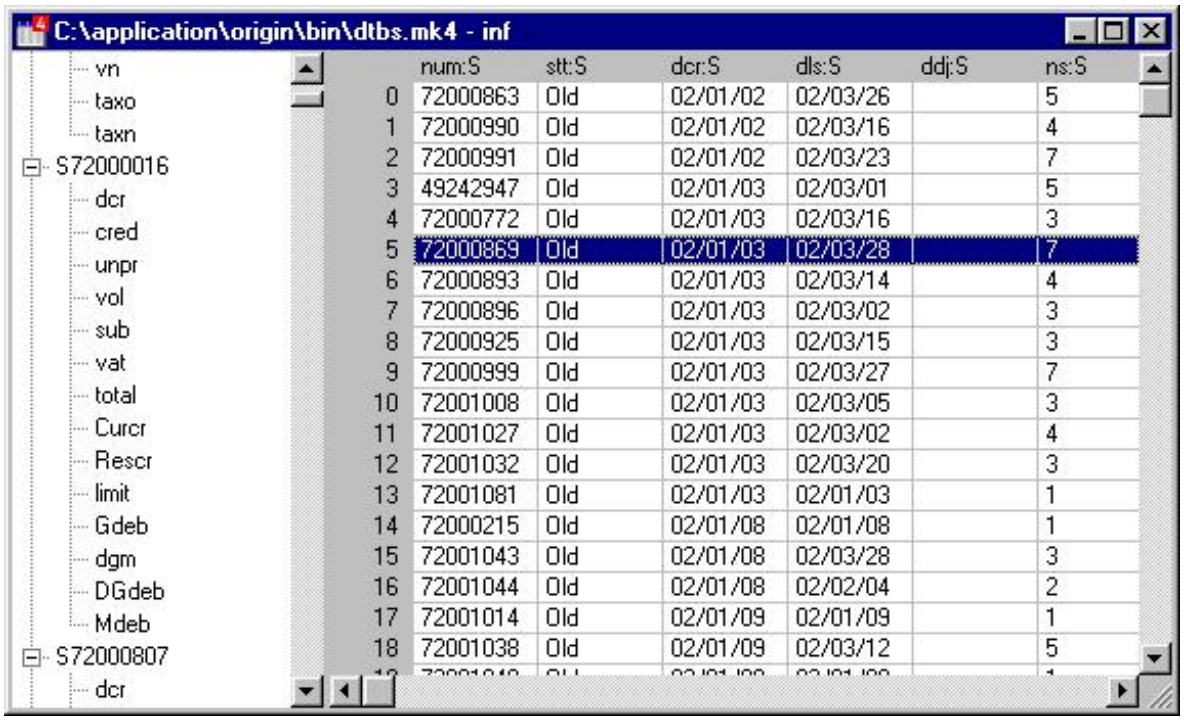
mk::file open *tag filename*
mk::file close *tag*

Several files can be opened at the same time, each with a different "tag".

- **View** (**Table**) - A table is a means of allocating a data file in one or more areas, each of which can support different types of data (symbolic, integer, real, floating, binary, and double-precision). The description of what data a particular table supports is called a structure. The table is specified using a descriptor.table (tag.viewname). The term "view" is equivalent to the term "table" in many other databases.
- **Row** – the row maintains the data associated with the same object. In other databases, this term is equivalent to the term "record". Or a row is a data record. The table is an array of rows, the first row is marked with the number 0. Access to a single row is done using the descriptor.table! Index (tag.viewname! Index).
- **Properties** - properties are described individually to each of the data. Each row can support data with different properties. Simply put, this is the programmatic description of the data in the single-line fields. If the names and telephone numbers of a group of subscribers are recorded, a line is formed for each subscriber, which contains two fields - name and telephone number. Denoted by "name" "tel", for each row, the properties of the variables are described.

In Fig. 1. a general view of a table containing subtables and variables is shown. A list of subtablets and the variables (fields) they contain is displayed on the left.

