CERN
CH-1211 Geneva 23
Switzerland

the
Large
Hadron
Collider
project

LHC Project Document No.
LHC-UNICOS

CERN Div./Group or Supplier/Contractor Document No.
EN / ICE

EDMS Document No.

Date: 2009-07-16

FUNCTIONAL SPECIFICATION

USING CMW WITH UNICOS
A STEP BY STEP EXAMPLE

Abstract

This document describes how to use CMW with a UNICOS like application.

Prepared by:	***Checked by:***	***Approved by:***
Hervé Milcent (EN/ICE)		

History of Changes

Rev. No.	*Date*	*Pages*	*Description of Changes*
1.1 Draft	10-July-2009		First version
1.2 Draft	16-July-2009		New feature: set the CORBA name server

Table of Contents

1. INTRODUCTION

1.1 PURPOSE OF THIS DOCUMENT

This document describes how to configure the PVSS CMW client interface and the PVSS CMW Server interface based on a step by step example.

1.2 DEFINITIONS, ACRONYMS AND ABBREVIATIONS

- UNICOS: UNified Industrial COntrol System.
- unicos-pvss.x.y.y: version of the unicos-pvss package.
- DS: UNICOS Data Server, a computer running a PVSS project (PVSS managers: Event Manager, Database manager, etc.).
- OWS: UNICOS Operator WorkStation, a computer running one (or many) PVSS graphical interface.
- Application: set of DS and PVSS projects.
- DP: PVSS data point type.
- DPType: PVSS data point type.
- CMW: Common MiddleWare
- CORBA name server: service provided by CORBA and used by CMW to find the server hodling the publication

1.3 PRE-REQUISITE

1. PVSS 3.8 SP1.

2. unicos-5.0.1b

3. UNICOS project with the following components installed: fwCore, fwAccessControl, fwTrending, unCore v5.0.1b, unCPC v5.0.0 and unLHCServices v5.0.1b

2. PVSS CMW CLIENT WITH CMW SIMULATOR

In this example, a CMW front-end with its devices will be imported into the application. The devices and front-end will subscribe to data simulated by the simulator: simCMWServer.

4. import the CMW front-end:

– copy the file 'example_cmw_import.txt' from the unicos-pvss-5.0.1b/PVSS/data to your project folder data

– add in the PVSS console 'PVSSsim –num 2' and 'PVSS00CMWClient -num 2 -db CMWClient_Driver_02' in manual mode

– start unicosHMI, log as admin

– create the CMWClient config (Figure 1 and Figure 2) for driver 2

– start the PVSS00sim –num 2 and import the file 'example_cmw_import.txt' with driver 2 and 'CMW' as front-end type (Figure 3).

5. start the PVSS00CMWClient and get data

– stop the PVSS00sim –num 2

– edit the config file of the project and in the section '[CMWClient_2]' uncomment the line '#cfgRdaTestServ = "UNICOSDEMO"' (remove the '#' character), the client will then try to connect to the server UNICOSDEMO (this server name must be the same as the one used with the simulator, see below), if another CORBA name server (dns) is used, replace 'pslxmcr:5020' by the CORBA name server and port number

– stop the PVSS00sim and start the PVSS00CMWClient –num 2 (unblock the program if needed), the errors shown in Figure 4 will appear in the log (these errors means that the CMW Client cannot find the device because the CMW Server is not reachable

– in the unicosHMI, open the tree device overview utility and choose front-end, do a right click on the CMW_TEST front end and enable the front-end

– open the application configuration (Figure 5 and Figure 6) and set the navigation panel to 'vision/unCMWDeviceTestAddress.pnl' (located in the installed_components/panels folder)

– open the navigation panel and select '*' as front-end, use server name and set the name to 'UNICOSDEMO' (same name as the one in the config file), set use sync and Update period to 2000 (Figure 7), if another CORBA name server (dns) is used, replace 'pslxmcr:5020' by the CORBA name server and port number (the same as the one above). The data will be updated every 2 sec. Click on the button load sim (unblock the program if needed), a DOS console is opened in minimized mode (Figure 8)

– the data start to oscillate between two values

– Select one device and do force mode, the command will be sent to the simulator (Figure 9)

The device data set and value can be modified: edit the files DeviceTest-SIM.xml and/or DeviceTest.xml in the folder 'installed_components/bin/CMW-Sim/bin/configTest' and restart the CMW Server simulator.

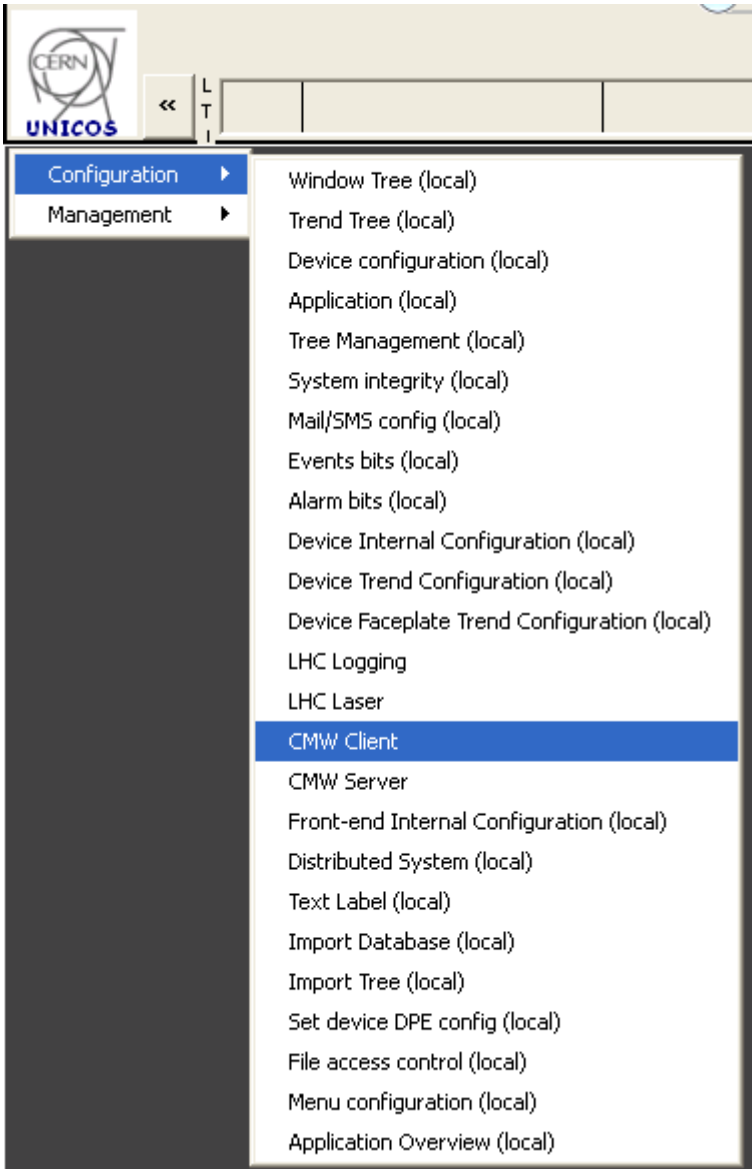

Figure 1: open the CMW Client configuration.

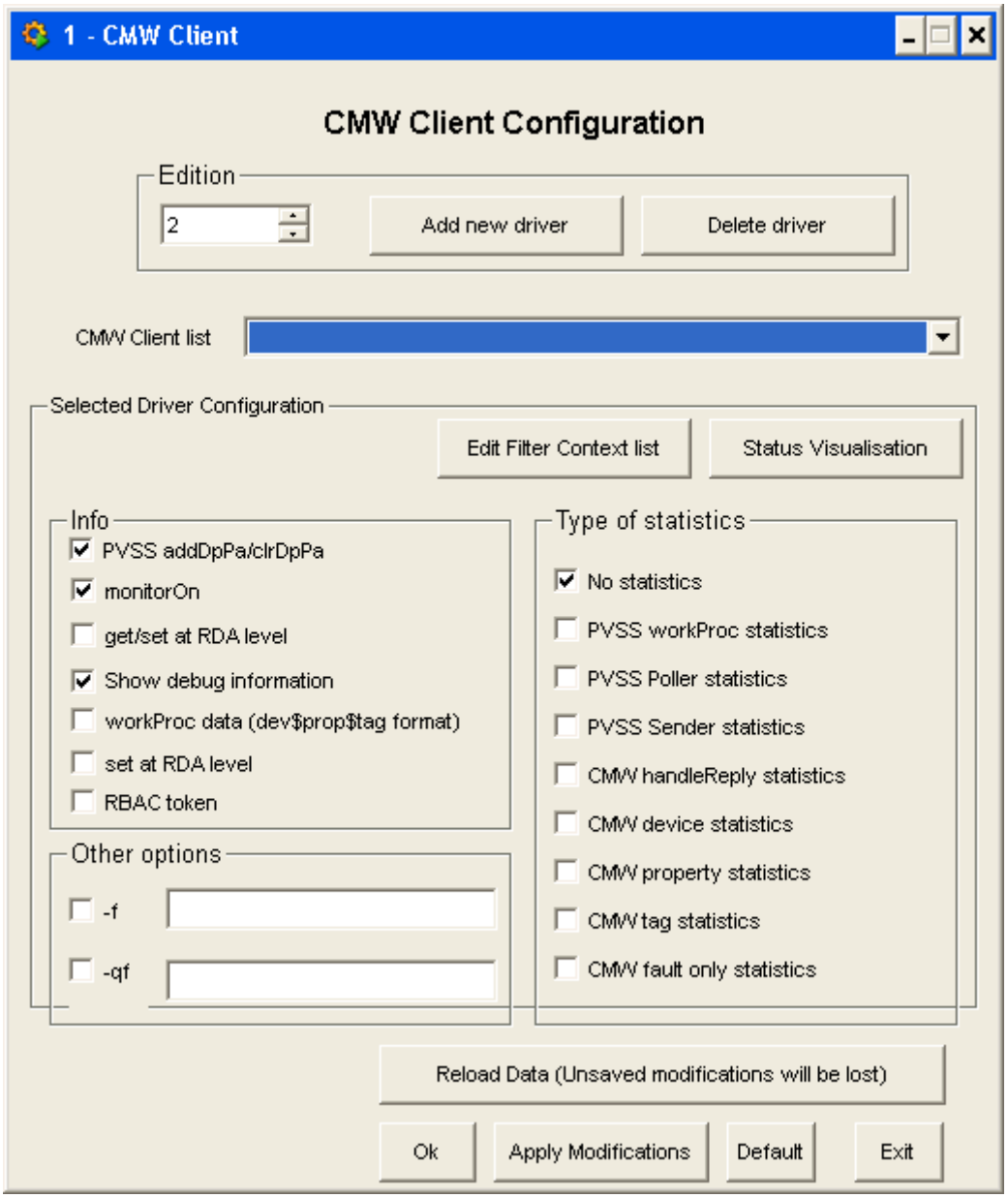

Figure 2: create the CMW config for driver 2.

