Manufacturing Performance Edition
Quick Reference Booklet

Lanner Group specialize in giving organizations a clearer view of the future. To date they have helped several thousand companies world-wide to manage and understand the process of change and as a result, significantly improve business performance.

is a trademark of Lanner Group

Ref: 2010/MAN/1

CONTENTS

Using the quick reference booklet

This booklet lists the most commonly used features of WITNESS and gives a brief description of their purpose. Refer to your on-line Help (F1) for more detailed explanations.

Help & Support

On-line help

Press the **F1** key or use the **Help** menu to display the on-line help. If you are currently displaying a dialog, WITNESS displays help specifically for that dialog; otherwise choose the topic from the help index.

Support

On occasion, whether you are a new or an experienced user, you may need support to achieve the maximum benefit from your WITNESS investment. Lanner provides a full package of helpdesk, web and consultancy support.

To contact Lanner please email or telephone your nearest location, or visit our web site for further contact options.

EUROPE

Help desk:	+ 44 (0) 1527 551 320	
Telephone:	+ 44 (0) 1527 403 400	
Fax:	+ 44 (0) 1527 404 452	FAO: “WITNESS Help Desk”
E-Mail	support@lanner.co.uk info@lanner.co.uk	

USA

Help desk:	+ 1 888 757 1611 (Toll Free)	
Telephone:	+ 1 713 532 8008	
Fax:	+ 1 713 260 9602	FAO: “WITNESS Help Desk”
E-Mail	support@lanner.com info@lanner.com	

Web page

The Lanner Web site is located at: **www.lanner.com**

Menus

File Menu

New
Clears the current model, so that you can begin a new model.

Open
Retrieves a previously saved file (e.g. model, module, picture gallery or designer elements).

Initialize from file…
Enables the loading of a status file (.sta) to initialize the model.

Save
Saves the current model or other data using the current model name.

Save As
Saves the current model or other data under a new name.

Send
Sends the model and all related data by e-mail.

Directories
Specifies directories to be used when locating files.

Record to AVI
Activates AVI recording.

Print Model
Prints the model in the form of a model library file.

Print Screen
Prints the current window.

Exit
Exits WITNESS.

Displayed above Exit are the last four models accessed. Select a model name to open that model.

Edit Menu

Undo Paste (Ctrl+Z)
Removes the last paste.
This may be;

- Details - An element's details
- Display - An element's display
- Graphics - Selected backdrop graphics
- Text - Limited text editing "undo" when using dialogs.

Cut (Ctrl+X)
Cuts to the clipboard.
This may be;

- Text - Text on dialogs
- Graphics - Selected backdrop graphics "Cut Graphics" is particularly useful, when moving backdrop graphics between layers

Copy (Ctrl+C)
Copies
This may be;

- Text - Text on dialogs
- Graphics – Selected Backdrop graphicsto the clipboard.

Paste (Ctrl+V)
Pastes current data from the clipboard
This may be to;

- Text - Text fields on dialogs
- Graphics - Add backdrop graphics
- Details - Set an elements details
- Display – Redefine an elements display
- Element details and display are copied using commands on the Element Menu. The name of the copied element is displayed in the menu.

Paste To: "Layer #"
Pastes backdrop graphics items on the clipboard to the currently selected layer (Ctrl+T).
The "layer #" is that of the last layer drawn to from the display bar.
(Use View – Screen Editor to access the display bar for the Back drop element.)

Rule Assistant
Enables rule keywords to be pasted to the current cursor position.

Action Assistant
Enables action keywords to be pasted to the current cursor position.

Function Assistant
Enables functions to be pasted to the current cursor position.

Distribution Assistant
Enables distributions to be pasted to the current cursor position.

ElementName Assistant
Enables element names to be pasted to the current cursor position.

Clipboard Text
Edit text data held in the clipboard.

View Menu

Tool bars Shows / Hides the standard toolbars.
Standard – New , open and save
Model – User & Immediate Actions
Element – All modeling commands that take elements as a context
Views – Zoom, Pan, associate views, Process view
Reporting Run – Reports and Reporting controls
– Run controls, trace and meteor trail generation & run time control and display.
Assistant – Rule, Action, Function, Distribution & Element Assistants.
Display Edit – Alignment of display items, grid configuration and changing layers, fonts and color of fonts and lines.
Customize – Allows you to customize content and appearance.

Element Selector – Shows/ hides the element selector tree view.
Screen Editor – Allows graphics to be drawn, associated with the Backdrop element.
Picture Gallery – Manages the 500 pictures available for reference by the model. Imported from : .BMP,.WMF,.EMF,.GIF,.JPG and .DXF formats.
You can also select pictures from a previously saved Picture gallery files (.ICN).
Graphical Editing – Switches graphical editing on/off allowing protection of model layout from accidental change.
Layers – Renames, or turns on or off, any of the graphical display layers.
Drawings – Displays the Drawings dialog so you can group layers into a Drawing.
Keys – Lets you add a color key to explain the meaning of colors representing element states.
Set Associate View – Allows you to associate a window and a view of the virtual screen with the currently selected element.
View Associate View – Displays view previously associated with the currently selected element.
Element Flow – Shows the movement of parts or fluids through the model as a series of lines (with arrows indicating direction). The ultimate destination of parts and fluids is shown at the end of the line (E.g. SCRAP,SHIP). You can also choose to display rule names at appropriate points.
Process View – Displays a separate window, containing a process view of the model. Elements are shown as boxes, and the direction of flow is shown as lines with arrows. You can choose to display rule names at appropriate points.

Model Menu

Options – Sets global and default Display and Detail values.
Clock – Customizes the clock display to the format of your choice.
Title – Lets you change the name, title and author of the current model.
Initialize Actions – Specifies the Actions that define the model’s start-up conditions.
User Actions – Specifies the Actions executed when you select the ***Run/User Actions*** option.
User Command – Allows you to create user commands that you can place on a special model toolbar.
Customize 3D… - Allows you to change 3D settings for VR & Quick 3D
Quick 3D - Creates 3D VR View of current mode
Experiment – Old WITNESS methodology to setup and run experiments
Random Numbers – Alters the behavior of the random number streams.
Scenario Manager – Opens the WITNESS Scenario Manager for experimentation
Scenario Manager Wizard – Opens the WITNESS Scenario Manager through a wizard to define an experiment
Actions on Force Breakdown – Specifies the Actions executed when a breakdown is forced on any element.
Actions on Force Repair – Specifies the Actions executed when a repair is forced on any element.

Watch – Selects attributes, variables and functions so that you can track their values as the model runs.
Remove Debug Data – Removes any trace of debug information from the model

Elements Menu

Define – Allows you to define new elements.
Detail – Allows you edit the details for the selected elements.
Display – Allows you edit the display features for the selected elements.
Update Graphic – Enable direct editing of the selected graphic set up dialog.
Run Properties – Runs the properties actions set on the selected element’s actions page.
Delete Elements – Allows you to delete elements providing they are not referenced by other elements.
Delete Graphics – Deletes the graphical items currently selected.
Grouping – Changes the display locking option controlling how graphics are selected.
Visual Output Rule – Enables an output rule to be specified by selecting destinations and rule type with the mouse.
Visual Input Rule – Enables an input rule to be specified by selecting sources and rule type with the mouse.
Visual Labor Rule – Enables a labor rule to be specified by selecting labor and rule type with the mouse.
Force Breakdown – Causes the selected element to breakdown immediately
Force Repair – Causes the forced breakdown to be repaired immediately.
Clone – Creates exact copies of the selected elements.
Copy Detail – Copies the details of a single selected element to the clipboard.
Copy Display – Copies the display features of a single selected element to the clipboard.

Reports Menu

Statistics - Produces a statistical report on the current selection.
Report Using - Generates a customized report from the selected Report element.
Used - Produces a report of where the selected elements and or types are used.
Summary - Produces a summary report on the current selection.
Explode - Produces an exploded current detail report on the currently selected element and or types.
Cost Revenue & Sustainability Reports - Produces summary report for all measures defined
Random Numbers - Displays the Random Numbers Usage grid, which indicates where all the random number streams and sub-streams currently deployed by the model are used, the number of times that they have been used and a simple analysis of how they have been used.
Current Status - Displays a list of future events and ‘blocked’ or ‘idle’ elements and a list of outstanding jobs waiting for labor.
Meteor Trail - Activates and details the display of part flow tracking records already collected.
By On-Shift Time - Switches reporting between total time and on-shift time.
Output - Controls the destination of all reporting output:
Screen, Printer, File, HTML, DIF or CSV File(s)
Reset Selection - Resets the statistical reports, for the selected elements.
Reset All - Resets statistical reports, for all elements

Run Menu

Stop – Halts the simulation run. The right-hand mouse button (or the **<Esc>** key) has the same effect.
Begin – Resets the simulation time back to 0.0 and restores the initial states of the elements.
Run – Animated simulation, with instant movement.
Step – Animated simulation, one step at a time.
Batch – Simulation with no animation, until a specific time or event is reached.
Advance – Animated simulation, until a specific time or event is reached.
Walk – Turns walk mode on or off. Does not run model.
Trace – Turns trace mode on or off. When on logs each step to a **.trc** file.
Time Scaling – When selected (shown by a tick next to the option) simulation time units are related to real time units.
Meteor Trail – Turns meteor trail part flow tracking record generation on or off.
Watch – Switches the Model Watch feature on or off. Model Watch tracks the values of selected attributes, variables and functions that have been selected using the ***Model/ Watch*** command.
Immediate Actions – Prompts for a single action command, which is executed immediately.
User Actions – Execute the User Actions specified in the ***Model/User Actions*** option.

Window Menu

1: window 1… – Displays or hide Windows 1…8.
8: Window 8 The Window title currently defined is shown in the menu.
Interact Box – Displays or hides the Interact box
Clock – Displays or hides the analog Clock window
Time – Displays or hides the model defined formatted Time window.
Designer Elements – Displays or hides the Designer Elements window.
Control – Allows adjustment of the current windows features:
– Title
– Layers visible
– Background color
– Zoom factor
Co-ordinates – Controls the display of the cursor co-ordinates in the current window title bar.

Help Menu

Contents – Online help Contents.
Search for help on . . . – Find help topics by word search.
Technical Support – Provides information about where to get product support.
Plugin Information – Gives a list of available plugin modules
About WITNESS – Displays the WITNESS copyright message and version number.

Toolbars

The contents of all toolbars are fully customizable, and the user can create new toolbars. As supplied, WITNESS is pre-configured with the following toolbars. Note: although you can add new toolbars, only the standard toolbars appear on the view menu.

Standard toolbar

New model – Deletes the current model and prepares for building a new one.

Open Model – Lets you open a model or other file.

Save Model – Saves the current model to a file with a **.mod** extension.

Model toolbar

Run User Actions – Runs the actions entered at Program User Actions.

Run Immediate Actions – Runs the action defined in the edit box.

Print Var1

Immediate Action – Edit box allowing you to enter an immediate action. (*Enter* runs the action, as does the toolbar button).

Views toolbar

Zoom In – Increases the size of your image on the virtual screen.

Zoom Out – Reduces the size of your image on the virtual screen.

Pan to Selection – Centralizes the current selection within the current window. Also finds graphics for selections made in the tree view / Toolbar edit field.
With ***SHIFT:*** all graphics for the selected elements are located.
With ***CTRL***: the View will Zoom to fill the window with the selection.

The **mouse zoom mode** button switches the mouse's zoom mode on or off. When the mouse zoom mode is selected, you can:

- Zoom into the modeling window by scrolling the middle wheel upwards. WITNESS zooms towards the top left-hand corner of the virtual screen.
- Zoom out of the modeling window by clicking the right-hand mouse button or by scrolling the middle wheel downwards. WITNESS zooms away from the top left-hand corner of the virtual screen.
- Zoom to the centre of the modeling window, by clicking the left-hand mouse button.

Parent Module – If the current element is contained in a module, the display pans to display to the parent module (if drawn).

Display Associate View – Displays the view and window associated with the currently selected element.

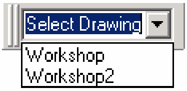
Set Associate View – Allows you to associate a window and a view of the virtual screen with the selected element.

Process View – Displays the **process view** dialog, allowing you to display an overview of the flow of processes in your model.

Element Flow – Displays or hides the Element Flow, which shows parts and fluids moving through the model as a series of lines.

Assistant – Displays or hides the Assistant toolbar.

Activate Graphical Editing – Allowing protection of model layout from accidental change.



Select Drawing – select a Drawing (or collection of layers) to display in the active window.

Element toolbar

Element Selector – Shows / Hides the element selector tree.

Current Defined Selection – Shows a summary of the current selection and allows you to enter an element name or type.

Add Defined Selection – groups elements together in a Defined Selection.

Selected Element box – shows the name of the element that is currently highlighted in the Element Selector (that is, an element's name is selected, not its check box selected).

Visual Output Rules – Lets you define an output rule for the selected element using the mouse.

Visual Input Rules – Lets you define an input rule for the selected element using the mouse.

Labor Rule – Lets you define a labor rule for the selected element.

Quick Rule Mode – A mode for the fast graphical definition of rules.

Define New Elements – The elements are by default added at the module level of the current selection.

Detail – Edit details for the currently selected elements.

Display – Edit display features for the currently selected elements.

Delete Elements – Delete elements providing they are not referenced by other elements.

Delete Graphics – Delete currently selected graphics items.

Clone Selected Elements – Creates copies of the selected elements. Requires you to place the copied graphics as a set for each element.

Copy Element Detail – Copies the details of a single element to the clipboard.

Copy Element Display – Copies the display features of a single element to the clipboard.

Paste – Pastes current data to:
– Text fields on dialogs
– Set an elements details
– Redefine an elements display / Add backdrop graphics

Undo Paste – Removes the last paste.
This may be:
– An element's details
– An element's display
– Selected Backdrop graphics

Create Designer Elements – From each of the selected elements.

Create Module – Creates a single module containing the selected simulation elements.

Force breakdown – causes an instant breakdown on any appropriate element.

Force Repair – An instant repair on any element that has been broken down.

If you wish to move displays in the modeling window, you can use the **lock mode** button to switch the lock mode on so that when you move one display item for an element, all the other display items associated with that element move too. You can select one of several locking modes.

Individual Lock locks all display items for element instances.

Element Lock locks all display items for all instances of elements.

Super Lock locks all display items for the contents of a module.

Unlock switches lock mode off. Any display item you move will move independently of associated display items.

Run toolbar

Begin – Resets the model and sets elements to an idle state.

Autostep Back – Allows you to step backwards in time to the previous **.sim** file in a sequence of saved **.sim** files. It also takes ***run until*** instructions into account.

Stop – Stops a run in any of the chosen modes.

Auto Save – Records the simulation progress to a set of **.sim** files.

