

# WP 4 Project Deliverables D4.8

## User Guide



Project Number	IST-2000-29266
Project Title	Virtual Real Time Fire Emergency Simulator
Deliverable Type	Report
Deliverable Class	Restricted

Deliverable Number	D4.8
Title of Deliverable	User Guide
Nature of the Deliverable	Report
Contributing WPs	WP 2, WP 3
Contractual Date of Delivery	2004-04-30
Actual Date of Delivery	2004-04-30
URL	<a href="http://www.virtualfires.org/">http://www.virtualfires.org/</a>
Authors	Gian Luca Volpato (KTH)
Contact Details	Center for Parallel Computers Department of Numerical Analysis and Computer Science Royal Institute of Technology SE-100 44 Stockholm Sweden

Abstract	
Keywords	

## **Summary**

Summary .....	2
Introduction.....	3
References.....	3
Glossary .....	3
Functionality .....	4
User guide for the PDA-GUI .....	4
Background .....	4
How to start.....	5
Main window .....	7
Scenarios .....	8
Timecontrol.....	10
Mission.....	11
Visualization .....	13
Computations .....	16
UIC config .....	18
Quit .....	19
Press the “Quit” button to exit the GUI. Appendix A: Communication Protocol	
UIC-GUI .....	19
Appendix A: Communication Protocol UIC-GUI .....	20

## ***Introduction***

The interface to control the interaction with the Virtual Fires system has been realized as a two-dimensional GUI with menus and buttons. This GUI can be used in a desktop environment as well as in a portable device; the graphic appearance is the same in both environments.

The portable device is a personal digital assistant; the chosen model is an HP iPAQ Pocket PC with embedded support for wireless LAN communication. A position tracker is attached to the PDA and therefore this device is used also for pointing and selection in the 3D space of the CAVE.

The GUI is implemented in Java. While the desktop computer can support the latest Java 2 Standard Edition (version 1.4.2) the PDA is limited to JDK1.1. AWT is the only graphical package available in JDK1.1; this package offers complex functionality that is not appropriate to the resource-limited environment of the PDA. To overcome this problem, the whole Java Swing package, a light-weight graphical package that partly replaces AWT, was separately compiled and included in the PDA software.

The GUI communicates with the User Interface Controller, a plugin of Covise that takes care of dispatching the GUI messages to the correct recipient. The UIC works as a coordination module that relays messages to the database, the visualization engine and the computation engine. The communication protocol between the GUI and the UIC is based on a request-response model and is fully described in Appendix A.

## ***References***

[1] WP4 Project Deliverable D4.1: Specification of methods of displaying CFD data.

[2] WP4 Project Deliverable D4.2: Specification of immersive user interface.

Appendix A: Communication Protocol UIC-GUI

## ***Glossary***

GUI    Graphical User Interface

PDA    Personal Digital Assistant

LAN    Local Area Network

JDK    Java Development Kit

AWT    Abstract Windowing Toolkit

UIC    User Interface Controller

VCR    Video Cassette Recorder

## ***Functionality***

Deliverables D4.1 and D4.2 [1,2] specify the functions the GUI has to implement. The GUI has to support both control of the simulation and control of the visualization.

The list of implemented functions contains:

- Loading pre-defined configurations and scenarios.
- Describing a mission.
- Starting, stopping and continuing a simulation.
- Detaching from and attaching to an ongoing simulation.
- Selecting alternate mission timelines and timesteps within them.
- Choosing visualization method.
- Navigating in the visualization.

## **User guide for the PDA-GUI**

### ***Background***

A personal digital assistant (PDA) is the device where the GUI is deployed. The model chosen is an HP iPAQ Pocket PC with embedded support for wireless LAN communication.

The GUI is implemented in Java. The PDA supports only JDK1.1, thus AWT is the only graphical package available. To overcome this limitation, the whole Swing package was compiled and included as a jar file, together with the GUI classes.

## ***How to start***

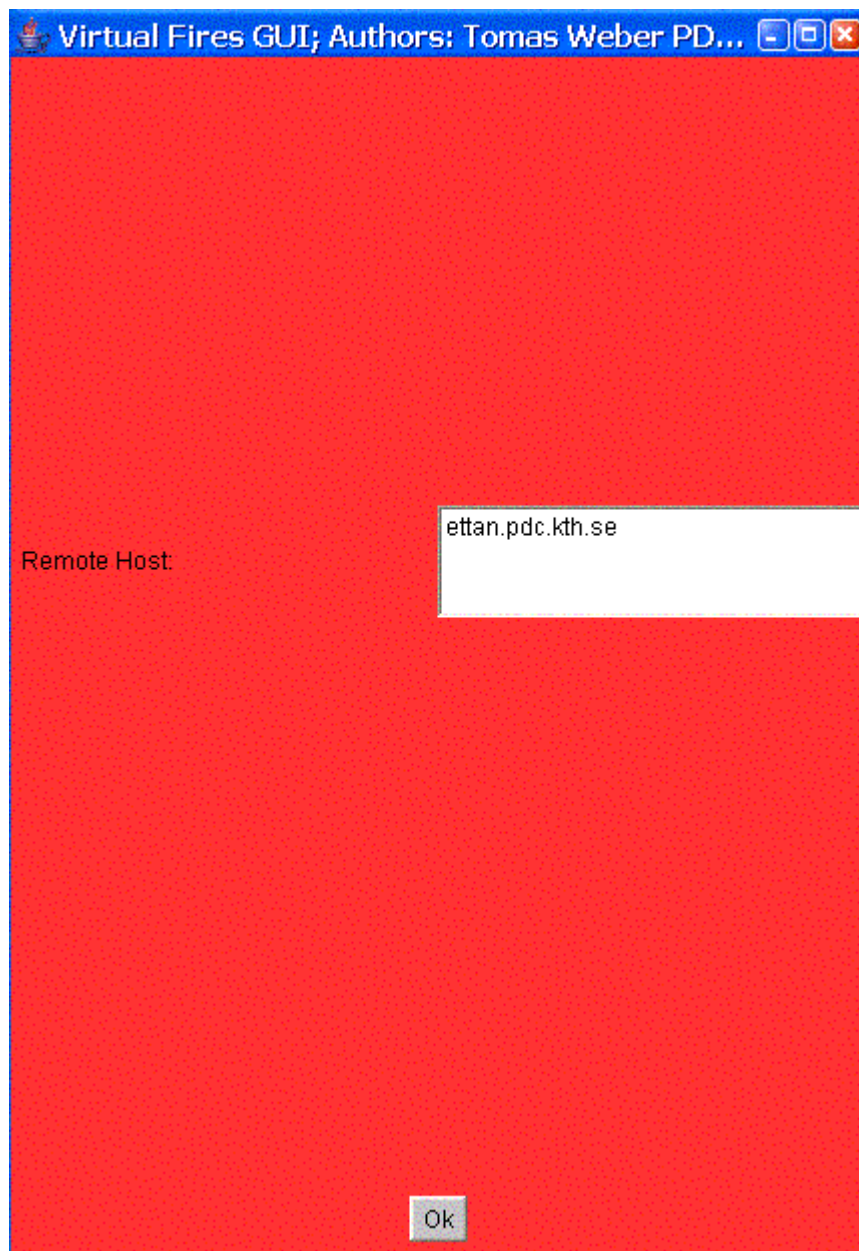
Switch on the PDA and wait until the wireless connection is established (green light is blinking in the upper-left corner).