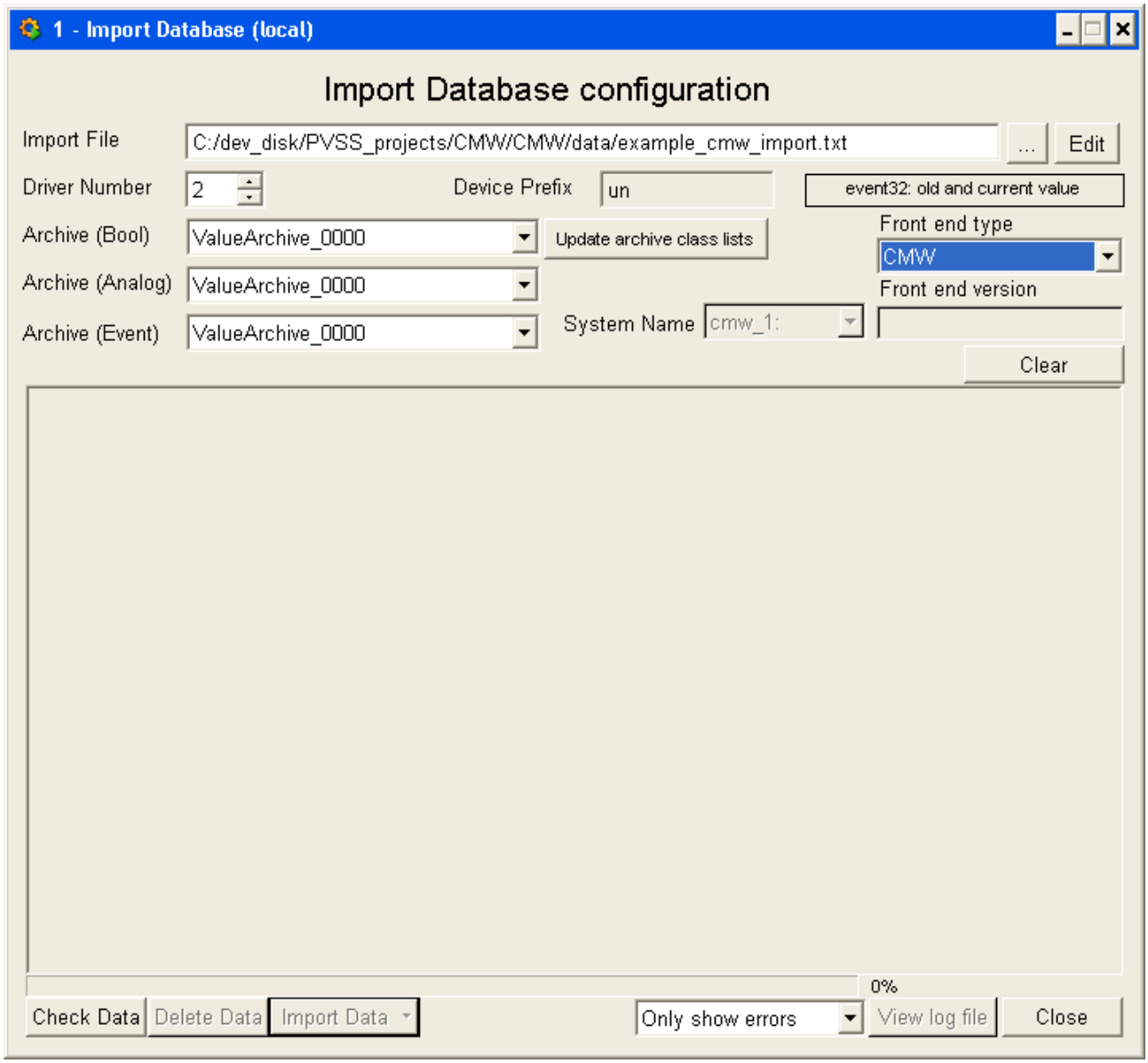

Figure 3: import the CMW front-end and devices.

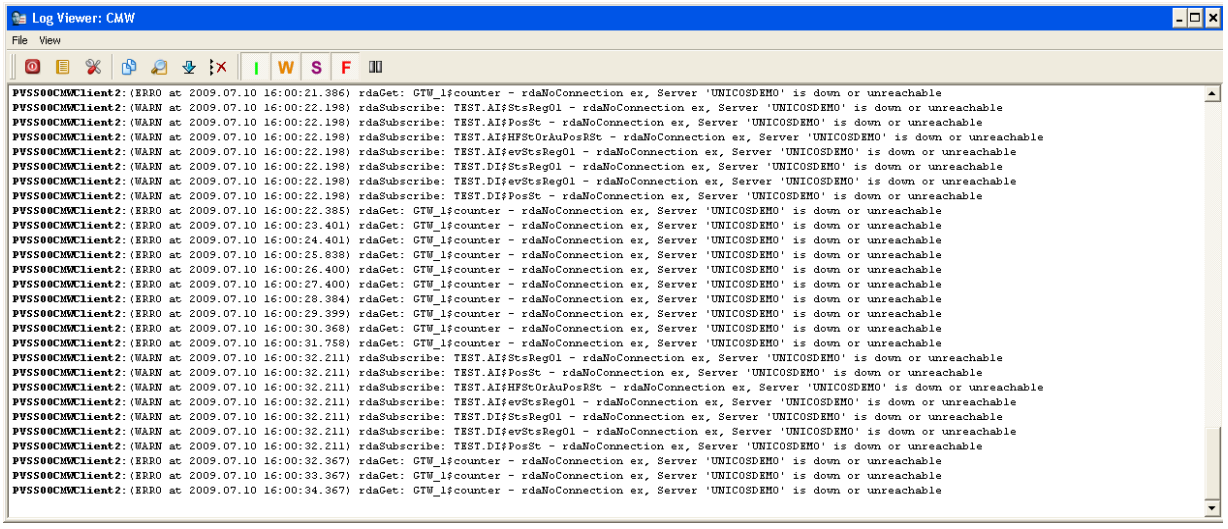

Figure 4: Error when starting the PVSS00CMWClient.

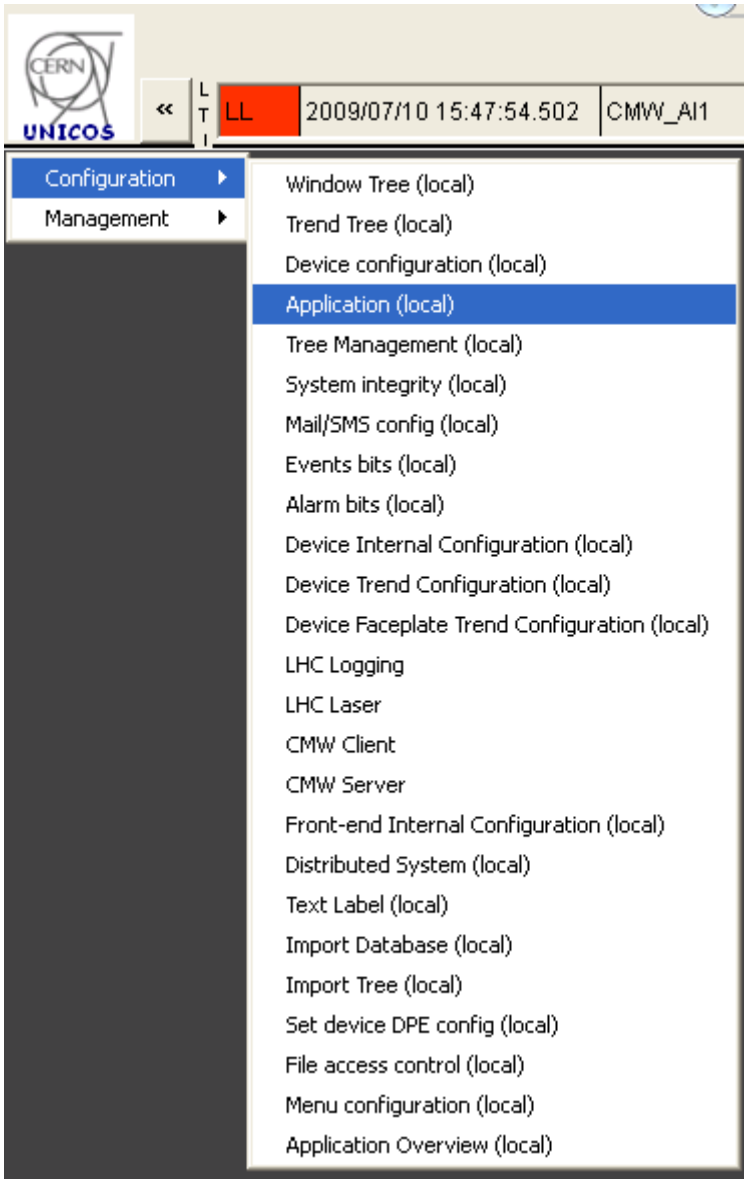

Figure 5: open the application configuration.

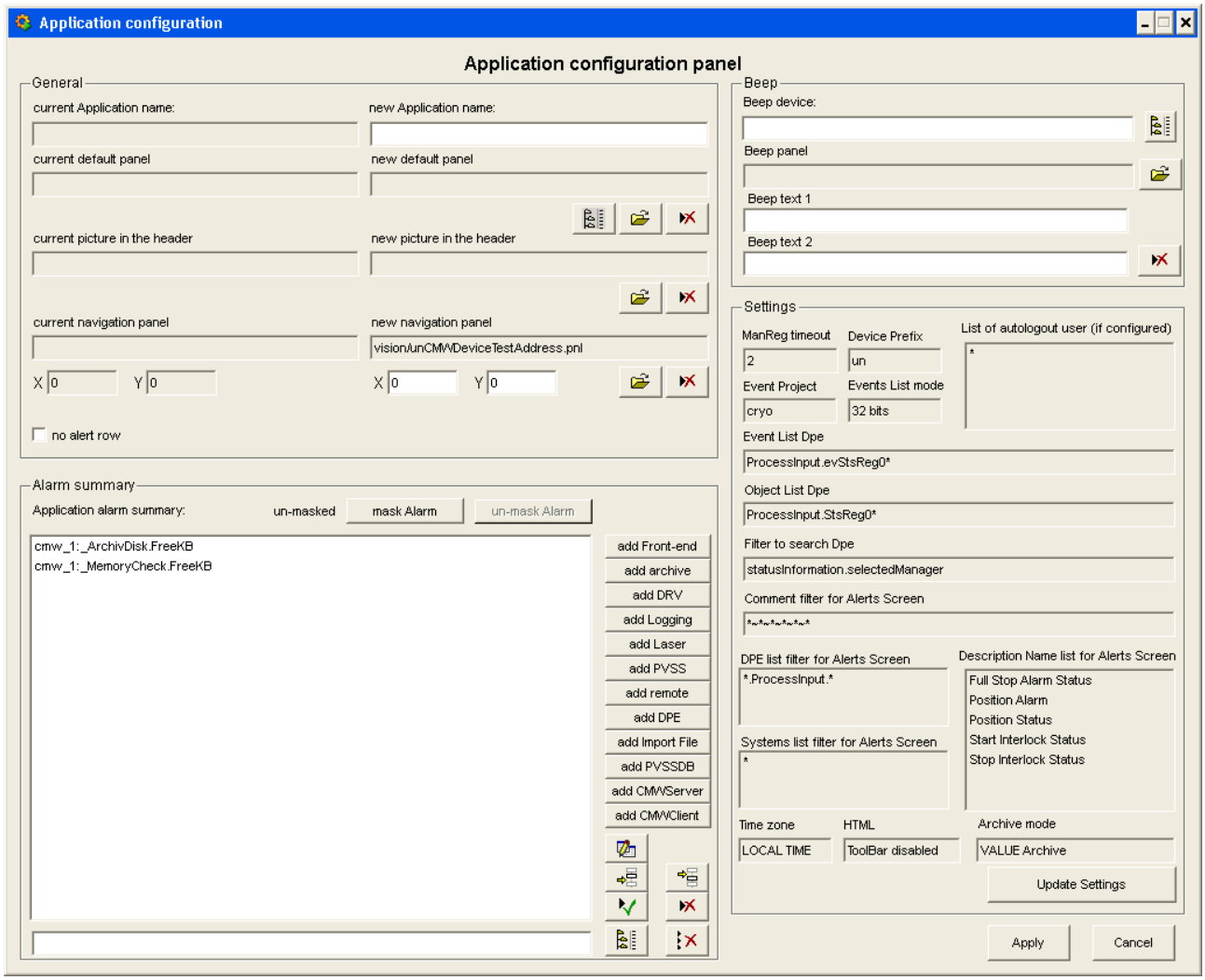

Figure 6: application configuration.