Step – Lets you examine one by one each step that is taken as the run proceeds.

Run – Runs your model with simulation visible on all currently opened windows.

Autostep Forward – Allows you to step forwards in time to the next **.sim** file in a sequence of saved **.sim** files. It also takes ***run until*** instructions into account.

Batch – Runs your model in batch mode, closing the simulation windows. ***Batch*** takes ***run until*** instructions into account. If you have not entered run until instructions the model indefinitely or until you stop it.

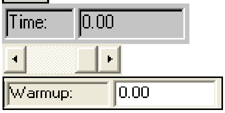
Run Until – Enter a time to batch to, or a time to step forwards or backwards to in which case the simulation runs until that time is reached. Alternatively, enter the name of an element and, the simulation runs until an event occurs for that element.

Walk Speed – Selects *Walk* run mode and controls the speed that the model runs at when selected.

Trace – Reports to a text file with the extension **.trc** and gives textual record of each step in the run.

Log Meteor Trail – Turns the Meteor Trail on or off.

Watch – Turns the Model Watch on or off.



Time: 0.00

Warmup: 0.00

Simulation Time – Displays the current simulation time.

Slow Motion Slider – An easy way to slow and speed up a model run.

Warmup Period – The definition of the time at which statistics are reset.

Reporting toolbar

Statistics Report – Produces a statistical report on the currently selected elements and or types.

Summary Report – Produces a summary report on the currently selected elements and or types.

Explode Report – Produces an exploded current detail report on the currently selected element and or types.

Used Report – Produces a report of where the selected elements and or types are used.

Random Number Report – Produces a report on Random number streams usage.

Display Meteor Trail – Activates and details the display of part flow tracking records already collected.

Report Basis – Toggles the basis for statistical reports between Total Simulation Time and On Shift Time.

Reset Reports for Selection – Resets the reports for the selected elements.

Reset All – Resets all reports.

Send to Minitab – Sends the selected variables to the Minitab statistical analysis program.

Cost, Revenue & Sustainability Report – Produces a report detailing user defined measures generation & usage

Assistant toolbar

Activates the **Rule Assistant**.

Activates the **Actions Assistant**.

Activates the **Function Assistant**.

Activates the **Distribution Assistant**.

Activates the **Element Names Assistant**.

Display Edit toolbar

Display Elements – displays the Display Bar for each of the selected elements in turn.

Update graphics - displays the appropriate Display dialog for the selected element so you can update the display item. This feature is only available for the last display item you selected; if you have selected several display items, each item is surrounded by a single-line box. The most recently-selected item is surrounded by a double line box.

If you wish to move displays in the modeling window, you can use the **lock mode** button to switch the lock mode on so that when you move one display item for an element, all the other display items associated with that element move too. You can select one of several locking modes.

Individual Lock locks all display items for element instances.

Element Lock locks all display items for all instances of elements.

Super Lock locks all display items for the contents of a module.

Unlock switches lock mode off. Any display item you move will move independently of associated display items.

Font – changes / applies the current font to text for selected elements

Font Color – changes / applies the current color to text for selected elements.

Line Color – changes / applies the current color to lines for selected elements.

Fill Color – changes / applies current color to the interior of rectangles, ellipses etc for selected elements.

Grid Snap – When activated, display items are aligned to the nearest line on a grid.

Display Grid – Shows or hides the grid. Hiding the grid does not affect the status of the Grid Snap feature.

Configure Grid – Changes grid dimensions and display.

Reference Point – Shows or hides the second Reference Point, if set up on the Model Options dialog's Coordinates page.

Layers – Displays the layer that a selected element is on. Also changes the layer for a group of selected elements.

Layers – Displays the Layers dialog, which lets you make layers movable, visible and selectable.

Drawings – Displays the Drawings dialog, which lets you add layers to a Drawing.

Align Left – Aligns the display of all selected elements with the left-hand edge of the first selected element.

Align Right – Aligns the display of all selected elements with the right-hand edge of the first selected element.

Align Top – Aligns the display of all selected elements with the top edge of the first selected element.

Align Bottom – Aligns the display of all selected elements with the bottom edge of the first selected element.

Path Edit toolbar

(For editing path displays of paths, queues, conveyors, tracks and power and free sections)

Add Straight – The plus button adds a straight segment to the currently selected path display of the length selected or typed into the box on the right.

Add Curve – The plus button adds a curved segment to the currently selected path display of the radius and arc size selected or typed into the box on the right. The curved arrow button determines the direction of the curve.

Delete Segment – Deletes the currently selected path segment

Catalogs Dialog – Opens up a display to show all defined path catalogs and their contents. Allows addition and deletion of segments from the pull down lists.

Change Length/Radius and Angle – when adding segments allows the current segment details to be edited

Path Information Dialog Opens up a dialog detailing the parameters of each segment in the currently selected path. Allows editing of height of path and length overrides.

Data Types

WITNESS supports four data types. Throughout this booklet the following symbols are used to indicate what data type a variable, function or parameter is:

Real $^{(R)}$
Integer $^{(I)}$
String $^{(S)}$
Name $^{(N)}$

$^{(V)}$ is used to indicate that a function is Void (that is, it returns nothing).

Rules

Input/output rules

WAIT

Entities or items will wait until they are pulled from or pushed to another element.

PUSH

Entities or vehicles are sent to the first available element in a preferred-order list which is capable of accepting them. Examples:
PUSH to m1,m2,m3
PUSH a to m1, b to m2

PULL

Entities are taken from the first available element in a preferred-order list which is capable of supplying them. Examples:
PULL from m1, m2, m3
PULL a from m1, b from m2

LEAST

Entities or items are sent to (or taken from) the element with the least entities, items or free capacity. Examples:
LEAST PARTS m1,m2
LEAST FREE m2(1),m2(2),m2(5)
LEAST FLUID t1(2),t2 VOLUME 6 RATE 2

MOST

Entities or items are sent to (or taken from) the element with the most entities, items or free capacity. Examples:
MOST FREE act3,act4
MOST FLUID pr1,pr2 RATE 34
MOST FLUID s2,s3,s4 VOLUME 10 RATE 5

MATCH

Entities or resource are matched on an activity according to a **condition**, **attribute** or **any** Entity/Resource selection. You can also MATCH entities to load onto a vehicle. Examples:

```
MATCH/ATTRIBUTE ATT1 a#(1) and b#(1)
MATCH/ANY (a#(1) AND b#(2)) OR c#(3)
MATCH/CONDITION (ATT2 = 5) buf1#(2)
MATCH/CONDITION SKILL>5 ResA#(1) OR ResB#(1)
```

PERCENT

Entities or vehicles are sent to (or taken from) several elements on a random percentage basis. Example:

```
PERCENT /3 a1 2.5, a2 97.5
```

SEQUENCE

Entities or vehicles are sent to (or taken from) several elements, in a cyclic sequence. Examples:

```
SEQUENCE /WAIT act1#(3), act2#(5)
SEQUENCE /NEXT act2#(3), act4#(1)
SEQUENCE /RESET act1#(3), act2#(5)
```

SELECT

Entities or vehicles are sent to (or taken from) several elements, according to an integer value. Example:

```
SELECT on X act1,act2,act3
```

IF

Route to or obtain from an element, if certain conditions for the element are true. Example:

```
IF NPARTS(act1) > 5
    PUSH to act2
ELSEIF NPARTS(act1)=5
    PUSH to act3
ELSE
    WAIT
ENDIF
```

DESTINATION

Vehicles are routed according to their current destination (output rule only). Example:

```
DESTINATION t1,t2,t3
PUSH to track5
```

BUFFER

Adds a dedicated input or output queue to an activity which previously had no queue. Example:

```
BUFFER (5)
```

FLOW

Used to move items between continuous processing elements (you must specify the rate). Examples:

```
FLOW to p5 RATE 10.0, t5 RATE 2.0
FLOW to p5 RATE 35,p4,p7,p6
DEFAULT RATE 24.5
```

CONNECT

Used to link streams to other continuous processing elements (no rate is specified). Example:

```
CONNECT with p2,t2,pi5(7)
```

RECIPE

Used to get known volumes of items from a number of specified sources, or to send known volumes to a number of specified destinations. Examples:

```
RECIPE SERIAL
  t1 VOLUME (15) RATE (3),
  p2 VOLUME(9) RATE(4.5)
RECIPE PARALLEL
  p3 VOLUME(2) RATE (4),
  p4 VOLUME(5) RATE (15)
```

Compounded rules

Rules may be combined, except where results could be ambiguous. The most commonly used combinations are:

- An IF rule may be followed by ELSEIF or ELSE to specify alternative conditional routing.
- Paths may be used in conjunction with rules in the form:
  PUSH to machine1 With labour1 Using Path.
- To pull or push specific part types, use rules of the form:
  PULL part1 From buff2, PUSH part3 To machine4.

To get specified fluid from world

- FLOW fluid WORLD (rate)
- CONNECT fluid WORLD
- MOVE (volume) fluid from WORLD to ELEMENT   (N.B. This is an **Action**, not a **Rule**).
- RECIPE (SERIAL or PARALLEL) fluid WORLD VOLUME (volume) RATE (rate)

Attributes

Standard attributes

TYPE $^{(N)}$ Name of the part or labor.
DESC $^{(S)}$ Character description of the part or labor.
PEN $^{(I)}$ Color of the part or labor (code number).
ICON $^{(I)}$ Icon of the part or labor (code number).

Dimension attributes

LENGTH $^{(R)}$ Part Length
WIDTH $^{(R)}$ Part Width
HEIGHT $^{(R)}$ Part Height

Fluid-related attributes

CONTENTS $^{(R)}$ Volume of fluid in a part (if the part contains fluid).
FLUID$^{(N)}$ The name of the fluid contained within a part. If the part contains a mixture of fluids, this attribute will return ‘MIXTURE’.

Routing-related attributes

STAGE $^{(I)}$ Number of the current part route destination in the route.
NSTAGE $^{(I)}$ Total number of part route stages.
R_CYCLE $^{(R)}$ The cycle time for the current element.
R_SETUP $^{(R)}$ The setup time for the current element.

Machine-related attributes

LaborElement Contains the name of the labor element that will be allocated to a machine or (if the labor has already been allocated) the name of the labor element that will be released from a machine.

Changing an attribute’s value remotely

To change the value of a part, carrier, vehicle or machine attribute (for example, from another element’s detail dialog or from the model’s Initial or User Actions) use the following syntax in an action:

Element {AT position} : attribute = Value

If a part is not present at the specified position or the part does not have the specified attribute, an error message is displayed. ***For parts***, position 1 is the front of the element that the part is in, position 2 is the second position and so on. Position 0 is the rear of the element. If you do not specify a position, WITNESS assumes position 1. ***For carriers, vehicles and machines***, specify 0 to change the value of the carrier's (or vehicle's or machine's) attribute, or 1 to change the value of the attribute of the part in the carrier (or vehicle or machine). Machines can contain more than 1 part, so position 2 changes the attributes of the second part in the machine, and so on.

You cannot change a labor unit's attribute value directly from its own detail dialog; you can only change it via actions on another element's detail dialog, or via actions elsewhere in the model (such as Initial or User Actions).

For specific instances of labor at machines, conveyors, tanks, processors or pipes the system function **LABORAT** may be used.

Variables

System variables

TIME (R)	– Current simulation clock value.
I (I)	– Index value of the current vehicle.
M (I)	– Batch or lot number of the current part, or the current timeseries observation number.
N (I)	– Index value of the current element.
VTYPE (N)	– Type of current vehicle.
ELEMENT (N)	– The name of the current element.
WARMUP(R)	- Time set for resetting reports – The initial Warmup period
REPLICATION(I)	- The current replication number used for Experimentation, Scenario Manager & Optimizer

Distributions

(Parameters are shown in *italic* type).

BETA $^{(R)}$*(shape* $^{(R)}$*, scale* $^{(R)}$*, stream* $^{(I)}$*, substream*$^{(I)}$*)* Beta distribution.

BINOMIAL $^{(I)}$*(probability* $^{(R)}$*, trials* $^{(I)}$*, stream* $^{(I)}$*, substream*$^{(I)}$*)* Binomial distribution.

ERLANG $^{(R)}$*(mean* $^{(R)}$*, K value* $^{(I)}$*, stream* $^{(I)}$*, substream*$^{(I)}$*)* Erlang K distribution.

GAMMA $^{(R)}$*(shape* $^{(R)}$*, scale* $^{(R)}$*, stream* $^{(I)}$*, substream*$^{(I)}$*)* Gamma distribution.

IUNIFORM $^{(I)}$ *(minimum* $^{(I)}$*, maximum* $^{(I)}$*, stream* $^{(I)}$*, substream*$^{(I)}$*)* Integer Uniform distribution.

LOGNORML $^{(R)}$*(mean*$^{(R)}$*, std_deviation*$^{(R)}$*, stream* $^{(I)}$*, substream*$^{(I)}$*)*Log Normal distribution.

NEGEXP $^{(R)}$*(mean* $^{(R)}$*, stream* $^{(I)}$*, substream*$^{(I)}$*)* Negative Exponential distribution.

NORMAL $^{(R)}$*(mean* $^{(R)}$*, std_deviation* $^{(R)}$*, stream* $^{(I)}$*, substream*$^{(I)}$*)* Normal distribution.

POISSON $^{(I)}$ *(mean* $^{(I)}$*, stream* $^{(I)}$*, substream*$^{(I)}$*)* Poisson distribution.

RANDOM $^{(R)}$*(stream* $^{(I)}$*, substream*$^{(I)}$*)*Uniform 0.0 - 1.0 distribution.

TNORMAL $^{(R)}$*(mean*$^{(R)}$*, std_deviation*$^{(R)}$*, minimum*$^{(R)}$*, maximum*$^{(R)}$*, stream*$^{(I)}$ *,substream*$^{(I)}$*))* Truncated Normal distribution.

TRIANGLE $^{(R)}$*(min* $^{(R)}$*, mode* $^{(R)}$*, max* $^{(R)}$*, stream* $^{(I)}$*, substream*$^{(I)}$*)* Triangular distribution.

UNIFORM $^{(R)}$*(minimum* $^{(R)}$*, maximum* $^{(R)}$*, stream* $^{(I)}$ *substream*$^{(I)}$*)* Uniform distribution.

WEIBULL $^{(R)}$*(shape* $^{(R)}$*, scale* $^{(R)}$*, stream* $^{(I)}$ *, substream*$^{(I)}$*)* Weibull distribution.

All stream and substream arguments are optional

Functions listed in alphabetical order

ABS $^{(R)}$*(expression* $^{(R)}$ *)*
Returns the absolute value of the given expression.

ACARR $^{(R)}$ *(p&f_section* $^{(N)}$*)*
Returns the average time that carriers have spent on the specified section.