The content of the respective variables in the subtable (by rows) is displayed in the right part. This screen is implemented using a program designed specifically to monitor the status and content of data structures implemented and maintained with Mk4Tcl.



	num:S	stt:S	dcr:S	dls:S	ddj:S	ns:S
0	72000863	Old	02/01/02	02/03/26		5
1	72000990	Old	02/01/02	02/03/16		4
2	72000991	Old	02/01/02	02/03/23		7
3	49242947	Old	02/01/03	02/03/01		5
4	72000772	Old	02/01/03	02/03/16		3
5	72000869	Old	02/01/03	02/03/28		7
6	72000893	Old	02/01/03	02/03/14		4
7	72000896	Old	02/01/03	02/03/02		3
8	72000925	Old	02/01/03	02/03/15		3
9	72000999	Old	02/01/03	02/03/27		7
10	72001008	Old	02/01/03	02/03/05		3
11	72001027	Old	02/01/03	02/03/02		4
12	72001032	Old	02/01/03	02/03/20		3
13	72001081	Old	02/01/03	02/01/03		1
14	72000215	Old	02/01/08	02/01/08		1
15	72001043	Old	02/01/08	02/03/28		3
16	72001044	Old	02/01/08	02/02/04		2
17	72001014	Old	02/01/09	02/01/09		1
18	72001038	Old	02/01/09	02/03/12		5

Fig. 1. General view of the table in Mk4Tcl

3. Basic command types in MK4Tcl

The commands in MetaKit can be grouped into several main groups [5]:

- **Commands for working with files** – These are commands of the type mk :: file and are used to open and close the data file. In addition, they are used to confirm recent changes, discard recent changes, and accept (send) a file on a channel, including a socket;
- **Commands for working with tables** – These are commands of the type mk :: view and are used to query or change the structure of the table or size and in the data file;
- **Commands for work by position** - These are commands of the type mk :: cursor and are used to determine the order in which you are currently working. This is an efficient way to determine the position of a field in the data structure;
- **Commands for working in rows** - These are commands of the type mk :: row r and are used to create, delete, insert or add one or more rows to the table. Here is the place to remind that the row contains records with data for one object and all rows from a certain table have the same structure (the properties of the variables are the same, but not their values);
- **Commands to access a specific cell** – These are the commands of the type mk :: get and mk :: set and are used to write (read) a specific field in the table, specified as parameters in the command. Depending on the parameters in the command, it is possible to control the reading (writing) of more than one cell at a time.
- **Selection and sort commands** – These are the mk :: select commands and are used to execute queries for sorting, searching or selecting depending on the parameters specified in the command. The result is a list of line numbers that meet a certain condition (depending on the command);
- **Additional commands** – these are commands that allow you to organize loops in rows and define channels for reading (writing) fields, rows or tables.

All commands (described above) are available for execution in programs written in Tcl / Tk [6].

The idea of using standalone applications and libraries (in high-level languages) is not new and we are constantly working on the possibility of embedding and batch processing the software. As mentioned (above), this is a way to solve a wide range of software applications, without strong dependence on the operating system used and without serious requirements for system installation.

Internationally, this approach is known as "Scripted Documents" and is used in a wide range of areas such as:

- process control;
- information systems;
- systems for accounting and control of energy, gas consumption and others;
- systems for security and access control, etc.

4. Applications

With the considered extension to the Tcl language (MetaKit for Tcl) various applications have been made in the field of surgery, space research, energy and others.

The following sections will describe some of the developed and tested applications.

4.1. Using of Mk4Tcl application in program resources for design of laparoscopic application to robots

It is very important for surgeons to be able to touch and feel the tissue/organs/ stones while operating since the sense of touch is one of the primary sources of information that guides the surgeon during surgery. It was designed and produced a model of instrument for laparoscopic surgery where two force sensors was incorporated.

This sensors measurement the interaction between instrument tip and organs/tumors/tissues/stones and returns information to the operator' fingers Computer program is designed to control of four laparoscopic instruments which can work together or individually. It is using Mk4Tcl application to Tcl language for development of databases.and Software (program resources) consists of various commands for manipulation of the instrument (insertion and retraction of the tool, start and stop machine) with contact surfaces, and date obtained from the experimental module which is used to find the difference between previous measuring and received information in real time too.

Another signification advantage of the proposed program solution is the graphical visualization of the measuring and comparing the results. Therefore the surgeon can submit the adequate command to force interaction between the instrument and tissue. The initial dialog box is activated after starting of command file. After start up, a dialog box appears on the screen Fig 2, [7, 8, 9].

