



# PVSS-UNICOS Module 1

---



# Module – 1

---

- ➔ ■ Overview of PVSS
  - presentation of the most important concept
  - PVSS manager
- UNICOS-PVSS
  - Concept, principle
  - Internal organization
  - Delivery, release plan of work, future work
  - Description of the UNICOS components



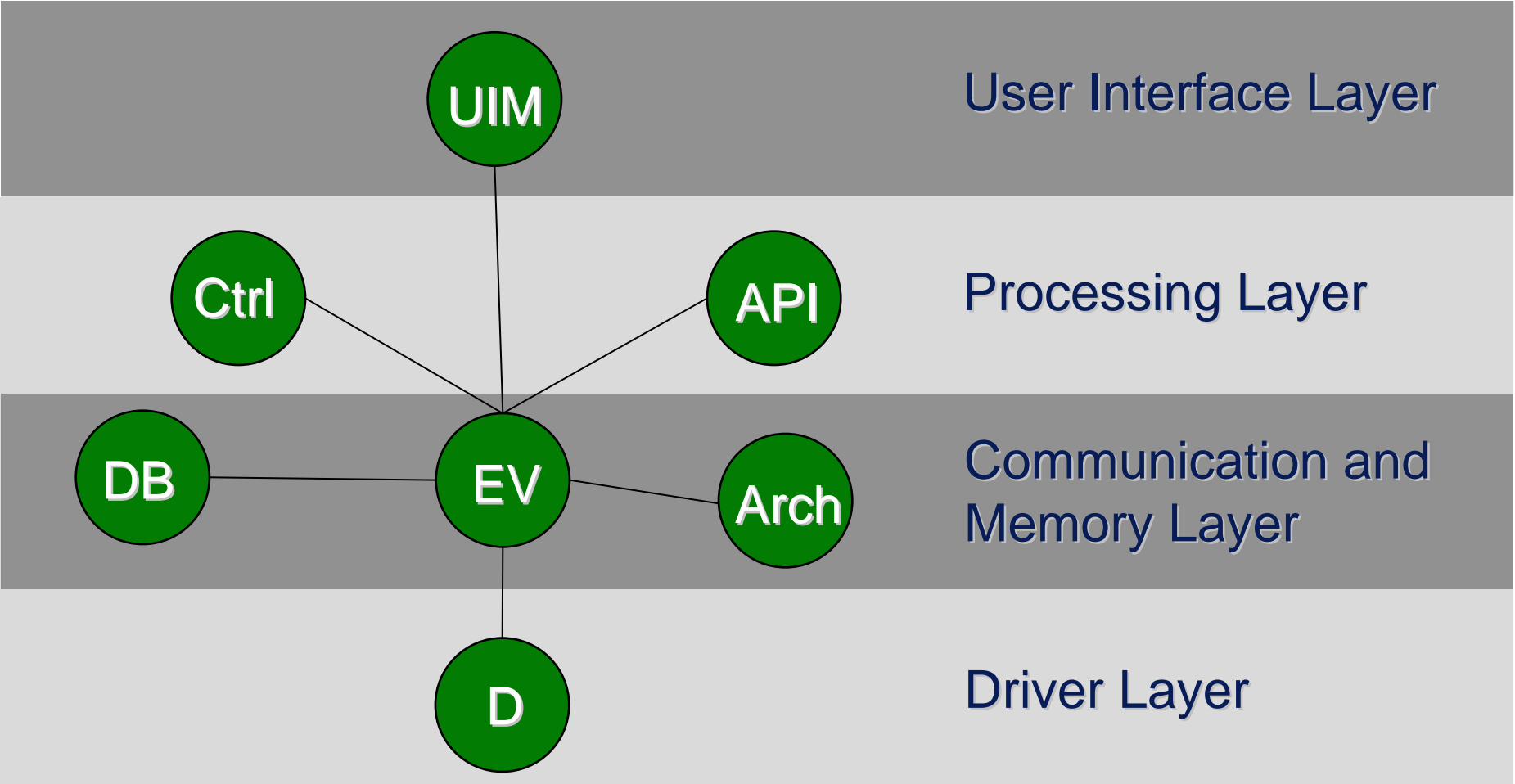
## PVSS II

---

- Toolkit for development
  - Like BridgeView
  - Interpreter of PVSS scripting language: C like
- Open:
  - New feature easily added
- Few configuration tools
- Permanent data base
  - On disk
  - Data are persistent
  - At startup, latest saved value before shutdown is restored



# Manager overview –1





## Manager overview – 2

---

- Connection between manager via tcp-ip
  - Configurable port number
  - No secure socket
  - Hackers can corrupt the system
- Each manager of same type connected to the Ev must have a unique number
- PVSS is considered event driven
- But...
  - Internally this is a polling system
  - All the managers inherit from base class:
    - A wait is done periodically on the socket connections
    - Any message on the socket will exit the wait and will be dispatched.
- All the messages are time stamped and processed in order of reception



# Platform

---

- Linux and Windows
  - Same PVSS script on both platform
- UIM:
  - Motif on linux platform
  - Windows activeX only on windows platform



# PVSS vocabulary – 1

---

- Data point type (DPT): C structure like
- PVSS system: Ev+Db
- Data point (DP): instance of a data point type, unique within a system
  - Cannot be renamed
  - DP deleted → archive lost event if created again afterwards
- Data point element (DPE): node or leaf node of a data point
  - Basic: struct, int, float, string, bool, etc.
  - Arrays: basic element of same type can be added
  - Dyn list
- Config on data point element
  - To define the behavior of the data point element
- Alias on data point and data point element
  - Can be changed, no impact on the archive



## PVSS vocabulary – 2

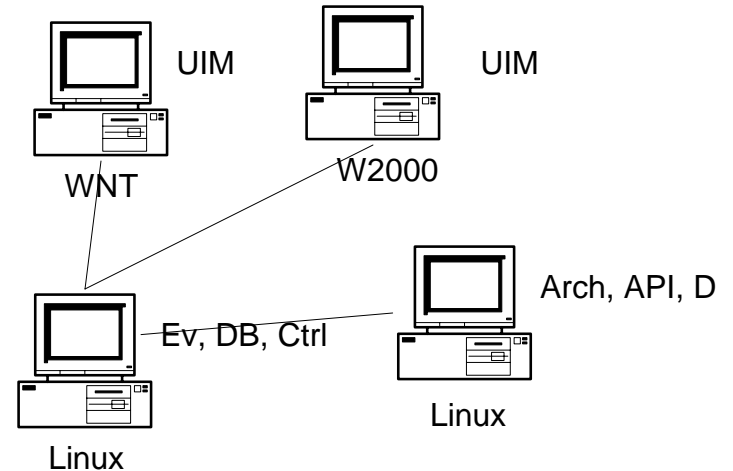
---

- Panel: graphical view (window in PcVue, VI in LabVIEW):
  - Script interpreted, valid only in the panel
  - No panel exchange of data within panels at run-time except through the Db.
  - Can have parameters, evaluated at startup
- Script: sequence of PVSS function and code
- Script: when executed by Ctrl manager it is a file containing a sequence of PVSS function and code with a main function.
- Library: function reusable in the panels, in other library or in the scripts
- Project path: set of folder to locate the panel, script, etc.
  - Modified file always saved in the last project path
  - First occurrence found is used.