Press the Menu button and choose the icon “VirtualFires GUI”. The PDA requires 5-6 seconds to load the Java Runtime Environment and display the GUI window.

Enter the username and password (this security function is presently not implemented, i.e. any username/password pair is accepted).



Enter the name of the remote host computer where the UIC is running.

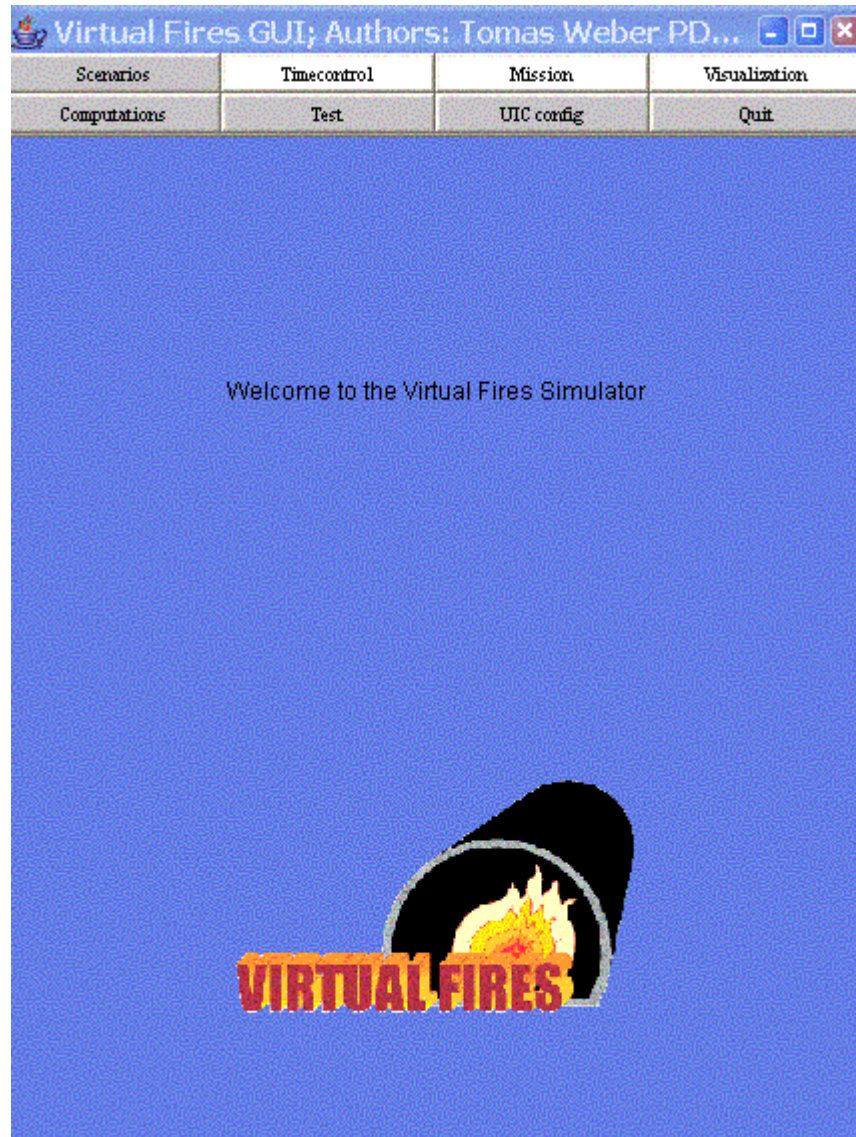


## ***Main window***

The main window consists of a group of eight buttons and a display area.

Active buttons have a grey background colour; inactive buttons have a white background colour.

The display area is used to show additional information when a button is pressed.



