For: wpm
Printed on: Tue, Sep 14, 1993 15:26:50

Document: caWingz
Last saved on: Tue, Feb 23, 1993 11:42:41

CaWingz User's Guide
(by Ben–chin Cha)

This document assumes that you have read and understood the Wingz user's manuals.

CaWingz is an external Wingz program which, when combined with a set of script files, provides easy–to–use channel access interface functions for Wingz users. The new function *run* will be available with release 3.8 of epics. The functions, script files, and usage are briefly described in this document. The script files supplied here serve as examples only. Users are responsible for generating their own spreadsheet and script files.

External Program caWingz

Program caWingz contains 5 functions: *caGet, caPut, caGetA, caPutA,* and *run*. The *caGet* and *caPut* functions handle getting and putting a real (integer or floating point) value from or to a channel device. The *caGetA* and *caPutA* functions handle getting and putting a string value (maximum length of 40) to a channel device field. The *run* function provide the system command invocation feature to caWingz users.

caGet

The function *caGet* accepts the channel device name corresponding to the highlighted spreadsheet cell as an input argument and returns a numerical value which is the current value obtained from IOC.

caPut

The function *caPut* accepts the channel device name and current highlighted spreadsheet cell value as input arguments and writes the value to the IOC.

caGet

The function *caGetA* accepts the channel device name corresponding to the highlighted spreadsheet cell as an input argument and returns the obtained string value.

caPutA

The function *caPutA* accepts the channel device name and current cell string value as input arguments and writes the string value to the IOC.

run

The function *run* accepts a command string as input and then automatically invokes the command string. The string can be a system command or a name of script file which contains more than one command string.

Script Files

The script files used with caWingz are caWingz.txt, caOpen.txt, caClose.txt, caGet.txt, caPut.txt, caGetA.txt, and caPutA.txt. They are described below.

caWingz.txt

This script file contains the execution steps associated with the work sheet.

```
on activate
    Run script "/home/phebos1/epics/share/src/toolsComm/wingz/scripts/caOpen.txt"
end activate
on deactivate
    Run script "/home/phebos1/epics/share/src/toolsComm/wingz/scripts/caClose.txt"
end deactivate
```

By attaching this script file to a work sheet, the Wingz program automatically runs the script file caOpen.txt when the work sheet is activated and automatically runs the script file caClose.txt when the work sheet is deactivated.

caOpen.txt

This script file gets the external program caWingz so that the functions defined in caWingz can be accessed from Wingz.

```
.   GET EXTERNAL "/home/phebos1/epics/Unix/sun4/bin/caWingz"
```

caClose.txt

This script file removes the external program caWingz so that the functions defined in caWingz can-no longer be accessed from Wingz.

```
REMOVE EXTERNAL "caWingz"
```

caGet.txt

This script file calls the caGet function which gets new values from the IOC and updates the values of a work sheet for the selected column and range of rows.

```
DEFINE rangeRef,rowN,row1st,rowI,colI,pvName,rtnVal

rangeRef=REFERENCE(SELECTION(1))
rowN=ROWS(RANGE(rangeRef))
row1st=ROWOF(RANGE(rangeRef))
colI=COLOF(RANGE(rangeRef))

FOR rowI=row1st TO (row1st+rowN−1) STEP 1
   pvName=INDIRECT(MAKECELL(1,rowI))
   rtnVal=caWingz:caGet(pvName)
   PUT rtnVal INTO MAKECELL(colI,rowI)
END FOR
```

caPut.txt

This script file calls the caPut function which sends values from the selected column and range of rows in a work sheet to the IOC.

```
DEFINE rangeRef,rowN,row1st,rowI,colI,pvName,sendVal,rtnVal

rangeRef=REFERENCE(SELECTION(1))
rowN=ROWS(RANGE(rangeRef))
row1st=ROWOF(RANGE(rangeRef))
colI=COLOF(RANGE(rangeRef))

FOR rowI=row1st TO (row1st+rowN–1) STEP 1
   pvName=INDIRECT(MAKECELL(1,rowI))
   sendVal=INDIRECT(MAKECELL(colI,rowI))
   rtnVal=caWingz:caPut(pvName,sendVal)
END FOR
```

caGetA.txt

This script file calls the caGetA function which gets string values from the IOC and updates the values of a work sheet for the selected column and the range of rows. The contents of this script file is exactly the same as the caGet.txt script except that caGet is replaced by caGetA.