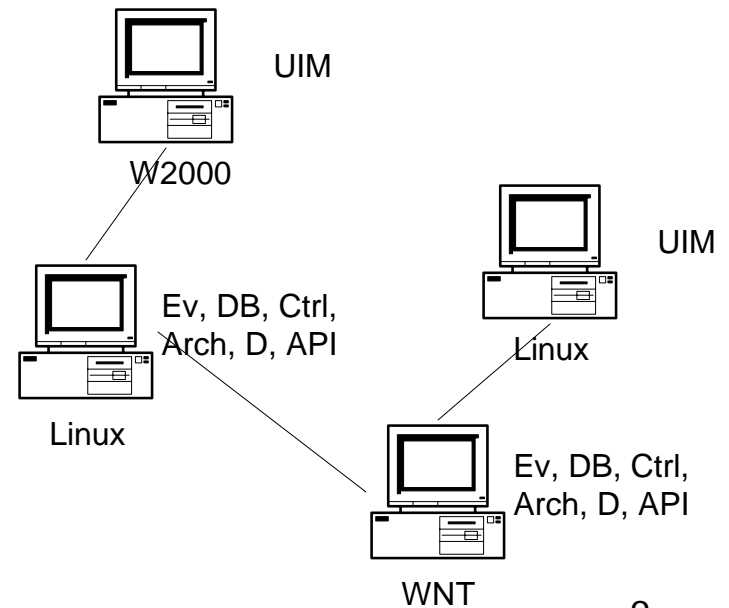


# PVSS Manager vocabulary – 1

- PVSS scattered system:  
1 Ev, 1 Db, and other managers distributed

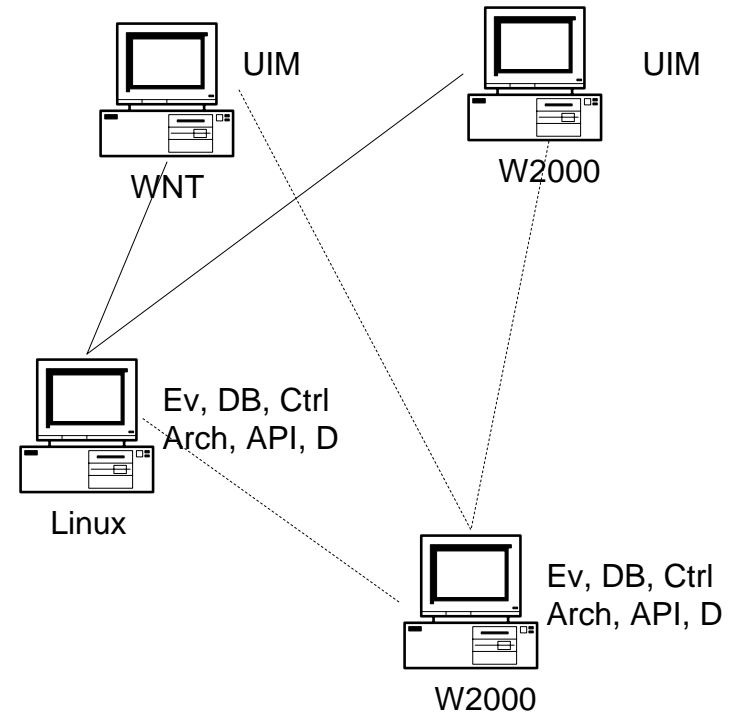


- PVSS distributed system:  
many scattered systems interconnected via a Dist:  
distributed manager



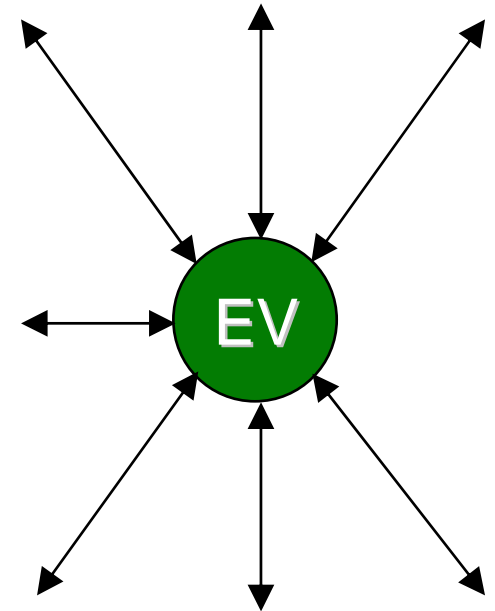
# PVSS Manager vocabulary – 2

- PVSS redundant system:
  - Via a Rm: redundant manager
  - One passive, one active
  - Software redundancy
  - Redundancy of all the managers
  - Distributed system
- PVSS project:
  - Scattered, distributed, redundant
  - config file
    - per scattered system, read at startup of the manager
    - A remote UIM can also have a config file
    - Section per manager in the config file



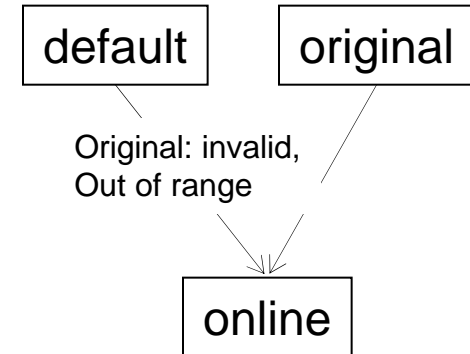
# Event manager: Ev

- Administrator of:
  - data point, data point element, configs
  - User access
  - Current process image of the data point
- Receive, evaluate and distribute the message
- Bottleneck of the system
  - Any request (set, get, connect, etc.) is via the Ev.
- Redundancy:
  - the passive Ev blocks message,
  - Update of the passive Ev by the active Ev



# Ev data point element config

- Value:
  - Original: from hardware, set by operator
  - Online: the one currently used
  - Default: default value for the online config if original is invalid or out of PVSS range
  - Value time stamped
- Pvss range: range of the value, invalid if outside





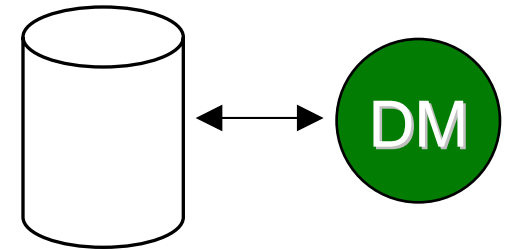
## Ev: other config

---

- Alert\_class: define the behavior of the alert
  - Archive, acknowledge, color, etc.
- Alert\_hdl: the alert itself
- Dp\_fct: to do calculation based other data point element:
  - Calculation, average, log, sum, etc.
- Offline: retrieve the archived value
- Archive: define the archiving
- Corr: to add correction value to an archived data