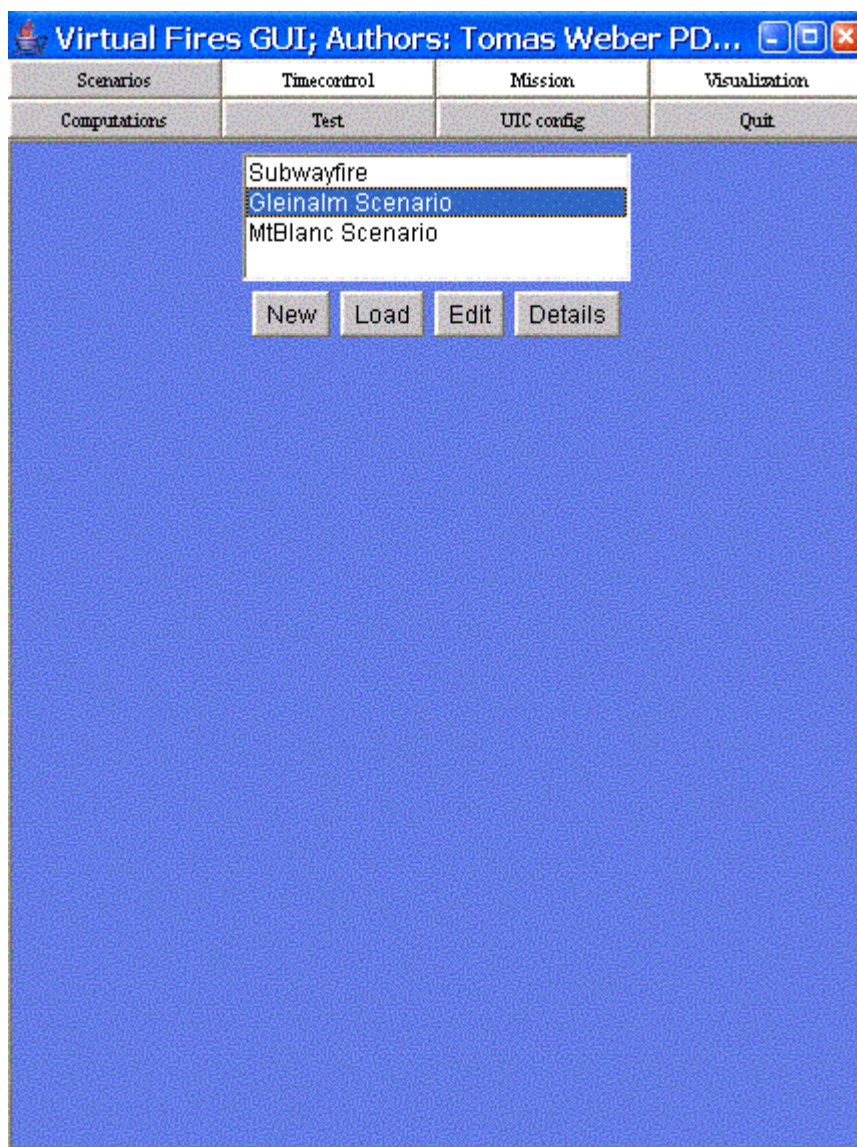


## Scenarios

Press the “Scenario” button to get a list of available, pre-defined, configurations and scenarios.

When one scenario is selected press:

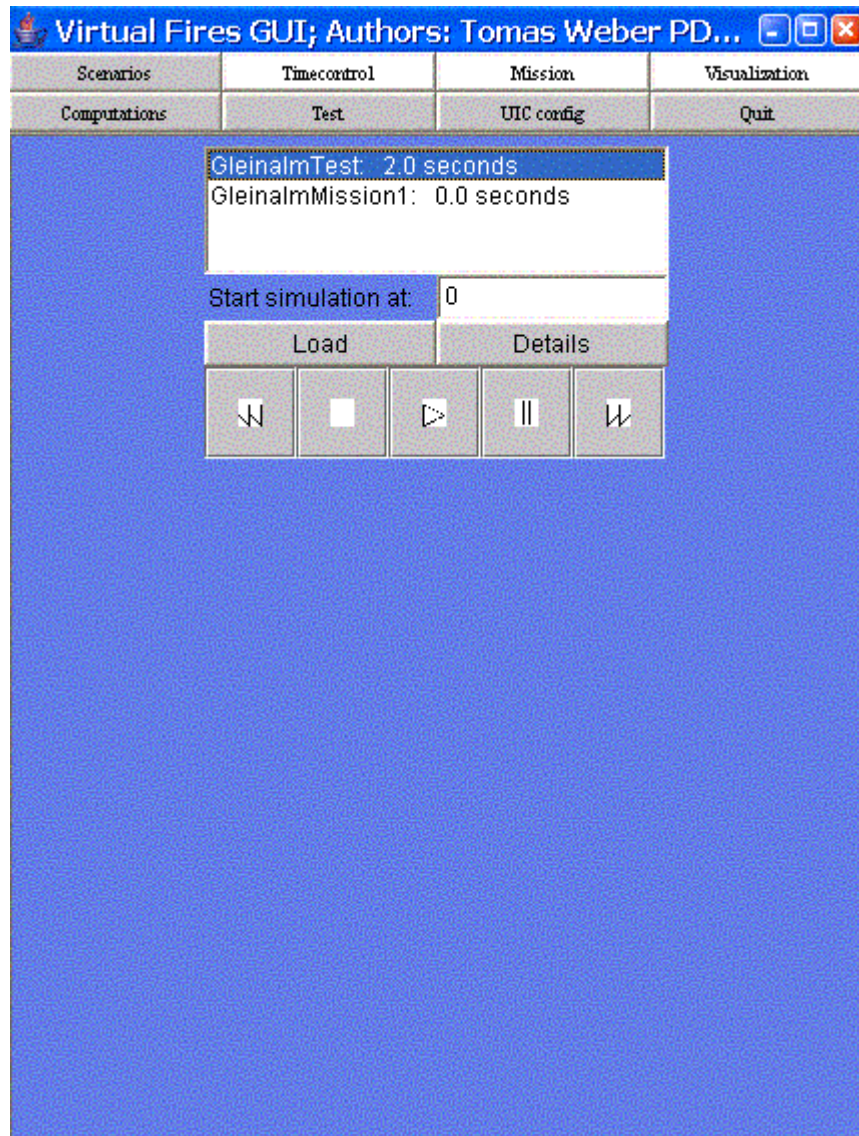
- “New”: Not Implemented.
- “Load” to get a list of missions for this scenario.
- “Details” to get a short description of this scenario.
- “Edit” to get a list of vehicles and fixed equipment that are (can be) placed in this scenario.





When one mission is selected:

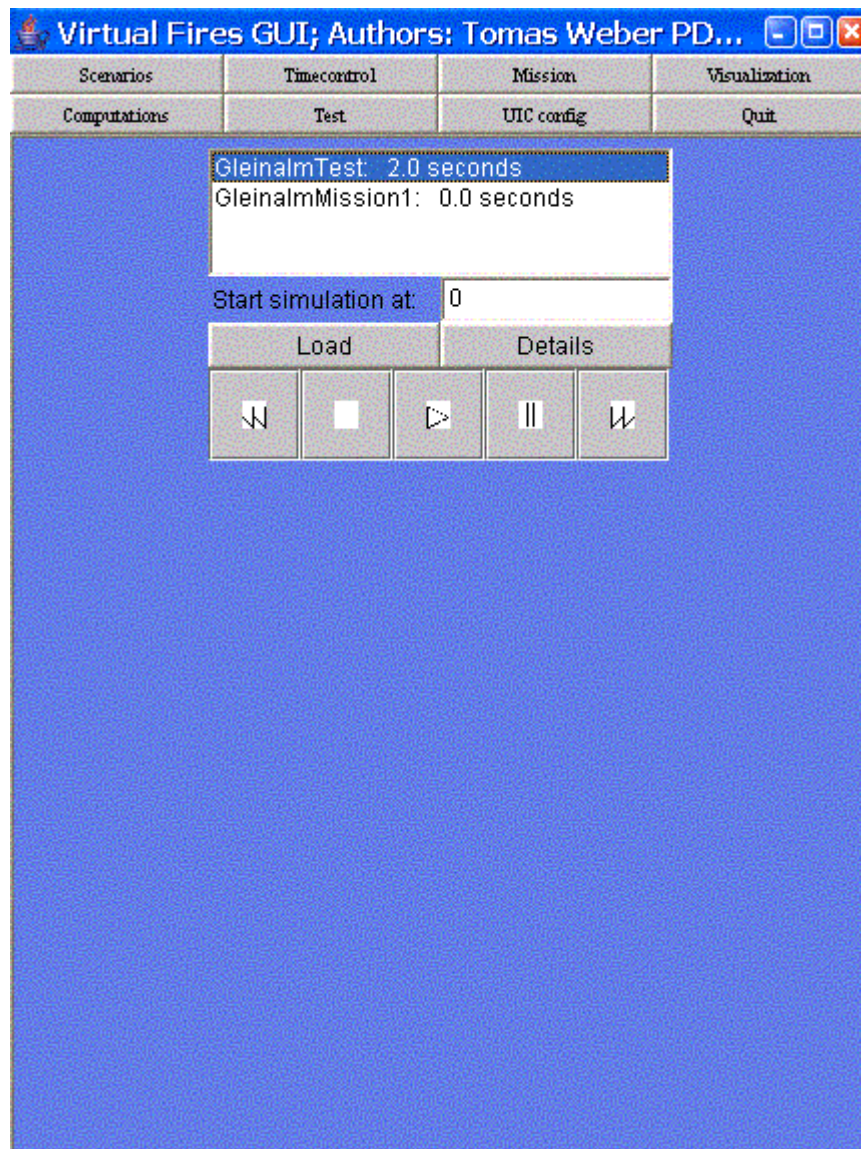
- Enter the time-step where the visualization starts.
- Press “Load” to place the scenario in the visualization environment and activate the time-control buttons (Play, Stop, Pause, Rewind, Fast Forward).
- Press “Details” to get a short description of this mission.



## ***Timecontrol***

Press the “Timecontrol” button to get the name of the mission that is currently visualized (it is the active item in the selection list) and the time-control buttons (Play, Stop, Pause, Rewind, Fast Forward).

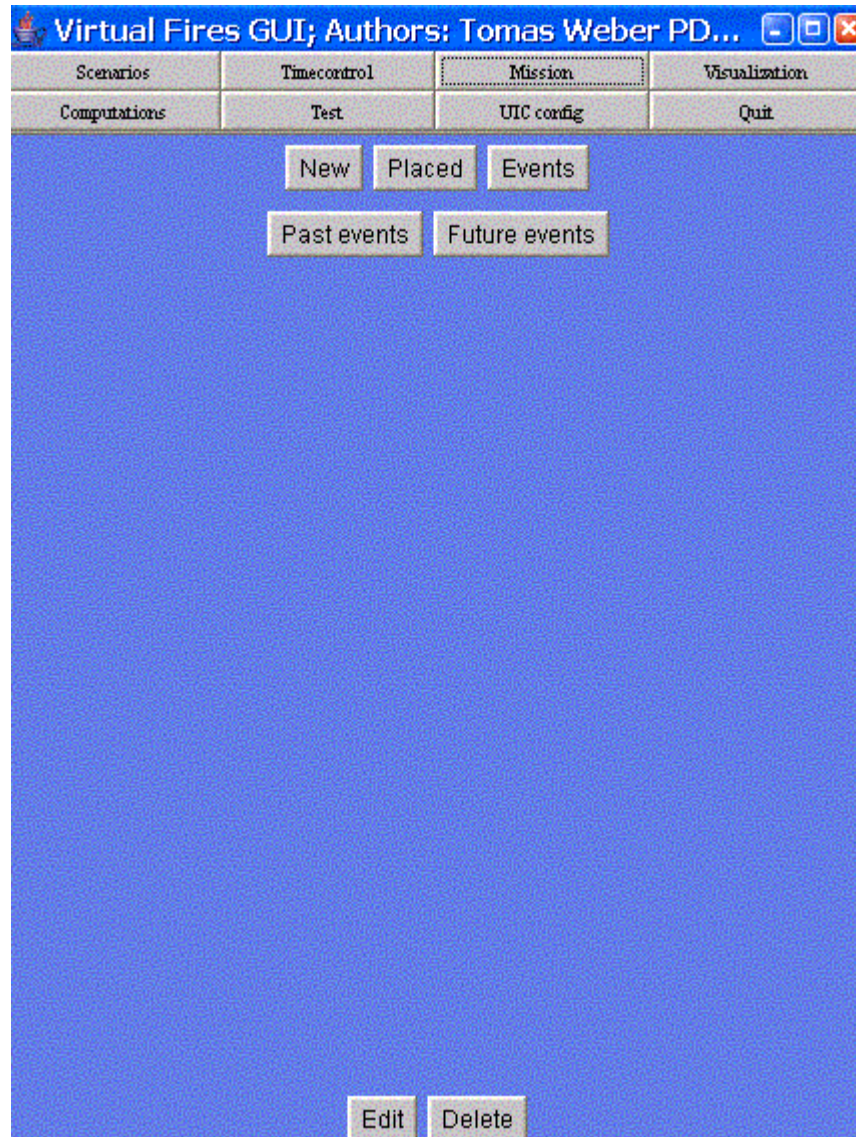
Press one of the time-control buttons to start, stop and change time point in the currently running visualization.





## ***Mission***

Press the “Mission” button to get a list of objects and events that are placed in the current mission.



Press:

- “New” to add a firefighter or an equipment item to the mission.  
Press “Firefighters” or “Equipment” according to the item to be added. Select one item from the list and press “Select”, this shows the item’s properties. Adjust the values and press “Save” to add the item to the mission.
- “Placed” to get a list of objects already placed in the mission. These objects can be firefighters, mobile objects and fixed objects.  
Press “Firefighters”, “Mobile objects” or “Fixed Objects” according to the items to be listed.

- “Events” to get a list of events that are scheduled in the past or in the future for the mission.

Press “Past events” or “Future events” according to the events to be listed.

The mission card shows the objects/events associated to the current mission and allows the user to add objects/events into the mission.