ACTCARR $^{(I)}$ *(p&f_element* $^{(N)}$*)*
Activates the carrier(s) specified in P&F_element and returns the number of carriers successfully activated.

AddLayerToDrawing $^{(I)}$ *(Layer_number* $^{(I)}$*, Drawing_number* $^{(I)}$*)*
Adds the specified layer to the specified Drawing.

AddSensor *(element_name*$^{(N)}$*, position*$^{(R)}$*)*
Adds sensor to a Conveyor element. The conveyor must be set as Continuous.

AdjustPriority $^{(I)}$ *(elt_name* $^{(N)}$*, new_priority* $^{(I)}$*)*
Changes the priority of the specified elements in the idle and jobs lists to the new priority. ALL applies the same priority to all elements.

AdjustShift $^{(I)}$ *(element_name* $^{(N}$*, shift_name* $^{(N}$ *)*
Changes the shift that applies to the specified element. ALL applies the shift to all elements.

AFCARR $^{(R)}$ *(p&f_section* $^{(N)}$*)*
Returns the average time that carriers have spent in and empty state on the specified section.

AFLOW $^{(R)}$*(part_or_fluid* $^{(N)}$*)*
The average time the part or fluid spent in the model, not including parts or fluids which were initially rejected, or fluid contained in parts.

AJOBTIME $^{(R)}$*(labor* $^{(N)}$*)*
The average time spent on jobs by the labor unit.

AMAX $^{(R)}$*(value* $^{(R)}$ *{,value* $^{(R)}$*...}*
The maximum value of a series of real numbers.

AMIN $^{(R)}$ *(value* $^{(R)}$ *{,value* $^{(R)}$*...}*
The minimum value of a series of real numbers.

AOVER $^{(R)}$*(buffer* $^{(N)}$*) ...})*
The average number of items in the buffer which remained in the buffer beyond the delay time.

AOVERT $^{(R)}$*(buffer* $^{(N)}$*)*
The average time that parts remained in the buffer beyond the delay time.

APARTS $^{(R)}$*(element* $^{(N)}$*)*
The time-weighted average of the number of parts in a buffer or conveyor.

APPEXEC $^{(I)}$*(application path* $^{(S)}$*, finish mode* $^{(I)}$*, start mode* $^{(I)}$*)*
Invokes the application found on application path.

ASC$^{(I)}$*(string* $^{(S)}$*)*
Returns an integer containing the ASCII code of the first character in the specified string

ASTIME $^{(R)}$*(element* $^{(N)}$*)*
Average on-shift time that parts or fluid spent in the specified element (pipe or tank for fluids).

ATIME $^{(R)}$*(element* $^{(N)}$*)*
Returns the time-weighted average of the time parts or fluid(s) have spent in a buffer, conveyor, pipe or tank.

ATIME2 $^{(R)}$*(part* $^{(N)}$*)*
Returns the average time that the specified part has spent in the model, only including those parts which have been shipped. If no instances of the part have been shipped, the return value is 0.0.

ATLOADED $^{(R)}$*(p&f_carrier*$^{(N)}$*)*
Returns the average time that the specified carrier has spent in the loaded state.

ATSTATE $^{(R)}$ *(p&f_carrier* $^{(N)}$ *,state*$^{(I)}$ *)*
Returns the average time that the carrier has spent in the specified state.(pg.**Error! Bookmark not defined.**)

AttachLabor $^{(I)}$ *(element* $^{(N)}$*, labor* $^{(N)}$*)*

Attaches the labor unit to the specified machine or vehicle.

AttachLaborToPartAt $^{(I)}$ *(element* $^{(N)}$*, position* $^{(I)}$ *labor* $^{(N)}$*)*

Attaches the labor unit to the part at the specified position in the machine, vehicle, buffer, path, conveyor or carrier.

AttrMatchSetInteger $^{(V)}$ *(attribute1* $^{(N)}$*, attribute2* $^{(N)}$ *new_value* $^{(I)}$*)*

Finds the value of *attribute1* for the current part, then sets the value of *attribute2* for all matching parts to the *new_value*.

AttrMatchSetName $^{(V)}$ *(attribute1* $^{(N)}$*, attribute2* $^{(N)}$ *new_value* $^{(N)}$*)*

Finds the value of *attribute1* for the current part, then sets the value of *attribute2* for all matching parts to the *new_value*.

AttrMatchSetReal $^{(V)}$ *(attribute1* $^{(N)}$*, attribute2* $^{(N)}$ *new_value* $^{(R)}$*)*

Finds the value of *attribute1* for the current part, then sets the value of *attribute2* for all matching parts to the *new_value*.

AttrMatchSetString $^{(V)}$ *(attribute1* $^{(N)}$*, attribute2* $^{(N)}$ *new_value* $^{(S)}$*)*

Finds the value of *attribute1* for the current part, then sets the value of *attribute2* for all matching parts to the *new_value*.

AVOL $^{(R)}$*(element* $^{(N)}$*)*

Returns the time-weighted average of the volume of fluid present in a pipe or tank.

AWIP $^{(R)}$ *(part_or_fluid* $^{(N)}$*)*

Returns a time-weighted average of the total number of work-in-process parts or total volume of fluid of the specified type.

AWIP2 $^{(R)}$*(part* $^{(N)}$*)*

Returns a time-weighted average of the total number of work-in-process parts of the specified type. Unlike AWIP, this function only includes parts that have been shipped.

BMAX $^{(R)}$ *(buffer* $^{(N)}$*)*

Maximum number of parts that have been in a buffer.

BMAXTIME $^{(R)}$ (*buffer* $^{(N)}$)

Returns the longest period of time that any part has spent in the specified buffer, this includes parts which are still in the buffer. If no parts have yet entered the buffer, the value returned is 0.0.

BMIN $^{(R)}$*(buffer* $^{(N)}$*)*

Returns the minimum number of parts that have been in a buffer.

BMINTIME $^{(R)}$ *(buffer* $^{(N)}$*)*

Returns the shortest period of time that any part has spent in the specified buffer. If no parts have yet left the specified buffer, -1.0 is returned.

BREAKDWN $^{(I)}$ *(element* $^{(N)}$)

Forces a breakdown on the specified element.

CARRAT $^{(N)}$ *(p&f_element* $^{(N)}$*, position* $^{(I)}$*)*

Returns the name of the carrier that is currently in the section or station specified in P&F_element at the specified cardinal position (for example, the carrier at position 1).

CARRPART $^{(N)}$ *(carrier* $^{(N)}$*)*

Returns the name of the part currently on the specified carrier. If the specified carrier does not contain a part, CARRPART returns NONE.

CHR $^{(S)}$ *(ascii_code* $^{(I)}$*)*

Returns a string containing the character corresponding to the ASCII code given.

CLOSE $^{(I)}$*(file* $^{(N)}$)

Closes a WITNESS write file.

CLOSEWIN $^{(V)}$ (*Window* $^{(I)}$) *)*

Closes the specified window on the screen. Specify the window number or:
-1 = Designer Elements
-2 = Time
-3 = Clock
-4 = Interact box

ComInvokeInt $^{(I)}$*(…)*
….. ComRelease$^{(v)}$ *(…)*

A range of functions have been provided to invoke methods in COM objects. Other function names include CLSIDFromProgID and DispIDFromName

COMBODLG $^{(S)}$ *(title* $^{(S)}$*, dialog text* $^{(S)}$*, combo items* $^{(S)}$*, combo default* $^{(S)}$*)*

Allows you to create a combination box that is displayed at a point in the model run.

CONCENT $^{(R)}$ *(element, fluid, position* $^{(R)}$*)*

Percentage of fluid in the mixture at that position in the specified element.

CopyVar $^{(I}$ *(To_var* $^{(N)}$*, From_var* $^{(N)}$ *[, DestDim1Start*$^{(I)}$*, DestDim1End*$^{(I)}$ *SourceDim1Start* $^{(I)}$ *[,DestDim2Start*$^{(I)}$*, DestDim2End*$^{(I)}$ *SourceDim2Start* $^{(I)}$ *[, . . .]]])*

Copies all (or part) of a specified array to another of the same type and dimensions.

COS $^{(R)}$*(radians* $^{(R)}$*)*

Cosine.

CountItemsInteger $^{(I)}$*(ivar*$^{(N)}$ *, value* $^{(I)}$ *, Mode*$^{(I)}$*)*
CountItemsName $^{(I)}$*(nvar*$^{(N)}$ *, value* $^{(I)}$ *, Mode*$^{(I)}$*)*
CountItemsReal $^{(I)}$*(rvar*$^{(N)}$ *, value* $^{(I)}$ *, Mode*$^{(I)}$*)*
CountItemsString $^{(I)}$*(svar*$^{(N)}$ *, value* $^{(I)}$ *, Mode*$^{(I)}$*)*

Returns the number of values in the named array that match or do not match the value specified.

Mode = 0 : Items that do not match
Mode = 1 : Items that do match

CPosition $^{(R)}$ *(conveyor element* $^{(N)}$*), EntityIndex* $^{(I)}$ *)*

Returns the current physical position of the n'th entity on the conveyor where n is specified as EntityIndex.

CreateDrawing $^{(I)}$*(drawing_name* $^{(S)}$*)*

Creates an empty Drawing so you can add layers to it.

CreateFromDesigner $^{(I)}$ *(DesignerName* $^{(N)}$*, New_elt_name* $^{(S)}$*, Module_X* $^{(I)}$*, Module_Y* $^{(I)}$*, Child_X* $^{(I)}$*, Child_Y* $^{(I)}$*)*

Creates a new instance of the element and allows you to position its display.

CreateLayer $^{(I)}$*(layer_name* $^{(S)}$*)*

Creates an empty layer so you can add display items and graphics to it.

CYCLE $^{(I)}$*(machine)*

Current processing cycle of a multi-cycle machine.

DBAdd$^{I)}$ *(...)*
….. DBUpdate$^{(I)}$*(…)*

Numerous functions have been provided to connect to, read from and write to databases. Please consult the online help for more details.

DeleteDisplayItem $^{(I)}$ *(DisplayItem* $^{(I)}$*)*

Deletes the specified display item.

DeleteSensorByIndex *(element_name*$^{(N)}$*, index_position*$^{(I)}$*)*

Deletes a sensor from a Conveyor element, The sensor deleted is the n'th defined against the conveyor (as shown in the conveyor detail dialog) where n is the index position entered. Can only be used for continuous conveyors

DeScheduleExpression *(Tag String*$^{(S)}$*)*

Is a void function. It removes the specified event in the simulation event queue. Only events which have been added using the ScheduleExpression function can be removed.

DESTOF1 $^{(N)}$ *(vehicle* $^{(N)}$*)*

Returns the immediate destination of a vehicle. If the vehicle has no destination, NONE is returned.

DESTOF2 $^{(N)}$ *(vehicle $^{(N)}$)*

Returns the second (unload) destination of a vehicle. If the vehicle has no unload destination, NONE is returned.

DetachLabor $^{(I)}$ *(element $^{(N)}$, labor $^{(N)}$)*

Detaches the labor unit from the specified machine or vehicle.

DetachLaborFromPartAt $^{(I)}$ *(element $^{(N)}$, position $^{(I)}$, labor $^{(N)}$)*

Detaches the labor unit from the part at the specified position in the machine, vehicle, buffer, path, conveyor or carrier.

DISTANCE $^{(R)}$ *(vehicle $^{(N)}$)*

Returns the total physical distance traveled by vehicle.

DrawLine $^{(I)}$ *(Layer $^{(I)}$, RGBColor $^{(I)}$, x1 $^{(I)}$, y1 $^{(I)}$, x2 $^{(I)}$, y2 $^{(I)}$)*

Draws a line on the specified layer and adds it to the BACKDROP element. The color of the line is expressed as an RGB value and is specified by the *RGBColor* parameter. The RGB value is a combined color value that you can obtain using the RGBColor function. The line extends from position (x1, y1) to position (x2, y2).
Each item displayed on the layer is allocates an integer. DrawLine returns the integer indicating which display item this line is.

ELTCOUNT $^{(I)}$*(elt_type $^{(I)}$)*

Returns the number of designer or simulation elements in the model (or both).

Enter *elt_type* as 0 to return the total number of simulation and designer elements in a model.
Enter *elt_type* as 1 to return the total number of simulation elements in a model.
Enter *elt_type* as 2 to return the total number of designer elements in a model.

ELTNAME $^{(N)}$*(position_number $^{(I)}$)*

Returns the name of the element at the position in the element table specified by position_number. Returns NONE if vacant.

ELTTYPE $^{(I)}$ *(elt_name $^{(I)}$)*

Returns a type number which identifies the element type for the element specified in *elt_name*. 0 = error.

EVALERR $^{(I)}$

Returns the status of the last evaluation function (EVALINT, EVALREAL, EVALSTR, or EVALNAME). If the evaluation function returned something successfully, EVALERR returns 1, otherwise it returns 0.

EVALINT $^{(I)}$ *(string_exp $^{(S)}$)*
EVALNAME $^{(N)}$ *(string_exp $^{(S)}$)*
EVALREAL $^{(R)}$ *(string_exp $^{(S)}$)*
EVALSTR $^{(S)}$ *(string_exp $^{(S)}$)*

These functions evaluate string_exp. If the string expression evaluates to the appropriate type the value is returned. If an error occurs, or if the string expression does not evaluate to the type of the function used, 0, NONE, 0.0, “” is returned respectively. Use the EVALERR function to check whether has returned a value successfully, as of course the return values are legal results.

EX $^{(R)}$ (expression $^{(R)}$)

Returns e^{x}, where x = expression.

ExtendDelayTime $^{(I)}$ *(buffer $^{(N)}$, extra_time $^{(R)}$, mode $^{(I)}$, position $^{(I)}$)*

Extends the delay time originally set on the buffer’s Detail dialog (if the buffer is on-shift).

FileExists $^{(I)}$ *(file $^{(N)}$)*

Checks whether the file associated with the specified WITNESS file element exists.

FindElement $^{(N)}$*(Place_index $^{(I)}$, Type $^{(I)}$, Module_name $^{(N)}$)*

Finds the name of an element of the specified type within a model.

FLOAT $^{(R)}$ *(integer $^{(I)}$)*

Converts an integer to a real number.

FormatAsDate $^{(S)}$ *(date $^{(R)}$, option $^{(I)}$)*

Extracts the calendar date portion of a real date value and converts this into a string in the format "DD/MM/YYYY" (or whatever date format your PC has been set to).

FormatAsDateAndTime $^{(S)}$ *(date_and_time* $^{(R)}$*, option* $^{(I)}$*)*

Converts a real date value into a string in the format "DD/MM/YYYY HH:MM:SS". Note that the date format depends on the date format set on your PC.

FormatAsTime $^{(S)}$ *(time* $^{(R)}$*)*

Converts a real time value into a string in the format "HH:MM:SS" . Note that this function returns the total number of hours (that is, HH can be a value greater than 24) unlike the FormatTimeOfDay function, which returns an actual time of day.