On the left side are situated some graphical tools. Some of basic program functions are Commands for Motion - Start and Stop machine, command for insertion and retraction linear of the tools, Mode-Automatic and manual, current step positions of the motor, save in samples or save in results, visualization and comparison of the measuring and etc

Four individually selectable buttons for working of the laparoscopic instruments are included. The first point is to be selected which instrument has to work . Every one instrument is connected to a radio-button, as a specific correspondence between the type of instrument and a number of radio–button is indicated.

The interval for scan of both sensors in milliseconds is set up in the field Scan Time in ms. This interval determines the time between two consecutive reports, by embedded ADC into the microcontroller, for each of them its channels, connected to this sensors-ADC0 and ADC1. The information is stored in corresponding arrays, with a length specified by the Step Time in Scan Periods parameter, which field can be updated by the operator if desired.

The last field specifies the interval between two steps of steps of the stepper motor, expressed in scan periods of the sensors.

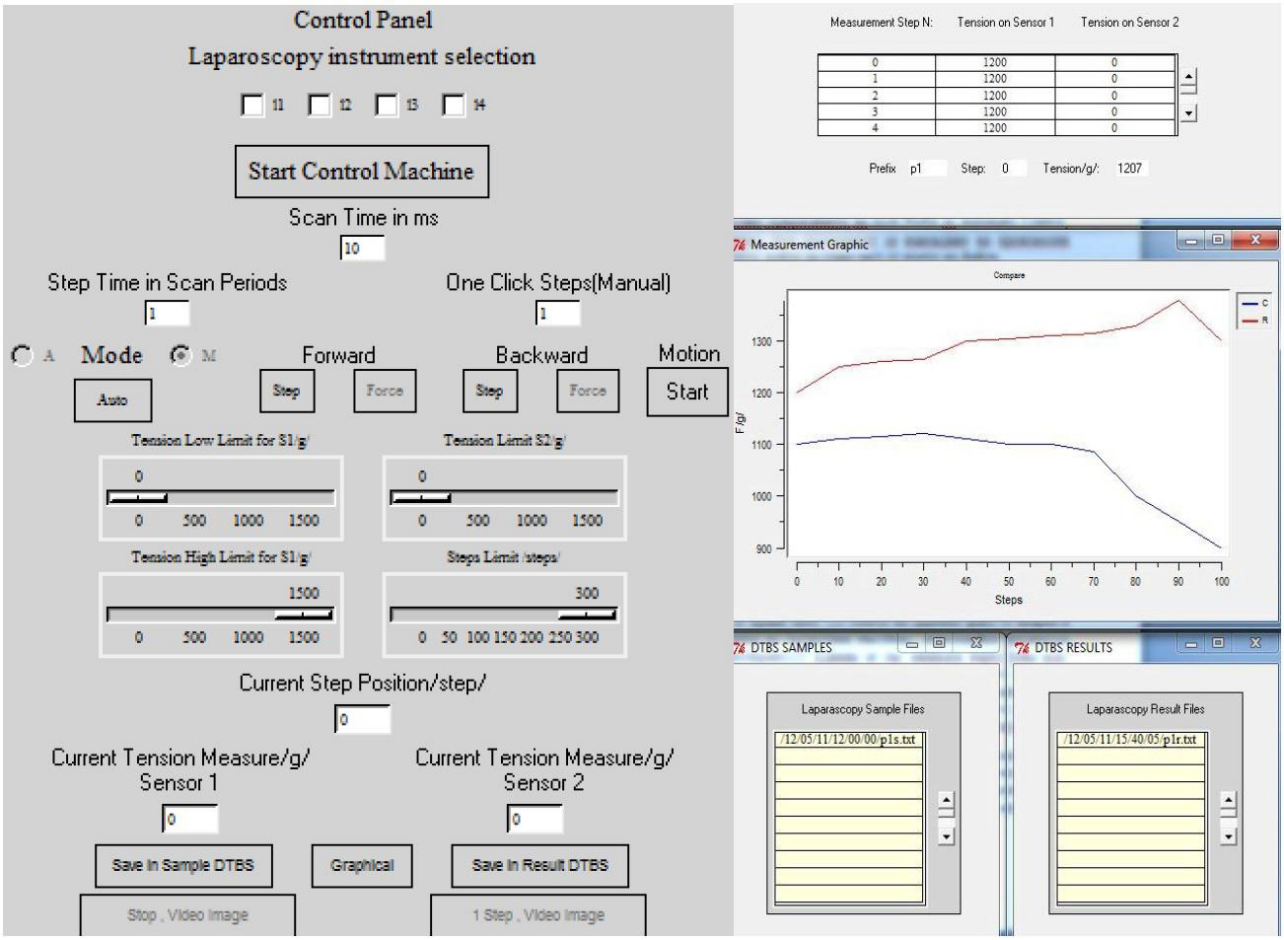


Fig. 2. An initial dialog box

Set parameters of the examined area of the object. It is the area, in which the measurements generated by accomplished steps are made. The data from these measurements are displayed as text in a table (Automatics Control) and graphical (in Dynamic Measurement Graph), at the time of receive in an operating station. The parameters of the examined area of the object are set with the sliders of the four objects of scale type: Tension Low Limit for S1 g defines the beginning of the area (where the value of sensor S1 is overall the Tension Low Limit defined by slider for S1 g). After that point, the measurements from the next steps are displayed in a table (Automatics Control and Dynamic Measurement Graph). The step at which this event happened, is defined as 0 step, and next steps increase by one incrementally.

Mode is basic function with two possibilities -Auto mode and Manual mode. It allows the operation with pair of identical buttons- the first marked as Backward, the second as Forward. Selecting ,,Manual,, allows only the work with those buttons, marked with Step –one for everyone direction. Selecting

Information from measurements of the two sensors and the number of the current step performed by the linear actuator is displayed in the fields Current Tension Measure- Sensor1, Current Tension Measure- Sensor 2 and Current Step Position during the execution of each moving.

To save information from the examined area into a file included in one of the two databases, it must click on one of the buttons - Save in Sample DTBS or Save in Result DTBS. After the saving is complete, the text box header corresponding to that database will contain a number incremented by 1. The operator can display the names of these files and select a file for graphical visualization with the text box slide. It is possible to select one or two files, 1 from each database. Selecting is