An object is an item placed in the mission; it can belong to any of the following three categories: firefighters, mobile objects, fixed objects. The parameters of an object are strictly dependent from the category it belongs to.

An event is a modification in the status of an object. Every event is therefore associated with the time when the modification occurs and the object whose status is modified.

To create a new object:

1. Press “New Object”;
2. Press either “Firefighters” or “Equipment”;
3. Choose one element from the list and press “Select”;
4. Write the values for the parameters and press “Save”.

To see the list of objects placed in the current mission:

1. Press “Placed Objects”;
2. Press any of “Firefighters”, “Mobile object” or “Fixed objects”;
3. Choose one element from the list;
4. Press “Select” to examine and modify the parameters of the object.

To create a new event:

1. Press “Placed Objects”;
2. Press any of “Firefighters”, “Mobile object” or “Fixed objects”;
3. Choose one element from the list;
  - 3.1. Press “New event” to create a new event associated to the object; specify the parameters of the event and press “Save”;
  - 3.2. Press “New realtime event” to create a new event associated to the object that occurs immediately in the simulation, specify the parameters of the event and press “Save”.

To see the list of existing events:

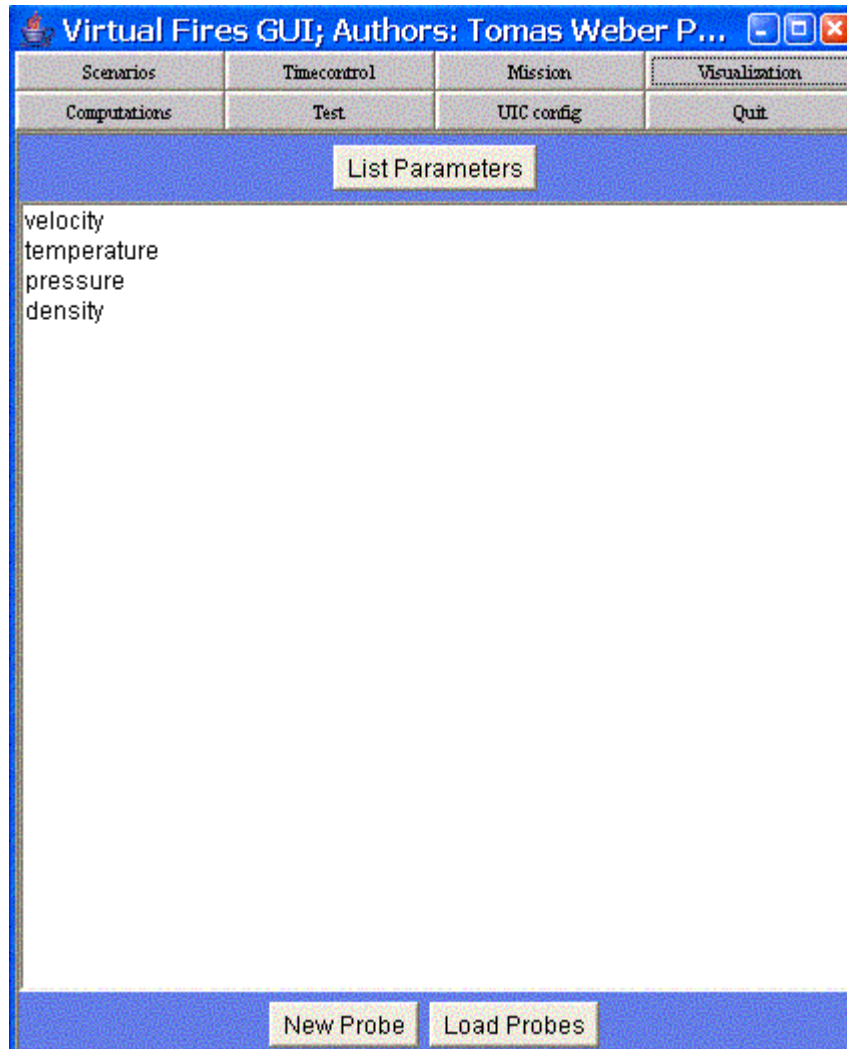
1. Press “Events”;
2. Press either “Past events” or “Future events”;
3. Choose one element from the list;
  - 3.1. Press “Edit” to examine and modify the parameters of the event;
  - 3.2. Press “Delete” to remove the event.

To get information about an object placed in the visualization environment:

1. Points the object and press “Pointed Object”.
2. Examine and modify the parameters of the object.

## Visualization

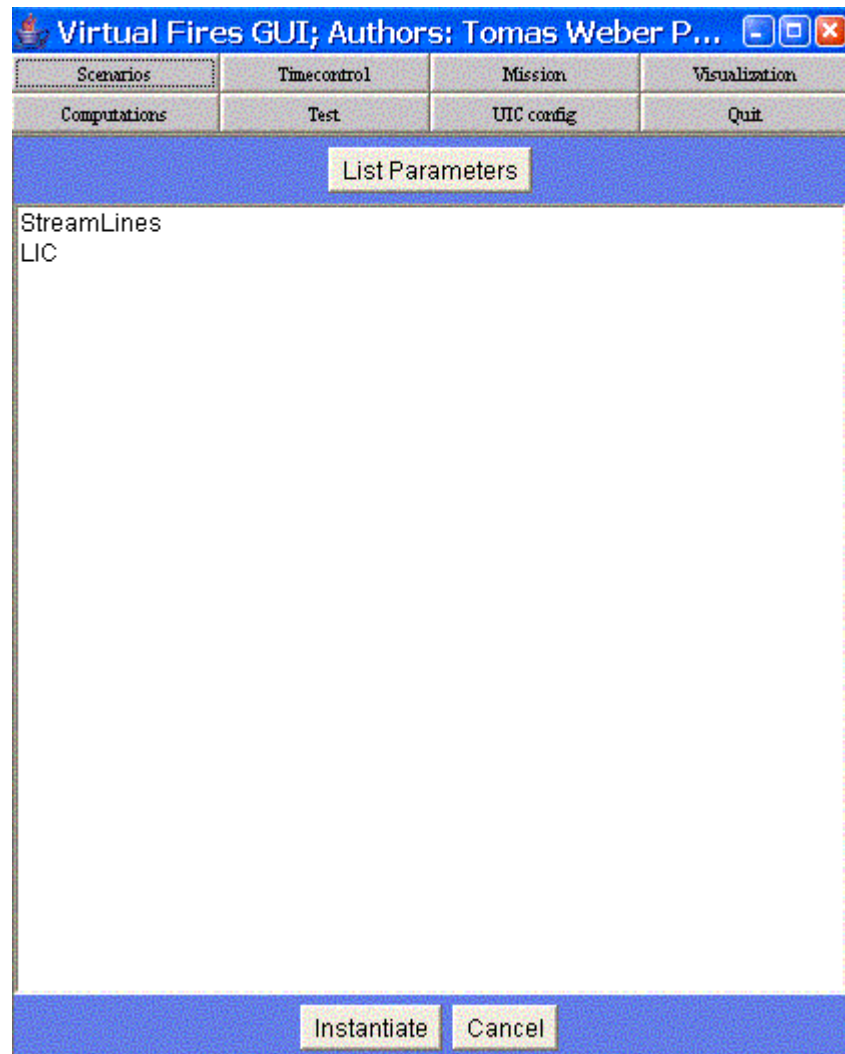
Press the “Visualization” button to get a list of variables that can be visualized in the environment.



