

# *DABC* user manual

---

Titel: DABC: User Manual

Document	Date	Editor	Revision	Comment
DABC-user	2009-01-05	Hans G.Essel	1.0.0	First scetch



# Contents

---

<i>DABC user manual</i>	<b>1</b>
<b>2 <i>DABC User Manual: Overview</i></b>	<b>5</b>
2.1 User manual overview . . . . .	5
<b>3 <i>DABC User Manual: Plugins</i></b>	<b>7</b>
3.1 <i>DABC</i> plug-in mechanism . . . . .	7
<b>4 <i>DABC User Manual: Parameter</i></b>	<b>9</b>
4.1 <i>DABC</i> parameters . . . . .	9
<b>5 <i>DABC User Manual: Commands</i></b>	<b>11</b>
5.1 <i>DABC</i> commands . . . . .	11
<b>6 <i>DABC User Manual: Setup</i></b>	<b>13</b>
6.1 Setting up system . . . . .	13
<b>7 <i>DABC User Manual: GUI</i></b>	<b>15</b>
7.1 GUI Guide lines . . . . .	15
7.2 GUI Panels . . . . .	15
7.2.1 <i>DABC</i> launch panel . . . . .	15
7.2.2 <i>MBS</i> launch panel . . . . .	15
7.2.3 Combined <i>DABC</i> and <i>MBS</i> launch panel . . . . .	15
7.2.4 Command panel . . . . .	15
7.2.5 Parameter selection panel . . . . .	15
7.2.6 Monitoring panels . . . . .	15
7.2.7 Parameter table . . . . .	15
7.2.8 Logging window . . . . .	15
7.3 GUI save/restore setups . . . . .	15
<b>8 <i>DABC User Manual: Example MBS</i></b>	<b>17</b>
8.1 Example <i>MBS</i> event building . . . . .	17
8.1.1 <i>MBS</i> upgrade for <i>DABC</i> . . . . .	17
8.1.2 New LMD file format . . . . .	17

<b>9</b>	<b><i>DABC User Manual: Example Bnet</i></b>	<b>19</b>
9.1	Example network event building . . . . .	19
<b>10</b>	<b><i>DABC User Manual: Example ROC</i></b>	<b>21</b>
10.1	Example ROC event building . . . . .	21
	<b>References</b>	<b>23</b>

## 2 *DABC* User Manual: Overview

---

[user/user-overview.tex]

### 2.1 User manual overview



## 3 *DABC* User Manual: Plugins

---

[user/user-plugin.tex]

### 3.1 *DABC* plug-in mechanism





## **4    *DABC* User Manual: Parameter**

---

[user/user-parameter.tex]

### **4.1    *DABC* parameters**



## 5 *DABC* User Manual: Commands

---

[user/user-command.tex]

### 5.1 *DABC* commands



## **6   *DABC* User Manual: Setup**

---

[user/user-setup.tex]

### **6.1   Setting up system**



# 7 *DABC* User Manual: GUI

---

[user/user-gui.tex]

## 7.1 GUI Guide lines

## 7.2 GUI Panels

### 7.2.1 *DABC* launch panel

### 7.2.2 *MBS* launch panel

### 7.2.3 Combined *DABC* and *MBS* launch panel

### 7.2.4 Command panel

### 7.2.5 Parameter selection panel

### 7.2.6 Monitoring panels

#### 7.2.6.1 Rate meters

#### 7.2.6.2 Histograms

#### 7.2.6.3 States

#### 7.2.6.4 Information

### 7.2.7 Parameter table

### 7.2.8 Logging window

## 7.3 GUI save/restore setups





## 8 *DABC* User Manual: Example *MBS*

---

[user/user-exa-mbs.tex]

### 8.1 Example *MBS* event building

#### 8.1.1 *MBS* upgrade for *DABC*

To have minimal changes, we use standard collector and transport. Two changes:

##### 8.1.1.1 Increased buffer size support

This is done in a completely compatible way. The only problem is the 16 bit `i_used` field in the buffer header keeping the number of 16 bit data words (behind buffer header). With a new rule we store this number in 32 bit field `l_free[2]`. Only if `dlen` is less equal `MAX__DLEN` defined as `(16K-sizeof(bufhe))/2` this number is also stored in `i_used` as before. Modifications have to be made in all *MBS* modules accessing `i_used`. Modules outside *MBS* can be modified on demand to support large buffers. Current buffers still can be handled without change.

When *MBS* writes large buffer files only the used part of the file header is written. Number of 16 bit words behind buffer header structure is stored in `filhe_used`. Event API `f_evt` is updated to handle large buffers on input. Note: By setup the number of buffers per stream can be set to one. This suppresses event spanning. Large buffers can be used by standard *MBS*.