FormatAsTimeOfDay $^{(S)}$ *(time_of_day* $^{(R)}$*, option* $^{(I)}$*)*

Extracts the time of day (from 00:00am to 12:00pm) from a (real) date value and converts this to a string in the "HH:MM:SS" 24 hour clock format. Note that this function ignores how many days result from the value, unlike the FormatAsTime function.

FPUTIL $^{(R)}$ *(element* $^{(N)}$*, state* $^{(I)}$*)*

The percentage of total time (on and off-shift) that the specified pipe, tank or processor spent in the specified state.

FreeSectionLength $^{(R)}$ *(section_name* $^{(N)}$*)*

The number of units of section length that are currently empty at the start of the specified section.

FSUTIL $^{(R)}$ *(element* $^{(N)}$*, state* $^{(I)}$*)*

Percentage of on-shift time that the pipe, tank or processor spent in the state.

FTYPE $^{(N)}$*(element* $^{(N)}$*, position* $^{(R)}$*)*

Returns the type of fluid in the element.

GenerateReportForType $^{(I)}$ *(report_name* $^{(N)}$*, element_type* $^{(I)}$*)*

Generates a customized report for the specified element type. The type is indicated by an integer value (see help text for details).

GenerateUserReport $^{(I)}$ *(report* $^{(N)}$*, element* $^{(N)}$*)*

Generates a customized report for the specified element.

GetActualTime $^{(R)}$ ( )

Returns the current actual time, taken from the clock on your PC.

GetAttachedItemAtPartAt $^{(N)}$ *(element* $^{(N)}$*, position* $^{(I)}$*, index* $^{(I)}$*)*

Returns the name of the part or labor unit that is currently attached to the part at the specified position in the specified machine, vehicle, buffer, path, conveyor or carrier.

GetAttachedLabor $^{(N)}$ *(element* $^{(N)}$*, index* $^{(I)}$*)*

Returns the name of the labor unit that is currently at the specified index in the specified element.

GetByQtyCostsByElement $^{(R)}$*(element*$^{(N)}$ *{,element*$^{(N)}$*...})*

Returns the costs (based on each job / task) calculated in the model for the element list supplied

GetByUseCostsByElement $^{(R)}$*(element*$^{(N)}$ *{,element*$^{(N)}$*...})*

Returns the costs by use (based on utilizations) calculated in the model for the element list supplied

GetByQtyCostsByType $^{(R)}$*(type* $^{(I)}$ *{, type* $^{(I)}$*})*

Returns the costs (based on each job / task) calculated in the model for the element types supplied (see Element Type Table)

GetByUseCostsByType $^{(R)}$*(type* $^{(I)}$ *{, type* $^{(I)}$*})*

Returns the costs by use (based on utilizations) calculated in the model for the element types supplied (see Element Type Table)

GetCarrierAtPosition $^{(N)}$ *(element* $^{(N)}$*, position* $^{(I)}$*)*

Returns the name of the carrier at the specified ordinal position in the specified section or station (eg. First carrier on a section).

GetCarrierSpeed $^{(R)}$ *(carrier_name* $^{(N)}$*)*

Returns the speed that has been set on the carrier’s detail dialog (this might be less than the actual speed).

GetCurrentCarrierSpeed $^{(R)}$ *(carrier_name* $^{(N)}$*)*

Returns the actual speed of the current carrier (this might be different to the speed specified on the carrier detail dialog).

GetCurrentDirectory $^{(S)}$ *()*

Returns the name of the current directory.

GetCurrentShift $^{(N)}$ *(element_name* $^{(N)}$*)*

Returns the name of the shift to which the specified element is working.

GetDay $^{(I)}$ *(Time* $^{(R)}$*)*

Extracts the day of the month from a variable or time/date expression and returns an integer in the range 1 to 31.

GetDayOfWeek $^{(I)}$ *(Time* $^{(R)}$*)*

Converts the day of the week from a variable or time/date expression into an integer in the range 1 (Sunday) to 7 (Saturday).

GetDayOfWeekAsString $^{(S)}$ *(Time* $^{(R)}$*)*

Converts the day of the week from a variable or time/date expression into a string, e.g. “Monday”.

GetDayOfYear $^{(I)}$ *(Time* $^{(R)}$*)*

Converts the day from a variable or time/date expression into an integer indicating a day of the year from 1 (January 1st) to 365 (December 31st) or 366 (December 31st in a leap year).

GetEventBreakdownNo *(Element_Name*$^{(N)}$*, Event_Index*$^{(I)}$*)*

Returns an integer value that identifies the breakdown number of the specified event. If the event is not a breakdown event then zero is returned. The element name must contain the index number for the element even if the quantity of the element is one (machine elements only)

GetEventCount *(Element_Name*$^{(N)}$*)*

Returns an integer value that is the number of events that are currently scheduled in the Event Queue for the specified element. The element name must contain the index value for the element even if the element has a quantity of 1 (machine elements only)

GetEventTime *(Element_Name*$^{(N)}$*, Event_Index*$^{(I)}$*)*

Returns a real value that is the time that the specified index event is scheduled to occur. The element name must contain the index number for the element even if the quantity of the element is one (machine elements only)

GetHour $^{(I)}$ *(Time* $^{(R)}$*)*

Extracts the hour from a variable or time/date expression and returns an integer in the range 0 (midnight) to 23 (11 pm).

GetMinute $^{(I)}$ *(Time* $^{(R)}$*)*

Extracts the minute from a variable or time/date expression and returns an integer in the range 0 to 59.

GetMonth $^{(I)}$ *(Time* $^{(R)}$*)*

Extracts the month from a variable or time/date expression and returns an integer in the range 1 (January) to 12 (December).

GetMonthAsString $^{(S)}$ *(Time* $^{(R)}$*)*

Converts the month from a variable or time/date expression into a string.

GetSecond $^{(I)}$ *(Time* $^{(R)}$*)*

Extracts the second from a variable or time/date expression and returns an integer in the range 0 to 59.

GetYear $^{(I)}$ *(Time* $^{(R)}$*)*

Extracts the year from a variable or time/date expression and returns an integer in the range 100 to 9999.

GETDCARR $^{(N)}$ *(carrier* $^{(N)}$*)*

Returns the destination of the specified carrier. If the carrier has no destination, GETDCARR returns NONE.

GetFixedCostsByElement $^{(R)}$*(element*$^{(N)}$ *{,element*$^{(N)}$*...})*

Returns the fixed costs calculated in the model for the element list supplied

GetFixedCostsByType $^{(R)}$*(type* $^{(I)}$ *{, type* $^{(I)}$*})*

Returns the fixed costs calculated in the model for the element types supplied (see Element Type Table)

GETICON $^{(I)}$ *(elt_name* $^{(N)}$ *,index* $^{(I)}$*)*

Returns the icon number of the element specified in *elt_name* (and with the specified index). If an error occurs, the function returns 0.

GetIconHeight $^{(I)}$ *(element_name* $^{(N)}$*, instance* $^{(I)}$*)*
Returns the height (in pixels) of the specified instance of the specified icon.

GetIconWidth $^{(I)}$ *(element_name* $^{(N)}$*, instance* $^{(I)}$*)*
Returns the width (in pixels) of the specified instance of the specified icon.

GETINFO $^{(S)}$ *(elt_name* $^{(N)}$ *,lst_file_string* $^{(S)}$*)*
This function allows you to retrieve details about a particular element that are stored in a model library (*.lst) file and return them as a string. The equivalent WCL command to GETINFO is OUTPUT.
GETINFO finds the details for the element *elt_name* in the *.lst file, then searches the file for the string lst_file_string. If the function finds the string, it returns element details from that point in the file onward. If the function doesn't find the string, it returns all the detail for that element.

GETLAYER $^{(I)}$
Returns the number of the current simulation layer (0 = simulation layer, 1 = layer 1 and so on).

GetMappingFactor $^{(R)}$ *( )*
Returns the conversion factor as entered on the Coordinates page of the Model Options dialog.

GetMappingUnit $^{(S)}$ *( )*
Returns the units of measurement as entered on the Coordinates page of the Model Options dialog.

GetModelAuthor $^{(S)}$ *( )*
Returns a string containing the name of the model's author.

GetModelDate$^{(S)}$ *( )*
Returns a string containing the model's date.

GetModelName$^{(S)}$ *( )*
Returns a string containing the model's name.

GetModelTitle$^{(S)}$ *( )*
Returns a string containing the model's title.

GetNumSensors *(element_name*$^{(N)}$*)*
The GetNumSensors function returns the number of sensors defined for the specified conveyor element. Can only be used for continuous conveyors

GetRevenueByElement $^{(R)}$*(mode* $^{(I)}$*,element*$^{(N)}$ *{,element*$^{(N)}$*…})*
Returns the revenue calculated in the model for the element list supplied

GetRevenueByType $^{(R)}$*(mode* $^{(I)}$*, type* $^{(I)}$ *{, type*$^{(I)}$*})*
Returns the revenues calculated in the model for the element types supplied (see Element Type Table)

Mode 0: All Revenue
Mode 1: All Revenue from shipped parts
Mode 2: All Revenue from scrapped parts

GetSensorPosByIndex *(element_name*$^{(N)}$*, index_position*$^{(I)}$*)*
Returns a real value in distance units from the head of the conveyor. Can only be used for continuous conveyors

GetStartTime*()* $^{(R)}$
Returns the start time set for the model

GetTotalCost*()* $^{(R)}$
Returns the total cost calculated in the model

GetTotalProfit*()* $^{(R)}$
Returns the total profit calculated in the model. i.e. revenues - costs

GetTotalRevenue $^{(R)}$ *(mode* $^{(I)}$*)*
Returns the total revenue calculated in the model

Mode 0: All Revenue
Mode 1: All Revenue from shipped parts
Mode 2: All Revenue from scrapped parts

GETXPOSN $^{(I)}$*(element* $^{(N)}$*, display type* $^{(I)}$*, Instance* $^{(I)}$*)*
Gives the X co-ordinate of the specified element display object. Specify the *instance* of the display type required. See the section describing **Display Types**.

GETYPOSN *$^{(I)}$(element $^{(N)}$, display type $^{(I)}$, Instance $^{(I)}$)*

Gives the Y co-ordinate of the specified element display object. Specify the *instance* of the display type required. See the section describing **Display Types**.

HALTCARR *$^{(I)}$ (p&f_element $^{(N)}$)*

Stops the carrier(s) in the network, section or station specified in *P&F_element* and returns the number of carriers that have been deactivated successfully. When carriers are stopped, they enter the free state.

If *P&F_element* is a *network*, HALTCARR stops all carriers that are on sections and stations within the network.

If *P&F_element* is a *section*, HALTCARR stops all carriers on that section.

If *P&F_element* is a *station*, that station finishes its operation on the carrier and pushes the carrier on to the next element (via its connection rule). As soon as the carrier reaches the carrier reaches the next element, HALTCARR stops the carrier.

If *P&F_element* is a *carrier*, HALTCARR stops that carrier.

HMAX *$^{(R)}$ (histogram $^{(N)}$)*

Returns the maximum observation recorded in the histogram.

HMAXTM *$^{(R)}$(histogram $^{(N)}$)*

Returns the time at which the maximum observation was recorded in the histogram.

HMEAN *$^{(R)}$(histogram $^{(N)}$)*

Returns the mean value recorded in the histogram.

HMIN *$^{(R)}$(histogram $^{(N)}$)*

Returns the minimum observation recorded in the histogram.

HMINTM *$^{(R)}$(histogram $^{(N)}$)*

Returns the time at which the minimum observation was recorded in the histogram.

HMODE *$^{(R)}$(histogram $^{(N)}$,mode)*

Returns the middle value of the range which occurs most frequently in the histogram (that is, the highest bar).

Mode 0: Return 0 if more than one highest bar
Mode 1: Use last highest bar
Mode 2: Use first highest bar

HSTD *$^{(R)}$(histogram $^{(N)}$)*

Returns the standard deviation of values recorded in the histogram.

IABS *$^{(I)}$ (expression $^{(I)}$)*

Returns the absolute value of the integer expression.

IFIX *$^{(R)}$ (expression $^{(R)}$)*

Converts the value of the real expression to an integer.

IHCELL *(histogram $^{(N)}$, cell $^{(I)}$)*

Returns the number of observations recorded in the histogram cell.

IHNUM *$^{(I)}$(histogram $^{(N)}$)*

Returns the number of observations recorded in the histogram.

ImportState *$^{(V)}$(Filename $^{(S)}$)*

Reads model initialization data from a text file and changes a model accordingly.

INAMEVAL *$^{(I)}$(variable $^{(N)}${,position $^{(I)}$ {,position $^{(I)}$....}})*

Retrieves the integer value held in the variable named by *variable*. The *position* parameters allow the integer variable to be indexed.

INDEX *$^{(I)}$ (element $^{(N)}$)*

The index value of the specified element. Used to extract index values from a name variable. Returns 0 if the name does not contain an index.

INPUTDLG *$^{(S)}$ (title $^{(S)}$, dialog text $^{(S)}$, field default $^{(S)}$, field width $^{(I)}$, type id $^{(I)}$)*

Allows you to create an edit dialog that is displayed at a point during the model run.

InvokeDialog $^{(I)}$ *(Module_name* $^{(N)}$*)*

Runs a user-defined dialog, setting all of the variables in the specified Module.
For format, validation and prompting control: see on-line help.

IPOSIVAL $^{(I)}$ *(Var_name* $^{(N)}$ *, Search_value*$^{(I)}$*, dimension*$^{(I)}$*, start_pos*$^{(I)}$*)*
IPOSNVAL $^{(I)}$ *(Var_name* $^{(N)}$ *, Search_value*$^{(N))}$*, dimension*$^{(I)}$*, start_pos*$^{(I)}$*)*
IPOSRVAL $^{(I)}$ *(Var_name* $^{(N)}$ *, Search_value*$^{(R)}$*, dimension*$^{(I)}$*, start_pos*$^{(I)}$*)*
IPOSSVAL $^{(I)}$ *(Var_name* $^{(N)}$ *, Search_value*$^{(S)}$*, dimension*$^{(I)}$*, start_pos*$^{(I)}$*)*

Searches the specified dimension for the first occurrence of the specified value. Returns the index position if found.
0 = not found.

ISampleFromVariable $^{(I)}$*(ivar*$^{(N)}$ *, rvar*$^{(N)}$ *, index, count, type, stream* $^{(I)}$ *, substream*$^{(I)}$*)*

Returns an integer value sample from the distribution defined by the two arrays. The distribution starts at the index position and contains 'count' values.

Type=0: Continuous
Type=1: Discrete

Stream and Substream arguments are optional

ISNAME $^{(I)}$ *(name* $^{(S)}$*)*

Checks whether a string can be converted to a valid WITNESS name.

