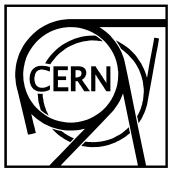
CERN
CH1211 Geneva 23
Switzerland



EN Engineering Department

EDMS NO.	REV.	VALIDITY
0000000	**1.0**	**DRAFT**

REFERENCE
XXXX

Date : 2011-05-04

UNICOS Device Access Control

Concept and example

This doc describes the concept.

DOCUMENT PREPARED BY:	DOCUMENT CHECKED BY:	DOCUMENT APPROVED BY:
Hervé Milcent: EN/ICE	[Checkers]	[Approvers]

HISTORY OF CHANGES

REV. NO.	DATE	PAGES	DESCRIPTIONS OF THE CHANGES

TABLE OF CONTENTS

1. Configuration

Log as root, in the access control setup, set Domain, Groups and user administration to the AuthControl privilege (Figure 1): only users in with AuthControl (typically root) will be allowed to configure the access control.

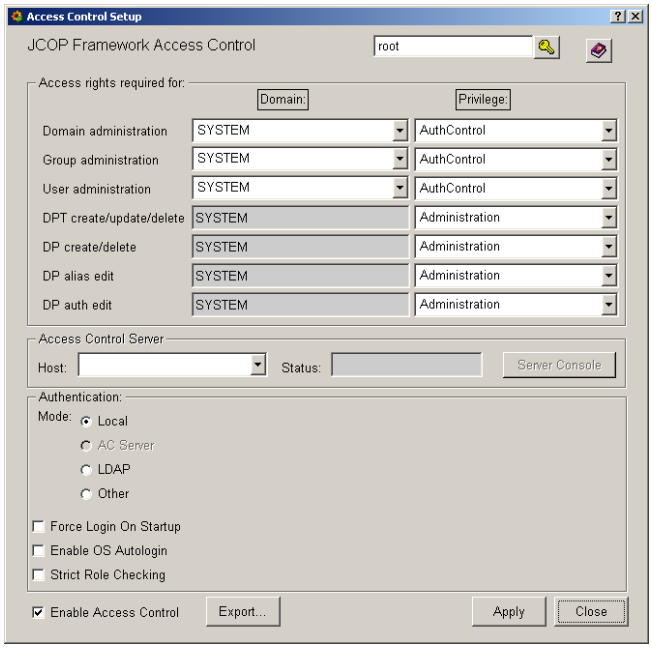


Figure 1: access control setup.

2. Test setup

2.1 Access control domain

– Define an access control domain (Figure 2): "VAC" with privileges

- 21: monitor
- 22: operator
- 23: expert
- 24: admin

– Define an access control domain: "VAC1" with privileges

- 21: monitor
- 22: operator
- 23: expert
- 24: admin

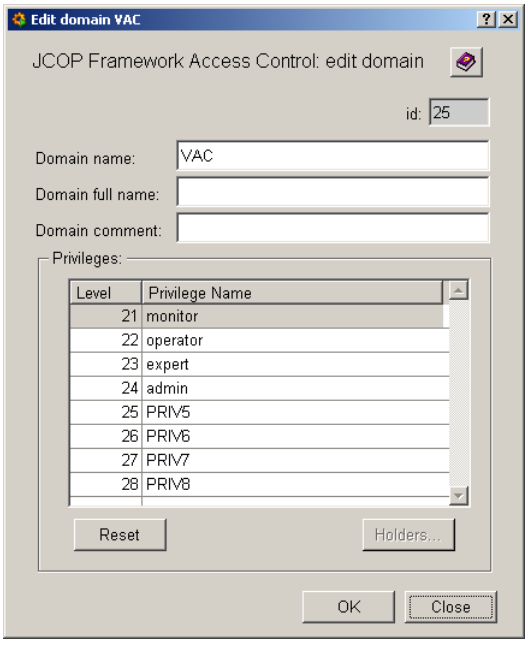


Figure 2: domain.