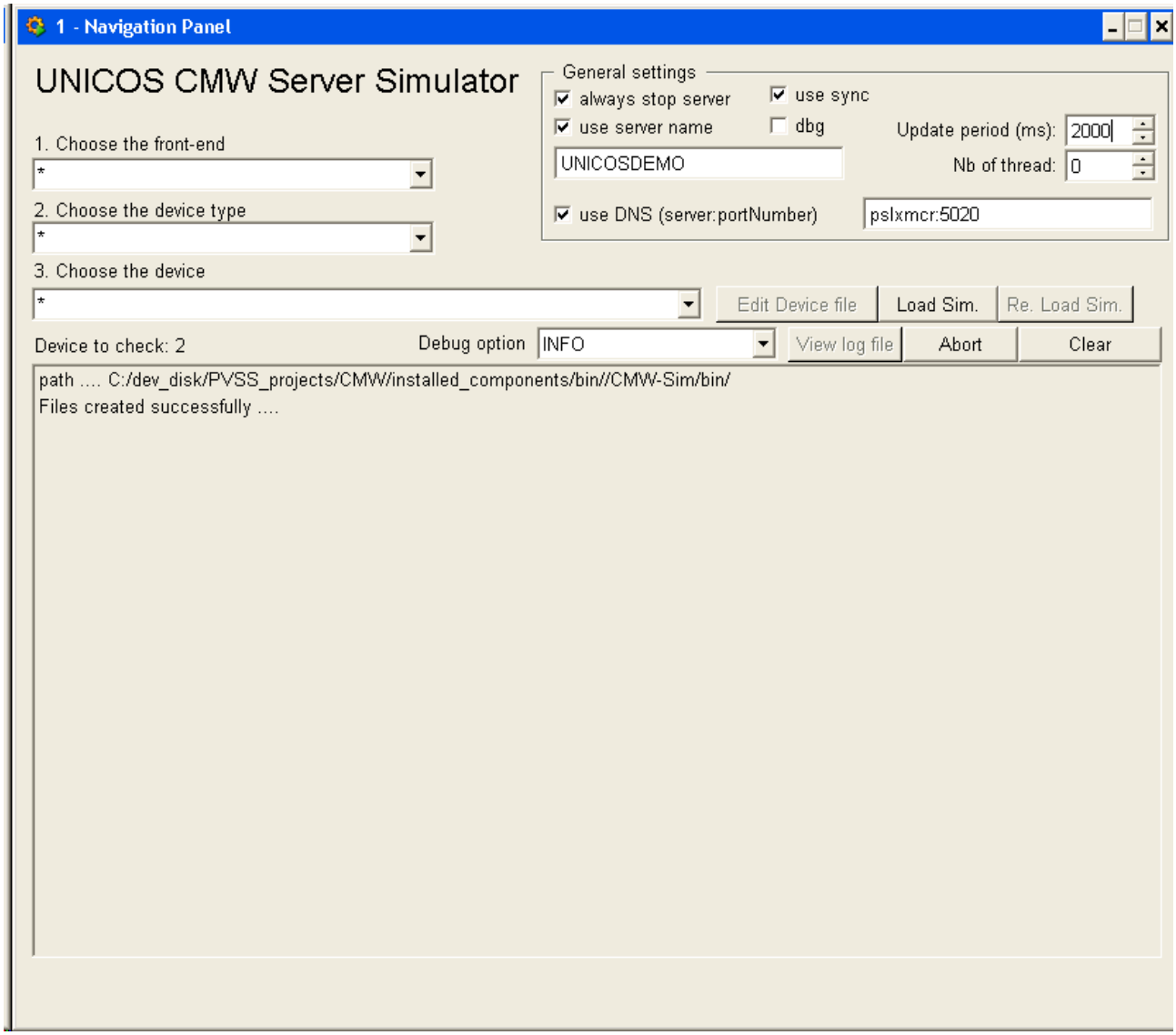

Figure 7: UNICOS CMW Server simulator.

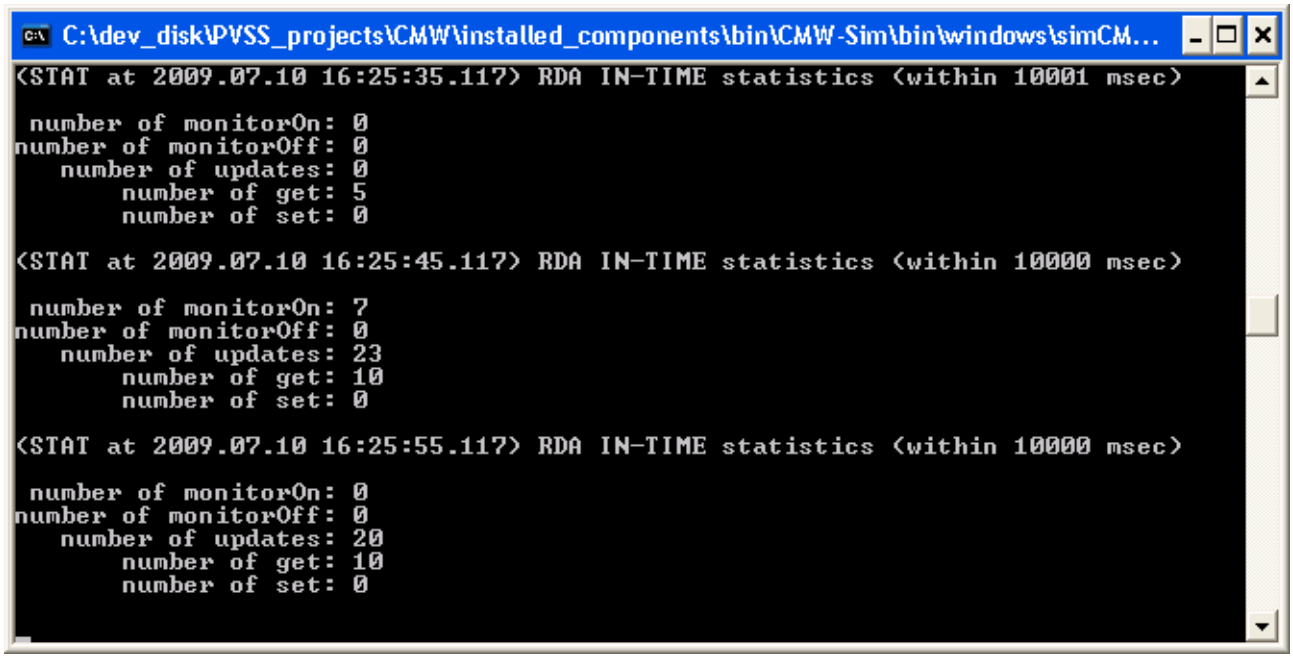

Figure 8: simulator console.

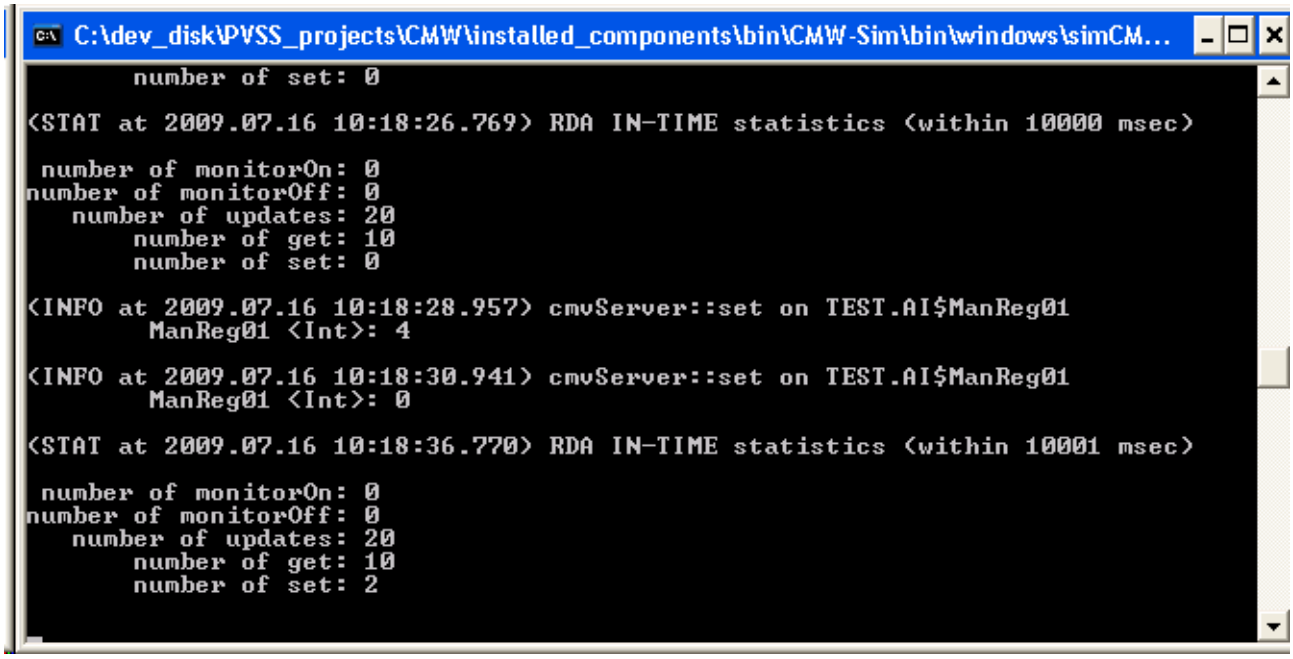

Figure 9: simulator console (set).

3. PVSS CMW CLIENT WITH PVSS CMW SERVER

In this example, a SOFT_FE front-end with two devices will be imported into the application: AI_SERVER and AI_CLIENT. A PVSS00CMWServer will be created and will publish the PosSt DPE of the device AI_SERVER as in/out and the 'MPosR' DPE of the device AI_SERVER as out. The device AI_CLIENT will subscribe via a PVSS00CMWClient to data published by the PVSS00CMWServer via monitorOn and set.

1. import the SOFT_FE front-end:

– copy the file ‘example_soft_fe_import.txt’ from the unicos-pvss-5.0.1b/PVSS/data to your project folder data

– add in the PVSS console ‘PVSSsim –num 3’ and ‘PVSS00CMWClient -num 3 -db CMWClient_Driver_03’ in manual mode

– start unicosHMI, log as admin

– create the CMWClient config (Figure 1 and Figure 2) for driver 3

– start the PVSS00sim –num 3 and import the file ‘example_soft_fe_import.txt’ with driver 3 and ‘SOFT_FE’ as front-end type (Figure 10)

– start the front-end utility and enable the SOFT_FE front-end

2. configure the PVSS00CMWServer:

– create the CMWServer config (Figure 11 and Figure 12) for manager 3, set the configuration file name to data/CMWServerData.txt

– open the expert setting of the CMW Server (Figure 13), set the CMW Server to ‘UNICOS_PVSS_DEMO’ (this name must be the same as the one in the config file for the CMW Client driver 3), the Integrity period to ‘30’ (the value is in sec not in msec.), the CMW Data Tag Name and CMW Time Tag Name will be the ones used for the configuration of the CMW address config (see below), if another CORBA name server (dns) is used, replace ‘pslxmcr:5020’ by the CORBA name server and port number

– click on Manage file; in the configuration file panel click add DP; click on PVSS DP and select the last AnalogInput device PostSt DPE (Figure 14 and Figure 15) set Publication Type=‘in/out’ (get/monitorOn and set), device=‘AI’, property=‘PosSt’, RDA Value Format=‘float’, RDA Time Format=‘longlongnano’; click on PVSS DP again and select the last AnalogInput device MPosR DPE (Figure 16 and Figure 17) set Publication Type=‘out’ (get/monitorOn), device=‘AI’, property=‘ MPosR’, RDA Value Format=‘float’, RDA Time Format=‘longlongnano’. An example of the configuration file is in the folder unicos-pvss-5.0.1b/PVSS/data.

– The configuration must be like in Figure 18.