ISNUMBER $^{(I)}$ *(number* $^{(S)}$*)*

Checks whether a string can be converted to a valid number.

ISTATE $^{(I)}$ *(element* $^{(N)}$*)*

Returns a code number indicating the current state of the element. See **Element States** section for details.

IVARSET$^{(I)}$ *(var_name* $^{(N)}$*, value*$^{(I)}$ *[,dim1start*$^{(I)}$*, dim1end*$^{(I)}$ *[,dim2start*$^{(I)}$*, dim2end*$^{(I)}$ *[, . . .]]])*

Sets all (or part) of a specified integer array to a specified value.
Example: Integer variable V(5)
After IVARSET(V,0) V will =0,0,0,0,0

Integer Variable V(3,4)
After IVARSET(V,0) IVARSET(v,3,0,0, 1,2) would update the array to :-

```
3    3    3
3    3    3
0    0    0
0    0    0
```

as dimension 1 varies from 1 to 3 (0=start to 0=end) and dimension 2 varies between 1 and 2. Using 0 for end values, allows you to future-proof your actions, so initialization remains valid should the dimension change

LABORAT $^{(N)}$ *(element* $^{(N)}$*, position* $^{(I)}$*, task* $^{(I)}$*)*

Returns the name of the labor item at the specified queue position at the specified element. The task is used to select the queue during multiple setups or multiple breakdowns. Use 1 if not applicable.

LaborPUtil $^{(R)}$ *(labor_name* $^{(N)}$*, job_index* $^{(I)}$*)*

Returns the percentage utilization for the specified labor unit for the specified job (total time).

LaborSUtil $^{(R)}$ *(labor_name* $^{(N)}$*, job_index* $^{(I)}$*)*

Returns the percentage utilization for the specified labor unit for the specified job (on-shift time only).

LaborUtilJobType $^{(I)}$ *(labor_name* $^{(N)}$*, job_index* $^{(I)}$*)*

Returns an integer that indicates what type of job a labor unit did for a specified job index.

LaborUtilName $^{(N)}$ *(labor_name* $^{(N)}$*, job_index* $^{(I)}$*)*

Returns the name of the element at which the specified labor unit performed the specified job.

LaborUtilNoOfJobs $^{(I)}$ *(labor_name* $^{(N)}$*)*

Returns the number of job types that a labor unit has performed at each element.

LAYEROFF $^{(V)}$ *(layer_no* $^{(I)}$*)*

Makes the layer specified in layer_no invisible (0 = simulation layer, 1 = layer 1 and so on).

LAYERON $^{(V)}$ *(layer_no* $^{(I)}$*)*

Makes the layer specified in layer_no visible. (0 = simulation layer, 1 = layer 1 and so on.)

LCARR $^{(I)}$*(p&f_section* $^{(N)}$*)*
Returns the number of loaded carriers that have passed through the specified section.

LEFTSTR $^{(S)}$ *(string* $^{(S)}$*, number_of_chars* $^{(I)}$*)*
Returns a string containing the number of characters specified taken from the start of the input string.

LIFE $^{(I)}$ *(element* $^{(N)}$*, setup number* $^{(I)}$*)*
Returns the remaining life (expressed in number of operations) of a setup, that is, number of cycles that will occur before the specified setup takes place.

LN $^{(R)}$ *(expression* $^{(R)}$*)*
Returns the natural log of the specified value.

LoadExternalIcon $^{(I)}$ *(icon_name* $^{(S)}$*)*
Loads an icon into the WITNESS picture gallery.

LoadModule $^{(I)}$ *(MDL_filename* $^{(S)}$*, New_module_name* $^{(S)}$*, Module_X* $^{(I)}$*, Module_Y*$^{(I)}$*, Child_X* $^{(I)}$*, Child_Y* $^{(I)}$*)*
Loads a module from an *.mdl file and displays it at the specified position.

LOCATION $^{(I)}$ *(element* $^{(N)}$*)*
Returns the index position of the specified vehicle on it's track.
Or
Returns the percentage of the current section traveled by the specified carrier.

LOG $^{(R)}$ *(expression* $^{(R)}$*)*
Returns the Log_{10} of the specified value.

MAX $^{(I)}$ *(expression*$^{(I)}$ *{,expression* $^{(I)}$*...})*
Returns the maximum value of a series of integer expressions.

MaxDim $^{(I)}$*(variable* $^{(N)}$*), dimension* $^{(I)}$*)*
Returns the number of values in the specified dimension (that is, its index).

MaxVariable $^{(R)}$ *(variable* $^{(N)}$*)*
Returns the maximum of all the values in a variable.

MeanVariable $^{(R)}$ *(variable* $^{(N)}$*)*
Returns the mean of all the values in a dynamic variable.

MEMBER $^{(N)}$*(name* $^{(N)}$*, index* $^{(I)}$*)*
Returns the specified name with the specified index instead of any index already stored in the name.

MIDSTR $^{(S)}$*(string* $^{(S}$*, first_char* $^{(I)}$*, number_of_chars* $^{(I)}$*)*
Returns a string containing the number of characters specified taken from the middle of the input string to the right of the first character specified.

MIN $^{(I)}$ *(expression*$^{(I)}$ *{,expression* $^{(I)}$*...})*
Returns the minimum value of a series of integer expressions.

MinVariable $^{(R)}$ *(variable* $^{(N)}$*)*
Returns the minimum of all the values in a variable.

MOD $^{(I)}$*(numerator* $^{(I)}$*, denominator* $^{(I)}$*)*
Returns the remainder of *numerator* divided by *denominator*. For example: MOD(8,5) gives 3.

MoveDisplayItem $^{(I)}$ *DisplayItem* $^{(I)}$*, NewX* $^{(I)}$*, NewY* $^{(I)}$*)*
Moves the specified display item from its current position to a new position specified by the *NewX* and *NewY* parameters.

MoveItemToLayer $^{(V)}$ (*element_name* $^{(N)}$*, display_type* $^{(I)}$*, instance* $^{(I)}$*, layer_number* $^{(I)}$)
Moves the specified instance of the element's display item to the specified layer.

MoveSensorByIndex *(element_name*$^{(N)}$*, index_position*$^{(I)}$*, position*$^{(R)}$*)*
Moves the position of a sensor to a new location. The sensor moved is the n'th defined against the conveyor (as shown in the conveyor detail dialog) where n is the index position entered. Can only be used for continuous conveyors

MSGDLG $^{(I)}$ (*title* $^{(S)}$*, icon id* $^{(I)}$*, dialog text* $^{(S)}$*, button text* $^{(S)}$*, default button* $^{(I)}$)
Allows you to create a message box that is displayed at a point during the model run.

NAME2STR $^{(S)}$(name $^{(N)}$)
Returns a string containing the name specified.

NameExist $^{(I)}$ *(String* $^{(S)}$*)*

Checks whether the string is the name of a current WITNESS element.

NASSEMBL $^{(I)}$*(part* $^{(N)}$*)*

Returns the number of parts of this type that have been assembled.

NAVAIL $^{(I)}$ *(labor* $^{(N)}$*)*

Returns the number of units of the specified labor type which are available (that is, on-shift and not currently in use).

NCARR $^{(I)}$ *(p&f_element* $^{(N)}$*)*

Returns the number of carriers in the section, station or network specified in *P&F_element*.

NCREATE$^{(I)}$*(part* $^{(N)}$*)*

Returns the number of parts of this type which have been created, including those which were rejected on arrival.

NDEMANDS $^{(I)}$*(Vehicle_name* $^{(N)}$*)*

Returns the vehicle's demand list.

NDim $^{(I)}$*(variable* $^{(N)}$*)*

Returns the number of dimensions in the specified variable.

NearestLabor $^{(N)}$*(element* $^{(N)}$*, mode* $^{(I)}$*, labor* $^{(N)}$ *(, labor* $^{(N)}$*))*

The function examines the specified labor unit(s) to see whether any are idle. If so, it checks whether any of those idle elements could reach the specified element along a path or pseudo-path. The function then uses the specified mode to calculate which labor unit could reach the element most quickly. Finally, the function returns the name of that labor unit.

Mode = 1: searches for the nearest labor unit to the element in terms of number of pixels
Mode = 2: searches for the nearest labor unit to the element in terms of travel time.

NFREE $^{(I)}$*(element* $^{(N)}$*)*

Current free capacity of the element.

NJOB $^{(I)}$*(labor* $^{(N)}$*, code* $^{(I)}$*)*

Information on jobs performed by labor. *Code* selects that you require the Number of Jobs:
1 : Started
2 : Finished
3 : Using this labor now
4 : Pre-empted

NLAB $^{(I)}$*(element* $^{(N)}$*, labor* $^{(N)}$*)*

Returns the amount of labor at the element.

NLOADS $^{(I)}$*(vehicle name)*

Returns the number of times that a vehicle has loaded parts.

NLSHIFT $^{(I)}$*(labor* $^{(N)}$*,shift* $^{(N)}$*)*

Returns the number of *labor* members available on the specified *shift*.

NOCC $^{(I)}$*(conveyor* $^{(N)}$*, start position* $^{(I)}$*, end position* $^{(I)}$*)*

Returns the number of parts on the conveyor between the specified positions.

0 for Start = Front, 0 for end = Rear
Positive is taken as relative to front
Negative is taken as relative to rear.

NoOfAttachedItemsAtPartAt $^{(I)}$ *(element* $^{(N)}$*, position* $^{(I)}$*)*

Returns the number of parts and/or labor units that are attached to the part in the specified position in the specified machine, vehicle, buffer, path, conveyor or carrier.

NoOfAttachedLabor $^{(I)}$ *(element* $^{(N))}$*)*

Returns the number of labor units that are currently attached to the specified machine or vehicle.

NOPS $^{(I)}$*(element* $^{(N)}$*)*

Returns the total number of operations completed by the element. If no index is supplied the sum of all operations for the group is given.

NPARTS $^{(I)}$*(element* $^{(N)}$*)*

Number of parts in the element.

NPARTS2 $^{(I)}$*(element* $^{(N)}$*,part* $^{(N)}$*,mode* $^{(I)}$*)*

Returns the number of the specified part in the *element*.

Part can be "ALL" or a specific part name.
Mode indicates whether all or part of the element is to be included in the count.
Mode 0: All.
Mode 1: Those in the element itself.
Mode 2: Those in the input buffer.
Mode 3: Those in the output buffer.

NPCARR $^{(I)}$ *(p&f_element* $^{(N)}$*)*

Returns the number of carriers in the section, station or network specified in *P&F_element* that are in a moving (that is, powered) state.

NPERIOD $^{(I)}$*(shift* $^{(N)}$*)*

Returns the current period number of the specified shift pattern (Main shifts only). If the shift is not in progress or the initial offset being carried out, 0 is returned.

NQTY $^{(I)}$*(element* $^{(N)}$*)*

Returns the quantity defined of the element. This is not applicable to parts.

NREJECT $^{(I)}$*(part* $^{(N)}$*)*

Returns the number of parts of this type that were rejected on arrival.

NSampleFromVariable $^{(I)}$*(nvar*$^{(N)}$ *, rvar*$^{(N)}$ *, index, count, type, stream* $^{(I)}$ *, substream*$^{(I)}$*)*

Returns a name value sample from the distribution defined by the two arrays. The distribution starts at the index position and contains 'count' values.

Type=0: Continuous
Type=1: Discrete

Stream and Substream arguments are optional

NSCRAP $^{(I)}$ *(part*$^{(N)}$*)*

Returns the number of parts of this type that have been scrapped.

NSHIFTS $^{(I)}$*(shift* $^{(N)}$*)*

Returns the total number of completed shifts for the specified *shift*. The initial offset counts as a completed shift.

NSHIP $^{(I)}$ *(part* $^{(N)}$*)*

Returns the number of parts of this type that have been shipped.

NUM2STR $^{(S)}$ *(expression* $^{(R)}$*)*

Returns a string containing the numeric value given.

NVARSET$^{(I)}$ *(var_name* $^{(N)}$*, value*$^{(I)}$ *[,dim1start*$^{(I)}$*, dim1end*$^{(I)}$ *[,dim2start*$^{(I)}$*, dim2end*$^{(I)}$ *[, . . .]]])*

Sets all (or part) of a specified name array to a specified value. E.g. : See IVARSET.

NVEHICLE $^{(I)}$*(track* $^{(N)}$*)*

Returns the number of vehicles on this *track*.

NWAIT $^{(I)}$*(buffer* $^{(N)}$*)*

Returns the number of parts still in the *buffer* that have exceeded the delay time.

NWIP $^{(I)}$*(part* $^{(N)}$*)*

Returns the number of the specified part type currently in progress.

OPENWIN $^{(V)}$ (*Window* $^{(I)}$)

Opens the specified window on the screen. Specify the window number or:

-1 = Designer Elements -2 = Time
-3 = Clock -4 = Interact box

ParentOf $^{(N)}$*(ElementName*$^{(N)}$*)*

Returns the name of the parent module of the element named. NONE is returned at the top level of hierarchy.

PartSigmaRating $^{(R)}$ *(part* $^{(N)}$*)*

Returns the Sigma Rating of the process being modeled for the specified part.

PathDestination()

Returns a Name value, which is the destination for the current element travelling along the current path.

PathLength $^{(I)}$*(path_name* $^{(N)}$*)*

Returns the distance in pixels along the path.

PathLengthX $^{(I)}$*(path_name* $^{(N),}$ *mode* $^{(I)}$*)*

Returns the length of a horizontal path in pixels (that is, a path positioned in the X direction). The mode parameter determines how to calculate the distance of a path with more than one segment. *Mode 0*: total distance of the path in a horizontal direction (that is, horizontal or diagonal segments). *Mode 1*: WITNESS calculates the horizontal distance between the start and end points of the path.

PathLengthY $^{(I)}$*(path_name* $^{(N)}$***,*** *mode* $^{(I)}$*)*

Returns the length of a vertical path in pixels (that is, a path positioned in the Y direction). The mode parameter determines how to calculate the distance of a path with more than one segment. *Mode 0*: total distance of the path in a vertical direction (that is, vertical or diagonal segments). *Mode 1*: WITNESS calculates the vertical distance between the start and end points of the path.

PathSource()

Returns a Name value, which is the source element for the current element travelling along the current path. The source reported is the source on the current path only

PhysicalPathLength $^{(I)}$ *(path* $^{(N)}$*)*

Returns the actual length of the path, if you have mapped the pixels to units on the Coordinates page of the Model Options dialog.

PopFromAttributeStack $^{(V)}$ *(attribute_name* $^{(N)}$*)*

Removes the first value from the specified string attribute of the current part (or the top value in the stack, in programming terms).