done by clicking on the element, including the name of the file. The element includes the name of the file is colored in grey.

On the right side are instruments for visualization and set up of the work of the tools in the process of measuring and date analyzing. DTBS Samples и DTBS Results are Graphical tools that provide the operator access to the files stored in the two databases for eventual visualization. They have the same organization and ways of working. Each one includes file names which are supported by the appropriate base at the current time, a sheet for locating a visible part of the list, and methods for selecting and positioning them in the lists, using several embedded program buttons [8].

Mk4Tcl extension to Tcl - was used in the development of the application database

The user or physician can perform graphical processing and analysis of the research results by Measurement Graphic too.

4.2. Using Mk4Tcl in the development of Software Package for Accounting, Processing and Management of Energy Consumption

The information system for data reading and management of energy consumption is an important tool for building of accounting and control systems. Using the electronic energy meters increases the reliability and accuracy of the user accounted data. HHU reading of the electronic energy meters and PC processing with present software package represent a base for full automation of the accounting process [10].

The software package has modular structure. The modules are structured as bellow:

- *Module for configuration of data collector terminal* – this operation is compulsory when the terminal is started for the first time. It creates in the default directory a text file, containing a sequence of operators for the terminal configuration;
- *Module for creating group of electronic meters* – it creates file for a group of electronic meters (which will be read and processed together). Using special designed form, operator need to enter group name and data field, which will be observed;
- *Module for initial loading of electronic meters group into the data collector terminal* – the terminal is connected to the active port of the computer and writes data in the terminal;
- *Module for terminal data reading* - reading of collected data from terminal to the PC;
- *Module for data base queries* - it operates with collected data from terminal and executes data base queries (searching, sorting, viewing);
- *Module for printing of the results of the queries* - this module supports a lot of types of printing forms.

The program modules are written on Tcl/Tk, working under Windows and use MK4Tcl database (DTBS).

Figure 3 shows the main menu of the program and some of the falling menus [10].