– Add in the PVSS console ‘PVSS00CMServer -num 3 -db CMWServer_Manager_03’ in manual mode

– In the CMWServer configuration (Figure 11 and Figure 12), open Status Visualisation, select the CMWServer (double click), right click and choose ‘Start PVSS00CMWServer –num 3’ (unblock if necessary), right click and choose ‘Commands’ and click on ‘Start’. Close and reopen the Status Visualisation, the status of the CMW Server should be green as in Figure 19. The PVSS console log looks like in Figure 20.

3. configure the PVSS00CMWClient

– open the tree device overview utility

– right click on the AI_CLIENT device and open the Device configuration

– click on the Address cell of the ProcessInput.PostSt DPE row of the DPE table; click on ‘S”; set Device Name=‘AI’, Property Name=‘MPosR’, Tag name=‘value’ (same name as the one in the server configuration), APM Tag=‘time’ (same name as the one in the server configuration), Time calculation method=‘longlong (nano)’; set driver number to 3, Data transformation to ‘float’ (Figure 21 and Figure 22)

– click on the Address cell of the ProcessInput.MPosR DPE row of the DPE table; click on ‘S”; set Device Name=‘AI’, Property Name=‘PosSt’, Tag name=‘value’ (same name as the one in the server configuration), Time calculation method=‘no’; set driver number to 3, Data transformation to ‘float’, Direction to ‘Output (individual)’, deselect ‘monitorOn (subscription)’ (Figure 23 and Figure 24)

4. start the simulation

– edit the config file of the project and in the section '[CMWClient_3]' uncomment the line '#cfgRdaTestServ = "UNICOSDEMO"' (remove the '#" character) and replace UNICOSDEMO by UNICOS_PVSS_DEMO (the one you set above in the CMW Server configuration), the client will then try to connect to the server UNICOS_PVSS_DEMO, if another CORBA name server (dns) is used, replace 'pslxmcr:5020' by the CORBA name server and port number (the same as the one above)

– stop the PVSS00sim and start the PVSS00CMWClient –num 3 (unblock if necessary)

– open the PARA module, set the 'StsReg01' DPE of the last AnalogInput device to 0 (Figure 25)

– in the unicosHMI, open the tree device overview utility, open the faceplate of AI_CLIENT and AI_SERVER

– select AI_CLIENT, open Set Value and set a value, the AI_SERVER Active value is updated (set operation in the PVSS00CMWServer)

– select the AI_SERVER, open Set Value and set a value, the AI_CLIENT Active value is updated (monitorOn operation in the PVSS00CMWServer with timestamping from the PVSS00CMWServer), open the PARA module, the timestamp of the MPosR DPE of the AI_SERVER (un-SOFT_FE_TEST-APPL-AnalogInput-00002.ProcessOutput.MPosR) and the timestamp of the MPosR DPE of the AI_CLIENT (un-SOFT_FE_TEST-APPL-AnalogInput-00001.ProcessInput.PosSt) must be equal (because the PosSt address config was configured to used the time stamp from the property). If the APM time calculation of the PosSt AI_CLIENT is set to 'no' the time stamps will be different (it will the one of the reception of the device by the PVSS00CMWClient).

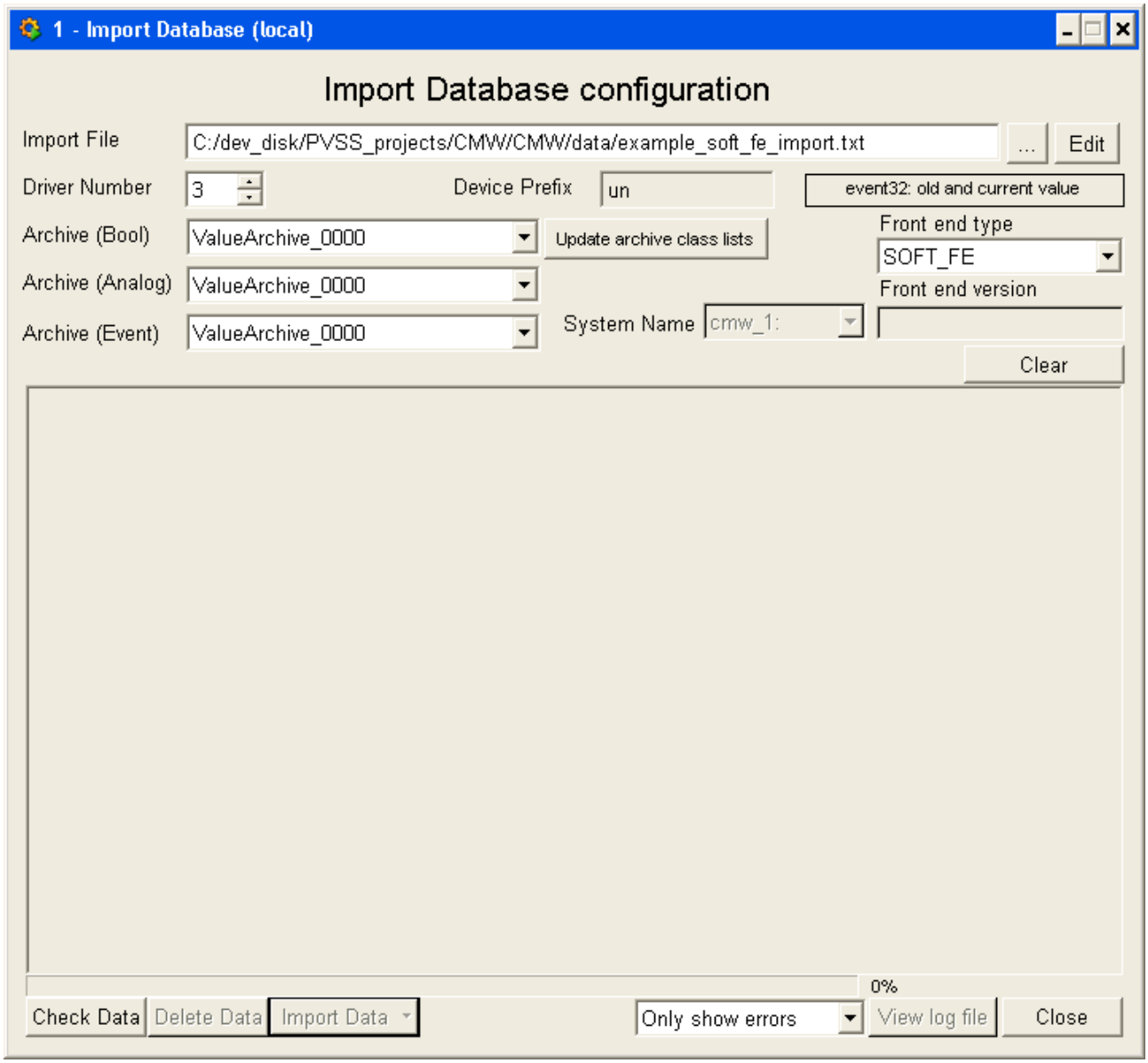

Figure 10: import the SOFT_FE front-end and devices.

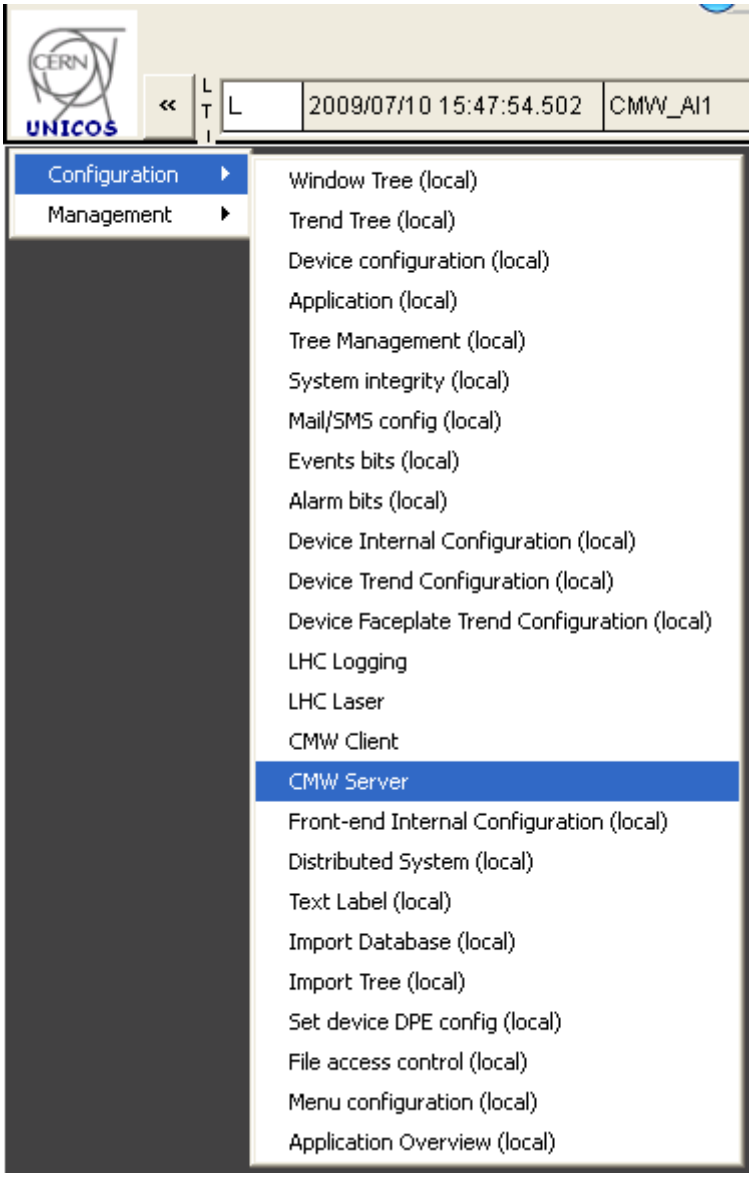

Figure 11: open CMW Server configuration.

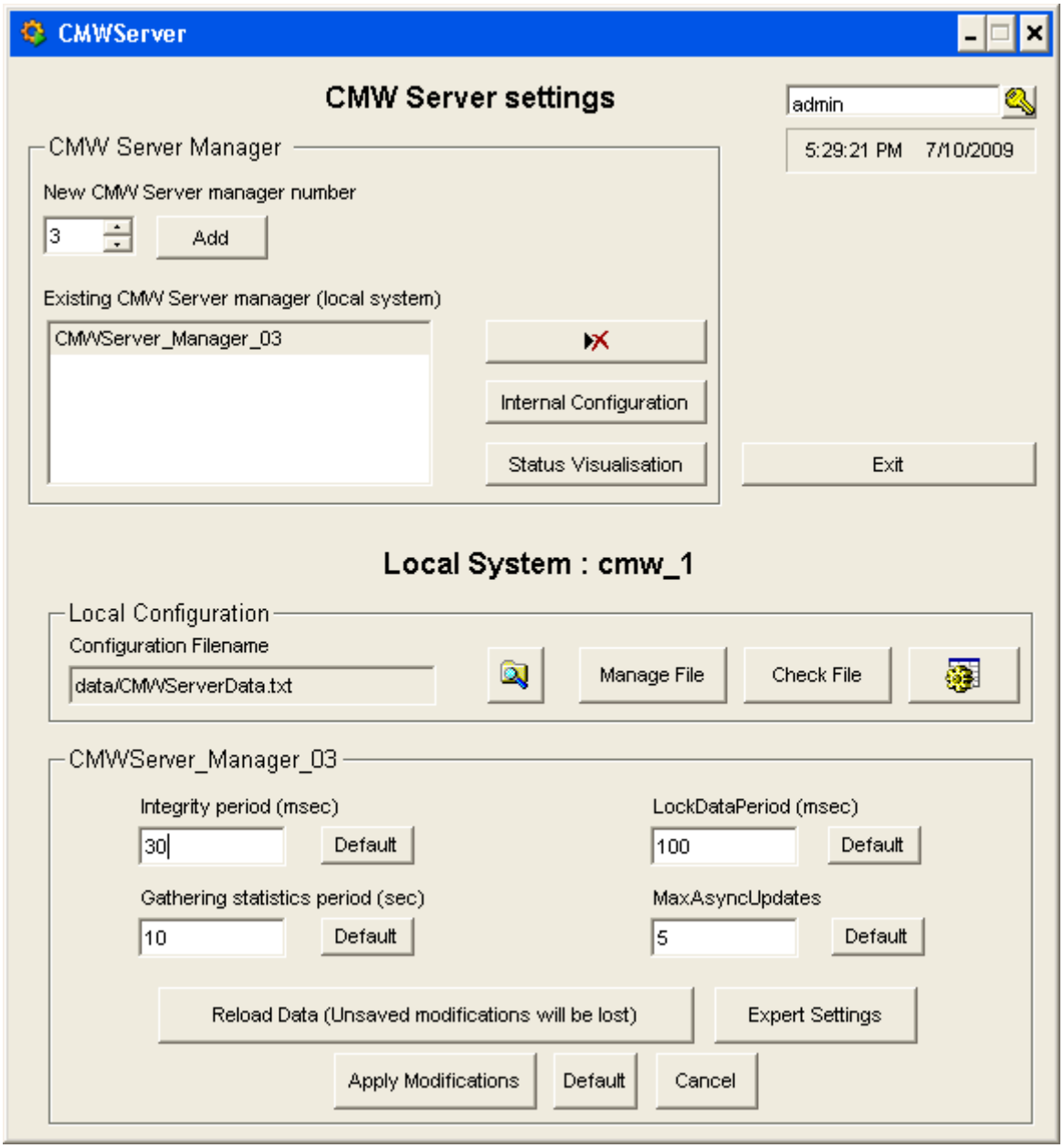

Figure 12: CMW Server configuration.