PushToAttributeStack $^{(V)}$ *(attribute_name* $^{(N),}$ *value* $^{(R)}$*)*

Pushes the specified value into the specified string attribute of the current part, so that value becomes the first value in the string (or the top value in the stack, in programming terms).

PUTIL $^{(R)}$*(element* $^{(N)}$*, state* $^{(I)}$*)*

Returns the percentage utilization of an element in a particular state.

If an index of 0 is specified, it returns the average of all these elements. The code number to use for each state appear under ISTATE.

PUTIL applies to machines, conveyors, tracks, vehicles, shifts, labor, processors, pipes, tanks, networks, carriers, sections and stations.

QLength $^{(I)}$ *(element* $^{(N)}$*)*

Returns the length of a conveyor's part or labor queue or a track's display length.

RATEIN $^{(R)}$*(element* $^{(N)}$*)*

Returns the flow rate of fluid entering the element.

RATEOUT $^{(R)}$*(element* $^{(N)}$*)*

Returns the flow rate of fluid leaving the element.

READDIST $^{(V)}$*(distribution* $^{(N)}$*,filename* $^{(S)}$*)*

Reads distribution data from the specified text file and allocates it to the specified distribution.

READPFL $^{(V)}$*(Part_name* $^{(N)}$*,filename* $^{(S)}$*)*

Reads data from a text file and imports it into a part arrival profile.

ReadProfile $^{(R)}$*(Element* $^{(N)}$*,Time* $^{(R)}$ *, Mode* $^{(I)}$*))*

Returns a value from a profile distribution.

1 Linear Spline
2 Cubic Spline
3 Linear line of best fit
4 Quadratic line of best fit
5 Cubic line of best fit

READSHFT $^{(V)}$*(shift* $^{(N)}$*,filename* $^{(S)}$*)*

Reads shift data from the specified text file and allocates it to the specified shift. The text file extension should be **.sft**

RecordNameValue $^{(I)}$ *(variable* $^{(N)}$*, value* $^{(N)}$*)*

Adds another name value to the end of the current list of a dynamic variable's values.

RecordRealValue $^{(I)}$ *(variable* $^{(N)}$*, value* $^{(R)}$*)*

Adds another real value to the end of the current list of a dynamic variable's values.

RecordStringValue $^{(I)}$ *(variable* $^{(N)}$*, value* $^{(S)}$*)*

Adds another string value to the end of the current list of a dynamic variable's values.

REPAIR $^{(I)}$*(element* $^{(N)}$*)*

Forces a repair on the specified element.

RepairElement $^{(I)}$*(element* $^{(N)}$*), mode* $^{(I)}$ *)*

Forces the specified element to repair.

Mode 0: On resumption the next stoppage is set based on half the value of the element breakdown interval
Mode 1: On resumption the next stoppage is set based on the whole of the value of the element breakdown interval

RESETRE $^{(I)}$ (*element* $^{(N)}$ {, *element* $^{(N)}$…)

Resets the statistics of the specified element(s), or all elements if ALL is specified.

ResetReportsByElement $^{(I)}$ (*element* $^{(N)}$ {, *element* $^{(N)}$…)

Resets the statistics of the specified element(s), or all elements if ALL is specified.
(New name for the function RESETRE)

RESETRT $^{(I)}$ (*type* $^{(I)}$ {, *type* $^{(I)}$…)

Resets the reports of the specified element types to zero (or resets all reports if 0 is specified).

ResetReportsByType $^{(I)}$ (*type* $^{(I)}$ {, *type* $^{(I)}$…)

Resets the reports of the specified element types to zero (or resets all reports if 0 is specified).
(New name for the function RESETRT)

RGBColor $^{(I)}$ *(red* $^{(I)}$*, green* $^{(I)}$*, blue* $^{(I)}$*)*

Returns the combined RGB value for the red, green and blue intensity values specified.

RIGHTSTR $^{(S)}$*(string* $^{(S)}$*, number_of_chars* $^{(I)}$*)*

Returns a string containing the number of characters specified taken from the end of the input string.

RNAMEVAL $^{(R)}$*(variable* $^{(N)}$ *{,position* $^{(I)}$ *{,position* $^{(I)}$*....}})*

Retrieves the real value held in the variable named by *variable*.
The *position* parameters allow the real variable to be indexed.

ROUND $^{(R)}$*(realno*$^{(R)}$*,decimals* $^{(I)}$*)*

Rounds value of *realno* to the number of decimal places specified.
If decimals is negative rounding will be to the nearest **10** ^{abs}(decimals).

RSampleFromVariable $^{(I)}$*(rvar*$^{(N)}$ *, rvar*$^{(N)}$ *, index, count, type, stream* $^{(I)}$ *, substream*$^{(I)}$*)*

Returns a real value sample from the distribution defined by the two arrays. The distribution starts at the index position and contains 'count' values.

Type=0: Continuous
Type=1: Discrete
Stream and Substream arguments are optional

RunCommandFile $^{(I)}$ *(file_name* $^{(N)}$*)*

Runs a file that contains WCL commands.

RunProperties$^{(V)}$ *(Module_name* $^{(N)}$*)*

Allows you to run the properties actions of a simulation module. You can call the Runproperties of a lower level module in your top level RunProperties, allowing all modules to be configured in one place.

RVARSET$^{(I)}$ *(var_name* $^{(N)}$*, value*$^{(I)}$ *[,dim1start*$^{(I)}$*, dim1end*$^{(I)}$ *[,dim2start*$^{(I)}$*, dim2end*$^{(I)}$ *[, . . .]]])*

Sets all (or part) of a specified real array to a specified value. See IVARSET for example

SAFLOW $^{(R)}$*(part_or_fluid* $^{(N)}$*)*

Returns the average on-shift time spent in a model by parts or fluid.

SAPARTS $^{(R)}$*(element* $^{(N)}$*)*

Returns the time-weighted average of the number of parts in the specified buffer or conveyor during on-shift time.

SAVOL $^{(R)}$*(element* $^{(N)}$*)*

Returns the time-weighted average of the volume of fluid in the specified tank or pipe during on-shift time.

SAWIP $^{(R)}$*(part_or_fluid* $^{(N)}$*)*

Returns the average on-shift work-in-progress of a part or fluid.

ScheduleExpression *(Tag String(S), Actions String(S), Time(R))*

Is a void function. It sets a new event in the simulation event queue. The event is called the **Tag string** value entered. The event is scheduled at the time specified, and the action is the **Actions String** which could be a function or calculation

SearchIVar *(I)(variable (N)), value (I), dimension (I), index (I), start_position (I))*
SearchNVar *(I)(variable (N)), value (N), dimension (I), index (I), start_position (I))*
SearchRVar *(I)(variable (N)), value (R), dimension (I), index (I), start_position (I))*
SearchSVar *(I)(variable (N)), value (S), dimension (I), index (I), start_position (I))*

Searches a variable for a specified value in a specified dimension of the specified variable and returns the position of that value in relation to the specified index. (2-dimensional variables only).

SensorState *(I)(conveyor element(N)), Sensor No.(I))*

Returns 1 if sensor covered, 0 if not covered.

SetActiveDesignerPage *(I) (page_number (I))*

Makes the specified page the current page in the Designer Elements window.

SetAttributeOfCall *(I) (attribute (N), expression (S))*

Sets the value of an attribute for a vehicle when it is summoned by a CALL action.

SetCarrierSpeed *(I) (carrier_name (N), speed (R))*

Sets the speed of the specified carrier, overriding any value that has already been set on the carrier detail dialog.

SETDCARR *(I) (p&f_element (N) ,p&f_dest (N))*

If *P&F_element* is a carrier, SETDCARR sets the destination parameter of the carrier specified in *P&F_element* to the section or station specified in *P&F_dest*. SETDCARR then returns the number of carriers whose destination has been set successfully.

If *P&F_element* is a network, section or station, SETDCARR sets the destination parameter of all the carriers on that network, section or station to the section or station specified in *P&F_dest*. SETDCARR then returns the number of carriers whose destination has been set successfully.

SetDisplayItemColor *(I) (display_item (I), RGBColor (I))*

Sets the color of the specified display item to the specified RGBColor value. The RGBColor value is a combined RGB color value that you can calculate using the RGBColor function.

SetDynamicVariableSize *(I) (variable (N), size (I))*

Specifies the number of values that a specified dynamic variable contains.

SetEventTime *(Element_Name(N), Event_Index(I), New Event Time(R))*

Is a void function. It sets the scheduled time of the specified event to the new event time (Simulation Time NOT a delay time). The element name must contain the index number for the element even if the quantity of the element is one (machine elements only)

SETFILE *(V)(element (N),filename (S))*

Sets the actual file name of the specified file or part file element to the specified file name.

SetIconSize *(V)(element_name (N)), instance (I), width (I), height (I))*

Changes an icon's size in the display without changing its actual size in the picture gallery.

SetIconTransparency *(I)(icon_number (I)), red_value (I), green_value (I), blue_value (I))*

Makes parts of an icon appear transparent when displayed.

SETINFO *(I) (WCL_string (S))*

This function allows you to change a WITNESS model by using WCL commands. If the command is carried out successfully, SETINFO returns 1, otherwise it returns 0. You can use \n within the string to indicate a carriage return.

SETLAYER *(V) (layer_no (I))*

Sets the current layer to the layer number specified in layer_no (0 = simulation layer, 1 = layer 1 and so on).

SETPOSN *(V) (element (N), display type (I), Instance (I), X (I), Y (I))*

Sets the X and Y co-ordinates of the specified element display object. Specify the *instance* of the display type required. See the section describing **Display Types**.

SETSHIFT *(V)(element (N),shift(N))*

Sets the shift of the specified element. In the case of multiple labor elements, the member index can be specified and that member alone will have the shift applied.

SetSpeed $^{(V)}$*(conveyor element* $^{(N)}$*), speed* $^{(R)}$ *)*

Sets the speed of the selected continuous conveyor element to the value requested, Speed must be >=0

SetStartTime *(Time* $^{(R)}$ *)*

Sets the start time for a model to the time specified (in the base time units chosen for the model)

SETTITLE $^{(V)}$*(window* $^{(I)}$*,title* $^{(S)}$*)*

Sets the windows title to the given text. Specify the window number or:
-1 = Designer Elements
-2 = Time
-3 = Clock
-4 = Interact box

SetVehicleDestination $^{(I)}$*(Vehicle* $^{(N)}$*,Track* $^{(N)}$*)*

Sets the vehicle's destination to the specified track (executes a WAKEUP if parked).

SETVIEW $^{(V)}$*(window* $^{(I)}$*, X* $^{(I)}$*, Y* $^{(I)}$*,zoom* $^{(I)}$*)*

Alters the viewing properties of the specified window. X and Y represent the position required for the top left corner of the window. Zoom values of -3 to +5 are valid, -3 being *8 magnification.

SetWarnLevel $^{(I)}$ *(element* $^{(N)}$*, mode* $^{(I)}$*, level* $^{(I)}$*, value* $^{(I)}$*)*

Sets the value of the specified warning level defined against the element.

Mode 0: Falling warning levels
Mode 1: Rising warning levels
Level is the number of the warning level in defined order (i.e. as appears in the detail dialog)

SetWindowToDrawing $^{(I)}$*(window_number* $^{(I)}$*), drawing_number* $^{(I)}$*)*

Display the specified Drawing in the specified window.

Shuffle $^{(I)}$ *(variable* $^{(N)}$*, dimension* $^{(I)}$*, distance* $^{(I)}$*)*

Working on a single dimension of a variable array, values are moved up or down, with 0 being added at the tail end.
Example: Integer variable V(3) values 1,2,3
Shuffle(V,1,1) =2,3,0 Shuffle(V,1,-1)=0,1,2
(Replace lost values using the I/R/S/NVARSET functions).

SigmaRating $^{(R)}$ *(DefectsPerMillion* $^{(R)}$*)*

Calculates a Sigma Rating from the specified DefectsPerMillion value and returns that value.

SimbaTrigger *(string* $^{(S)}$*, integer* $^{(I)}$*)*

Generates the OnSimbaTrigger event in the WITNESS COM Object Model.

SIN $^{(R)}$ *(radians* $^{(R)}$*)*

Returns Sine of the expression.

SNAMEVAL $^{(S)}$*(variable* $^{(N)}$ *{,position* $^{(I)}$ *{,position* $^{(I)}$*....}})*

Retrieves the string value held in the variable named by *variable*. The *position* parameters allow the string variable to be indexed.

SortVar $^{(I)}$(var_name $^{(N}$, Dimension $^{(I)}$,SortOrder $^{(I)}$ *)*

Sorts the specified Array in the specified Dimension. Sort Order : 0 Descending, 1 Ascending.
Example: Integer variable IV(3,4)
SortVar(IV,1,1)

1	12	2		1	2	12
9	5	6	Becomes	9	6	5
4	8	7		4	7	8
10	3	11		10	11	3

Any variable type can be used.

SOVER $^{(R)}$*(buffer* $^{(N)}$*)*

Returns the average number of parts in the buffer, which remained in the buffer beyond the delay time. (on-shift time only).

SOVERT $^{(R)}$*(buffer* $^{(N)}$*)*

Returns the average on-shift time spent by parts in a specified buffer beyond the delay time.

StdDevVariable $^{(R)}$ *(variable* $^{(N)}$*)*

Calculates the standard deviation of all the values in a dynamic variable and returns the result.

STR *(S)(expression (R))*

Converts the expression given to a string.

STR2NAME *(N)(string (S))*

Returns a name variable containing the name specified by the string value given (if possible to convert).

STR2NUM *(R)(string (S))*

Returns a real number specified by the string value given (if conversion is possible).

STRLEN *(I)(string (S))*

Returns the length of the specified string.

STRSTR *(I)(stringa (S), stringb (S))*

Returns an integer indicating the first position of *stringb* in *stringa*. Not found = -1.

SumVariable *(R) (variable (N))*

Adds all the values in a dynamic variable together and returns the result.

SUTIL *(R)(element (N), state (I))*

Returns the percentage on-shift utilization of an element in a particular state.

If an index of 0 is specified, it returns the average of all these elements. The code number to use for each state appears under ISTATE. SUTIL applies to machines, conveyors, tracks, vehicles, labor, processors, pipes, tanks, networks, carriers, sections and stations.

Note: The percentage on-shift utilization of an element in all states may total more than 100%. The utilizations are calculated on the normal shift time, therefore any allowance to finish the job could cause utilization to exceed 100%.

SVARSET*(I) (var_name (N), value(I) [,dim1start(I), dim1end(I) [,dim2start(I), dim2end(I) [, . . .]]])*

Sets all (or part) of a specified string array to a specified value. E.g. : See IVARSET.

TCARR *(I)(p&f_element (N))*

Returns the total number of carriers that have passed through the specified section or station.

TimeEnteredModel *(R)( )*

Returns the simulation time at which the current part entered the model.