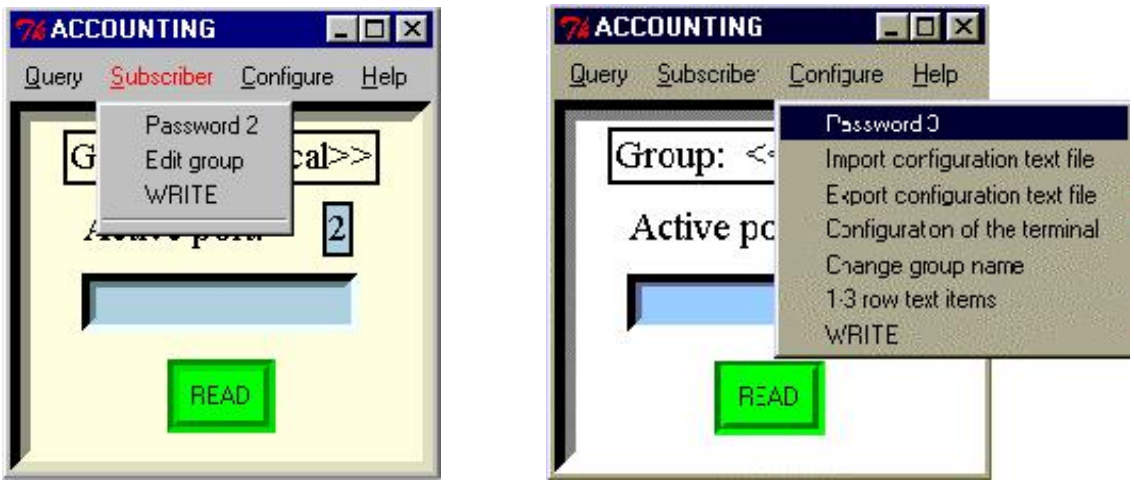


Fig.3 MAIN MENU and pull-down menus of the program

Each terminal can store data for up to 2000 electronic meter [10]. The data fields for each electronic meter consists: meter number, consumer number, accounted data, and data for 4 consumption tariffs.

The tariff data in to DTBS is structured in the following fields: new reading, old reading, difference value (between new and old reading) and summarized value.

Other set of measured data includes the current status of the collector terminal. This data is saved in to history archive file into the PC. Both the accounted data (from DTBS) and archive file are accessible for the computer.

More detailed information on the software can be found in [10]

4.3. Information system for primary pre-processing, intended for use with Bulgarian devices LP and DP, for operation of the International Space Station - ISS (International Space Station).

The participation of Bulgaria in the project includes the development and conduction of scientific research with four devices that work in open space – 2 devices from type DP for measurement of the disturbance of the electromagnetic field in the space near the station and two devices LP with the purpose to research the parameters of low-temperature plasma by using the method of Langmuir [7, 11, 12]. These devices, together with other specialized devices are installed in two containers, fixed by masts located on different distances from the board of ISS. Each container represents separate automatic measurement complex that conducts various physical measurements under the same conditions and same time.

The two containers are part of a local Ethernet network together with a specialized BSTM computer and a client computer on the board of the station. BSTM supports a database of scientific measurements on portable disk [13].

The information processing software was developed on the basis of the Tcl / Tk scripting language, and the database was developed using the Mk4Tcl application.

The software package includes a set of software tools that allow the operator to select, search, visualize, modify and graphically process the data for an experiment [11, 12].

In Fig. 4 and 5 are show the control screen and the table for tabular presentation of the data. There is also a screen for graphical presentation of the information from the measurements accumulated in the database [11].

The structure of Science Results Data Base supposes abilities for parallel processing of the incoming data areas.

Each “experiment_file” of measurements data of concrete instrument unit (LP1, LP2, DP1, DP2) is placed (into the DTBS folder) corresponding to the time of experiment starting. The file name includes the type of experiment processed with the unit.

There are two different base folders – one (LP1dtbs) for LP1 and one (LP2dtbs) for LP2. Each of them has tree hierarchy organization with 6 hierarchy levels (corresponded to year, month, day, hour, minute and seconds). By this way the file places into Database are sorted naturally. The lowest level (second) gives name of a folder, including corresponded “experiment_file” and empty “number_file“. The name of the last corresponds to the ordered number of the “experiment_ file”.
For example:

/LP1dtbs/14/10/01/10/01/00/(DE_2_2_1.txt ,1001.num),

here the text "14/10/01/10/01/00/" determinates the date and time of the experiment starting, DE_2_2_1.txt is "experiment_file" and 1001.num gives us the number of "experiment_file" into the sorted list of all files in LP1dtbs.