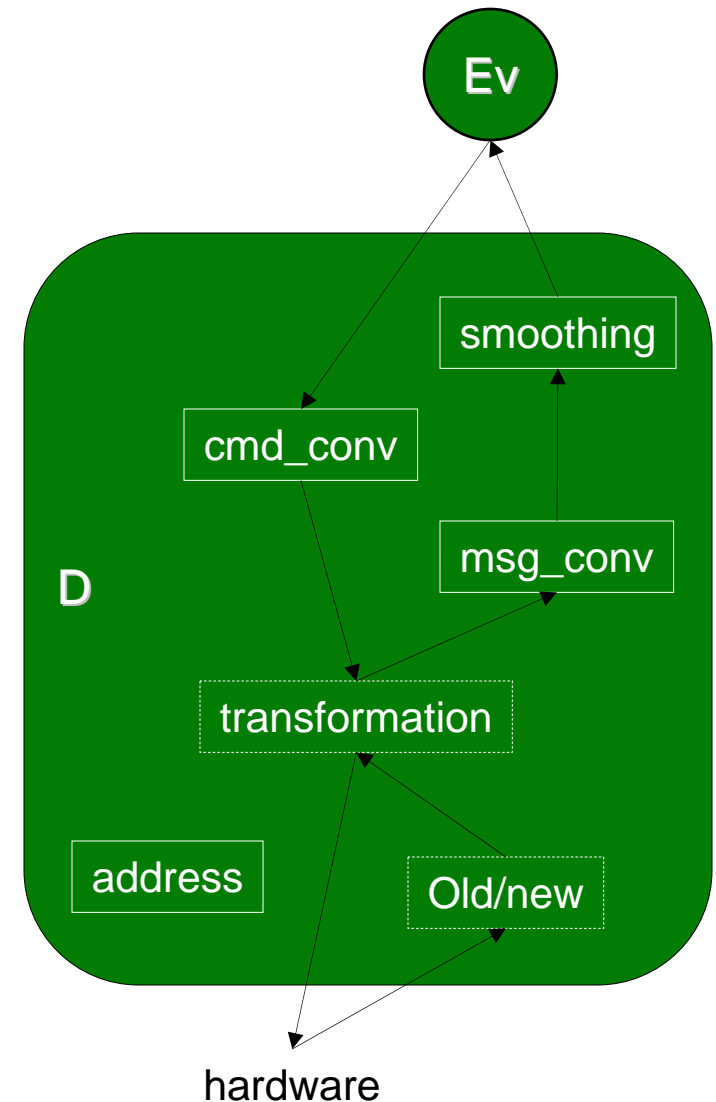
# Data base manager: Db

- Proprietary database:
  - Last value, configs
  - Archive of configs
  - Access through the Ev, no direct access
- SQL query supported
- At startup
  - the last value read from the data base and restored
  - The configs are sent to the managers
- Archive the alarm data
  - Purge of old alarm
- Archive history of configs → disk space problem
  - Not possible to delete old config from the DB
- Redundancy: via the Ev



# Driver manager – 1

- Manage the connection to the hardware
- Transformation to correct DPE type
- Configs hold by the driver:
  - Address, distrib: connection to the hardware via a driver
  - Smoothing: filtering, reduction of data flow with the Ev
  - Msg\_conv: raw to engineering conversion
  - Cmd\_conv: engineering to raw conversion
  - Data point element either from or to hardware, not both





## Driver manager – 2

---

- Startup:
  - Input DPE are read from hardware and sent to Ev (no smoothing, no old/new)
  - Ev is not sending the output DPE to the driver
- Single query of the DPE value (DPE as In)
- General query of all the DPE (DPE as In) hold by the driver





## Driver limitation

---

- Too many data from the hardware at too high rate may lead to a memory increase:
  - Ev too slow (generally Db)
  - The memory buffers increase up to a limit
    - Then only one value per data
    - Loss of value
- Driver not started, no access to the driver DPE config



# Ctrl Manager

---

- PVSS script: C like
- Interpreter
- Multithread
- Access to all the data point value, config
- Execution of process program





# API manager

---

- C++
- Same functionality as Ctrl but in C++
- Allow link to existing C++ software
- Integrated with PVSS functions





# Archive manager

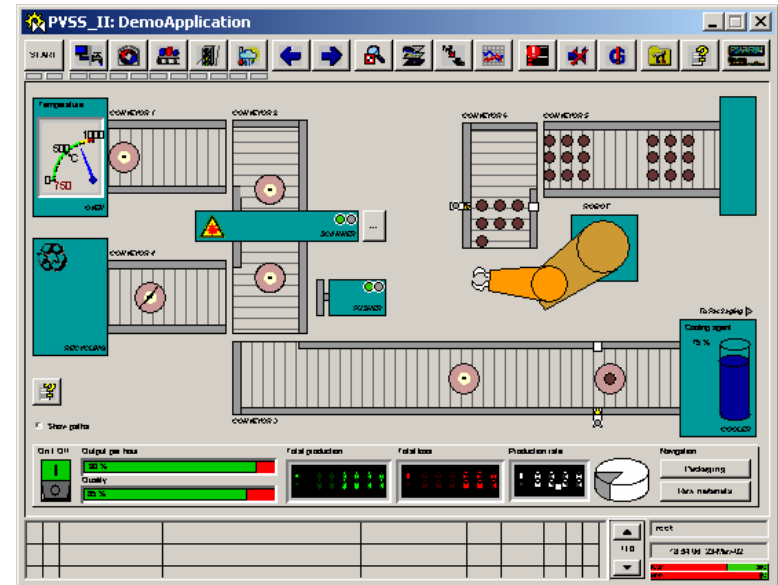
---

- Archive the data not the alarm
- Multi threaded
- One manager per archive class
- Handle the archive behavior
  - Switch archive files
  - Move the archive file to backup
  - Compress archive files
- Handle the archive retrieval

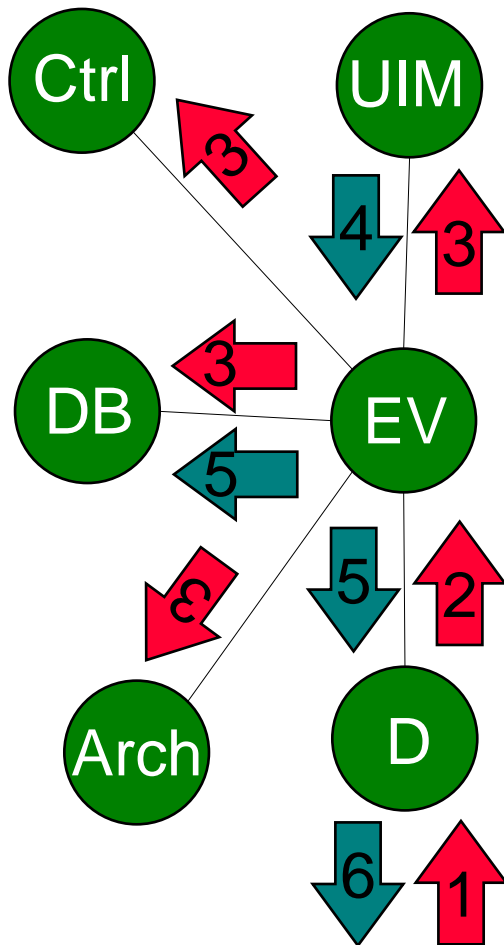


# UIM

- Graphical interface:
  - run panels, scripts
  - Animation of graphic object (event driven, polling)
  - Receive and send data to the Ev
  - Trend
  - Etc.
- User access
- 2 tools:
  - Development:
    - GEDI: panel, script and library development
    - PARA: data point configuration, creation, delete
  - Run time:
    - no panel development, no access to the PARA tool
    - Can run a panel that does data point configuration