Select one variable and press:

- “New Probe” to add a probe for this variable. According to the nature of the variable there are different visualization algorithms.

Select one visualization algorithm and press “Instantiate”. This shows the algorithm’s parameters. Adjust the values and press “Save” to add the probe.

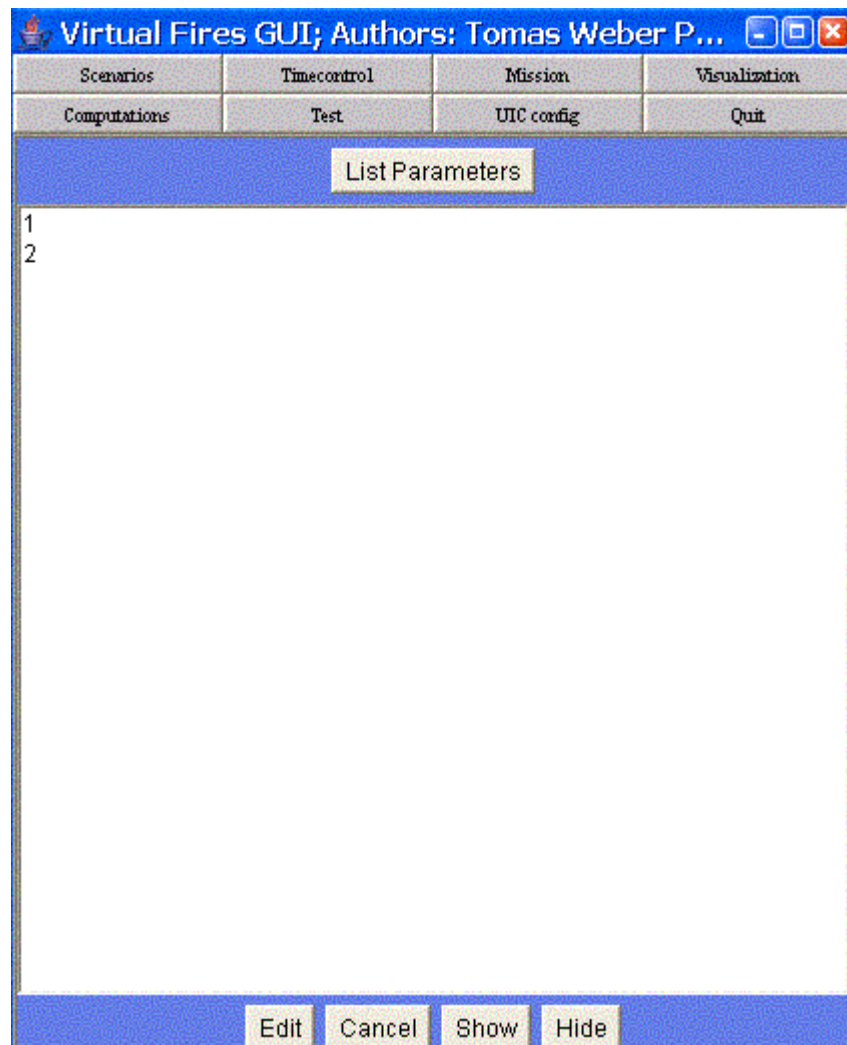




- “Load Probes” to get the list of currently active probes for this variable. If no probe is active the display window will be empty.

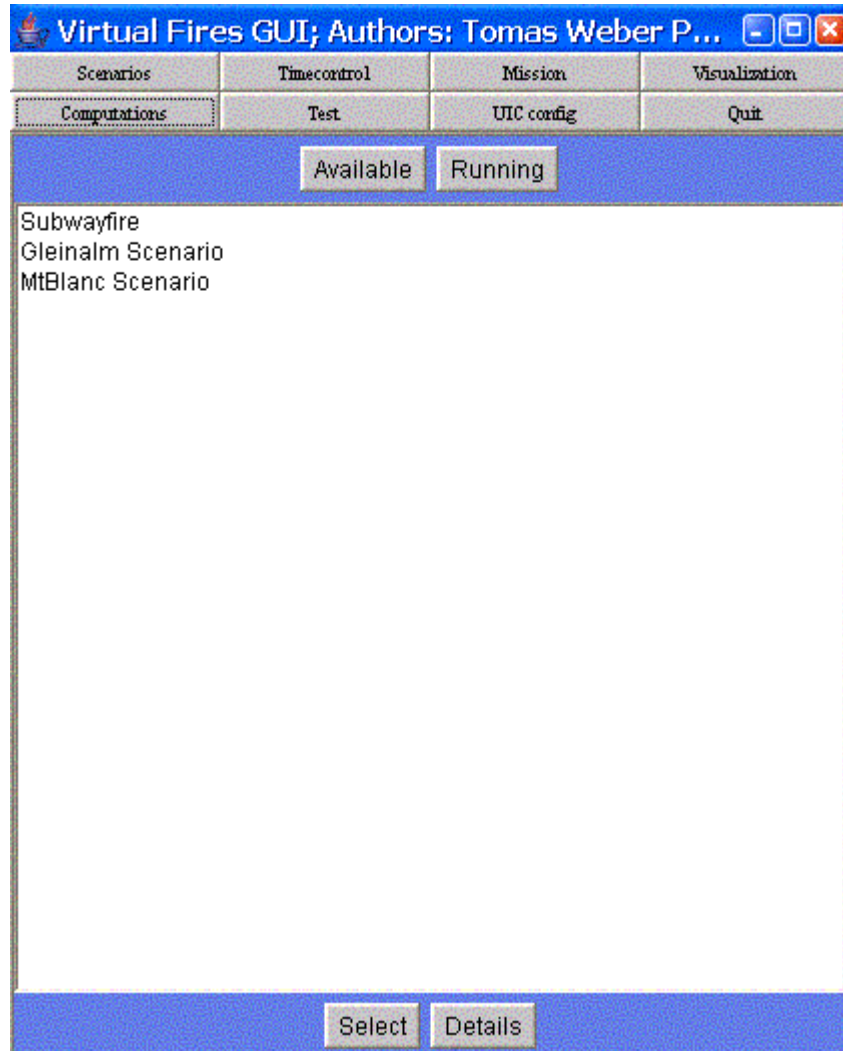
Select one active probe and press:

- “Edit” to show and adjust the algorithm’s parameters.
- “Cancel” to go back to the Visualization menu.
- “Show” to add this visualization algorithm in the environment.
- “Hide” to remove this visualization algorithm from the environment.






## Computations

Press the “Computations” button to get a list of available and running computations. The default is to show available computations. To show the running computations press the “Running” button.



Select one computation and press:

- “Select” to get the list of missions for this computation.
- Choose one mission and press “Select” to show mission’s properties. Press “Start” or “Stop” to change the status of the mission.

Virtual Fires GUI; Authors: Tomas Weber P...   

Scenarios	Timecontrol	Mission	Visualization
Computations	Test	UIC config	Quit

Stoptime

600.0

Resolution

1.0

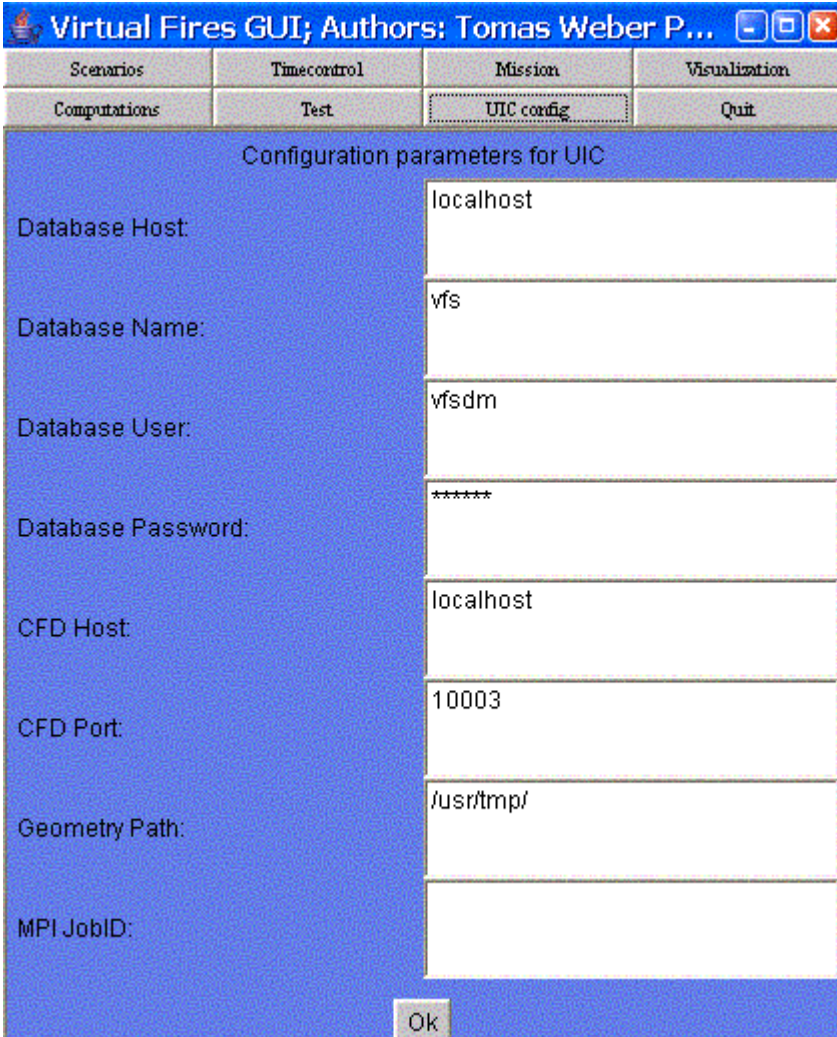
Start Cancel

## ***UIC config***

Press the “UIC config” button to get a list of the current values of configuration parameters for the UIC.

Enter one value for each parameter and press “Ok” to update the UIC configuration. Database, CFD and Geometry parameters are always compulsory; MPI JobID parameter is optional and can be left empty.

After submitting the new configuration values a message will indicate which parameters have been successfully updated.



The screenshot shows a window titled "Virtual Fires GUI; Authors: Tomas Weber P...". It has a menu bar with "Scenarios", "Timecontrol", "Mission", and "Visualization". Below the menu bar is a toolbar with "Computations", "Test", "UIC config" (highlighted with a dashed border), and "Quit". The main area is titled "Configuration parameters for UIC" and contains a list of parameters with text input fields:

Parameter	Value
Database Host:	localhost
Database Name:	vfs
Database User:	vfsdm
Database Password:	*****
CFD Host:	localhost
CFD Port:	10003
Geometry Path:	/usr/tmp/
MPI JobID:	

An "Ok" button is located at the bottom right of the configuration area.

## ***Quit***

Press the “Quit” button to exit the GUI.