2.2 Group

– Define group (Figure 3)

- VAC_operator with privilege VAC:operator and SYSTEM privilege id from 1 to 5
- VAC_expert with privilege VAC:expert and SYSTEM privilege id from 1 to 5
- VAC_admin with privilege VAC:admin and SYSTEM privilege id from 1 to 5
- VAC1_operator with privilege VAC1:operator and SYSTEM privilege id from 1 to 5
- VAC1_expert with privilege VAC1:expert and SYSTEM privilege id from 1 to 5
- VAC2_admin with privilege VAC1:admin and SYSTEM privilege id from 1 to 5

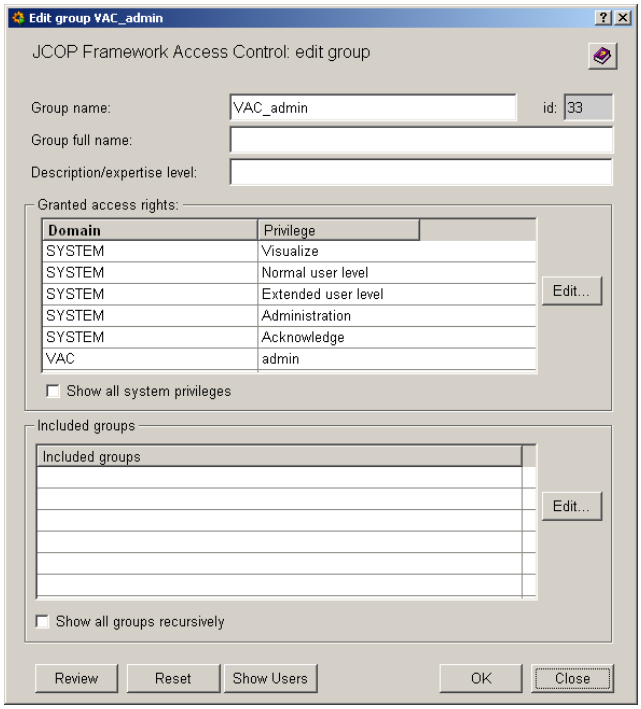


Figure 3: group.

2.3 User

– Define user (Figure 4)

- user_vac_operator in group VAC_operator
- user_vac_expert in group VAC_expert
- user_vac_admin in group VAC_admin
- user_vac1_operator in group VAC1_operator
- user_vac1_expert in group VAC1_expert
- user_vac1_admin in group VAC1_admin

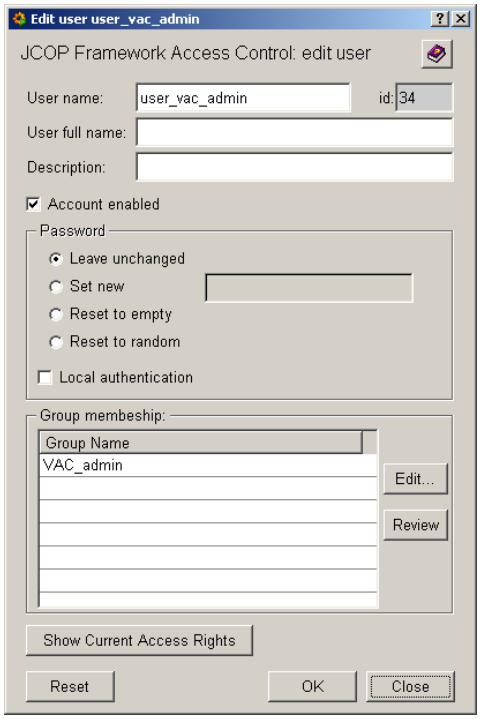


Figure 4: user.

3. Device

1. Create a dpType (Figure 5) and add a DPE statusInformation under root: type ref to _UnStatusInformation
2. Create a DP: vac_rr_1
3. Create a DP: vac_rr_2 and set (Figure 6):

- vac_rr_2.statusInformation.accessControlDomain to VAC1,VAC2
- vac_rr_2.statusInformation.operator to oper_custom1
- vac_rr_2.statusInformation.expert to expert_custom1,expert_custom2
- vac_rr_2.statusInformation.admin to admin_custom1,admin_custom2,admin_custom3

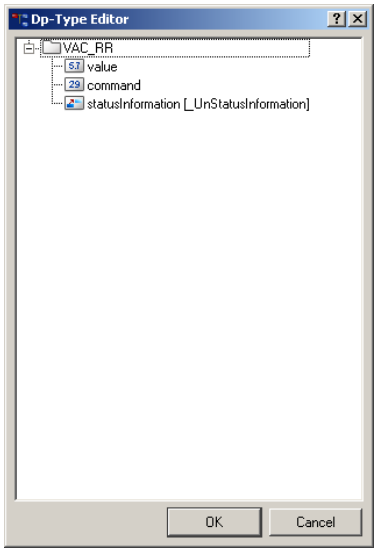


Figure 5: DP type.