TimeInModel *(R)( )*

Returns the amount of simulation time that the current part has spent in the model.

TimeLeft *(R)(pathname (N, Position(I))*

Returns the time left before the Part or Labor at ‘position’ on ‘pathname’ reaches its destination. Postion : 1= First, 2=second …

TimeOnShift *(R)(element (N), param1 (I) , param2 (I))*

Returns the time from the start of the currently allocated shift. Param1 – 0 to include breaks and 1 to not include breaks; Param2 – 0 for current sub shift and 1 for top level master shift.

TIMEVT *(R)(element (N))*

Returns the time remaining until the next scheduled event for the specified element.
If no event is scheduled -1.0 is returned.

TIMINBUF *(R)(buffer (N), position(I))*

Returns the length of time that the part at the specified *position* has spent in the *buffer*.

TMAX *(R)(timeseries (N), plot number (I))*

Returns the maximum value of a timeseries plot.

TMAXTM *(R)(timeseries (N), plot number (I))*

Returns the time at which the maximum value in the timeseries plot was recorded.

TMEAN *(R)(timeseries (N), plot number (I))*

Returns the mean value of the specified timeseries plot. If *plot number* is out of range, 0.0 will be given.

TMIN *(R)(timeseries (N), plot number (I))*

Returns the minimum value of a timeseries plot.

TMINTM *(R)(timeseries (N), plot number (I))*

Returns the time at which the minimum value in the timeseries plot was recorded.

TOLOWER *(S)(string (S))*

Returns a string containing the input string after conversion to lower case.

TOTALIN *$^{(R)}$(element $^{(N)}$)*
Returns the total number of parts that have been on a conveyor or in a buffer.

TOTALOUT *$^{(R)}$(buffer $^{(R)}$)*
Returns the total number of parts that have been output from a buffer.

TOUPPER *$^{(S)}$(string $^{(S)}$)*
Returns a string containing the input string after conversion to upper case.

TPARTS *$^{(I)}$(p&f station $^{(N)}$)*
Returns the total number of parts that have passed through the specified station.

TRACK *$^{(N)}$(vehicle $^{(N)}$)*
Returns the name of the track currently occupied by the specified vehicle.

TRACKUSE *$^{(R)}$(track $^{(N)}$)*
Returns the total number of vehicles that have been on the track.

TravelTime *$^{(R)}$(start $^{(N)}$, destination$^{(N)}$,location$^{(N)}$,position $^{(I)}$)*
Returns the time it would take for a part or labor unit to travel between the start and destination elements using paths.

For Parts:
Location = The current location of the part Position = The position within that element.
For Labor:
Location = The name of labor required.
Position is ignored and can be omitted.

TREMAIN *$^{(R)}$(element $^{(N)}$)*
Returns the time remaining of the current shift state for the specified element.

TRUNC *$^{(I)}$(expression $^{(R)}$)*
Returns as an integer value the specified real number truncated at the decimal point.

ValidateActionBlock *$^{(I)}$(test_string $^{(S)}$)*
ValidateIconExpression *$^{(I)}$(test_string $^{(S)}$)*
ValidateIntegerExpression *$^{(I)}$(test_string $^{(S)}$)*
ValidateNameExpression *$^{(I)}$(test_string $^{(S)}$)*
ValidateRealExpression *$^{(I)}$(test_string $^{(S)}$)*
ValidateStringExpression *$^{(I)}$(test_string $^{(S)}$)*
These functions allow you to check whether WITNESS will validate *test_string* as the appropriate expression action block. Most useful, when *test_string* is being validated from a user defined dialog. The User Defined dialogs provide auto validation options, making use of these functions only necessary when you require to customize the validation.

Pass = 1, Fail =0.

VCHANGE *$^{(R)}$(fluid $^{(N)}$)*
Returns the volume of fluid that has been changed.

VCONTS *$^{(R)}$(element $^{(N)}$)*
Returns the current volume of fluid in the element.

VCREATE *$^{(R)}$(fluid $^{(N)}$)*
Returns the volume of fluid that has been created.

VEHICLE*$^{(N)}$(track$^{(N)}$, number$^{(I)}$)*
Returns the name of the vehicle currently on the specified track in the specified position. 1 = Front, 2 = Second, and so on.

VFREE *$^{(R)}$(element$^{(N)}$)*
Returns the amount of free space in the element.

VIN *$^{(R)}$(element$^{(N)}$)*
Returns the volume of fluid that has entered the element.

VISLAYER $^{(I)}$ *(layer)no* $^{(I)}$*)*

Returns 1 if the layer specified in *layer_no* is visible (0 = simulation layer, 1 = layer 1 and so on). Returns 0 if the layer specified in *layer_no* is not visible.

VOUT $^{(R)}$*(element*$^{(N)}$*)*

Returns the volume of fluid that has left the element.

VREJECT $^{(R)}$ *(fluid*$^{(N)}$*)*

Returns the volume of the specified fluid that has been rejected.

VSHIP $^{(R)}$ *(fluid*$^{(N)}$*)*

Returns the volume of specified fluid that has been shipped.

VWASTE $^{(R)}$ *(fluid*$^{(N)}$*)*

Returns the volume of the specified fluid that has been wasted.

VWIP $^{(R)}$ *(fluid* $^{(N)}$*)*

Returns the volume of fluid currently in the model (except fluid contained in parts).

WalkTime $^{(R)}$ *( )*

Returns the amount of time that parts and labor take to travel along a path (or that parts, vehicles and carriers take to travel along a path display for a conveyor, track or section) based on the length of a path and a walk rate.

WHEREIS $^{(N)}$ *(element* $^{(N)}$*)*

Returns the name of the element that contains the specified carrier or vehicle, or the name of the element at which labor is present.

WRNLEVEL $^{(R)}$ *(element* $^{(N)}$*, mode* $^{(I)}$*, level* $^{(I)}$*)*

Returns the value of the specified warning level defined against the element.

Mode 0: Falling warning levels
Mode 1: Rising warning levels
Level is the number of the warning level required. If level is out of range -1.0 is returned.

XLCellToInteger$^{(I)}$ *(…)*
….. XLWriteArray$^{(I)}$*(…)*

Numerous functions have been provided to connect to, read from and write to Microsoft Excel™. Please consult the online help for more details.

Please note the following:

- Where an element with a quantity greater than one has been defined, specify a unique element by adding the index in parentheses. For example, BUFF(23), ISTATE(MACH(5)).
- ‘Element’ refers to physical elements, where the result of the function would be meaningful.
- It is not necessary to assign a variable to function's return value even though it may return a value. For some functions, the consequence of calling the function is more important than the return value

Functions listed by element type

Schedule Expressions

DeScheduleExpression
ScheduleExpression

Mathematical & string manipulation functions

ABS
AMAX
AMIN
ASC
CHR
COS
EVALERR
EVALINT
EVALNAME
EVALREAL
EVALSTR
EX
FLOAT
IABS
IFIX
ISNAME
ISNUMBER
LEFTSTR
LN
LOG
MAX
MIDSTR
MIN
MOD
NAME2STR
NUM2STR
RIGHTSTR
ROUND
SIN
STR
STR2NAME
STR2NUM
STRLEN
STRSTR
TOLOWER
TOUPPER
TRUNC

Window, Display & Interaction functions

AddLayerToDrawing
CLOSEWIN
CreateDrawing
CreateLayer
COMBODLG
DeleteDisplayItem
DrawLine
GETICON
GetIconHeight
GetIconWidth
GETLAYER
GetMappingFactor
GetMappingUnit
GETXPOSN
GETYPOSN
InvokeDialog
INPUTDLG
LAYEROFF
LAYERON
LoadExternalIcon
MoveDisplayItem
MoveItemToLayer
MSGDLG
OPENWIN
RGBColor
SetDisplayItemColor
SetIconSize
SetIconTransparency
SETLAYER
SETPOSN
SETTITLE
SETVIEW
SetWindowToDrawing
VISLAYER

General element functions

AdjustPriority
AdjustShift
CreateFromDesigner
INDEX
ISTATE
MEMBER
NameExist
RESETRE
RESETRT
TREMAIN

Costing Functions

GetByQtyCostsByElement
GetByQtyCostsByType
GetByUseCostsByElement
GetByUseCostsByType
GetFixedCostsByElement
GetRevenueByElement
GetRevenutByType
GetTotalCost
GetTotalProfit
GetTotalRevenue

General model functions

APPEXEC
ELTCOUNT
ELTNAME
ELTTYPE
FindElement
GenerateUserReport
GetCurrentDirectory
GETINFO
GetModelAuthor
GetModelDate
GetModelName
GetModelTitle
GetStartTime
ImportState
LoadModule
NameExist
RunCommandFile
SetActiveDesignerPage
SetStartTime
SETINFO

Part functions

AFLOW
AOVER
ASTIME
ATIME
ATIME2
AWIP
AWIP2
NASSEMBL
NCREATE
NOCC
NPARTS
NPARTS2
NREJECT
NSCRAP
NSHIP
NWAIT
NWIP
PartSigmaRating
READPFL
SAFLOW
SAPARTS
SAWIP
SOVER
SOVERT
TimeEnteredModel
TimeinModel
TOTALIN
TOTALOUT
TPARTS
TravelTime

Buffer functions

AOVER
AOVERT
APARTS
ASTIME
ATIME
BMAX
BMAXTIME
BMIN
BMINTIME
ExtendDelayTIme
NFREE
NPARTS
NPARTS2
NQTY
NWAIT
SAPARTS
SOVER
SOVERT
TIMEVT
TIMINBUF
TOTALIN
TOTALOUT

Machine functions

- ASTIME
- BREAKDWN
- CYCLE
- GetEventBreakdownNo
- GetEventCount
- GetEventTime
- LIFE
- NFREE
- NOPS
- NQTY
- PUTIL
- REPAIR
- RepairElement
- SetEventTime
- SetShift
- SUTIL
- TIMEVT

Conveyor functions

- AddSensor
- APARTS
- ASTIME
- ATIME
- BREAKDWN
- CPosition
- DeleteSensorByIndex
- NFREE
- NOCC
- NPARTS
- GetNumSensor
- GetSensorPosByIndex
- MoveSensorByIndex
- NPARTS2
- NQTY
- PUTIL
- QLength
- REPAIR
- RepairElement
- SAPARTS
- SensorState
- SetShift
- Setspeed
- SUTIL
- TIMEVT
- TOTALIN

Labor functions

- AttachLabor
- AttachLaborToPartAt
- AJOBTIME
- DetachLabor
- DetachLaborFromPartAt
- GetAttachedLabor
- LABORAT
- LaborPutil
- LaborSutil
- LaborUtilJobType
- LaborUtilName
- LaborUtilNoOfJobs
- NAVAIL
- NearestLabor
- NJOB
- NLAB
- NLSHIFT
- NoOfAttachedLabor
- NQTY
- PUTIL
- SetShift
- SUTIL
- TIMEVT
- TravelTime
- WHEREIS

Track functions

- ASTIME
- BREAKDWN
- NFREE
- NPARTS
- NPARTS2
- NQTY
- NVEHICLE
- PUTIL
- QLength
- REPAIR
- RepairElement
- SetVehicleDestination
- SUTIL
- TIMEVT
- TRACK
- TRACKUSE
- VEHICLE

Vehicle functions

- ASTIME
- DESTOF1
- DESTOF2
- DISTANCE
- LOCATION
- NDEMANDS
- NFREE
- NLOADS
- NPARTS
- NPARTS2
- NQTY
- NVEHICLE
- PUTIL
- SetAttributeOfCall
- SetShift
- SetVehicleDestination
- SUTIL
- TIMEVT
- TRACK
- TRACKUSE
- VEHICLE
- WHEREIS

Network functions

- ACTCARR
- HALTCARR
- NCARR
- NPCARR
- PUTIL
- SETDCARR
- SUTIL

Section functions

- ACARR
- ACTCARR
- AFCARR
- ATSTATE
- BREAKDWN
- CARRAT
- FreeSectionLength
- GetCarrierAtPosition
- HALTCARR
- LCARR
- Location
- NCARR
- NPCARR
- PUTIL
- RepairElement
- REPAIR
- SETDCARR
- SUTIL
- TCARR
- WHEREIS

Station functions

- ACTCARR
- BREAKDWN
- CARRAT
- GetCarrierAtPosition
- HALTCARR
- NCARR
- NFREE
- NPARTS
- NPARTS2
- NPCARR
- PUTIL
- REPAIR
- RepairElement
- SETDCARR
- SUTIL
- TCARR
- TPARTS

Carrier functions

- ACTCARR
- APARTS
- ASTIME
- ATLOADED
- ATIME
- ATSTATE
- CARRAT
- CARRPART
- GetCarrierAtPosition
- GetCarrierSpeed
- GetCurrentCarrierSpeed
- GETDCARR
- HALTCARR
- LCARR
- LOCATION
- NCARR
- NPARTS
- NPCARR
- PUTIL
- SAPARTS
- SETDCARR
- SetCarrierSpeed
- SUTIL
- TCARR
- WHEREIS

Fluid functions

- AFLOW
- ASTIME
- ATIME
- AVOL
- AWIP
- CONCENT
- PUTIL
- RATEIN
- RATEOUT
- SAFLOW
- SAVOL
- SAWIP
- SUTIL
- VCHANGE
- VCONTS
- VCREATE
- VIN
- VOUT
- VREJECT
- VSHIP
- VWASTE
- VWIP

Tank functions

- ASTIME
- ATIME
- AVOL
- CONCENT
- FPUTIL
- FSUTIL
- FTYPE
- NQTY
- PUTIL
- RATEIN
- RATEOUT
- SAVOL
- SETSHIFT
- SetWarnLevel
- SUTIL
- TIMEVT
- VCONTS
- VFREE
- VIN
- VOUT
- WRNLEVEL

Processor functions

- BREAKDWN
- CONCENT
- FPUTIL
- FSUTIL
- FTYPE
- NOPS
- NQTY
- PUTIL
- PUTIL
- RATEIN
- RATEOUT
- REPAIR
- RepairElement
- SETSHIFT
- SetWarnLevel
- SUTIL
- TIMEVT
- VCONTS
- VFREE
- VIN
- VOUT
- WRNLEVEL

Pipe functions

- ASTIME
- ATIME
- AVOL
- BREAKDWN
- CONCENT
- FPUTIL
- FSUTIL
- FTYPE
- NQTY
- PUTIL
- RATEIN
- RATEOUT
- REPAIR
- RepairElement
- SAVOL
- SETSHIFT
- SUTIL
- TIMEVT
- VCONTS
- VIN
- VOUT

Variable functions

- CopyVar
- CountItemsInteger
- CountItemsName
- CountItemsReal
- CountItemsString
- INAMEVAL
- IPOSIVAL
- IPOSNVAL
- IPOSRVAL
- IPOSSVAL
- ISampleFromVariable
- IVARSET
- MaxDim
- MaxVariable
- MeanVariable
- MinVariable
- NDim
- NSampleFromVariable
- NVARSET
- RecordNameValue
- RecordRealValue
- RecordStringValue
- RNAMEVAL
- RSampleFromVariable
- RVARSET
- SearchIVar
- SearchNVar
- SearchRVar
- SearchSVar
- SetDynamicVariableSie
- SHUFFLE
- SNAMEVAL
- SortVar
- StdDevVariable
- SumVariable
- SVARSET

Distribution functions

- READDIST
- ReadProfile

Timeseries functions

- NQTY
- SETSHIFT
- TMAX
- TMAXTM
- TMEAN
- TMIN
- TMINTM

Histogram functions

- HMAX
- HMAXTM
- HMEAN
- HMIN
- HMINTM
- HMODE
- HSTD
- IHCELL
- IHNUM
- NQTY

File functions

- CLOSE
- FileExists
- READSHFT
- SETFILE

Shift functions

AdjustShift
GetCurrentShift
NLSHIFT
NPERIOD
NSHIFTS
PUTIL
READSHFT
SETSHIFT
TimeOnShift

Path functions

NPARTS
PathLength
PathLengthX
PathLengthY
PathDestination
PathSource
PhysicalPathLength
PUTIL
SUTIL
TimeLeft
TOTALIN
TOTALOUT
TravelTime
WalkTime

Piechart functions

NQTY
SETSHIFT

Attribute functions

AttributeMatchSetInteger
AttributeMatchSetName
AttributeMatchSetReal
AttributeMatchSetString
NQTY

Actions

Logic control

IF/ELSEIF/ELSE/ENDIF Execute different statements, according to the value of logical expressions.