## Appendix A: Communication Protocol UIC-GUI

<i>Request</i>	<i>Response</i>	<i>Response types/comments</i>
Request Scenarios	Name0 ScenarioID0 Name1...	String Integer ...
Request Missions "ScenarioID" "TunnelID"	Name0 TimelineID0 EndingTime0 Name1...	String Integer Integer ...
Request Objects "TimelineID" "time"	"Ok"	
Computation Start "MissionID" "endTime" "resolutionTime"	"Ok"	
Computation Continue "MissionID"	"Ok"	
Computation Stop "MissionID"	"Ok"	
Computation Clear "MissionID"	"Ok"	
Computations Running	Name0 ID0 Name1 ...	String Integer ...
Request Description * "ID"	Description	String * can be:  Scenario  Mission  Tunnel?  Template  Computation  Object  Other values might be later added.
Request TemplateObjects "ClassID"	"Name0" ObjectID0 "Name1" ...	String Integer ...
Request PlacedObjects "ClassID"	"Name0" ObjectID0 "Name1" ...	
Request TemplateObject Mission "TemplateID"	ObjectID timestamp Parameters (mission_edit tags) (1)	According to the parameter protocol for template objects I would like for the name of the object to be sent as the first parameter. Since it is a template object there is no objectID, so the UIC should decide one
Request TemplateObject Scenario "TemplateID"	ObjectID Parameters (scenario_edit tags)	According to the parameter protocol for template objects I would like for the



<i><b>Request</b></i>	<i><b>Response</b></i>	<i><b>Response types/comments</b></i>
		name of the object to be sent as the first parameter. Since it is a template object there is no objectID, so the UIC should decide one
Request Object Mission "ObjectID"	ObjectID timestamp Parameters (mission_edit tags)	According to the parameter protocol
Request Object Mission Current	ObjectID timestamp Parameters (mission_edit tags)	Returns info about the object currently pointed in the scenario
Request Object Scenario "ObjectID"	ObjectID Parameters (scenario_edit tags)	According to the parameter protocol
Request Mission Placed	Name0 ID0 Name1...	Union of list of instantiated object of type firefighters and mobile firefighting equipment
Request Scenario Placed	Name0 ID0 Name1...	
Update Scenario "ScenarioID"	"Ok"	This command will be needed as users can still edit scenario parameters that don't affect the grid
****Save NewObject Parameters***	"Ok"	According to the parameter protocol  ***replaced by update objectID request
Update Object "ObjectID" timestamp Mission Parameters	"Ok"	Only the values of the parameters are sent. In the <b>same</b> order that they were received.  If the objectID indicates a template object being instantiated, then the first parameter will be the name.
Update Object "ObjectID" Scenario Parameters	"Ok"	Only the values of the parameters are sent. In the <b>same</b> order that they were received.  If the objectID

<i><b>Request</b></i>	<i><b>Response</b></i>	<i><b>Response types/comments</b></i>
		indicates a template object being instantiated, then the first parameter will be the name.
Dissociate Object "ObjectID" "MissionID"	"Ok"	
Create New Mission	MissionID timestamp parameters	
Create New Mission Branch "MissionID"	MissionID timestamp parameters	
Edit Mission "MissionID"	MissionID timestamp parameters	
Save Mission "MissionID" parameters	"Ok"	Only the values of the parameters are sent. In the <b>same</b> order that they were received.
Delete Mission "MissionID"	"Ok"	
Delete Scenario "ScenarioID"	"Ok"	
Set Mission Endpoint "MissionID" "endpoint"	"Ok"	
Set Mission Resolution "MissionID" "resolution"	"Ok"	
Request Computation Parameters "MissionID"	Endpoint Resolution	Float Float
Playback Play	"Ok"	For all these we need to define proper actions in the UIC
Playback Pause	"Ok"	
Playback Stop	"Ok"	
Playback FastForward	"Ok"	
Playback FastRewind	"Ok"	
	The following commands are more or less obsolete since the decision that scenarios will not be defined through the VF UI	
Request Tunnels	Name0 TunnelID0 Name1...	String Integer ...
Request Tunnel "TunnelID"	"Ok"	
Request Event List "MissionID"	ID0 ID1...	Passed on to the DMC with a mID added first
Request Event List Past "MissionID"	ID0 Time0 ID1 Time1 ...	Passed on to the DMC with a mID added first and the current time added last

<i><b>Request</b></i>	<i><b>Response</b></i>	<i><b>Response types/comments</b></i>
Request Event List Future "MissionID"	ID0 Time0 ID1 Time1 ...	Passed on to the DMC with a mID added first and the current time added last
Save Scenario "TunnelID" "Name" Description	"Ok"	
Request Visualization Data	FieldID0 FieldName0 UnitName0 Type0...	I think I would like fieldName and unitName to have be in double quotes
Get_Available_Vizmethods VisType	Name0 Name1...	VizType is scalar, vector or all. The names returned are unique.
Activate_New_Probe VizMethodName FieldID	ProbeID Name0 Range0 Type0 Value0 Name1...	I think, when a new probe is instantiated the UIC should request the parameters?
Deactivate_Probe id	"Ok"	
Get_Active_Probe_Parameters id	Name0 Range0 Type0 Value0 Name1...	( <sup>1</sup> )
Set_Active_Probe_Parameters id Value0 Value1 ...	Ok/error message	
Get_Active_Probe_Ids ReqType Name	id0 id1...	See UIC-VIZ protocol
Get_Active_Probe_Vizmethod_Na me id	Name	
Get_Active_Probe_Visibility_State id	visible/invisible	
Show_Active_Probe id	"Ok"	
Hide_Active_Probe id	"Ok"	
Create New Event "ObjectID"	EventID timestamp parameters	Create a new event
Create RTEvent "ObjectID"	EventID timestamp parameters	Create a new real time event
Request Event "EventID"	EventID timestamp parameters	
Save Event "EventID" parameters	"Ok"	Only the values of the parameters are sent. In the <b>same</b> order that they were received.
Save RTEvent "EventID" parameters	"Ok"	Only the values of the parameters are sent. In the <b>same</b> order that they were received.
Delete Event "EventID"	"Ok"	

<i><b>Request</b></i>	<i><b>Response</b></i>	<i><b>Response types/comments</b></i>
New Mission "MissionID" "Name" "Description" stoptime resolution	MissionID	MissionID sent is the mission the new mission branch is based upon.  MissionID returned is the id of the new branch.
Request UICParameters	UIC configuration parameters	Set of values indicating DBHost, DBName, DBUser, DBPasswd, CFDDHost, CFDDport, GEOMETRYPATH.  The values must be sent in this order. All values are String, except CFDDPort that is Integer
SetDBConnectionParameter "DBHost" "DBName" "DBUser" "DBPasswd"	"Ok"	Values must be sent in this order.
SetCFDDConnectionParameter "CFDDHost" CFDDPort	"Ok"	Values must be sent in this order.
SetGeometryPath "path"	"Ok"	

Errors are always sent as Error "....." and can always be the response instead of the specified response to a request.

Observe space between "Error" and error string.

(<sup>1</sup>) The format of parameters is:

name minrange maxrange type value

"time x NIL float y"

"time NIL NIL choice yes no selected no"