# Message flow



- 1 Hardware sends data for value change
- 2 Driver creates PVSS message
- 3 EV writes altered values to respective DP and sends updated DP to DB, Arch, Ctrl and UIM
  - 3.a UIM: creates visualization of value change
  - 3.b Ctrl: carries out calculations with new DP value
  - 3.c DB: archives value change
- 4 User reacts to value change and sends value change to for the hardware
- 5 EV processes user reaction and sends value change to DB for logging and to the driver
- 6 Driver converts value to a hardware



## Other PVSS features

---

- Groups of data point
- Recipes
- Panel hierarchical navigation
- ActiveX
- Web server
- ODBC
- OPC Client, OPC Server
- MODBUS/UNICOS driver
- DB backup (ORACLE, etc.) integrated via ODBC
- Excel macro
- Etc.



# PVSS scripting language – 1

---

- Similar to C
  - ; as the end of execution line
- Interpreted
  - No compilation, no link
  - Typical errors:
    - Wrong function call, wrong variable name, etc.
- Debugging:
  - Script debugger: not so easy to use, basic
  - Printout function
- Multithreaded:
  - no exclusive access on a variable (can be implemented)





## PVSS scripting language – 2

---

- Global variable to a manager or to a panel
- Constant, dynamic list, array, basic elements, etc.
- Callback function trigger on one or many value change (dpConnect function)
  - Sequentially called
  - Data point name and value given to the function
  - Should be fast
  - Not advised to query data because no coherency is guaranteed with the data given to the callback function
  - UIM: Number of pending call can be limited: the last one is executed.



# Identification of the data

---

[system:]dpName.[dpElement(s)]:config.[detail].attribute

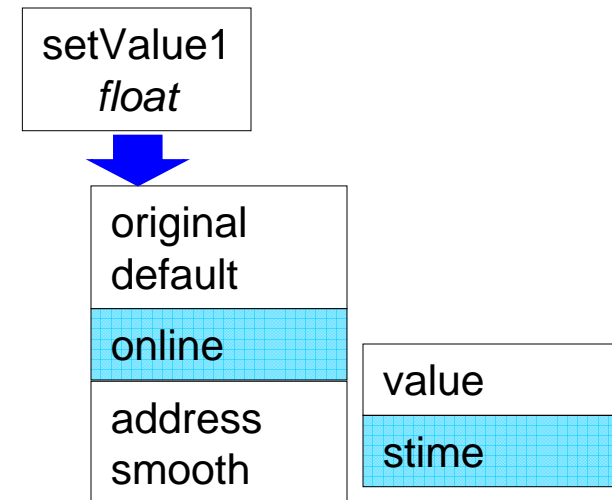
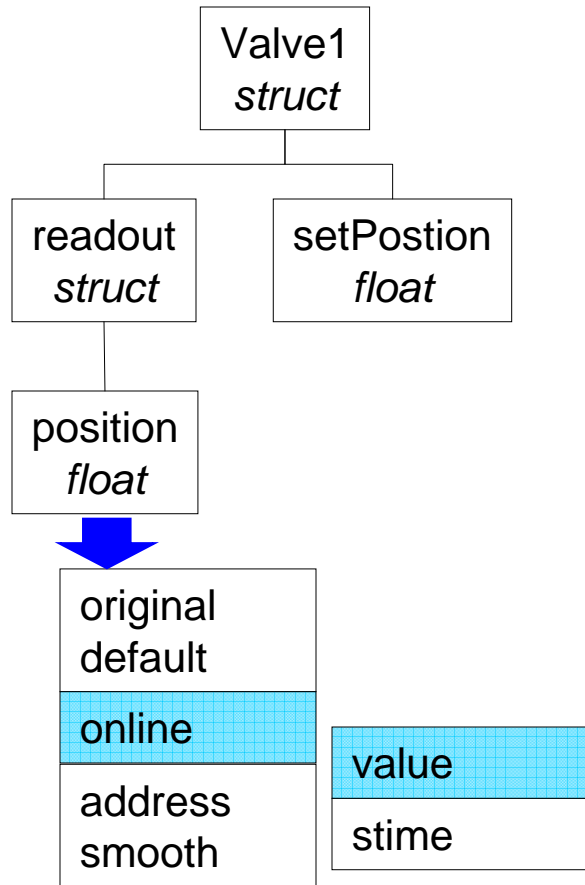
- system: name of the PVSS system
- dpName: data point name
- dpElement(s): data point element name(s)
- config: the type of config
- detail: number of the config
- attribute: attribute of the config

[] optional

Several dpElement are separated by .

In a distributed system, if system is not specified the current system name is used

# Example of data identification



System1:setValue1.:\_online..\_stime  
setValue1.:\_online..\_value

System1:valve1:readout.position:\_online..\_value  
valve1.readout.position:\_online..\_value



# Panel development

---

- Graphical view inside a module (container of a panel)
  - Child panel, root panel, etc.
  - Contains PVSS script: library call, etc.
- Typical drawing tools:
  - Default font, style, layer
- Basic graphical element:
  - Button, comboBox, text field, radio button, etc.
  - Each object has a unique name
- Add ActiveX:
  - Embed an ActiveX
- Add reference (symbol): generic panel
  - Embed an existing PVSS panel: executed in the context of the parent panel.
- The panel and the graphical attribute have associated events and attributes



# Panel event

---

- (general): global function of the panel, not known by the embedded panels
- EventClick: when one does a click in the panel
- EventClose: when one closes the module (x)
- EventDbClick: when one double click in the panel
- EventInitialize: at startup of the panel
- EventRightClick: when one does a right click
- EventTerminate: when the panel is close (e.g.: panelOff function call)
- EventZoom: when zooming



# Basic object event

---

- EventAcknowledge: when one acknowledge an alarm when in PVSS acknowledge mode
- EventClick: when one clicks on the object
- EventInitialize: at start of the panel
- EventRightClick: when one does a right click on the object

And other depending on the object:

- EventCommand: when one does enter a value
- EventChange: when one changes a position
- EventColumnCommand: when one changes a value in a table
- EventMouseDown, EventMouseUp: mouse up and down on a button
- ActiveX event: event published by the ActiveX
- Etc.