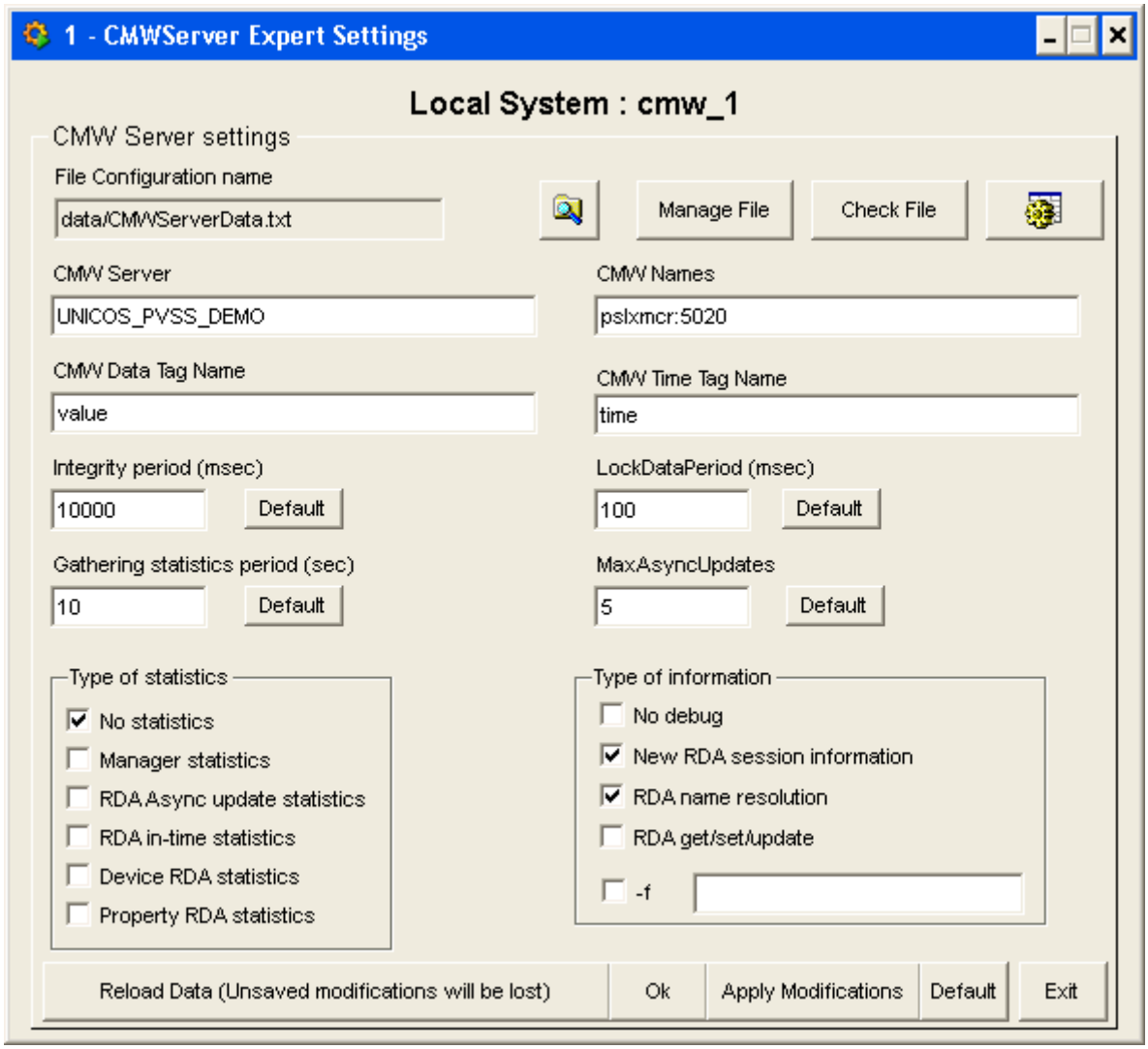

Figure 13: CMW Server expert configuration.

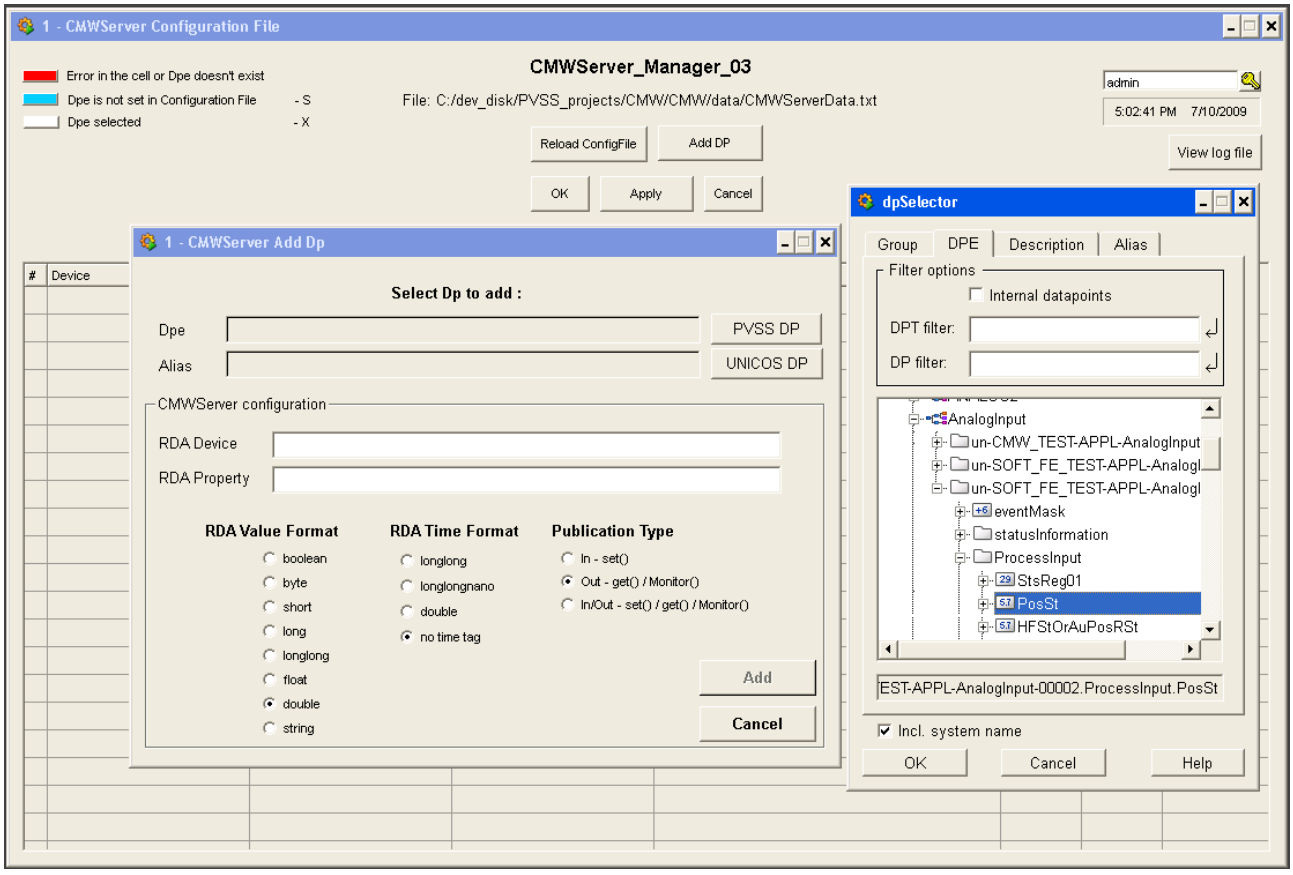

Figure 14: add device/property PosSt.

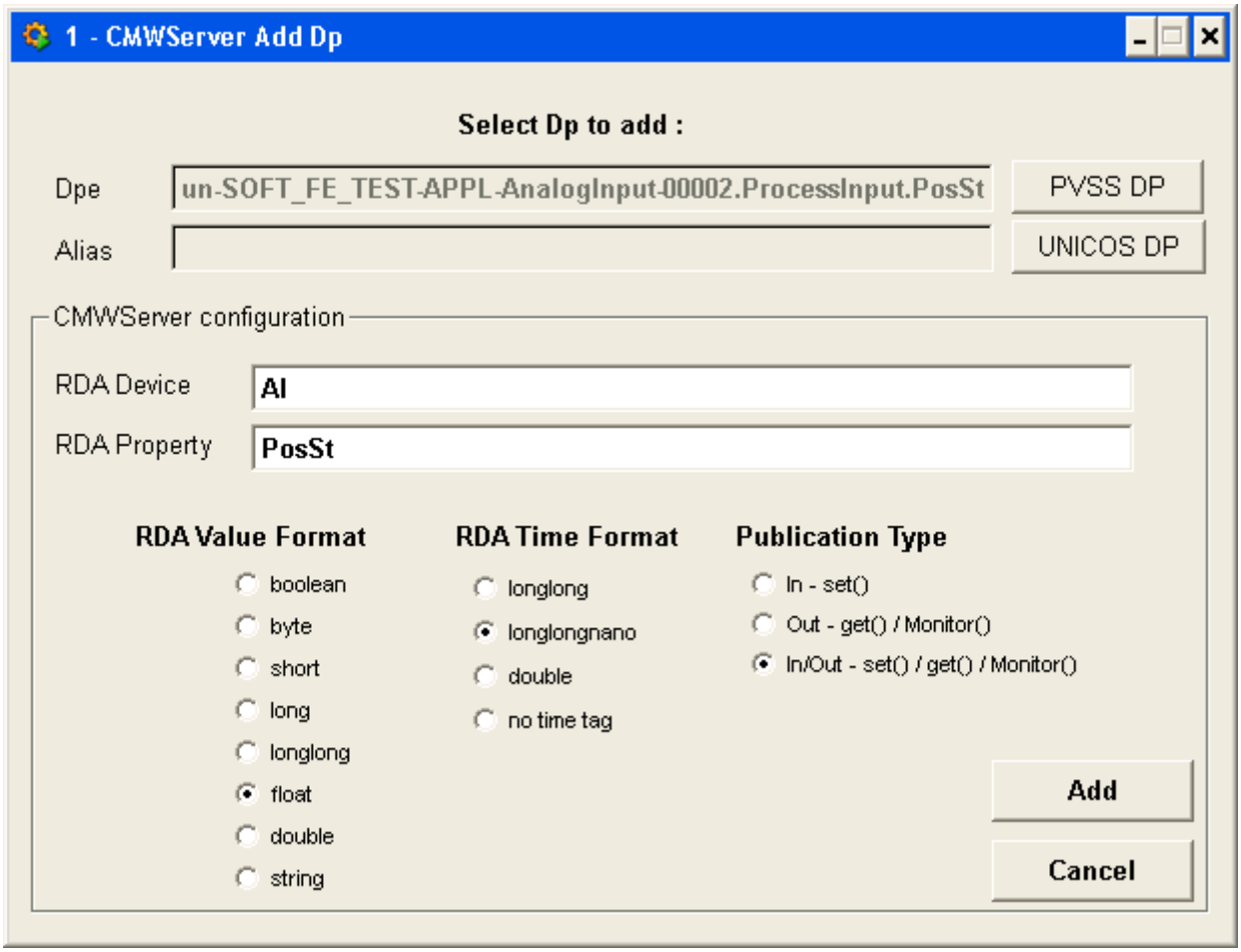

Figure 15: device/property PosSt config.