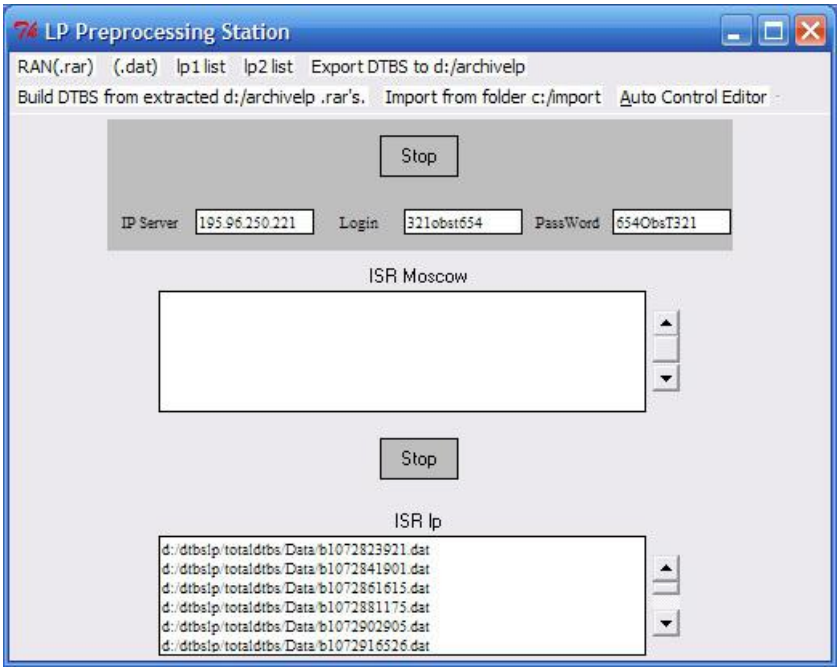


Fig4. Control panel

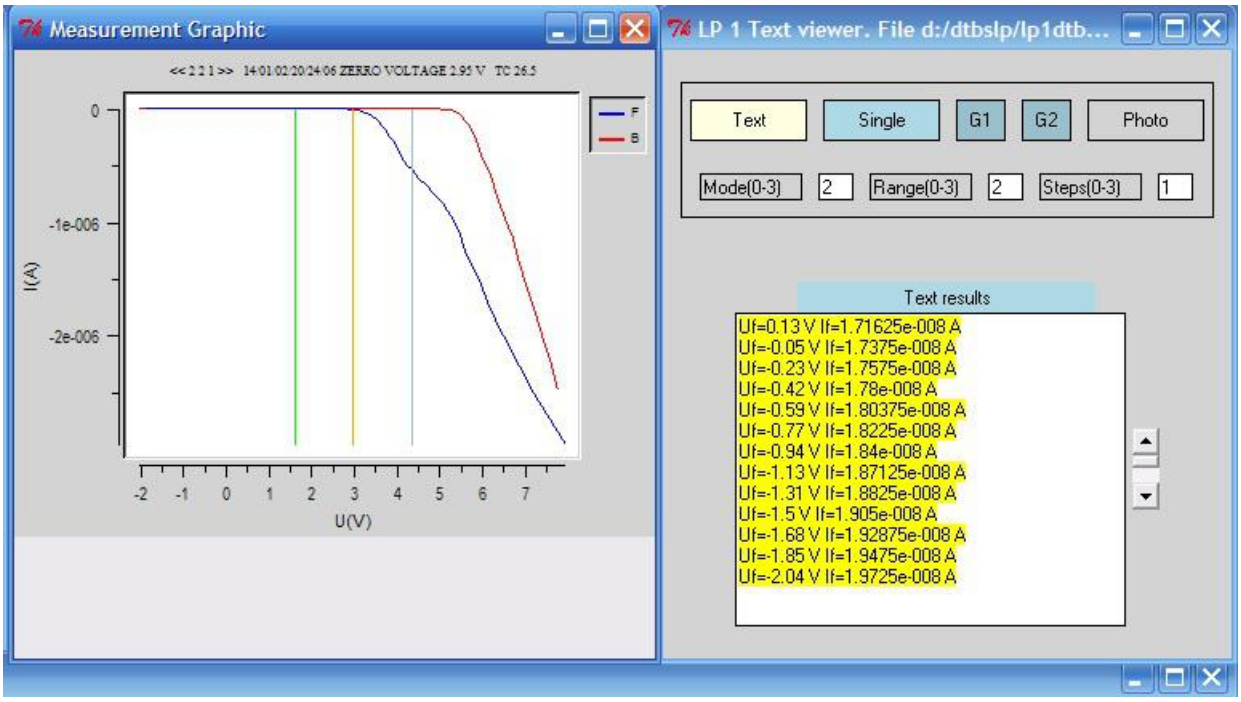


Fig. 5. Graphic and text screens

It is supporting of ordered file names list of included files into corresponded DTBS at current time. The adding of "experiment_file" into the "DTBS" forces creation of new entry into file structure or writing over existed one.