# Object attribute – 1

---

- Static attribute
  - E.g.: name
- Modifiable at run-time:
  - Visible, enabled, color, text, value etc.
- Access by:
  - setValue, getValue function call
  - . notation:
    - val = object1.text() equivalent to getValue
    - object1.text(val) equivalent to setValue



## Object attribute – 2

---

- Can be modified from a script
  - Executed in the context of the panel: dpConnect, event of the panel
  - Of an event of another object: dpConnect, event
  - Associated to the configs, value of a data point element (dpGet, dpSet, dpConnect, etc.)
    - Value of a data point element
    - Alert color of a data point element
    - Set a value of a data point element
    - Mask an alert of a data point element
    - Etc.
- Object attribute of an embedded panel can be modified from the parent panel (setValue, getValue)





## Symbol: reference panel

---

- Generic panel:
  - Same panel for many data point
- Parameters can be given to the panel: dollar parameters:
  - Data point name, value, etc.
  - Dyn\_string, dyn\_int cannot be passed (they are translated in a contiguous string)
  - Evaluated at initialization



## Panel as reference: development

---

- Panel can be embedded in a panel:
  - Catalog menu: one per project path.
    - Folders in project's catalog
    - Save in catalog, change icon, etc.
  - Via the reference button
- A panel can be opened when adding the reference panel to configure the dollar parameters if any.



## Panel as a reference: run-time

---

- Reference panel can be added/removed at run-time
  - addSymbol, removeSymbol function call
  - With dollar parameter
  - The configuration panel is not opened
  - Must have a unique name



## Panel limitation

---

- All the scripts are started in a separate thread
  - Take care to multiple user action
  - E.g.: enable/disable button
- All the thread are killed when the panel is stopped
  - Callback function are also killed: automatic dpDisconnect
- It is not possible to trigger an event script of a graphical object
- No dpConnect on a graphical object attribute



# Alert concept

---

- Alert\_class config: define the behavior of the alert:
  - Acknowledgeable, acknowledge old alert, archive, color, priority, etc.
  - Script to be executed
- Alert\_hdl config: the alert handler
- Alert transition:
  - CAME: transition to an alert state
  - WENT: transition from an alarm state
- One alert message created per transition

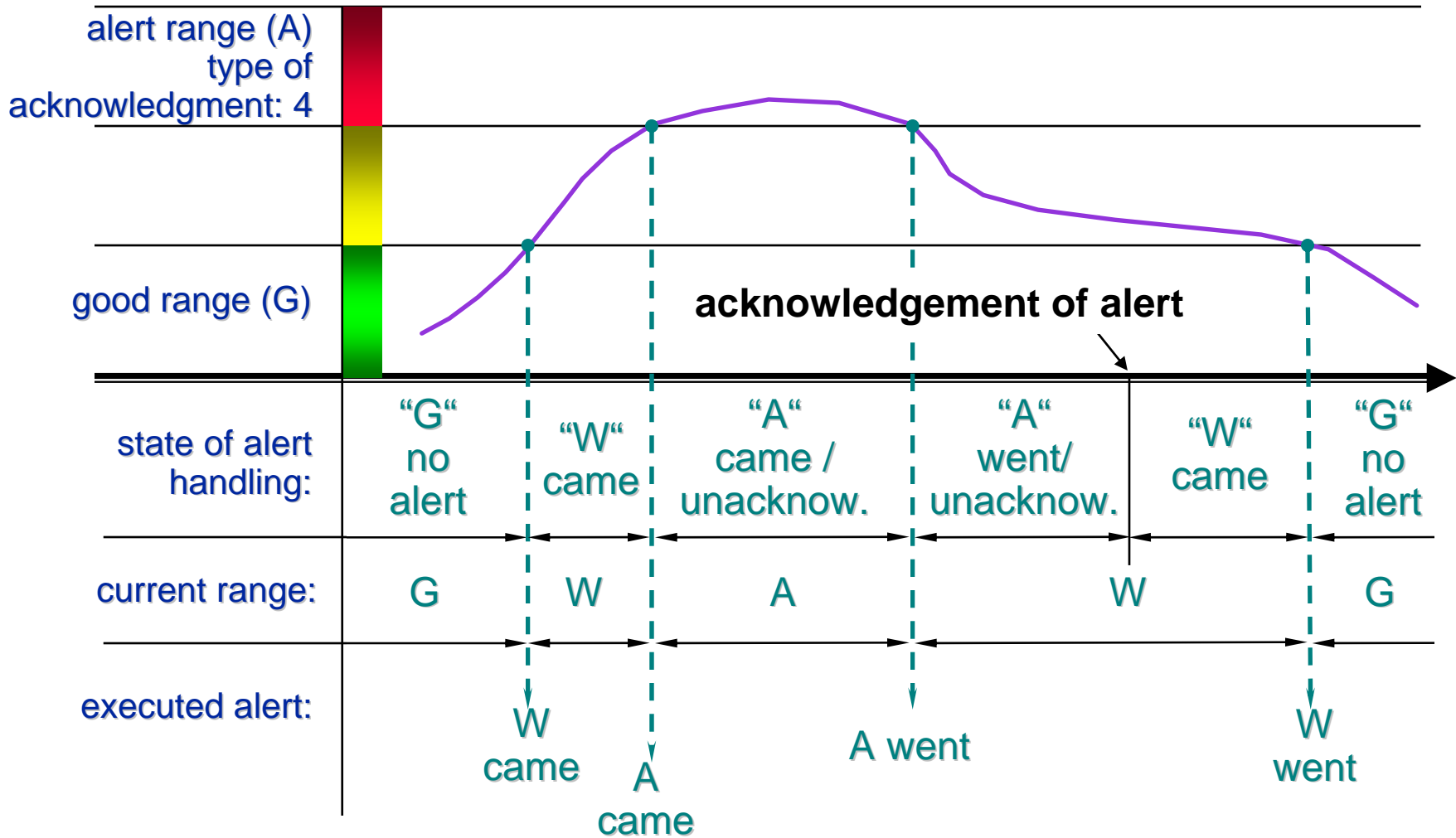


## Alert\_hdl config

---

- Ranges, hysteresis, alert text, panel to open, etc.
  - Can be modified
- Associated with alert\_class:
  - Of any data point element
- Executed when the data point element value is modified
- Time stamp of the data point element value
- All attributes can be modified via PVSS script
- Connected to the attribute of a graphical object
- Can be masked (de-activated) and un-masked (activated)

# Alert status





# Alert visualization

---

- Alert screen
  - Filter: short sign, priority
  - Time range: current, previous
  - Load, save, filter
  - Acknowledge single alert, not all
  - Masking not possible
- Alert row





## Alert limitation – 1

---

- An alert can be generated even if the value is invalid (no default configured, etc.): need programming to avoid it
- Alert range are not data point element
- It is not possible to configure the automatic mask/un-mask an alert based on a value of another data point element
  - Can be done by inventing a script
- No global acknowledge of the alert from the PVSS alert screen
- No masking of the alert from the PVSS alert screen



## Alert limitation – 2

---

- No list of data point element having an alert\_hdl but not activated (can be done by a PVSS script)
- No execution based on priority in case of avalanche
- Memory load
  - A lot of config attributes for an alert
- CPU Load
  - Alert evaluated by Ev
  - Script in alert class executed by Ev
  - Slow down the whole PVSS project