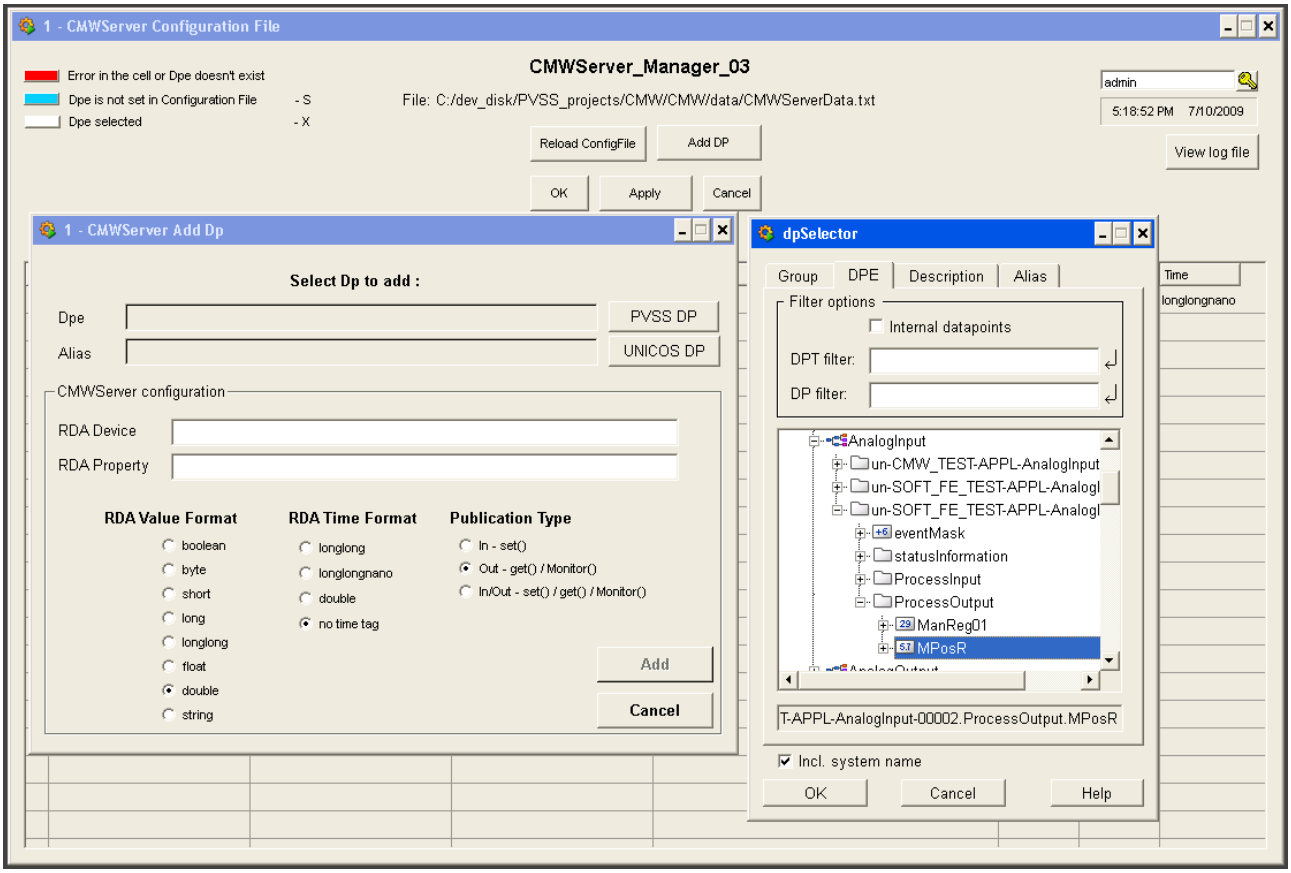

Figure 16: add device/property MPosR.

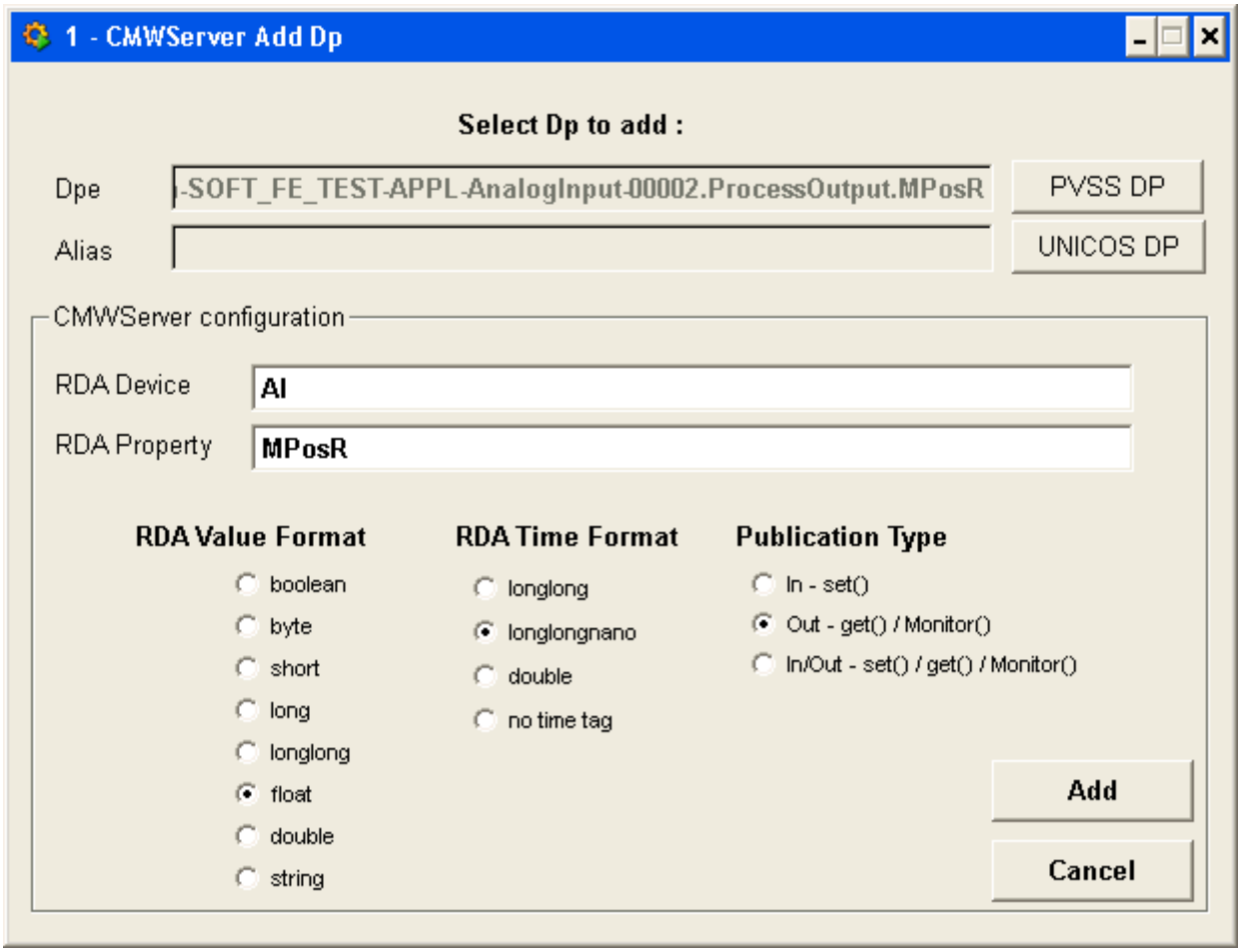

Figure 17: device/property MPosR config.

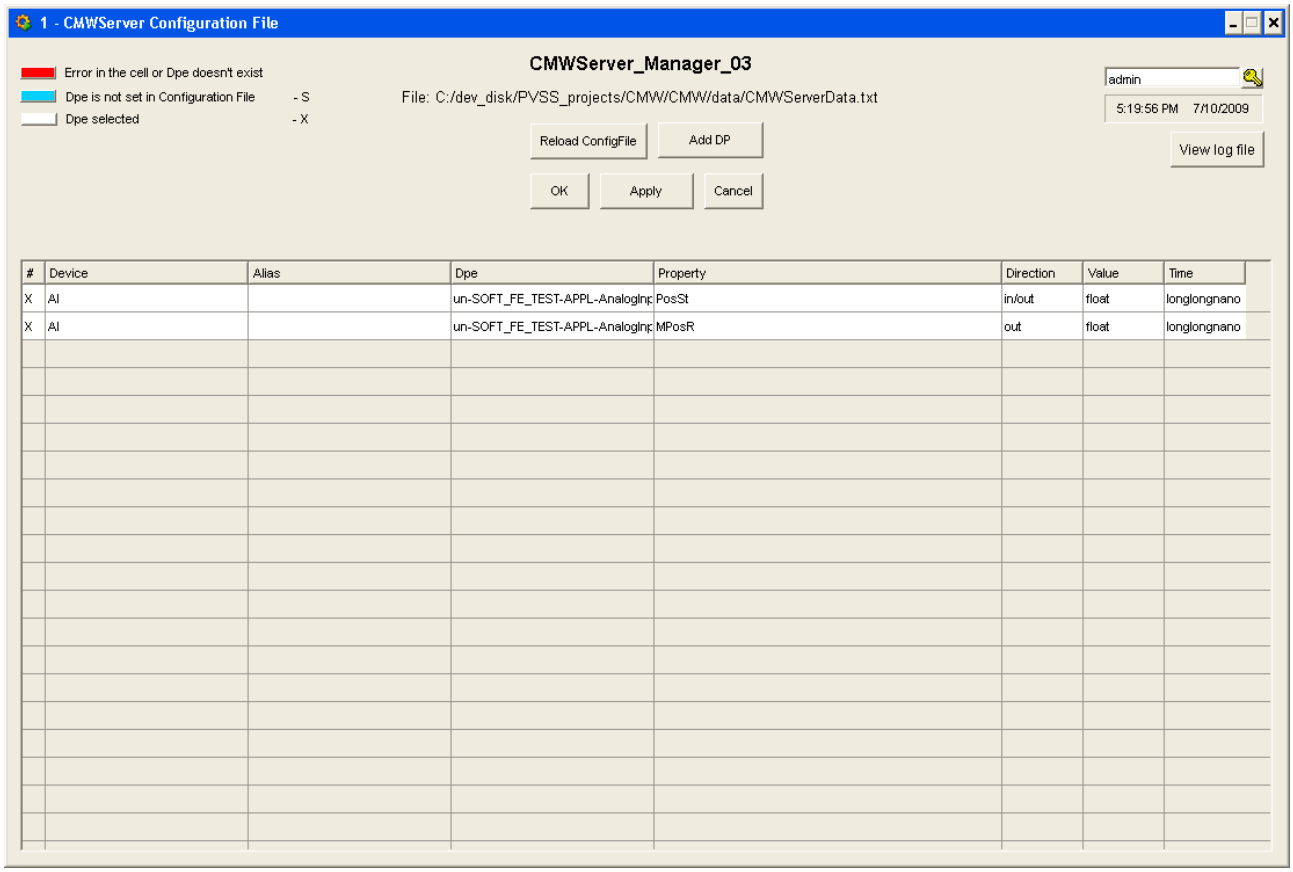

Figure 18: CMW Server device property configuration.