There are realized two function for searching of files into DTBS:

- By name of the “experiment_file” and the time of it creating to determinate it ordered number;
- By ordered number of the file to determine it file name and the time of it creation.
- Each time when primary file processing is finished is resorting the file names of DTBS list and renaming of the “number_file” associated with the “experiment_file”.
- Used data organization allows parallel processing of data-stream on two or more machines and monitoring at this time on other computer of scientific data.
- It is possible next scenario:
- The group of machines includes pre-processing ones and special scientific computers;
- On all of them are installed program packages;
- After starting and initialization of it all machines automatically create own DTBS file structure (empty initially);
- Each of pre-processing machines is connected to FTP server hosted in ISR RASc. (Moscow) and executes the task of receiving of set of primary files, after then pre-process them und update its local DTBS. It is doing in parallel of all preprocessing computers. At end of preprocessing task, each local DTBS is converting into self-extracted archive.

The researcher has ability to get these compressed local DTBS and move them to his Research computer, where they are extracting automatically and adding to global DTBS. All actions connected to internal file processing and transport are supporting by program package.

With the presented library for working with data structures (Mk4Tcl) in addition to those described in points 4.1, 4.2 and 4.3, other experiments were performed for installation, compilation, embedding and use in programs developed on Tcl/Tk, some of which are described and in [14, 15].

The cited applications also provide means for information protection using the capabilities of Tcl/Tk and the application Safe Tcl.

5. Conclusion

As mentioned at the beginning, MetaKit is a library for embedding in scripting languages, working with data sets. The idea of using modular and component-based approaches in the development process allows the use of different applications in one software package.

In this case, the use of MetaKit in Tcl scripts becomes a powerful tool for easy implementation of programs working with controllers, programs that require process history or CRM software packages..

The library itself includes many functions in a small package. Some of the most basic advantages are:

- works with files and maintains a file structure;
- self-expandability - each record is automatically added to the file, and this operation is quite quick;
- flexibility - possibility for select/sort operations on each of the fields, as well as search with different options;
- adaptability - the library file can be used under Unix, Windows and Mac, as well as as a standalone application;
- compactness - the whole library as a file (Mk4Tcl) takes only 184 Kb.

The tendency for the future is to design a Multiplatform operator station using the capabilities of Tcl/Tk and other program structures and tools such as Mk4Tcl, Save Tcl and others, on the basis of which to implement the individual functional blocks of the station. Language skills will also be used to mix C with Tcl codes (or other codes), add and work with pre-compiled Tcl extension packages (data protection, multifunction interpreters and multithreaded mechanisms, built-in

network connectivity of applications and multiplatform support. Mk4Tcl Application for Work with Data Structures is a good option for setup and control of smart instruments for Particle Radiation Therapy of tumors. This way has given physician the capability to personalize treatments for accurate delivery of radiation dose based on clinical parameters and anatomical information.

References

[1] Jean-Claude Wippler, (February 14-18, 2000), Scripted Documents, 7th USENIX Tcl/Tk Conference – Tcl/Tk, Austin, Texas, USA,.

[2] Tcl/Tk program, https://www.tcl.tk/ (last visited 20.06.2020)

[3] http://www.dgroth.de/tcl8.4.4/databases/mk4tcl/mk4tcl.html (last visited 20.06.2020)

[4] Jean-Claude Wippler, (2000), MetaKit for Tcl: The structured database which fits in the palm of your hand, Equi4 Software - Draft, http://www.equi4.com/metakit/tcl.html

[5] Roseman M., (2002), Meta Kit: Quick and Easy Storage for your Tcl Application, Equi4 Software - Draft.

[6] Welch B, (1998 - 2000), Practical Programming in Tcl and TK, part3 - TclHttpd Web Server, Ajuba Solutions.