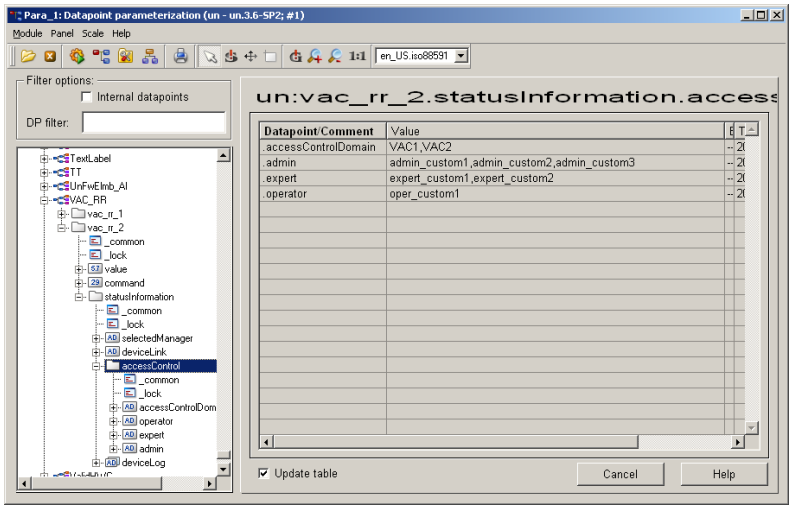


Figure 6: vac_rr_2

4. UI

4.1 design

In the panel which is not opened by unicosHMI you must add the two following lines:

– dpConnect("unGraphicalFrame_domainCB", "_Areas.UserName");

– g_dsPrivilegeAccessControl = makeDynString("monitor","operator","expert","admin");

Animation of the device action, for instance in a panel, put I the event Intialize a call to

unGenericDpFunctionsHMI_setCallBack_user("myUserAccessCB",iRes, exceptionInfo);

with the following

```
const string MY_DEFAULT_ACCESS_CONTROL_DOMAIN = "VAC";


myUserAccessCB(string sDp, string sUser)
{
  dyn_string dsAccess;


  myDeviceUserAccess("vac_rr_1", dsAccess);
  DebugN("vac_rr_1, list of allowed actions: ", dsAccess);
  myDeviceUserAccess("vac_rr_2", dsAccess);
  DebugN("vac_rr_2, list of allowed actions: ", dsAccess);
}
myDeviceUserAccess(string sDpName, dyn_string &dsAccess)
{
  bool operator, expert, admin, monitor;
  dyn_bool dbPermissions;
  dyn_string exceptionInfo;
  string sOperAction, sExpAction, sAdminAction;
  bool bActionDefined = false;


  unGenericDpFunctions_getAccessControlPriviledgeRigth(sDpName, g_dsDomainAccessControl, dbPermissions,
sOperAction, sExpAction, sAdminAction, exceptionInfo, MY_DEFAULT_ACCESS_CONTROL_DOMAIN);
  if((sOperAction != UNICOS_ACCESSCONTROL_DOMAINNAME) || (sExpAction !=
UNICOS_ACCESSCONTROL_DOMAINNAME) ||
        (sAdminAction != UNICOS_ACCESSCONTROL_DOMAINNAME))
    bActionDefined=true;
  monitor = dbPermissions[1]; // always true
  operator = dbPermissions[2];
  expert = dbPermissions[3];
  admin = dbPermissions[4];
  dsAccess = makeDynString();


  if(operator)
  {
    if(bActionDefined)
      dynAppend(dsAccess, strsplit(sOperAction, UN_ACCESS_CONTROL_SEPARATOR));
    else
      dynAppend(dsAccess, makeDynString("oper1", "oper2"));
  }
```

```
  if(expert)
  {
   if(bActionDefined)
     dynAppend(dsAccess, strsplit(sExpAction, UN_ACCESS_CONTROL_SEPARATOR));
   else
     dynAppend(dsAccess, makeDynString("exp1", "exp2", "exp3"));
  }
  if(admin)
  {
   if(bActionDefined)
     dynAppend(dsAccess, strsplit(sAdminAction, UN_ACCESS_CONTROL_SEPARATOR));
   else
     dynAppend(dsAccess, makeDynString("admin1", "admin2"));
  }
 }
```

4.2 Test

Log in as :

– user_vac_operator, result in log

```
PVSS00ui1:["vac_rr_1, list of allowed actions: "][dyn_string 2 items
PVSS00ui1:    1:"oper1"
PVSS00ui1:    2:"oper2"
PVSS00ui1:]
PVSS00ui1:["vac_rr_2, list of allowed actions: "][dyn_string 0 items
PVSS00ui1:]
```