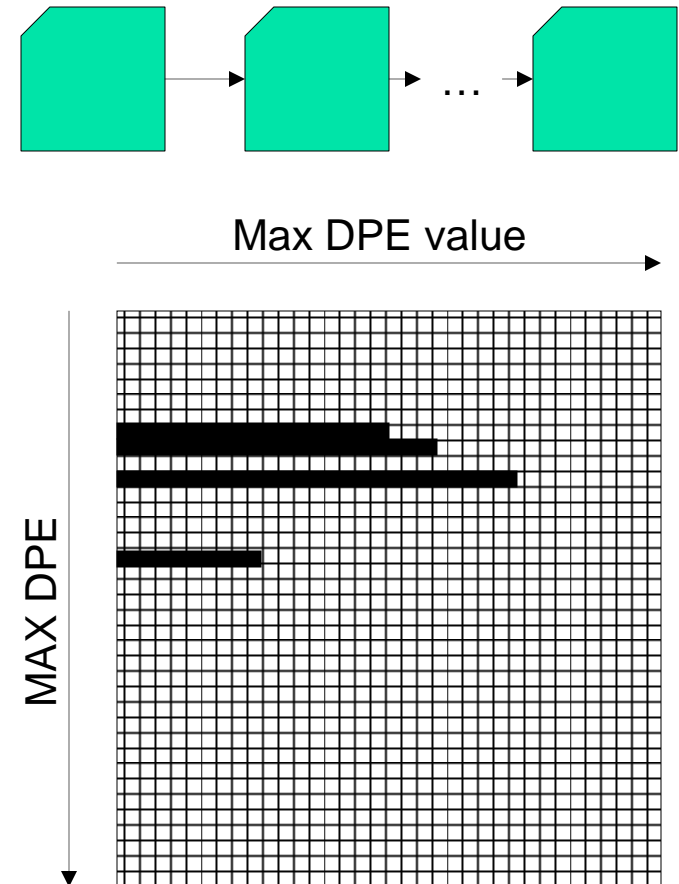
# Archive concept

---

- Archive class data point: define the behavior of the archive file
- Archive config: archive the data point element online value, time stamps, status bits
  - The archive class
  - The smoothing of the archive
- Offline config: to retrieve the archived value
- Corr config: to add correction value to a archive data
  - Replace an archived value
  - Not possible to add intermediate value between two archived value

# Archive class DP: HistoryDB

- Series of chronologically ordered archival files
- Maximum of DPE and maximum of DPE value change per archive file
- Constraint:
  - same change rates of all DPEs in the same archive to avoid empty block in the archive file
- Can be backup in another media
- Activity:
  - Backup and archived file
- Information:
  - Status, etc.





# Archive Limitation

---

- Can be slow:
  - Retrieval and archive is done by the same process
  - → memory increase (manager buffer size can be modified for optimization)
- Errors or archive offline or not running does not prevent an operator to configure the archive config or archive class
  - Max DPE reached, etc.
  - Errors are written to the log file when the value change
  - But no error from the a dpSet function



# PVSS limitation – 1

---

- UIM can set a data point element even if it has an address config
  - Modified by the driver if the value change again (most of the time)
  - Take care to inconsistencies
- Db is writing on file for each value change (not archiving)
  - Can be disabled but loosing functionality
  - Db on a powerful computer with a fast disk
  - Db file must be closed correctly to avoid corruption
    - Otherwise DB repaired at startup: possible risk of DB file corrupted
    - Db PC on UPS.



## PVSS limitation – 2

---

- Data point can be delete/added/modified online (in a production system)
  - Data point deleted means archive history deleted
- Ev kernel of PVSS: no Ev→no PVSS project running
  - All messages go through it (except few cases: getting archived data)
  - Ev on a powerful computer with a battery
- Message buffer for each manager connection
  - Message to process are buffered if the manager can not treat it
  - Increase of memory
- Emergency mode: to avoid swapping and disk full
  - Db must be re-started→project temporary stopped.
- No versioning of application in PVSS, no roll-back
  - Data point configs
  - Panel (only .bak file), lib, script



# PVSS II+UNICOS performance

---

- AB/CO HP Proliant
  - 2xPiv 2.8 Ghz hyperthreaded
  - 2.5 Gb RAM
  - RAID 0 disk
  - RedHat 7.3 or SLC3
  - PVSS II v3.0
- Data simulated by PVSS sim driver, extraction of data (simulation of the LHC Logging)
  - QPS: size of PVSS project for a sector <sup>1)</sup>
    - Small project 52000 DPE, 16000 archived and updated (30%).
  - Cryo: 1/3 of PVSS project in one cryo point <sup>1)</sup>
    - Big project 104000 DPE/40000 archived and updated (37%).

1) Contact UNICOS Support for more information





# Module – 1

---

- Overview of PVSS

- presentation of the most important concept
- PVSS manager



- UNICOS-PVSS

- Concept, principle
- Internal organization
- Delivery, release plan of work, future work
- Description of the UNICOS components



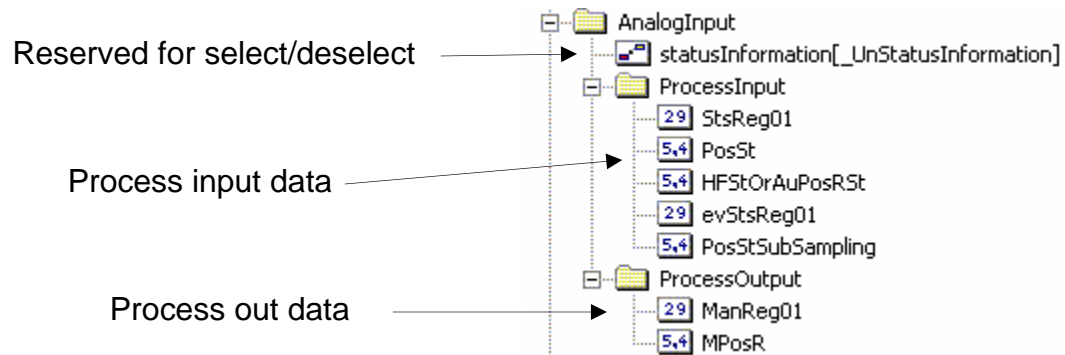
# UNICOS vocabulary

---

- Object type = Device type = PVSS DPT
  - UNICOS object are device type
  - AnalogInput, Analog, AnaDig, etc.
- Object, Device, device name = PVSS DP
- Device DPE, DPE = PVSS DPE
- Control script = script executed by a PVSS Ctrl manager
- Synoptic = PVSS panels
- Alarm = alert = PVSS alert
- Users, group of users:
  - Admin = sdeveloper
  - Expert = super operator
  - Operator
  - Monitor

# UNICOS convention

- PVSS alias: reference of the device
- DPE:



- Device name
  - Widget: `systemName:AliasName`
  - For the TrendTree:
    - `systemName:AliasName.leafProcessInputDPE`
    - `systemName:AliasName.leafProcessOutputDPE`
      - Ex: `tr1:ATVAC_PT100.PosSt`
- Alarm on the device:
  - Alarm are mapped to bits of the stsreg