[7] V. Ivanova, A. Boneva, Y. Doshev, S. Ivanov, P. Vasilev, (27 February 2020), Multifunctional Operating Station Based on Tcl/Tk and its Applications, Proc. of the 6th IEEE International Conference BdKCSE'2019, Sofia, Bulgaria, IEEE, Electronic ISBN: 978-1-7281-6481-6, pp. 1-7, DOI: 10.1109/BdKCSE48644.2019.9010662,

[8] V. Ivanova Z.Ilcheva, D Bachvarov, A.Boneva**.,** (2018), Control system and softweree pakage for an experimental module with force feedback capabilities"**,** Proc. in Manufacturing Systems, RA Publishing House, Vol. 13, Issue 1, ISSN 2343-7472, ISSN_L 2067-9238, pp. 3-10.
http://www.icmas.eu/Journal_archive_files/Vol_13-Issue1_2018_PDF/03-10_IVANOVA.pdf

[9] Ivanova Veronika, Dichko Bachvarov, Ani Boneva, (2018), Smart Surgical Instruments for Robots, j. Complex Control Systems., ISSN 1310-8255, pp. 98-103,
http://ir.bas.bg/ccs/2018/19_ivanova.pdf

[10] Simeon Angelov, Dichko Batchvarov, Kiril Belov, Fikret Calikoglu, Ani Boneva, Rumyana Krasteva, Andrey Zamanov, Veselin Georchev, (16.10.2002), Software Package for Accounting, Processing and Management of Energy Consumption, Academic Open Internet Journal (AOIJ), ISSN 1311-4360, Category: IT and Computing, Issue 8,
http://www.acadjournal.com/2002/v8/part5/p1/index.html

[11]. Dichko Bachvarov, Ani Boneva, Bojan Kirov, Yordanka Boneva, Georgi Stanev, Nesim Baruh, (2014), Primary information preprocessing system for LP, DP devices -project "Obstanovka"**,** Internacional Conference**,** BdKCSE'2014 – Big Data, Knowledge and Control Systems Engineering, November 5, Sofia, Bulgaria, Proceeding, pp. 65 - 74, http://css.iict.bas.bg/eLibrary/BdKCSE'2014_Proseedings.pdf

[12] B. Kirov, S. Asenovski, D. Bachvarov, A. Boneva, V. Grushin, K. Georgieva and S. I. Klimov, (2016), Langmuir Probe Measurements Aboard the International Space Station, Geomagnetism and Aeronomy, Vol. 56, No. 8, ISSN 0016-7932, 1555-645X (Online) , Pleiades Publishing, Ltd., pp. 1082-1089, DOI: 10.1134/S0016793216080120, IF 0,556

[13] Nagy János Zoltán, (2016), A Föld környezetében tudományos méréseket végző űrkutatási berendezés fedélzeti adatgyűjtő rendszerének és fedélzeti kommunikációs rendszerének kidolgozása, Doktori (PhD) értekezés, Alkalmazott Informatikai és Alkalmazott Matematikai Doktori Iskola, Óbudai Egyetem, Budapest, pp. 1-82 (in Hungarian)
http://lib.uni-obuda.hu/sites/lib.uni-obuda.hu/files/Nagy_Janos_Zoltan_ertekezes.pdf

[14] Simeon Angelov, Veselin Georchev, Angel Angelov, Dichko Batchvarov, Ani Boneva, Rumyana Krasteva, Elmira Bachvarova, Kiril Belov, (2005), Distributed system for collection and management of the information using wireless technology, Proceedings of the Third IEEE

IDAACS'2005, ISBN 0-7803-9446-1, Sofia, Bulgaria, IEEE, pp. 331-336, DOI: 10.1109/IDAACS.2005.282997

[15] D.Trifonov, O.Tzarnoretcki, Ditchko Batchvarov, Ani Boneva, Rumyana Krasteva (2005), A possibility for Investigation of Dissolved Oxygen Quantity in a Laboratory Aeration Module with Computer System, J. Cybernetics and Information Technologies, ISSN 1311 – 9702, Vol. 5, No. 1, Institute of Information Technologies – BAS, pp. 128-133, http://www.iit.bas.bg/Cit_en/CIT_05_en/CIT_51-05.html

Article received: 2020-06-20