– user_vac_expert, result in log:

```
PVSS00ui1:["vac_rr_1, list of allowed actions: "][dyn_string 3 items
PVSS00ui1:    1:"exp1"
PVSS00ui1:    2:"exp2"
PVSS00ui1:    3:"exp3"
PVSS00ui1:]
PVSS00ui1:["vac_rr_2, list of allowed actions: "][dyn_string 0 items
PVSS00ui1:]
```

– user_vac_admin, result in log:

```
PVSS00ui1:["vac_rr_1, list of allowed actions: "][dyn_string 2 items
PVSS00ui1:    1:"admin1"
PVSS00ui1:    2:"admin2"
```

PVSS00ui1:]

PVSS00ui1:["vac_rr_2, list of allowed actions: "][dyn_string 0 items

PVSS00ui1:]

– user_vac1_operator, result in log:

PVSS00ui1:["vac_rr_1, list of allowed actions: "][dyn_string 0 items

PVSS00ui1:]

PVSS00ui1:["vac_rr_2, list of allowed actions: "][dyn_string 1 items

PVSS00ui1: 1:"oper_custom1"

PVSS00ui1:]

– user_vac1_expert, result in log:

PVSS00ui1:["vac_rr_1, list of allowed actions: "][dyn_string 0 items

PVSS00ui1:]

PVSS00ui1:["vac_rr_2, list of allowed actions: "][dyn_string 2 items

PVSS00ui1: 1:"expert_custom1"

PVSS00ui1: 2:"expert_custom2"

PVSS00ui1:]

– user_vac1_admin, result in log:

PVSS00ui1:["vac_rr_1, list of allowed actions: "][dyn_string 0 items

PVSS00ui1:]

PVSS00ui1:["vac_rr_2, list of allowed actions: "][dyn_string 3 items

PVSS00ui1: 1:"admin_custom1"

PVSS00ui1: 2:"admin_custom2"

PVSS00ui1: 3:"admin_custom3"

PVSS00ui1:]

– logout, result in log:

PVSS00ui1:["vac_rr_1, list of allowed actions: "][dyn_string 0 items

PVSS00ui1:]

PVSS00ui1:["vac_rr_2, list of allowed actions: "][dyn_string 0 items

PVSS00ui1:]

You can configure as many domain, group and user as you want. You can also put a user in many groups. You can also combine the myDeviceUserAccess with the state of the device (action are enabled if the device is in a given sate, etc.)

5. Concept

The internal datapoint:

- statusInformation.accessControlDomain: list of domain in which a user has to be in in order to be able to act ona device
- statusInformation.operator: list of allowed action for a user having the operator privilege in the statusInformation.accessControlDomain
- statusInformation.expert: list of allowed action for a user having the expert privilege in the statusInformation.accessControlDomain
- statusInformation.admin: list of allowed action for a user having the admin privilege in the statusInformation.accessControlDomain

If all these fields are empty, the default will be taken, implemented by:

```
if(operator)
{
  if(bActionDefined)
    dynAppend(dsAccess, strsplit(sOperAction, UN_ACCESS_CONTROL_SEPARATOR));
  else
    dynAppend(dsAccess, makeDynString("oper1", "oper2"));
}
```

If you put in statusInformation.operator, statusInformation.expert, statusInformation.admin the name of the button in a panel, you can then directly do setValue("myButton", "enabled", dynContains(dsAccess, "myButton")>0);

As a rule in UNICOS (but it is not implemented in any code, it is just a convention), a user having the expert operator privilege has also the monitor one, a user having the expert privilege has also the monitor and operator one, and a use having the admin one has also the expert, operator and monitor one.

There is no place for the monitor as by convention monitor is just for reading. However if it is required one can use the function unGenericDpFunctions_getAccessControl4PriviledgeRigth instead of unGenericDpFunctions_getAccessControlPriviledgeRigth but there is no way to customize the monitor actions of a device.

6. Future evolution

Currently, the limitations are the following:

- Maximum 4 privileges and named monitor, operator, expert and admin
- The action are defined for a given privilege and for as many domain as needed.

For the end of the year it is planned to implement:

– Definition of the allowed action per domain:priv, for instance one could define admin_custom3=VAC1:admin,VAC:expert meaning that one has to have the expert privilege in the VAC domain or the admin privilege in the VAC1 domain.
– Allow up to 8 privileges.
– Allow different privilege name than monitor, operator, expert and admin.

Table of Figures