# Concept – 1

---

- UNICOS based on the JCOP framework
  - PVSS development framework used in the LHC experiments
  - Encapsulate PVSS internals:
    - Configs: libs provided
  - Hierarchy implementation
    - Tree: start of the hierarchy
    - Node: a virtual folder
    - Device: a PVSS DPT
  - Trend utility
  - Based on components
    - Components used by UNICOS:
      - fwCore
      - fwTrending: trending utility
        - Faceplate trend
        - Plot: one trend, maximum 8 curves
        - Page: multi-plot
        - Hierarchical classification
      - fwAccessControl



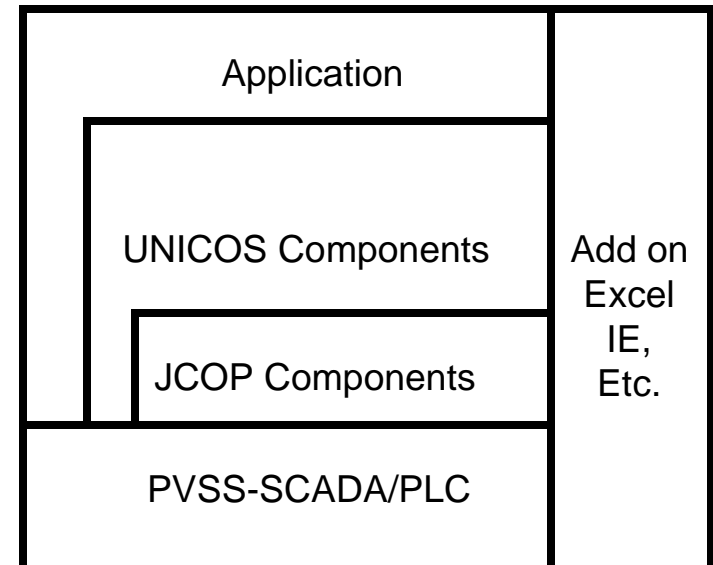
## Concept – 2

---

- UNICOS is an application framework
- The application developer should not need to access the PVSS PARA module
- Same look and feel for all the devices
- Composed of components:
  - unCore: core of the PVSS-UNICOS
  - unMessageText: utility to archive and display messages
  - unActiveX: activeX and libs for the window and trend tree
  - unSelectDeselect: select utility
  - unDistributedComponent: handle the distributed system, connection, disconnection, reconnection
  - unicosObjects: graphicalFrame, objects, etc.

# Internal organization – 1

- Development in layer
- Re-use of existing component as much as possible
- Collaboration with IT/CO: JCOP framework
  - As a consequence sometimes it is slow, may put some overload
  - But share work, share framework
    - Re-use of utilities from JCOP.
  - Not to redo everything
    - Support, maintenance not by UNICOS team





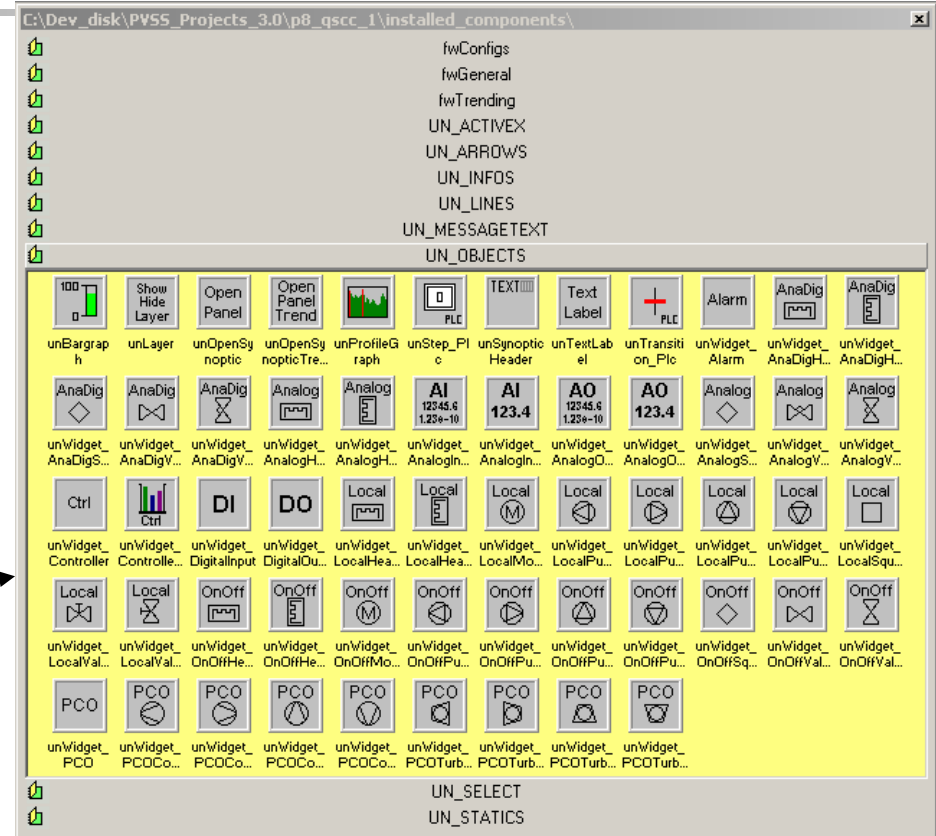
# UNICOS components – 1

---

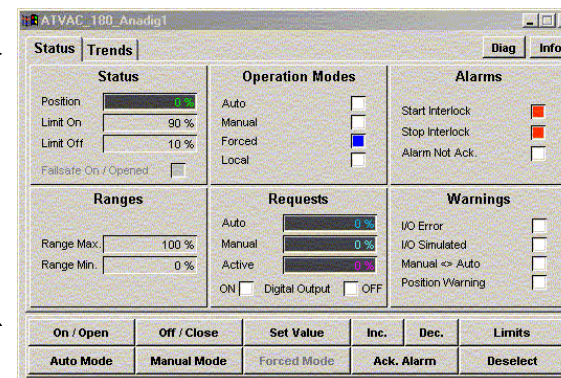
- unCore: Core of UNICOS
  - unDistributedComponent
    - Check the state of the remote PVSS systems
    - Guarantee the state of the remote PVSS system in all possible cases
    - Can trigger graphical animation when the remote system is disconnected
  - unMessageText
    - Messaging utility
    - Send message to operators:
      - UIM manager number
      - Signature of the sender of the message
      - Visualization:
        - 4 levels: INFO, WARNING, EXPERT and EXPERTINFO
        - History
- unTree
  - unActiveX
    - JCOP ActiveX for hierarchical tree
    - Used in WindowTree and TrendTree

# UNICOS components – 2

- unicosObject
  - UNICOS devices type
    - Alarm, AnalogInput, AnalogOutput, DigitalInput, DigitalOutput, Analog, AnaDig, OnOff, Local, Controller, ProcessControlObject (PCO)
  - Graphical representation
    - Catalog of widgets
      - Device widgets
      - open synotic/trend, layer, TextLabel
      - Bargraph, step, transition, ProfileGraph