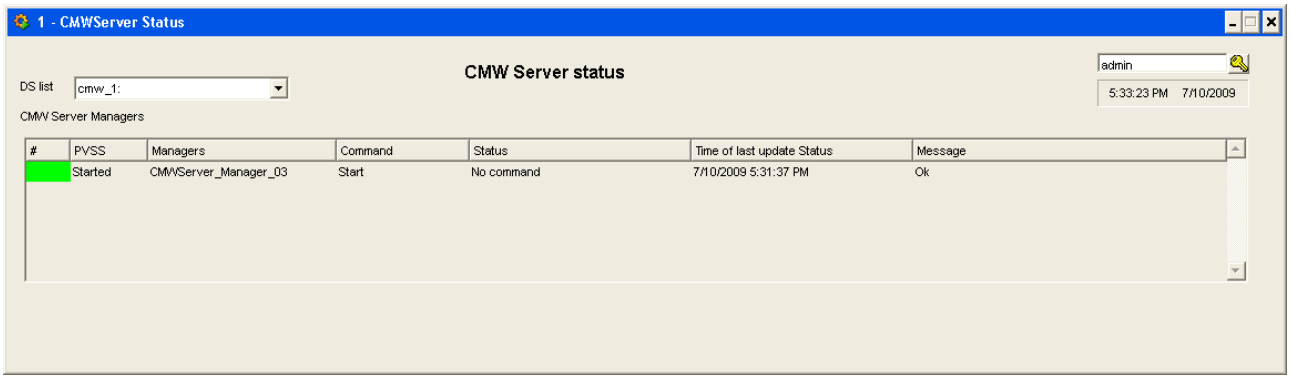

Figure 19: CMW Server status.

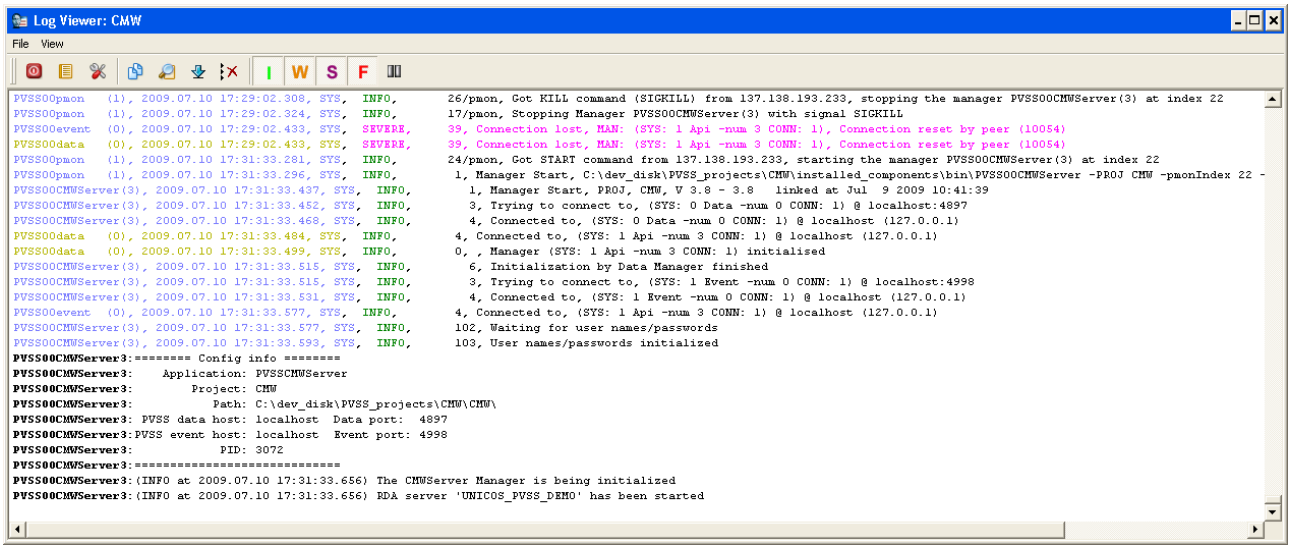

Figure 20: PVSS console log: PVSS00CMWServer.

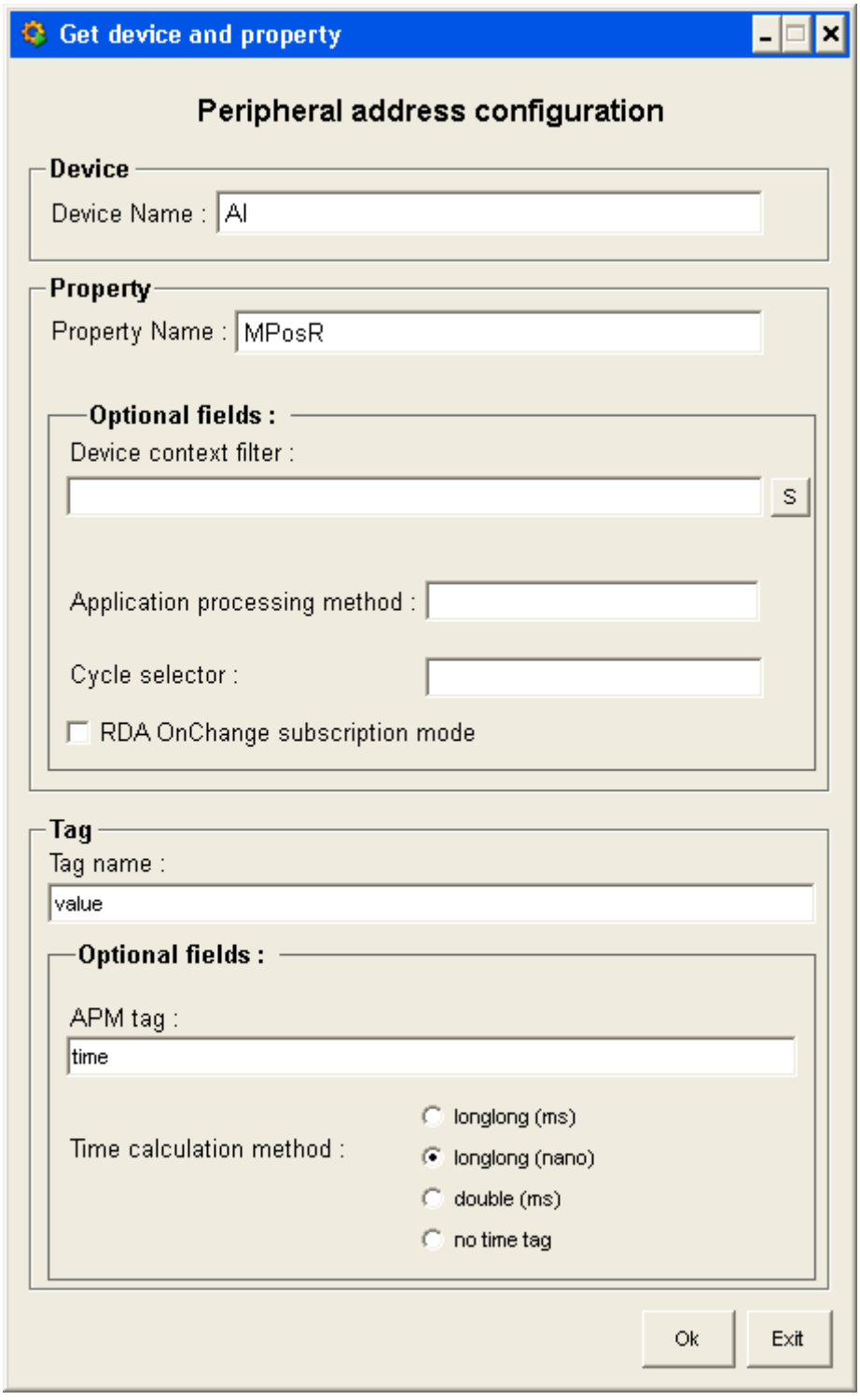

Figure 21: CMW PosSt address config, set device/property.

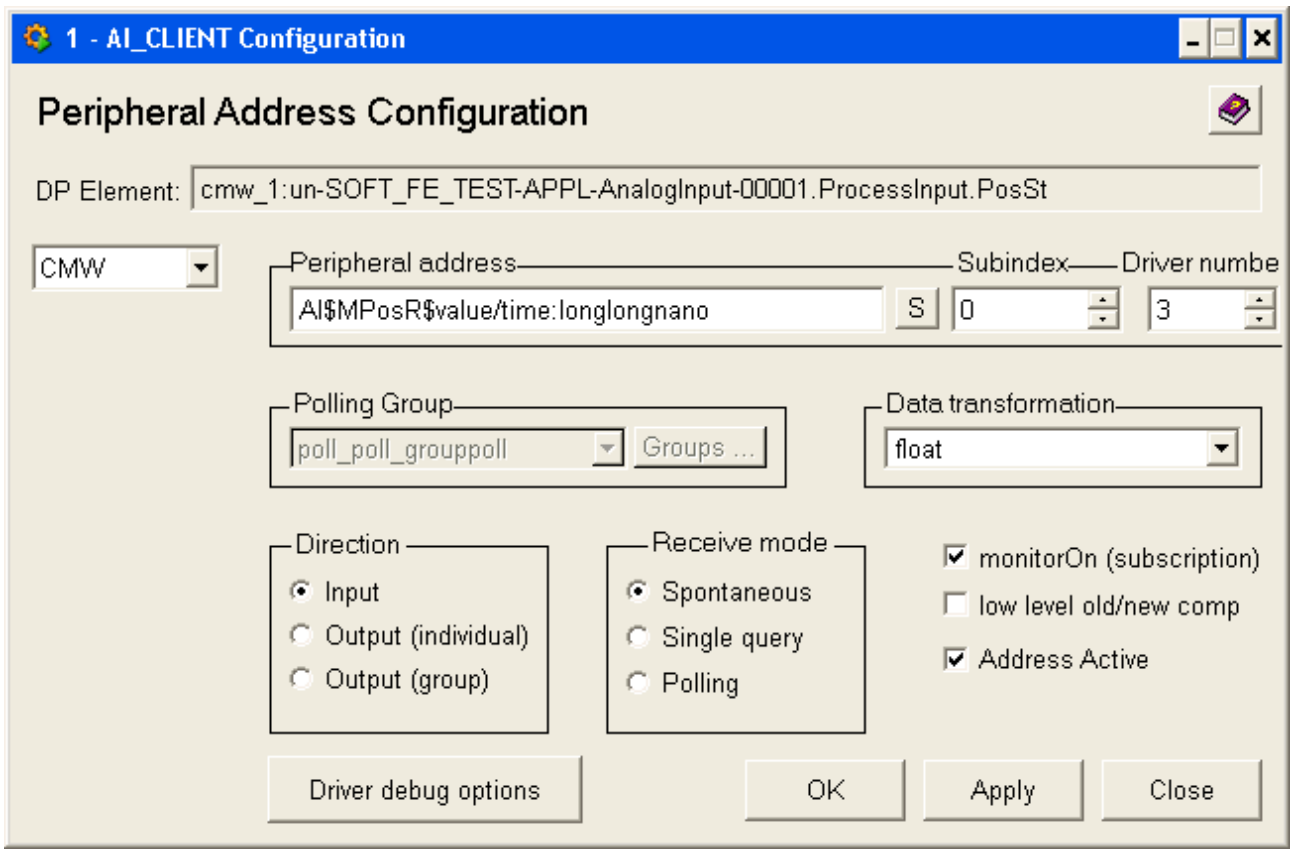

Figure 22: CMW PosSt address config.