##### 8.1.1.2 Variable sized buffers

As a second step variable sized buffers are implemented. They get a new type 100. The allocated buffers are still fixed length as before. However, the *MBS* transport would write only the used part of the buffers to clients. Processing these buffers a module must first read the header, then get the used size from `iUsedWords(free[2])` and read the rest. Modules outside *MBS* must be modified to process such buffers. In *MBS*, after stream buffers are created, buffer types are set to 100 by a new command `enable dabc` in transport. This command also sets the transport synchronous mode. In this mode transport processes streams only if a client is connected.

#### 8.1.2 New LMD file format

With *DABC* as event builder for *MBS* there is no need to write files in *MBS*. This gives more freedom to design a new file format. This format will be written by *DABC* and read by `fLmd` functions (`get event`). The format is quite simpler than the old one, because it has no buffer structure causing so much complications by event spanning.

A file has a file header as before, but with a fixed size and a variable part (size `iUsedWords`). Behind the header follow data elements with `sMbsHeader` headers (length, type, subtype) allowing to identify and process or skip them. Besides event data, time stamps may be inserted from the original *MBS* formatted buffers to preserve this time information. Writing/reading such a file is very straight forward. The file header contains the number of data elements and the maximum size of elements. This information is collected throughout the file writing and written on close into the file header. The file header is an

`sMbsFileHeader` structure. `iMaxWords` is the maximum size of elements, `iElements` is the number of data elements.

The file size is no longer restricted to 2GB. Optionally an element index is written at the end of the file. This allows for random access of elements in the file through this index table. The table itself has 32-bit values for the element offsets (in 32-bit). It can therefore address offsets up to 16GB in the file. If larger files are needed, the table can be created with 64-bit values giving unlimited addressing.

Note: This file format needs the rewind file function because the file header must be rewritten to store `iMaxWords`, `iElements`, and optionally the offset of the index table. This function is currently not implemented in the `RFIO` package, but will be done.

## 9 *DABC* User Manual: Example Bnet

---

[user/user-exa-bnet.tex]

### 9.1 Example network event building



## 10 *DABC* User Manual: Example ROC

---

[user/user-exa-roc.tex]

### 10.1 Example ROC event building



# References

---

- [1] CBM collaboration, "CBM Experiment: Technical Status Report", Januar 2005
- [2] CMS collaboration, <http://cmsinfo.cern.ch/outreach/>, "CMS Outreach", CERN 2006
- [3] The Experimental Physics and Industrial Control System website, <http://www.aps.anl.gov/epics/index.php>, Argonne National Laboratory 2006
- [4] C. Gaspar et al., "DIM - Distributed Information Management System" , <http://dim.web.cern.ch/dim/>, CERN May 2006
- [5] Y. Liu and P. Sinha, "A Survey Of Generic Architectures For Dependable Systems", IEEE Canadian Review, Spring 2003
- [6] The National Instruments Labview web site, <http://www.ni.com/labview/>, National Instruments Corporation 2006
- [7] L. Orsini and J. Gutleber, "The XDAQ Wiki Main Page" <http://xdaqwiki.cern.ch/index.php>, CERN 2006
- [8] L. Orsini and J. Gutleber, "I2O Messaging" [http://xdaqwiki.cern.ch/index.php/I2O\\_Messaging](http://xdaqwiki.cern.ch/index.php/I2O_Messaging) , CERN 2006
- [9] L. Orsini and J. Gutleber, "XDAQ Monitor application" [http://xdaqwiki.cern.ch/index.php/Monitor\\_CGI\\_interface](http://xdaqwiki.cern.ch/index.php/Monitor_CGI_interface) , CERN 2005
- [10] L. Orsini and J. Gutleber, [http://xdaqwiki.cern.ch/index.php/Configuration\\_schema](http://xdaqwiki.cern.ch/index.php/Configuration_schema) "XDAQ XML configuration schema", CERN 2006
- [11] D. Stenberg et al., The curl and libcurl web site, <http://curl.haxx.se/>, HAXX HB 2006
- [12] SystemC website, <http://www.systemc.org/>
- [13] The W3C Consortium, "SOAP Version 1.2 Part 1: Messaging Framework", <http://www.w3.org/TR/soap12-part1>, W3C Recommendation, 24 June 2003
- [14] K. Whisnant, R.K. Iyer, Z. Kalbarczyk, and P. Jones, "The Effects of an ARMOR-based SIFT Environment on the Performance and Dependability of User Applications", University of Illinois, 2006
- [15] The Wikipedia, "Finite State Machine", [http://en.wikipedia.org/wiki/State\\_machine](http://en.wikipedia.org/wiki/State_machine), Wikipedia 2006