- Faceplate
- Contextual button per device



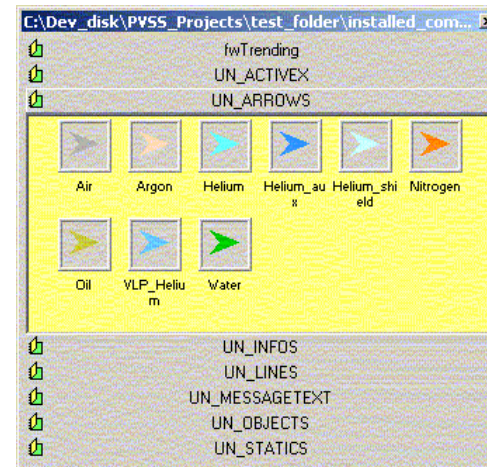
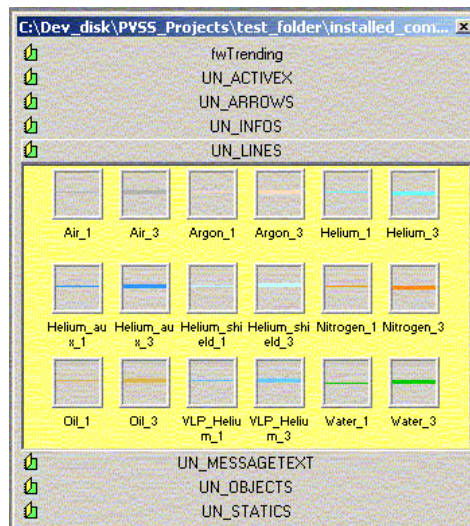
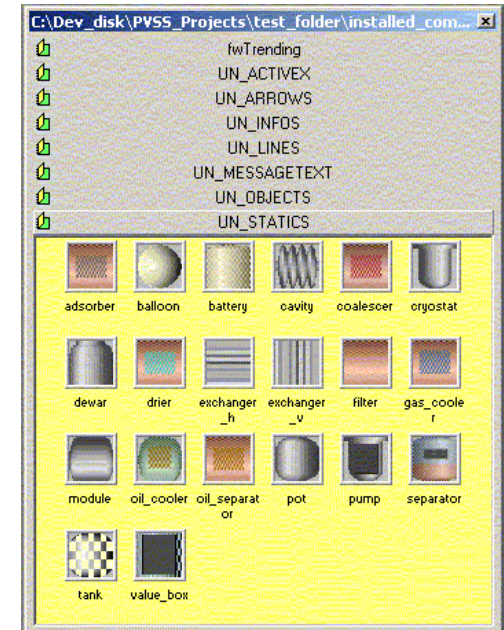


# UNICOS components – 3

- unicosObject (cont.)

- Static objects:

- Cryostat, cavities, coalescer, drier, etc.
- Lines, arrows: air, helium, etc.





## UNICOS components – 4

---

- unicosObject (cont.)
  - UnPanel: synoptic device = PVSS panel device
    - Open panel widget: pop-up and base panel
    - Contextual button
      - Horizontal navigation
      - Up to 10 panel, plot or page
    - Navigation:
      - Back, forward, last 20 opened panels, plots or pages, home, WindowTree
    - Hierarchical navigation



# UNICOS components – 5

---

- unicosObject (cont.)
  - unSelectDeselect: select utility, select the device
    - Script and lib:
      - Select/deselect the device
      - Automatic deselection after a configurable timeout
      - Automatic deselection on exit
    - Libs: UIM
      - Guarantee the uniqueness of selected device per module and per UIM manager graphicalFrame
      - Same look and feel
      - Frame for operation
      - Configuration, diagnostic menus and panels
  - Generation:
    - Generation of the devices and their configuration
  - systemIntegrity:
    - Check the state of the configured PVSS archive, driver manager and remote systems
    - Check the state of the MODBUS/UNICOS driver
    - Check the state of the configured PLCs
    - Check the state of the communication of the configured PLCs
    - Generate alarms
    - Send the DS ip to the configured PLCs
    - Send UTC time to the PLCs



# UNICOS: release v3.6

---

- All UNICOS objects
  - Widgets, faceplates
    - State of the PLC communication included in widgets and faceplates
    - AI/AO/DI/DO: alarm creation/deletion from faceplate
    - AI/AO: alarm letter in widget
  - New devices can be easily added.
  - Filtering (smoothing on deadband) on all the float data
- Event list, object list, alarm list
- WindowTree, TrendTree: support of \$param
- Trend:
  - Faceplate, plot, page
  - Navigation
  - Configuration
- Horizontal navigation: support of \$param
  - Panel ↔ plot
  - Panel ↔ page
  - Plot ↔ page
- graphicalFrame: unicosMHI
  - One alarm row per unicosHMI in multi screen
  - User defined panel to be opened as child panel
- Import
  - Front-end and device import function configurable



# UNICOS: release v3.6 (cont.)

---

- Send email/CERN SMS via email for systemIntegrity alarm
  - 250 emails max per days → others are not sent until next day, except the reports
- Report of the systemIntegrity alarm
  - Via email/CERN SMS
- Validated for distributed system
- Send UTC time to the PLC
- Performance test of the UNICOS package
  - Distributed system: Linux/WindowsXP
- Connect to the LHC-Logging: long term archiving database
  - Windows/Linux platform
- Two UNICOS event format for the eventList:
  - Old value and current value sent at the same time (32 bit)
  - Only current value sent (16 bit)
- Integrity: check the state of the critical elements of the application
  - Configurable
  - Value Archive errors, file switch frequency
  - Alarm archive size and file switch frequency
  - Driver, remote system, PVSS DB file size
- New front-end and new device can be added
- Connection to LHC-Alarm system: LASER
  - Windows/Linux platform



## UNICOS: release v3.6 (cont.)

---

- Optimization of the ObjectList/EventList
- Filtering on Alias for ObjectList/EventList
- ObjectList/AlarmList/EventList/MessageText:
  - UTC/Local (LTI) time format display
- Export of devices
- CMW/OPC/DIP implementation (basic as a demo)
- Import:
  - event archive, Archive configurable from import file
  - Smoothing on archive: old/new, time and/or old/new, no archive
- Linux SLC3
- Use of alias for LASER and LHCLogging interface
- Integration of S7
- WindowTree, TrendTree export/import utility (XML based)
- SMS CERN messages on UNICOS devices alarms
- Tree device overview: widget + snapshot mode
- CMW Client/CMW Server
- Device in more than one domain and nature



## UNICOS: release v3.6 (cont.)

---

- Link between devices
  - RC on device widget
  - Import/export
- AnaDig/Analog: R (regulation)
  - Linked to at least 1 Controller of the same PVSS system
  - New tab in faceplate: Controller data
- Backup utility triggered by the PVSS online backup
- Masking of event



# UNICOS: plan of work

---

- unicos-pvss 3.7
  - GCS devices
  - S7 Controller
    - Default value
- unicos-pvss 4.0 ... 4.1
  - QT
  - JCOP access control