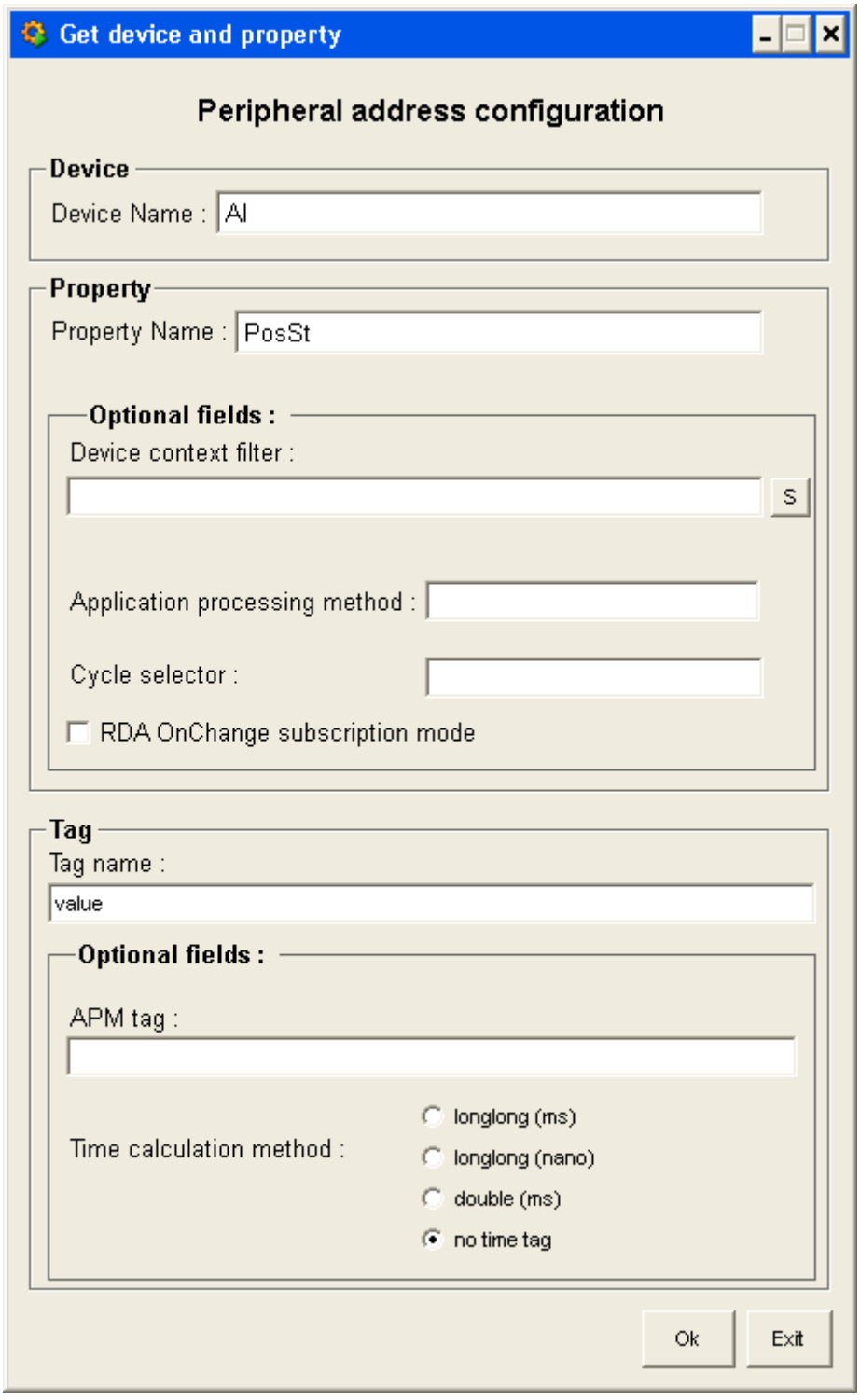

Figure 23: CMW MPosR address config, set device/property.

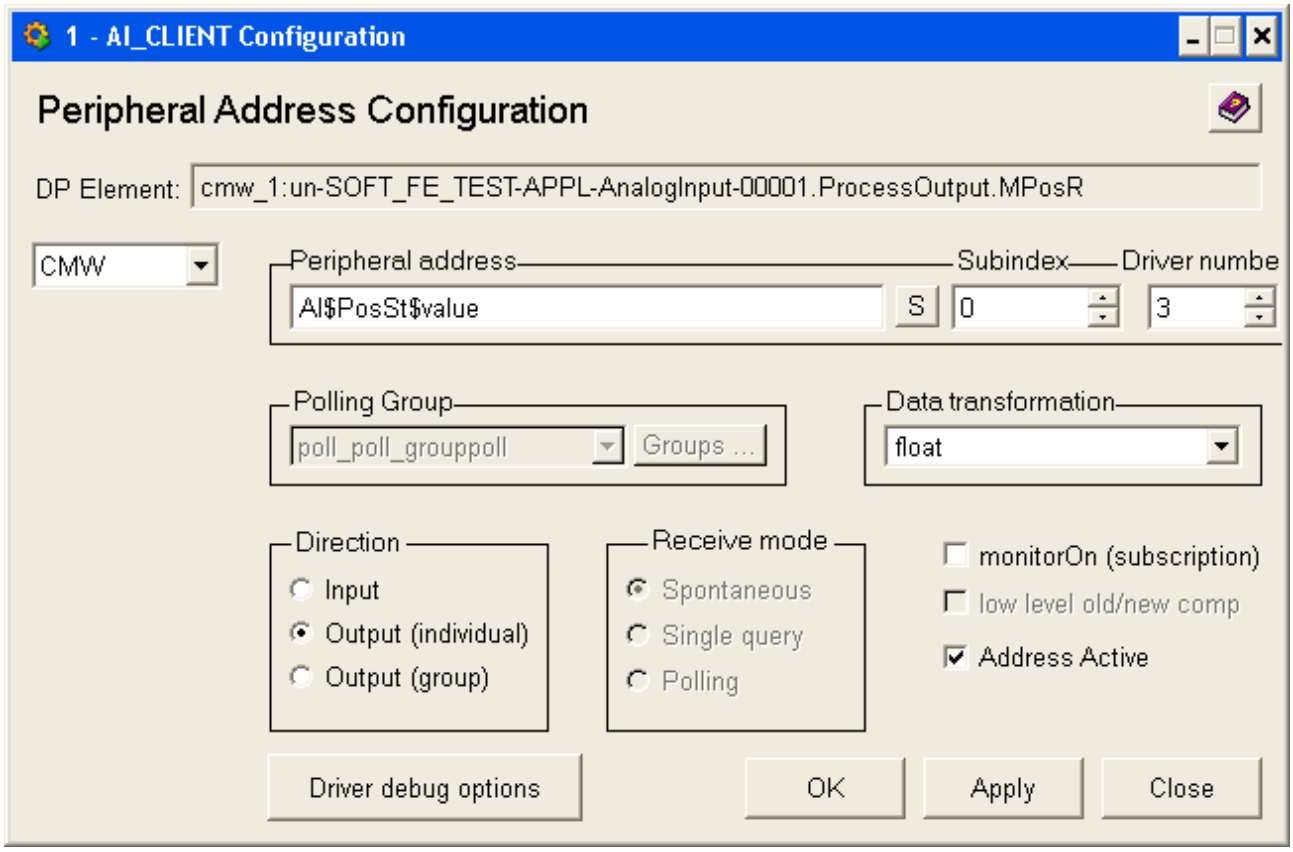

Figure 24: CMW MPosR address config.

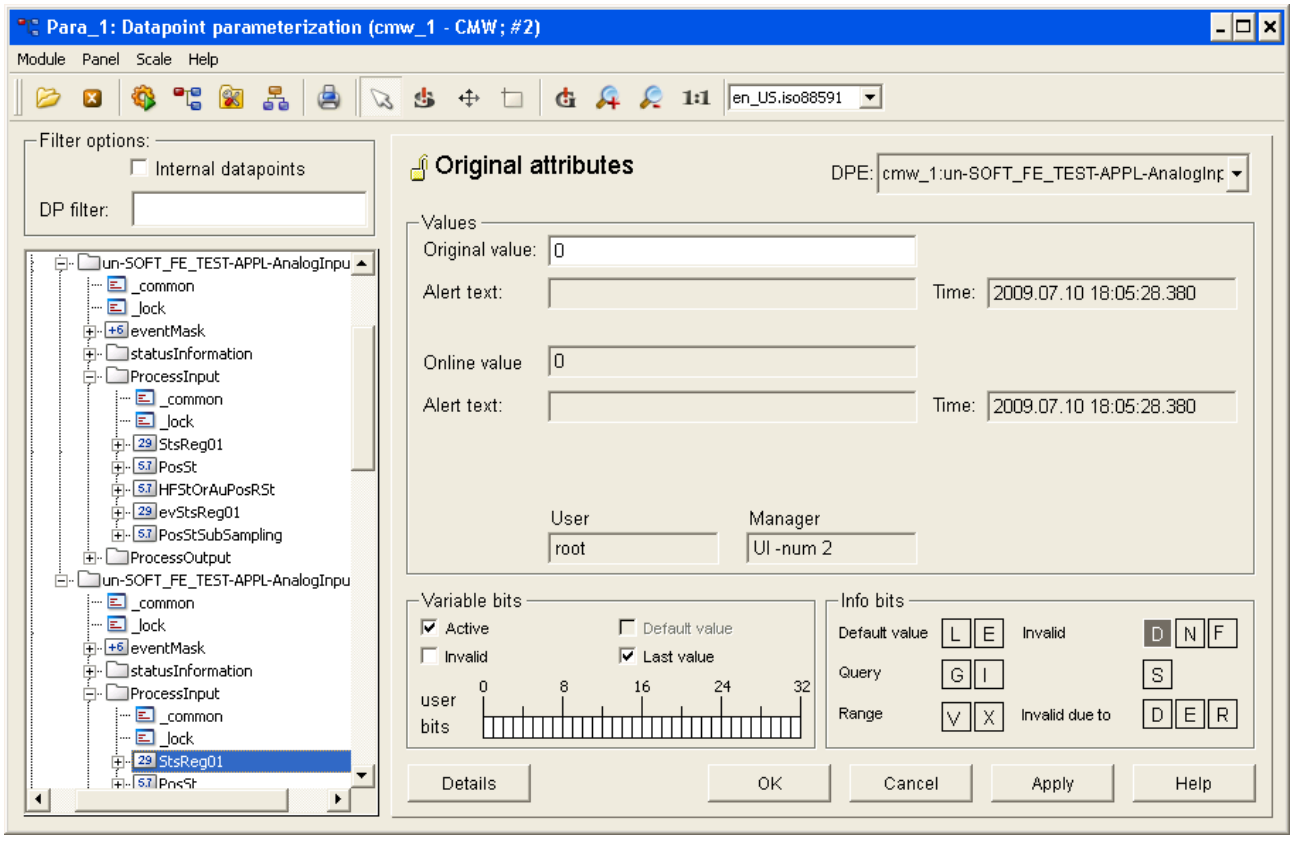

Figure 25: set StsReg01 to 0.

4. ADVANCED CONFIGURATION

SystemIntegrity can be set for both PVSS00CMWClient and PVSSCMWServer. Numerous options for debugging are provided. The PVSS00CMWServer can publish array of different types as in and out (set and monitorOn/get).

The PVSS00CMWClient:

- can use context filter in the address config of the DPE
- can use sub-index
- can subscribe to array of different types
- can detect if the device/property is not defined and set a DPE of type bool for the state of the device/property
- can be customized for high performance

Contact unicos.support@cern.ch for more information.

REFERENCE & USEFUL LINK

1. UNICOS: http://ab-project-unicos.web.cern.ch/ab-project-unicos/
2. CMW: http://proj-cmw.web.cern.ch/proj-cmw/

LIST OF FIGURES