GOTO/LABEL Continue actions at a point specified by the label.

FOR/NEXT Execute statements a specified number of times.

WHILE/ENDWHILE Execute statements while a condition remains true.

RETURN Exit from actions, returning a value in the case of a user-defined Function.

STOP Stop (interrupt) the simulation run.

Text input/output

PRINT *{expression {separator}...}*
Display a message in the Interact Box.

INPUT *{prompt* $^{(S)}$*, } variable or attribute*
Obtain user input from the Interact Box.

OPENBOX
Open the Interact Box

CLOSEBOX
Close the Interact Box.

WRITE *file* $^{(N)}$ *expression* $^{(I,R,N.or\ S)}$ *{{separator} expression* $^{(I,R,N\ or\ S)}$*{separator}...}*
Write values to a file on disk.

READ *file* $^{(N)}$ *variable or attribute* $^{(I,R,N\ or\ S)}$ *{, variable or attribute...}*
Read values from a file on disk.

REWIND *file* $^{(N)}$
Rewind (i.e. go to the top of) the file on the disk.

Track/vehicle control

CALL *vehicle*$^{(N)}$*,load track*$^{(N)}$*,unload track*$^{(N)}$*{, priority*$^{(I)}$*}*
Request a vehicle.

DEST *track* $^{(N)}$
Set the destination of the current vehicle.

DESTV *vehicle* $^{(N)}$*, track* $^{(N)}$
Set the destination of the specified vehicle.

FINDV *search track* $^{(N)}$*{,search track* $^{(N)}$*...}*
Try to meet the vehicle request immediately.

VSEARCH track $^{(N)}$*{,track* $^{(N)}$*...}*
Specify tracks to search to meet the vehicle request.

WAKEUP *vehicle* $^{(N)}$, *track* $^{(N)}${,*track* $^{(N)}$...}
Reactivate parked vehicles.

Tank/processor/pipe control

CLEAN *element* $^{(N)}$, *clean duration* $^{(R)}$
Executes a cleaning cycle on the specified continuous processing element.

PURGE *pipe* $^{(N)}$, *purge volume* $^{(R)}$
Executes a purge cycle of a specified volume through the specified pipe.

MOVE *volume* $^{(R)}$, *source* $^{(N)}$, *destination* $^{(N)}$
Moves fluids instantaneously from one element or location to another (tanks and processors only).

Miscellaneous commands

= Assign values to variables and attributes.
! Add a remark to an Action statement or a rule.

CHANGE *old part* $^{(N)}$, *new part* $^{(N)}$
Change type of the current part.

DIM *VariableName {AS DataType} {!Comment}*
Defines a local variable in a function or action. You must use the DIM statement before any executable statements in the rest of the action. The *VariableName* can contain a maximum of 24 alphanumeric characters, must start with a letter and is case-insensitive. The name must be unique within this action. However, local variables in separate actions can share the same name, and WITNESS elements or system elements can share the same name as a local variable (in this case, the action assumes the local variable rather than the element).

SET ICON *element* $^{(N)}$, *value* $^{(I)}$
Change the icon of the element (not parts).

SET LIFE *element* $^{(N)}$, *setup number* $^{(I)}$, *value* $^{(R)}$
Set the remaining life of a machine setup (no. of cycles to the next setup).

SET QUANTITY *element* $^{(N)}$, *quantity* $^{(I)}$
Set quantity of element name to new quantity.

SET QUANTITY LABOR *laborname* $^{(N)}$, *shift* $^{(N)}$, *new quantity* $^{(I)}$
Set quantity of labor laborname on shift to new quantity.

SET CAPACITY *element* $^{(N)}$, *new capacity* $^{(R)}$
Set capacity of element to new capacity.

RECORD *value* $^{(R)}$, *histogram name* $^{(N)}$
Record a value in a histogram.

ADDBAR *histogram* $^{(N)}$, *value* $^{(R)}$, *increment {,color* $^{(I)}$}
Extends bars (which have already been drawn) in different colors.

DRAWBAR *histogram* $^{(N)}$, *value* $^{(R)}$, *height* $^{(I)}$ *{, color* $^{(I)}$}
Records observations on a histogram.

INTERACT COLOR *number* $^{(I)}$
Sets the color of text in the Interact Box.

INTERACT CURSOR *x-coordinate* $^{(I)}$, *y-coordinate* $^{(I)}$
Sets the position of the cursor in the Interact Box.

INTERACT CLEAR *x_top_left* $^{(I)}$, *y_top_left* $^{(I)}$, *x_bottom_right* $^{(I)}$, *y_bottom_right* $^{(I)}$
Clears a specified area in the Interact Box.

Media control

MCIOPEN *mci_id* $^{(S)}$*, window_no* $^{(I)}$*, start_x* $^{(I)}$*, start_y*$^{(I)}$*, filename* $^{(S)}$

Opens devices (such as video).

MCICLOSE *mci_id* $^{(S)}$

Closes the identified device opened by a MCIOPEN action.

MCIPLAY *mci_id* $^{(S)}$ *{,REPEAT* | *WAIT}*

Plays the opened device. You have the option to WAIT until playing is complete, or REPEAT the play command until the end is reached.

MCIBEGIN *mci_id* $^{(S)}$

Sets the device back to the start.

MCISTEP *mci_id* $^{(S)}$ *{,frames* $^{(I)}$*}*

Steps the device forward. By default it is advanced by 1 frame. A number of *frames* can be specified (which can be negative).

MCI *command_string* $^{(S)}$

Sends the command string *command_ string* to the MCI command process.

MCISTOP *mci_id* $^{(S)}$

Stops a multimedia device playing.

Element states

Element states that the ISTATE, PUTIL and SUTIL functions use are listed below. PUTIL and SUTIL do not apply to buffers.

Machine states

State No.	**State**	**Color**
0	Off-shift	white
1	Waiting for parts	yellow
2	Busy	green
3	Blocked	magenta
4	Setting up	cyan
5	Being repaired	red
6	Waiting for labor to cycle	blue
7	Waiting for labor to set up	cyan
8	Waiting for labor to repair	red
9	Filling	green
10	Emptying	green
14	Waiting for parts to arrive at machine using path	yellow
15	Waiting for labor for cycle to arrive using path	blue
16	Waiting for labor for setup to arrive using path	cyan
17	Waiting for labor for repair to arrive using path	red
18	Waiting for labor to arrive to remove parts	magenta

Conveyor states

State No.	**State**	**Color**
0	Off-shift	white
1	Empty	yellow
2	Moving freely	green
3	Blocked	magenta
4	Moving and queued	cyan
5	Broken down	red
6	Restarting (if restart delay used)	dark green

Shift states

State No.	State	Color
0	Off-shift	(no color key)
1	On-shift	

Labor states

State No.	State	Color
0	Off-shift	White *
1	Idle	Yellow *
2	Busy	Green *

(* Not in modeling window; only available in graphical reporting)

Track states

State No.	State	Color
1	Empty	yellow
2	Busy	green
3	Blocked	magenta

(Only available in graphical reporting)

Section states

State No.	State	Color
0	Off-shift	white
1	Running	green
2	Broken down/repairing	red
3	Waiting for labor	blue

Vehicle states

State No.	State	Color
0	Off-shift	white
1	Idle	yellow
2	Demanded	cyan
3	Blocked	magenta
4	Loaded	green
5	Loading	blue
6	Unloading	blue
7	Stopped	red
8	Parked	dark green
9	Outside	No color – not yet entered simulation

Buffer states

State No.	State	Color
0	Off-shift	white
1	Empty	yellow
2	Partially full	green
3	Full	magenta

Network states

State No.	State	Color
0	Off-shift	white
1	Running	green

Carrier states

State No.	State	Color	P/F Status
0	Off-shift	white	N
1	Blocked	magenta	F
2	Parked	dark green	N
3	Loading	blue	N
4	Unloading	blue	N
5	Moving	green	P
6	Drive broken down	yellow	F
7	Carrier free on section	dark magenta	F
8	Carrier outside model	N/A	N
9	Parking	cyan	N
10	Unparking	cyan	N
11	Processing	green	N

F = Free, P = Powered, N = Neither

Station states

State No.	State	Color	Station Type
0	Off-shift	white	A
1	Idle	yellow	A
2	Processing carrier	green	A
3	Blocked	magenta	A
4	Repairing	red	A
5	Waiting for labor to process	blue	B
6	Parking	green	P
7	Unparking	green	P
8	Loading	green	L
9	Unloading	green	U
10	Waiting to load	blue	L
11	Waiting to unload	blue	U
12	Arriving	green	B
13	Departing	green	B
14	Waiting for labor to park	blue	P
15	Waiting for labor to unpark	blue	P
16	Waiting for labor to load	blue	L
17	Waiting for labor to unload	blue	U
18	Waiting for labor to process arrival	blue	B
19	Waiting for labor to process departure	blue	B
20	Waiting for labor to process repair	blue	A

Path states

State No.	State	Color
1	Idle	Yellow
2	Busy	Green

States for continuous processing elements used by ISTATE, PUTIL and SUTIL

Processor states

State No.	State	Color
0	Off-shift	white
1	Filling	yellow
2	Emptying	dark green
3	Processing	green
4	Cleaning	cyan
5	Broken down	red
6	Waiting for labor to cycle	blue
7	Waiting for labor to fill	yellow
8	Waiting for labor to empty	dark green
9	Waiting for labor to clean	cyan
10	Waiting for labor to repair	red

Tank states

State No.	State	Color
0	Off-shift	white
1	Running	green
2	Cleaning	cyan
3	Waiting for labor to flow	blue
4	Waiting for labor to clean	cyan

Pipe states

State No.	State	Color
0	Off-shift	white
1	Flowing	green
2	Cleaning/purging	cyan
3	Broken down	red
4	Waiting for labor to flow	blue
5	Waiting for labor to clean/purge	cyan
6	Waiting for labor to repair	red

States for continuous processing elements used by FPUTIL and FSUTIL

Processor states

State No.	State	Color
0	Off-shift	white
1	Filling, flowing	yellow
2	Filling, unable to flow	dark cyan
3	Processing	green
4	Emptying, flowing	dark green
5	Emptying, unable to flow	magenta
6	Empty	dark magenta
7	Waiting for labor to process	blue
8	Cleaning, or waiting for labor to clean	cyan
9	Repair, or waiting for labor to repair	red

Tank states

State No.	State	Color
0	Off-shift	white
1	Empty	yellow
2	Part full	green
3	Full	magenta
4	Cleaning, or waiting for labor to clean	cyan

Pipe states

State No.	State	Color
0	Off-shift	white
1	Flowing	green
2	Unable to flow	dark cyan
3	Cleaning/purging or waiting for labor to clean/purge	magenta
4	Repair, or waiting for labor to repair	red

Element Types

Each element in WITNESS is associated with a particular number (or class). The class is used in certain functions and OLE commands.

1	Part	8	Timeseries	15	Part file	22	Module
2	Machine	9	Histogram	16	Fluid	23	PF_Network
3	Conveyor	10	Distribution	17	Pipe	24	PF_Section
4	Buffer	11	Variable	18	Processors	25	PF_Station
5	Labor	12	Attribute	19	Tank	26	PF_Carrier
6	Track	13	Function	20	Shift	27	Path
7	Vehicle	14	File	21	Piechart	35	Report element

Pen colors

PEN CODE: Foreground/Background Color

0	As Part Style	7	White/Black	14	Black/Cyan	21	Magenta/Magenta
1	Red/Black	8	Black/Black	15	Black/White	22	Cyan/Cyan
2	Green/Black	9	White/Red	16	White/Black	23	White/White
3	Yellow/Black	10	Black/Green	17	Red/Red	24	Black/Black
4	Blue/Black	11	Black/Yellow	18	Green/Green		
5	Magenta/Black	12	White/Blue	19	Yellow/Yellow		
6	Cyan/Black	13	Black/Magenta	20	Blue/Blue		

Color codes: ADDBAR/DRAWBAR

0	Black	4	Blue	8	Gray	12	Dark Blue
1	White	5	Cyan	9	Dark Gray	13	Dark Cyan
2	Red	6	Yellow	10	Dark Red	14	Dark Yellow
3	Green	7	Magenta	11	Dark Green	15	Dark Magenta

Display Types: GETXPOSN/GETYPOSN

1	Icon	13	Timeseries	25	Line	37	Vehicle Style
2	Name	14	Histogram	26	Rectangle	38	Style
3	Part	15	Off Shift Labor	27	Ellipse	39	Idle Labor
4	Input Buffer	16	Vehicle	28	Polyline	40	Track
5	Output Buffer	17	Key	29	Flow rate	41	N/A (reserved)
6	Labor	18	Shift Status	30	Minimum	42	Max/Min
7	Text	19	Shift Period	31	Maximum	43	N/A (reserved)
8	Notes	20	Shift Sub-shift	32	Title	44	Path
9	Piechart	21	Pipe	33	Fluid	45	Station carrier queue
10	Variable Grid	22	Vessel	34	Track Length	46	Attached item queue
11	Variable Text	23	Vessel Contents	35	Function		
12	Expression	24	Vessel Mixture	36	Part Style		