caPutA.txt

This script file calls the caPutA function which sends string values from the selected column and range of rows in a work sheet to the IOC. The contents of this script file is exactly the same as the caPut.txt script except that caPut is replaced by caPutA.

run.txt

An example of script file which calls the run function is given below.

```
DEFINE command,rtnVal
command=”/home/phebos2/CHA/bin/testdm”
rtnVal=caWingz:run(command)
```

where ”testdm” is an executable shell script file, the content of ”testdm” is given below.

```
cd /home/phebos2/CHA/appl2/3.5.x3
dm test.dl &
cd /home/phebos2/CHA/appl2
xmca &
```

This shell script file ”testdm” will inoke two processors **dm** and **xmca** for user and return to the caller Wingz.

Procedure To Access caWingz

To use the external program caWingz, a Wingz user must follow the procedure below. The purpose here is only to provide an example of how to access caWingz functions from a user's point of view. A user has to refer to the Wingz documents for the detail information of how to use Wingz program.

1. Create a directory for Wingz application, and copy the script files to your own script directory.

   ```
   cd
   mkdir wingz
   cd wingz
   mkdir scripts
   cd  scripts
   cp /home/phebos1/epics/share/src/toolsComm/wingz/scripts/*.txt .
   chmod 644 *.txt
   ```

2. Modify the script file caWingz.txt to replace the script path directory

   /home/phebos1/epics/share/src/toolsComm/wingz/scripts/ . . .

   by your own script path directory , e. g.

   /home/phebos1/USERID/wingz/scripts/ . . .
3. Save the changes and go to the wingz directory, and invoke the Wingz program.

   ```
   cd
   cd wingz
   Wingz  &
   ```

4. Create a work sheet with channel names defined in column A.

5. Select the "Script" pull down menu from the Wingz window, then select the 'Attach Script' item from the menu to attach the 'caWingz.txt' file to the work sheet.

Steps of Creating Channel Access Buttons

1. Create a channel access get value button on your work sheet.

   Use the "button icon" from the work sheet to create and place a button on the work sheet.
   Select the "Format" pull down menu from the Wingz window, then select the 'Button Info' item from the menu. Name the button as "caGet" or any

name you prefer. You may change the appearance of this newly–created button by selecting the size, color, or font from the menu.
Select the "Script" pull down menu from the Wingz window, then select the 'Attach Script' item from the menu to attach the 'caGet.txt' script file to this newly created button.

2. Repeat step 1 for creating "caPut", "caGetA", and "caPutA" buttons (Only needed button should be created).
   Save the work sheet under your wingz directory.
3. Highlight the range of rows (i.e. corresponding to device names) of a given column where the new values to be obtained from IOC.
   Click the "caGet" button resulting that these newly–obtained values been written to the highlighted column of the work sheet.
4. Click the "caPut" button to put the highlighted values to IOC from the work sheet.

Steps of Creating Run Button

1. Use the "button icon" from the work sheet to create and place a button on the work sheet.
2. Select the "Format" pull down menu from the Wingz window, then select the 'Button Info' item from the menu. Name the button reflecting the processor you want to run.
3. You may change the appearance of this newly–created button by selecting the size, color, or font from the menu.
4. Select the "Script" pull down menu from the Wingz window, then select the 'Button Script' item from the menu to open the button object script window for this newly created button.
5. Type in the following button script

```
DEFINE command,rtnVal
command="command or name of script file"
rtnVal=caWingz:run(command)
```

6. Select the "Quit" option from the menu of the button object script window, and click the "Save" button before closing the window
7. Turn on the "selection icon" from the work sheet and click the newly created run button to test the command

Note: You should see the "Enter caWingz" printed on your terminal session, before you can use any of the channel access buttons. If you dont see the "Enter caWingz" message, quit the work sheet window and reopen the work sheet again. This should properly load the caWingz functions into Wingz. You will see the "Exit caWingz" message printed on your terminal session when you quit the work sheet window.